GC30-3008-5

Systems

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

Network Control Program/VS—Version 5
Emulation Program/VS—Version 3

IBM

GC30-3008-5

Systems

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

Network Control Program/VS—Version 5
Emulation Program/VS—Version 3

IBM

Sixth Edition (November 1976)

This is a major revision of, and renders obsolete, GC30-3008-4. This edition applies to:

- Network Control Program/VS–OS/VS (Program No. 5744-BA2), Version 5 Modification 0
- Network Control Program/VS–DOS/VS (Program No. 5747-AJ2), Version 5 Modification 0
- Emulation Program/VS–OS/VS (Program No. 5744-AN1), Version 3 Modification 0
- Emulation Program/VS–DOS/VS (Program No. 5747-AG1), Version 3 Modification 0

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

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

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

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

Technical Newsletter

This Newsletter No.	GN30-3088
Date	September 2, 1977
Base Publication No.	GC30-3008-5
File No.	S360/S370-30
Previous Newsletters	None

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

This Technical Newsletter provides replacement and added pages for the subject publication. These pages remain in effect until the next edition of the publication. The pages to be replaced and added are:

iii-vi
xvii, xviii
xix, (blank)
5-69, 5-70
5-73, 5-74
5-79, 5-80
5-80.1, (blank) (added)
7-15 – 7-18
L-1 – L-5 (added)
X-1 – X-6
X-9, X-10

Each technical change is marked by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter adds information needed by users of the IBM Airlines Control Program to specify communication lines on which airlines line control (ALC) is to be used. Also added is a new operand of the GROUP macro: QUIETCT.

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications Center, Dept. E01, P.O. Box 12195, Research Triangle Park, N.C. 27709

Printed in U.S.A.

Preface

This publication provides the information necessary to define and generate a network control program/VS or emulation program/VS for the IBM 3704 or 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 (1) defining a network control program to be used in communicating with a central processing unit (CPU) in which the OS/VS or DOS/VS Virtual Telecommunications Access Method (VTAM) or OS/VS Telecommunications Access Method (TCAM) is being executed or (2) defining an emulation program to be used in communicating with a CPU in which TCAM, BTAM, or QTAM (or an equivalent access method that supports IBM 2701, 2702, or 2703 transmission control units) is being executed.

Important: This publication applies only to [1] version 5 (and subsequent versions) of the network control program/VS and [2] version 3 modification 0 of the emulation program/VS. The basic (version 1) network control program is covered by the publication, *IBM 3704 and 3705 Communications Controllers, Network Control Program Generation and Utilities Guide and Reference Manual (for OS/MFT and OS/MVT TCAM Users),* GC30-3000. Version 2 of the Network Control Program/VS (used with TCAM) is covered by the publication *IBM 3704 and 3705 Communications Controllers, Network Control Program Generation and Utilities Guide and Reference Manual (for OS/VS TCAM Users),* GC30-3007.

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

Prerequisite and Related Manuals

Prerequisite to use of this publication is a basic understanding of teleprocessing and teleprocessing access methods. You should also have a general knowledge of the purposes of the IBM 3704 and 3705 Communications Controllers; this may be obtained from the publication, *Introduction to the IBM 3704 and 3705 Communications Controllers,* GA27-3051. See the *VTAM System Programmer's Guide* and *TCAM System Programmer's Guide,* respectively, for VTAM and TCAM information.

Within this publication, mention is made of the following publications:

Control Panel Guide refers to either of these two publications, as appropriate:

Guide to Using the IBM 3704 Communications Controller Control Panel (GA27-3086)

Guide to Using the IBM 3705 Communications Controller Control Panel (GA27-3087)

Program Reference Handbook refers to:

IBM 3704 and 3705 Program Reference Handbook (GY30-3012)

NCP Program Logic Manual refers to:

IBM 3704 and 3705 Communications Controllers Network Control Program/VS Program Logic Manual (SY30-3013)

VTAM System Programmer's Guide refers to any of these three publications, as appropriate:

- *DOS/VS VTAM System Programmer's Guide* (GC27-6957)
- *OS/VS1 VTAM System Programmer's Guide* (GC27-6996)
- *OS/VS2 System Programmer's Library: VTAM* (GC28-0688)

TCAM Programmer's Guide refers to any of the following publications, as appropriate:

- *OS/VS1 TCAM Programmer's Guide* (GC30-2054)
- *OS/VS2 TCAM Programmer's Guide* (GC30-2041)
- *OS/VS TCAM System Programmer's Guide* (GC30-2051).

The *Teleprocessing Preinstallation Guide for IBM 3704 and 3705 Communications Controllers,* (GC30-3020) and the *Teleprocessing Installation Record for IBM 3704 and 3705 Communications Controllers,* (GC30-3021) may also be used as aids to defining your emulation program. The *Preinstallation Guide* provides the appropriate values for many of the network configuration parameters that must be specified in the emulation program and gives the value(s) appropriate to each type of station and line set accommodated by the program. The *Installation Record* contains a set of formatted sheets representing the communications controller and the attached lines, with labeled spaces suitable for recording the parameter values appropriate for the teleprocessing network being documented. This record should be filled out as soon as the equipment configuration of your teleprocessing subsystem is known, and should subsequently be updated each time any changes are made to the configuration.

Please consult your IBM representative for the editions that are current and applicable.

CAUTION

Do not attempt to define a network control program (or emulation program) solely on the basis of information given in this publication.

This publication explains what functions the network control program (or emulation program) can perform, what teleprocessing network configurations are supported, and how to define and generate a program that will perform the functions your installation requires. This book does not contain the restrictions and programming considerations imposed by specific types of stations (terminals) or by other program components (such as CICS) with which the network control program may communicate.

Successfully defining a network control program (or emulation program) suitable for your installation requires that you consult the appropriate programming manuals for each of the program components and each of the types of teleprocessing equipment (controllers, stations) that make up your installation. Consult your IBM representative to determine the applicable publications that are available.

HOW TO USE THIS BOOK

This book contains 13 chapters, plus appendixes, grouped into seven parts as follows:

Part:	Chapter:
I. Introduction	1. Introduction
II. Defining Network Control Functions: SDLC Networks	2. Network Control Functions for SDLC Networks
	3. NCP Generation Macro Instructions for SDLC-Only Networks
III. Defining Network Control (and Emulation) Functions: BSC and/or Start-Stop Networks	4. Network Control (and Emulation) Functions for BSC and/or Start-Stop Networks
	5. NCP and PEP Generation Macro Instructions for BSC—Start-Stop—(SDLC) Networks
IV. Defining Emulation Functions: BSC and/or Start-Stop Networks	6. Emulation Functions for BSC and/or Start-Stop Networks
	7. EP Generation Macro Instructions for BSC and/or Start-Stop Networks
V. OS/VS Generation and Utilities	8. Program Generation under OS/VS
	9. Loader Utility under OS/VS
	10. Dump Utilities under OS/VS
VI. DOS/VS Generation and Utilities	11. Program Generation under DOS/VS
	12. Loader Utility under DOS/VS
	13. Dump Utilities under DOS/VS
VII. Appendixes	A. Types of Stations Supported by the IBM 3704 and 3705
	B. Network Control Program Generation Messages
	C. Utility Messages
	D. Coding Examples for Switched Lines and Multiple Terminal Access Operation
	E. Transmission Codes for World Trade Teletypewriter Terminals
	F. Required Coding Sequence for Program Generation Macro Instructions
	G. Multiple-Terminal-Access Sign-On Procedure for Terminal Operators
	H. Partial Program Generation
	I. Sample Network Control and Emulation Programs
	J. Procedure for Determining Line Interrupt Priorities
	K. Upper Scan Limits, Address Substitution, and High Speed Select Options
	L. Supplemental Information for Airlines Line Control Users

The foregoing arrangement, a guide to which appears on the two following pages, presents the content of this book in modular fashion to meet the needs of users having differing network configurations, network operating modes, and operating system utility requirements, as follows:

Chapters Used to Define a Program

The next page is a guide to the chapters needed to define a program that accommodates various combinations of network configuration and network operating mode:

Network Configuration:

- SDLC only (no BSC or start-stop stations)
- SDLC and non-SDLC (SDLC and BSC, or SDLC and start-stop, or SDLC, BSC, and start-stop)
- non-SDLC only (no SDLC stations)

Operating Mode for Non-SDLC Stations:

- Network Control Mode only
- Network Control and Emulation Modes
- Emulation Mode only

The seven valid combinations of the foregoing six options are graphically represented in the chart on the next page, *To Define a Program.* The chart shows, for each of the seven combinations, which chapters you should use (1) for descriptions of program functions, and (2) to define a program.

A related table, appearing on the foldout sheet at the end of the book, is an index to the program generation macro instructions applicable for each of the seven network configuration combinations.

Chapters Used for Generating, Loading and Dumping a Program

The second following page is a guide to the chapters needed to generate, load, and dump a network control or emulation program under OS/VS and under DOS/VS.

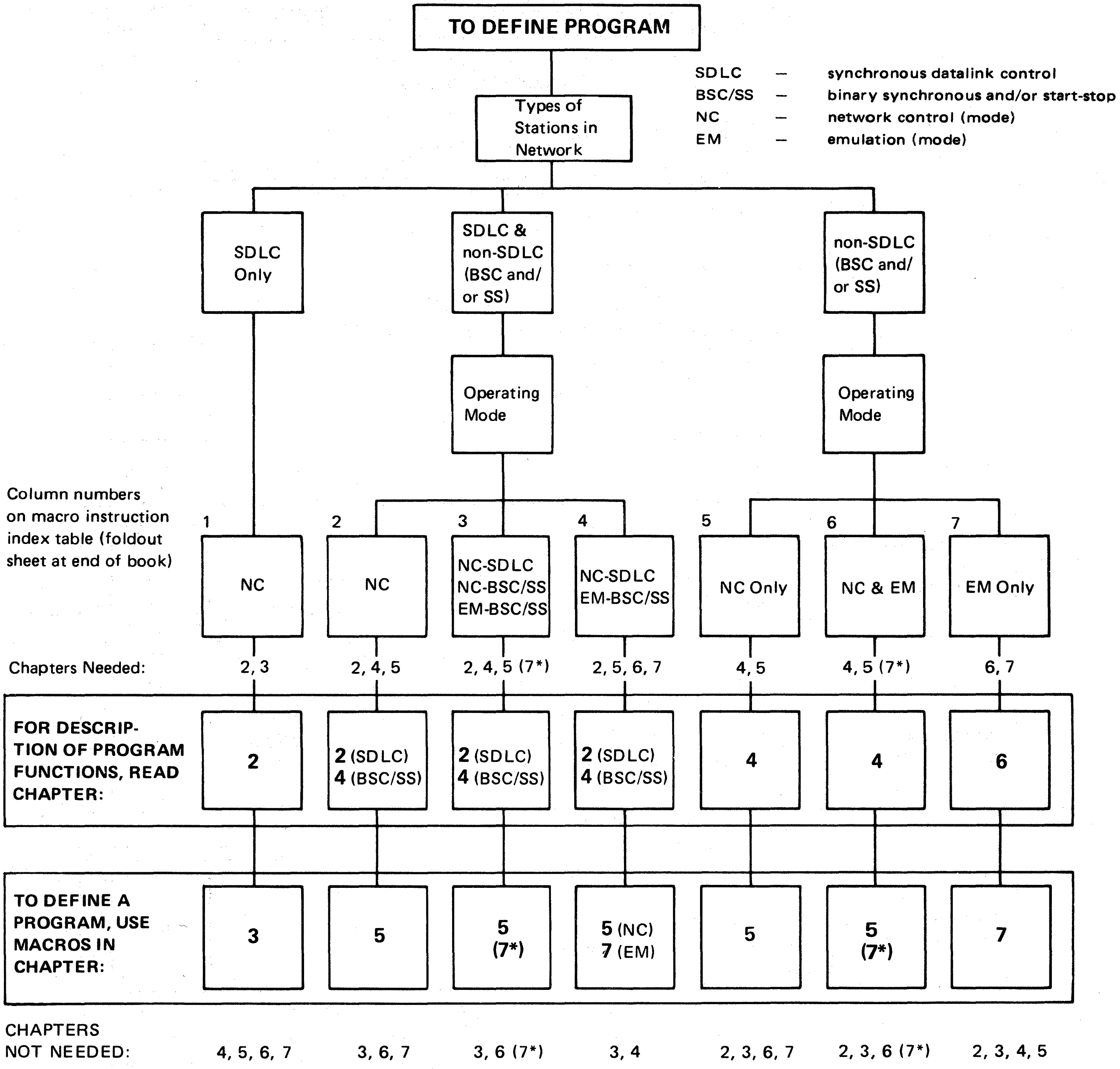
The coding conventions used in the macro instruction descriptions appear at the beginning of Chapters 3, 5, and 7.

Important: To define a network control program/VS that interacts properly with the access method in the host processor, you must be aware of the access method requirements imposed on the network control program. Refer to the *VTAM* or *TCAM System Programmer's Guides* mentioned in the Preface.

When defining a program that performs emulation functions, you should also be familiar with the IBM 2701, 2702, or 2703 transmission control units, as described in:

- *IBM 2701 Data Adapter Unit Component Description* (GA22-6864)
- *IBM System/360 Component Description: IBM 2702 Transmission Control* (GA22-6846)
- *IBM 2703 Transmission Control Component Description* (GA27-2703)

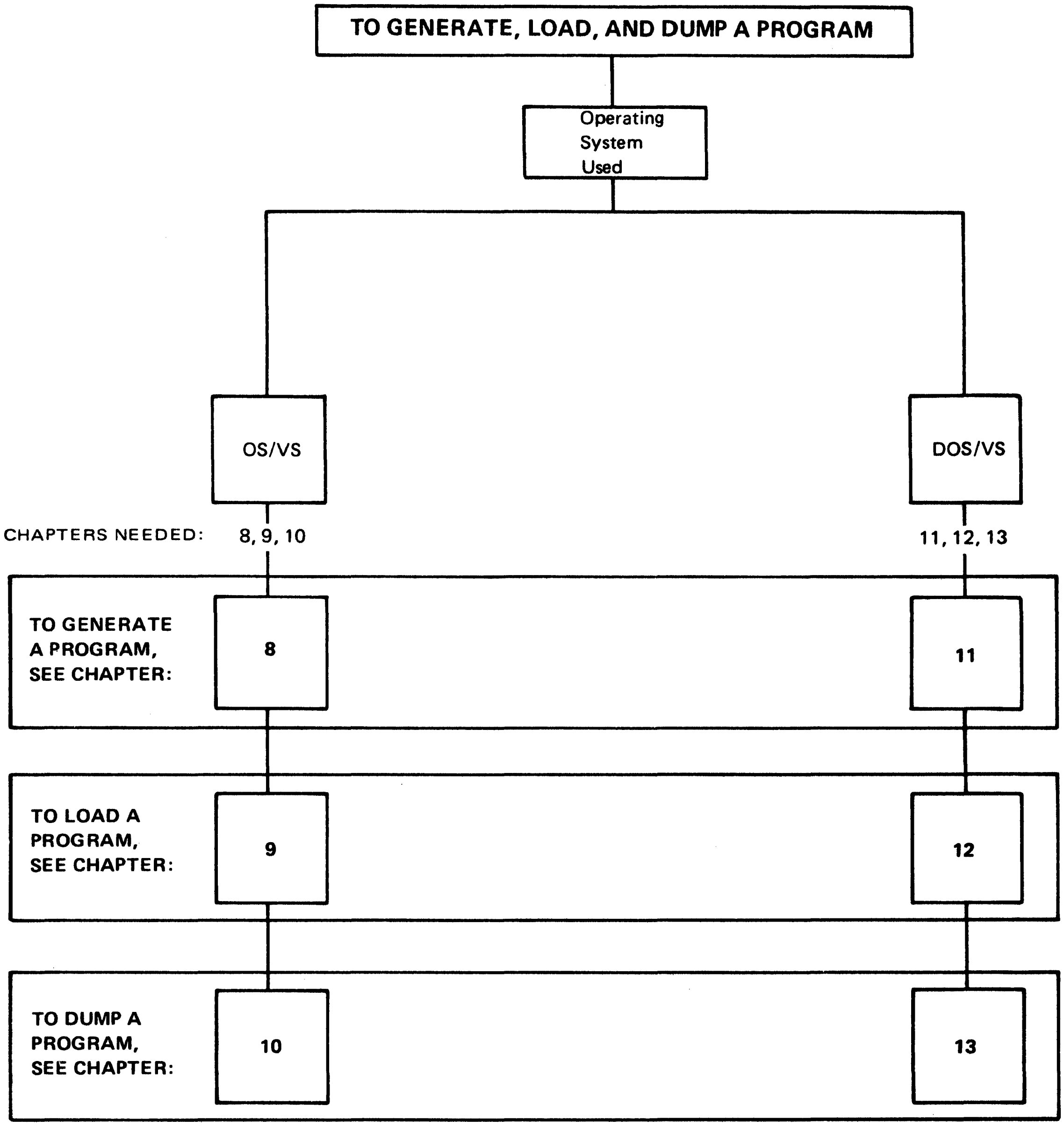
In addition, whether you are generating a program that performs network control functions or emulation functions, you should have available the component description manuals applicable to the types of terminals in your teleprocessing subsystem. Please consult your IBM representative for the manuals that are current and applicable.



*If you intend to define one or more BSC and/or start-stop lines or line groups exclusively in emulation mode, you may use the GROUP and LINE macro descriptions in Chapter 7 for defining such lines or groups. The macro descriptions in Chapter 7 include only the operands applicable to emulation mode.

Note: If you plan to use this book only to define programs corresponding to the path(s) you select above, you may remove the chapters listed directly above under the selected path(s) as you will not need these chapters. The elimination of extraneous material renders the information you need easier to find and use, and reduces the size of the book. If, for example, you plan to use this book only to define all-SDLC networks (represented by the leftmost path (column 1) above), you will not need and may therefore remove chapters 4 through 7. It is suggested that you file the removed chapters for possible later use, should your needs change.

SEE FOLDOUT INDEX AT END OF BOOK FOR GUIDE TO MACRO INSTRUCTIONS



Note: If you need only OS/VS information, you may remove Part VI (Chapters 11, 12, and 13). Conversely, if you need only DOS/VS information, you may remove Part V (Chapters 8, 9, and 10). It is suggested that you file the removed chapters for possible later use, should your needs change.

Summary of Amendments–Sixth Edition (GC30-3008-5)

This sixth edition applies to version 5 of the Network Control Program/VS and version 3 of the Emulation Program/VS.

This edition reflects the availability of OS/VS TCAM as an access method with which version 5 of the network control program/VS can directly communicate. (The previous edition reflected only OS/VS VTAM and DOS/VS VTAM as access methods for cummunicating with the network control program.)

Throughout this publication the term "access method" generally refers to whichever access method is used to communicate with the program in the communications controller. References to a specific access method appear where a network control program function is supported by only one access method or the NCP parameters required differ for different access methods.

In addition to making the foregoing changes, this edition:

- Adds the following new operands:

LNQTCNT	(LINE macro)
PARCHK	(LINE macro)
PECHAR	(GROUP macro)

- Makes corrections or clarifications to the following operands:

BFRPAD	(HOST macro)
LOCHAN	(BUILD macro)
MAXDATA	(PU, CLUSTER macros)
NRZI	(LINE macro)
UNITSZ	(HOST macro)

- Modifies the list of SYSCNTRL macro parameters.
- Adds guidelines for writing user block handling routines.
- Adds two messages (IFL515I and IFU515I) to Appendix C.

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

Summary of Amendments–Fifth Edition (GC30-3008-4)

The fifth edition:

- Adds the following operands to the macros indicated:

  BUILD: CA, OPCSB2
  LINE: BUFSIZE, NEGPOLP, PAUSE

- Adds information on the IBM 3705-II. This edition applies equally to the 3705-I and 3705-II, except where a distinction between the two is explicitly made.
- Adds information on the type 3 communication scanner and type 4 channel adapter for the IBM 3705.
- Adds a description of the multi-subchannel line access facility for channel operations in emulation mode.
- Clarifies the generation procedure for emulation programs, as described in Chapters 8 and 11.
- Adds Appendix J, which gives a method for determining the interrupt priority to be assigned to each line in the network.
- Adds Appendix K, which provides information on use of the upper scan limit, address substitution, and high speed select options as specified in the GENEND macro.
- Adds information on data set space provided if the UT1, UT2, or UT3 operand of the BUILD macro is omitted.
- States that the STATMOD operand is optional for release 33 (or higher) of DOS/VS and release 2 (or higher) of VTAM.
- Corrects and clarifies the description of the SERVLIM operand of the LINE macro.
- Corrects and clarifies the description of the MAXDATA and PASSLIM operands of the PU; CLUSTER, and INNODE operands.
- Adds and updates messages in Appendixes B and C.

Contents

PART I. INTRODUCTION

PART II. DEFINING NETWORK CONTROL FUNCTIONS: SDLC NETWORKS

Chapter 5: NCP Generation Macro Instructions for BSC–Start-Stop–(SDLC)

PART VI. DOS/VS GENERATION AND UTILITIES

PART VII. APPENDIXES

Figures

Part I.
Introduction

Chapter 1: Introduction

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

Control programs for the communications controller are first defined in the form of macro instruction 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/VS

The network control program/VS (hereafter referred to in this book as the network control program) controls the transmission of data between the host processor and the remote stations in the teleprocessing network connected to the communications controller. (The *host processor* is the central processing unit (CPU) in which is executed the access method that interacts with the network control program.)

A network control program/VS can execute two broad categories of functions: network control functions and emulation functions.

Network Control Functions

Network control functions include the wide range of capabilities for which the 3704 and 3705 controllers are principally designed, as follows.

The program recognizes and fulfills requests by the teleprocessing access method to transmit data to and receive data from the network. In so doing, the program performs whatever operations are needed to establish communication with stations (these operations include polling, addressing, dialing, and answering), as appropriate for the type of station and type of communication line linking the station and the controller. Then the program receives message data into buffers, inserts and deletes transmission control characters as required, and translates message data from processing code (EBCDIC) into transmission code, and vice versa. Finally, the network control program transmits the data from the buffers to the access method or to the stations in the network.

The network control program also governs many aspects of communication between the network and the host processor, such as the amount of data to be accepted from a station once connection is established, the number of devices on a multipoint line with which the access method can communicate concurrently, and exchange of identification sequences with stations on switched lines.

Network control functions also include: (1) automatic error recovery and statistical recording, (2) diagnosis of controller, line, and station malfunctions, and (3) changes to operating parameters during program execution upon request from the host processor.

Emulation Functions

Emulation functions comprise a more restricted range of functions, equivalent to those provided by the IBM 2701, 2702, and 2703 (collectively referred to in this book as *transmission control units*). These emulation functions permit most existing user application programs to operate unchanged when a 3704 or 3705 supplants one or more of the transmission control units mentioned. These functions are described below under *The Emulation Program/VS.*

You may generate a program that performs only network control functions or only emulation functions, or both, according to the needs of your teleprocessing installation. A program capable of both categories of functions is called a network control program with the partitioned emulation programming (PEP) extension. When generating a program having the PEP extension, you specify, for each communication line in the network, whether that line is to operate in network control mode or emulation mode, or both. Operation in network control mode means that all of the network control functions apply to data transmission over that line. Operation in emulation mode means that only the emulation functions are performed for that line. These are equivalent to the functions performed by the 2701, 2702, or 2703 to which the line was formerly attached. Because the functions performed by the three types of transmission control units differ in some respects, you specify for each line which of these units is to be emulated.

If you specify operation in both modes, operation can be changed from one to the other whenever desired by command from the access method that communicates with the network control program.

The Emulation Program/VS

The emulation program/VS (hereafter referred to as the emulation program) and the partitioned emulation program (PEP) extension of the network control program/VS allow many programs written for support of the IBM 2701, 2702, and 2703 transmission control units to operate with the IBM 3704 and 3705 controllers with no modification. These programs include IBM Type I access methods that support the 2701, 2702, and 2703, as well as IBM Type II and Type III programs and user-written programs that interface with the 2701, 2702, and 2703 in a manner equivalent to Type I access method programs. Programs that involve timing dependencies and support of certain special and custom features may, however, require modification.

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

The emulation program, in conjunction with the type 1 or type 4 channel adapter, permits the use of the same control sequences and data transfers as do the 2701, 2702, and 2703. It also provides most of the standard functions of these control units. Not supported are the parallel data adapter, synchronous data adapter type 1, programmable two-processor switch, 230,400 bps synchronous speed, direct attachment of the IBM 1032 Digital Time Unit, the IBM 2712 Remote Multiplexer attachment features, and the reverse channel feature. ASCII transparency is supported only for a communication line serviced by a type 3 communication scanner. (*Exception:* Programming RPQs [PRPQ] are available that permit the type 1 and type 2 scanners to accommodate ASCII transparency and six-bit transcode.)

Communication between Controller and Host Processor

The network control or emulation program interacts with one or more access methods executing in the host processor. The access method and any associated application programs must be designed to interact with the control program in the manner appropriate to the mode in which the lines served by the access method are to operate—that is, in network control mode or emulation mode, or both.

Communication in network control mode between the network control program and the access method generally consists of an exchange of requests issued by the access method and responses returned by the network control program. Each request and each response contains the control information necessary to identify the teleprocessing resource to which it applies, the operation required (for example, read or write), and status information pertaining to that operation. Requests and responses also contain the text of messages to be passed between the access method and the network. Some responses from the network control program are unsolicited; that is, not returned in response to a request. Unsolicited responses report error conditions and status information that may develop during operation of the controller.

All requests and all responses for lines operating in network control mode pass between controller and host processor over the network control subchannel. This subchannel is represented by a subchannel address on the CPU channel to which the controller is attached. There is always one network control subchannel regardless of how many lines are operated in network control mode. If the controller is to perform only network control functions, this is the only subchannel required. Over the network control subchannel also passes load module data sent by the access method loader or independent loader, and the contents of controller storage during the dumping process.

Communication in emulation mode between the access method and the network control program is essentially the same as between the access method and the transmission control unit being emulated. As is the case for transmission control units, each communication line operated in emulation mode requires its own CPU subchannel address. The subchannels associated with lines operated in this mode are called emulation subchannels. In defining a program that performs emulation functions, you associate each communication line with an emulation subchannel address. Use of the multi-subchannel line access (MSLA) feature of the program allows more than one subchannel address to be associated with a line, but communication is possible over only one subchannel at a time. (This feature is described in Chapters 4 and 6.) The converse, however, is not true: more than one line cannot be associated with a single subchannel address. (An exception is a program option for emulation of a 2701 that is equivalent to the dual communications interface feature of the 2701. This option allows either of two lines to be associated alternately with a single subchannel.)

For either the MSLA or dual communication interface facility, the line with which a subchannel is to communicate is established by command from the access method. Selection is possible only among the line/subchannel associations established when the program was defined. The source program must be modified and reassembled to change these associations.

A program that performs both network control and emulation functions requires (1) a single network control subchannel and (2) one or more emulation subchannels for each line to be operated in emulation mode (even if such a line will sometimes operate in network control mode). A program that is to perform only emulation functions still requires a network control subchannel (also called a "native" subchannel); however, this subchannel is used only for loading the network control program into the controller and for dumping the contents of controller storage.

This book applies to version 5 of the network control program/VS—that is, the version designed to communicate, in network control mode, with VTAM or TCAM—and to version 3 of the emulation program/VS. (Earlier editions of this book apply to previous versions of these programs.) Control program operations in emulation mode do not communicate with VTAM, but rather with TCAM, BTAM, or QTAM. Figure 1-1 illustrates the data flow relationships that can be established between the host processor and local and remote communications controllers.

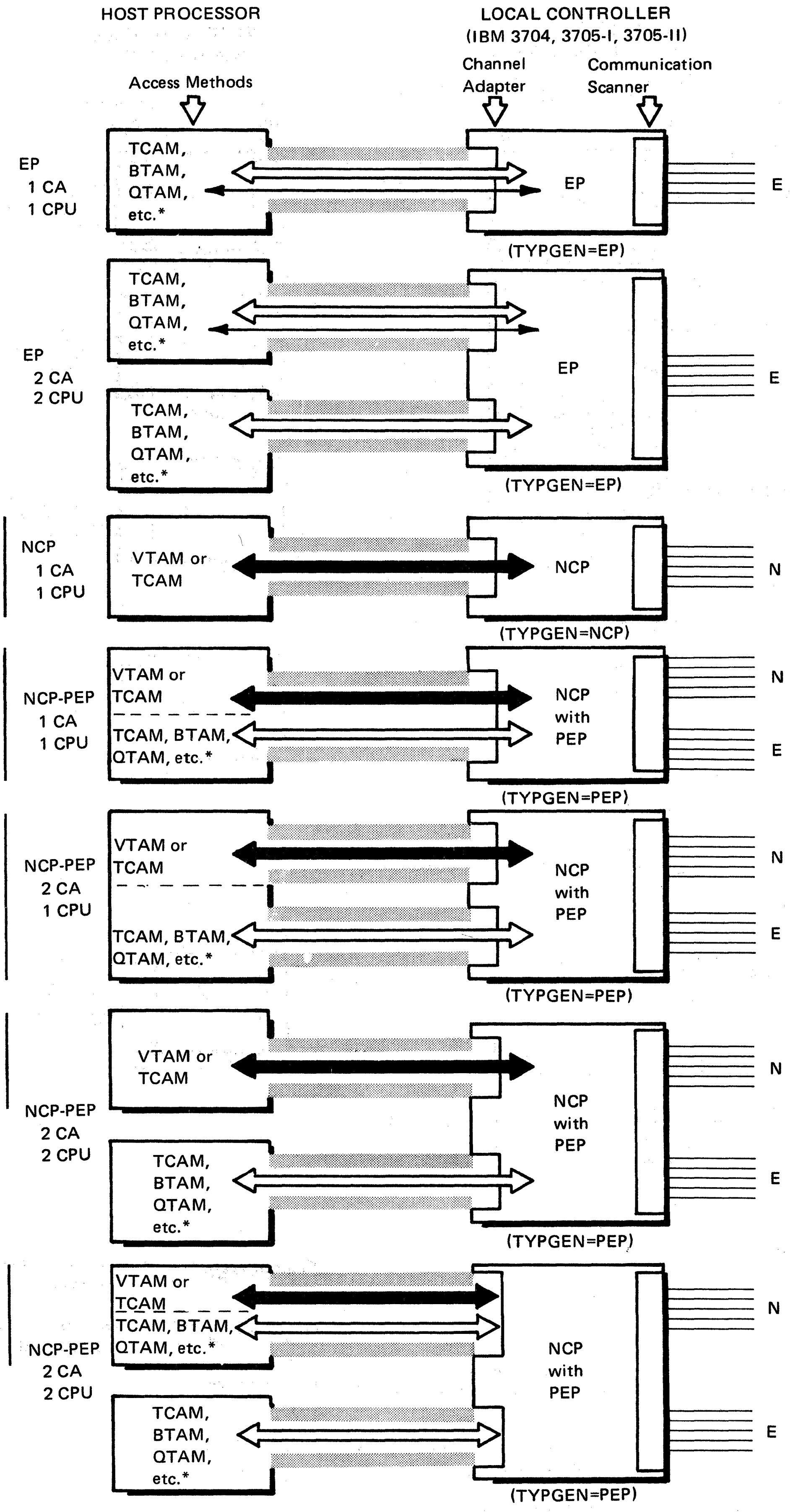
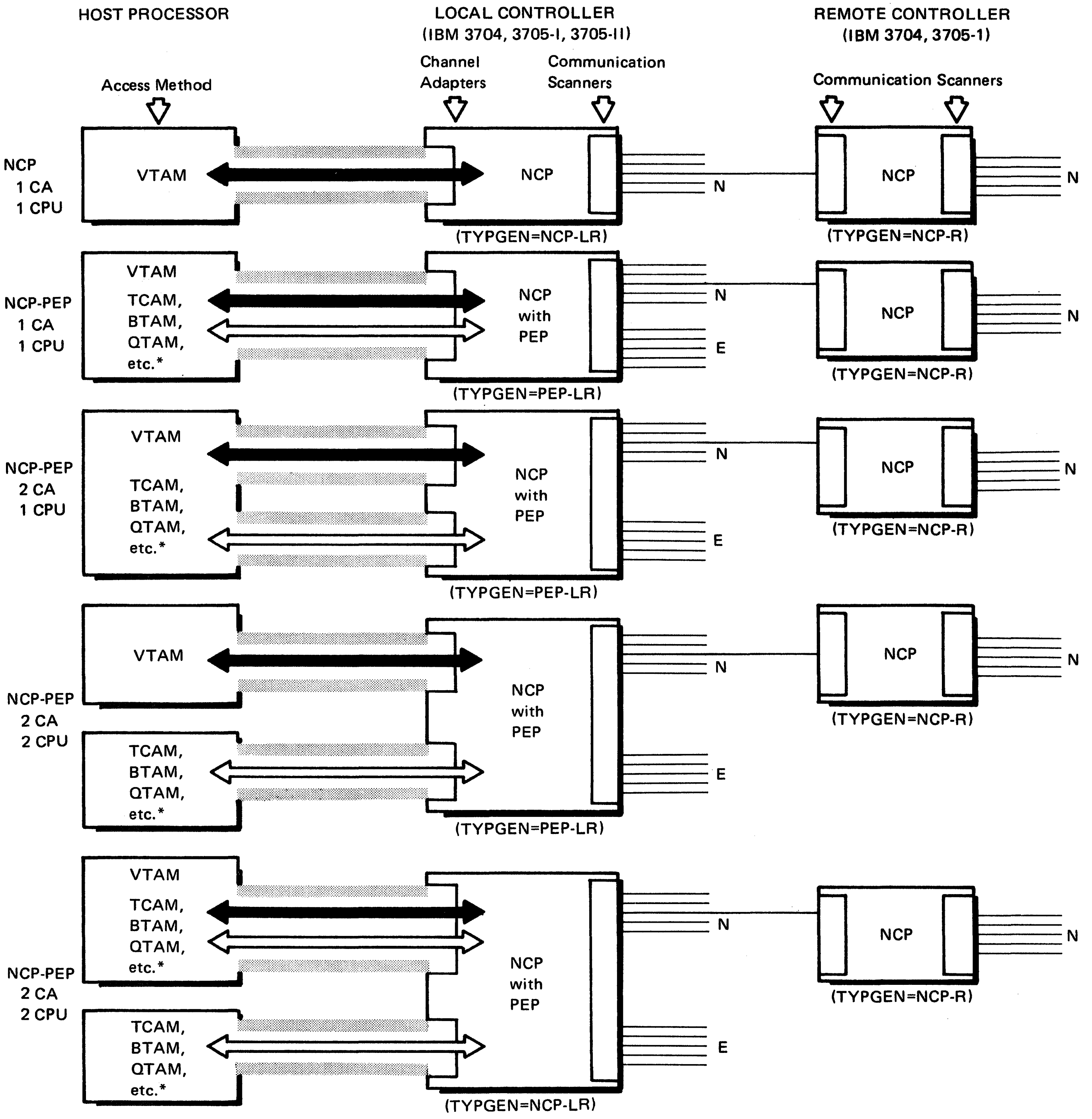


Figure 1-1. Host Processor/Communications Controller Data Flow Relationships (1 of 2)



*Other access methods and program products designed to communicate with the communications controller in emulation mode may be used (for example, HASP, JES, CICS, IMS).

Channel ▒

Message flow over channel:
- ⬌ in network control mode (single subchannel)
- ⇔ in emulation mode (multiple subchannels)

Native/network control subchannel: (used for program loading/dumping) ↔

Lines in
- network control mode ☰ N
- emulation mode ☰ E

Figure 1-1. Host Processor/Communications Controller Data Flow Relationships (2 of 2)

Defining the Network Control or Emulation Program

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

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

Chapter 4 describes each of the characteristics of the teleprocessing subsystem to be considered in defining a network control program that is to perform network control functions, or both network control and emulation functions for BSC and/or start-stop stations. Chapter 2 provides equivalent information for SDLC stations (network control functions only). Some of these characteristics are generally applicable to most teleprocessing subsystems. Other characteristics pertain to the capabilities and control techniques distinctive to the 3704 and 3705 communications controllers and the network control program.

Chapter 6 is for the convenience of the reader who wishes to define a program that performs only emulation functions or to define a line or group of lines that operate only in emulation mode. This chapter contains the same information about emulation functions as Chapter 4 does, but omits the large amount of information in Chapter 4 that pertains only to network control operation.

Once you understand the characteristics explained in Chapters 2, 4, and/or 6, as determined by your requirements, you may undertake the task of defining a network control program by referring to Chapters 3, 5, or 7. Chapter 5 contains the macro instructions required to define a BSC and/or start-stop network that is to operate in network control mode or in both network control and emulation modes. Chapter 3 contains the same information for SDLC stations as does Chapter 5, but omits BSC and start-stop stations and is thus more convenient for the reader wishing to define an SDLC-only network. Chapter 7 contains the information needed to define an emulation program for BSC and/or start-stop networks. You may also use the GROUP and LINE macro descriptions in Chapter 7 for defining a line or line group within a network control program (with the PEP extension) that is to operate only in emulation mode.

Note: See the *Caution* following the Preface before attempting to define a network control program.

Generating the Network Control or Emulation Program

After the network control or emulation program is defined in the form of a source program containing control program generation macro statements, it is ready to be generated. This is a compilation procedure consisting of assembly and link-editing steps. 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 link-editing requirements and (2) has access to the IBM-supplied network control program module libraries. These libraries, supplied by the IBM Program Information Department, must be added to the operating system before any network control or emulation programs can be generated. Documentation provided with the libraries explains how to add them to the operating system.

The primary output of the generation procedure is a network or emulation control program load module, ready for loading into the communications controller. Chapters 8 and 11 describe the generation procedure under OS/VS and DOS/VS, respectively.

The VTAM Initialization Process

The VTAM initialization process requires as input the same control program source statements as used for NCP generation, supplemented by several other source statements meaningful only to VTAM. These statements, which consist of separate macros and additional operands of existing NCP macros, may be placed in the network control program source deck either before or after the NCP is generated. Placing the VTAM-only statements in the deck before generating the network control program is recommended, however, because adding them to the deck after generating the program can result in introducing inadverent errors (such as misspelled operands, transposed cards) that would cause the information given to VTAM initialization procedure to differ from the generated network control program.

Chapters 3 and 5 list these VTAM-only macros and operands, and indicate where they must appear in the NCP source deck used for VTAM initialization. However, their use is not explained; the *VTAM System Programmer's Guide* tells how to use these VTAM-only source statements. The program generation procedure checks only the keyword part of VTAM-only operands for proper spelling. No check is made on the appropriateness or accuracy of the parameters specified, and no verification is made of the appropriateness of the operands coded.

Caution: Because the VTAM initialization does no validity checking of NCP parameters validity checked by the NCP generation procedure, it is imperative that the NCP source statements be entirely free of errors before being given to the VTAM initialization procedure. Therefore, the network control program must be assembled, via stage one of the generation procedure, and reassembled if necessary, until the stage one output listing shows no MNOTE statements having severity codes of 4 or 8.

Loading the Network Control or Emulation Program

The final step in achieving an operating network control or emulation program is loading the program load module into the communications controller. For a local communications controller, this requires that a loader utility program be executed in the host processor, with the controller on-line to the processor. For a remote communications controller, loading requires that a loader utility program be executed in the host processor and that a network control program be executing in the local controller to which the remote unit is connected. Apart from transferring load module data between the host processor and the remote controller, the program in the local controller does not participate in the loading process.

The loader utility program executed in the host processor may be an access method (VTAM or TCAM) facility or an independent utility program provided by IBM as part of the system support programs. The independent utility, use of which is explained in Chapters 9 (OS/VS) and 12 (DOS/VS), or the TCAM facility, may be used only for loading a local controller. The VTAM loader facility *must* be used for loading a remote communications controller and *may* be used for loading a local controller. See the appropriate VTAM or TCAM publications, of those listed in the Preface, for information on the access method loading facility.

Obtaining the Contents of Controller Storage

A utility called the dump program allows a selected portion or all of the contents of the controller storage (whether local or remote) to be transferred from the controller to the host processor, which then prints the contents in hexadecimal format. The dump program, like the loader program, may be an access method facility or an independent program supplied as part of the system support programs. Either dump program has two modules, one of which the host processor transfers to the controller before the dumping process begins. The two modules then interact to transfer the contents of controller storage to the host processor; the host processor module then formats and prints the storage contents.

Executing the dump utility ~~stops~~ operation of the network ~~control or emulation~~ program; after the dumping process is completed a control program must be reloaded into the controller before teleprocessing operations can resume. (An alternate means of obtaining the storage contents that does not require stopping the control program is explained below under *The Dynamic Dump Utility*. This utility is available only in an emulation program or a network control program with the PEP extension.)

The storage contents of a remote communications controller can be obtained only with the access method facility; the storage contents of a local controller can be obtained with either the access method facility or the independent utility program.

Chapters 10 (for OS/VS) and 13 (for DOS/VS) explain how to use the independent dump utility.

The Dynamic Dump Utility

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

Chapters 10 (for OS/VS) and 13 (for DOS/VS) explain how to use the dynamic dump utility program.

Part II.
Defining Network Control Functions: SDLC Networks

This Part (Chapters 2 and 3) is of interest only to users needing to define network control programs for SDLC networks. You may wish to remove and file one or both of these chapters if unneeded—see *How to Use this Book* and the associated chart *To Define a Program* at the front of this book.

Chapter 2: Network Control Functions for SDLC Networks

Described in this chapter are the many aspects of an SDLC teleprocessing subsystem that you must identify to the network control program to tailor it to your particular teleprocessing installation. You should read this chapter if you wish to define a program capable of operating SDLC links.

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

- The stations and lines of the SDLC network
- The communications controller hardware configuration
- Data transfer between the communications controller and the host processor
- Procedural options governing message traffic between the controller and the network

The remaining two major sections explain the optional diagnostic and service aid facilities that may be included in the network control program, and the program generation options and data sets (files) that the generation procedure will use in creating a network control program load module. The description of each characteristic and option is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the program generation macro instructions.

For many characteristics, especially those relating to the equipment configuration, the decisions about what to code in the macro instructions 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 pacing. Such characteristics, which affect the message-handling capacity and throughput of the teleprocessing subsystem, require careful consideration before specifying the corresponding parameters in the program generation macro instructions.

Once you are familiar with those characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions that define the network control program. At that point you should go on to Chapter 3 (macro instructions for SDLC-only networks) or Chapter 5 (for combined BSC/start-stop/SDLC networks).

SDLC Network Characteristics

This section applies only to SDLC network operation. See the section *BSC and Start-Stop Network Characteristics* in Chapter 4 for information on BSC and start-stop network operation.

Physical and Logical Units

In this book, *SDLC station* refers to any of the units with which the network control program can communicate using the synchronous data link control scheme, such as the IBM 3601 and 3650 control units (called cluster controllers), IBM 3767 and 3770-series terminals, and 3704 and 3705 communications controllers.

To the network control program (and to the access method), each SDLC station appears as a *physical unit.* A physical unit, which constitutes the logical "appearance" of the station to the rest of the network, represents a specific set of defined functions performed by programming or by hardware, depending on the type of physical unit. Associated with certain physical units may be one or more *logical units*–application programs within the SDLC station.

Each physical unit is represented within the network control program by a PU macro, which identifies it as a type 1, type 2, or type 4 physical unit. These type designations are in lieu of specific, numeric machine types. For example, IBM 3270 and 3767 SDLC terminals are both type 1 physical units. The type designations reflect the degree of program function performed by the physical unit. Functionally equivalent physical units bear the same type designation.

Other parameters associated with the physcial unit are the station address, the subarea address assigned to the unit (communications controllers [type 4 physcial units] only), and the procedural options that govern communication between the network control program and the physical unit. The procedural options are described under *Procedural Options*, later in this chapter.

Note: In version 3 of the network control program, physical units are represented by CLUSTER or INNODE macros. The CLUSTER macro is equivalent to a PU macro for a type 2 physical unit, and the INNODE macro is equivalent to a PU macro for a type 4 physical unit (that is, a communications controller). All SDLC stations added to an existing (version 3) source program that is to be re-generated under version 5 should be represented by PU macros. For compatibility with the previous version, version 5 continues to accept CLUSTER and INNODE macros; however, any subsequent changes or additions to physical unit parameters will be reflected only in the PU macro.

The logical units associated with an SDLC physical unit are defined by LU macros, if the physical unit is attached to a nonswitched link, and the logical unit parameters are defined in these macros. If, on the other hand, the physical unit is reached over switched communication facilities (VTAM only), the network control program allocates its logical units dynamically from a pool of logical units. A single LUPOOL macro defines the pool from which all logical units for switched SDLC links are obtained. The logical unit parameters in this case are defined and maintained in VTAM, rather than in the network control program. Upon establishing a session with an SDLC station, VTAM must pass the appropriate logical unit parameters to the network control program, which inserts them in the logical units it has allocated for that session. Thus initialized, these NCP logical units represent the logical units in the station for the duration of the session. When the session ends, the program returns the allocated NCP logical units to the logical unit pool for re-use in other sessions.

Communication Line Characteristics

A *communication line* as used in this book includes the entire transmission link between a station and the communications controller, including the modems (data sets), regardless of the actual transmission medium—physical conductors (wire), microwave links, satellite links, etc., or a combination of these media. Communication lines over which synchronous data link control procedures are used are called *SDLC links.*

Line characteristics refer to the functional attributes of the transmission path, for example, whether the communication facility is half-duplex or duplex; and to related aspects of the line such as the physical address, within the communications controller, to which it is attached.

SDLC cluster controllers may communicate with a local or remote communications controller over a nonswitched point-to-point link, or a nonswitched multipoint link, or over switched point-to-point facilities. Each SDLC link must be represented within the network control program by a LINE macro. This macro specifies to the program certain characteristics of the link.

A multipoint line discipline is used for all SDLC links attached to a communications controller, regardless of the number of stations on the link. The controller contacts a specific station by sending a control character (physical address) assigned to and recognized only by that station. The one station receiving that character responds appropriately; the other stations ignore the character.

Specify an SDLC link as requiring a multipoint discipline by coding POLLED=YES in the LINE macro.

The network control program requires a service order table for each SDLC link for which the controller in which the program is executed is the primary station. (The station that controls an SDLC link is the *primary station* for that link; the other station is the *secondary station.*) The service order table contains one or more entries for each SDLC station attached to the link.

A service order table is defined by a SERVICE macro. Code a SERVICE macro directly following each LINE macro that represents an SDLC link for which the controller is the primary station.

Half-Duplex vs. Duplex Links

The network control program must know whether a communication facility. is half-duplex or duplex (sometimes called full-duplex). You specify this in the DUPLEX operand of the LINE macro representing the line. This operand represents the characteristics of the entire communications path including common-carrier lines and equipment, and the modems at both ends of the path. The operand does *not* specify the mode of data transfer over the line. (It is important not to assume that a two-wire modem is necessarily a half-duplex modem; some modems are in fact duplex. In general, if the "clear-to-send" signal lead in the modem is continuously energized, the modem is duplex, regardless of whether it is a two-wire or four-wire modem. If in doubt, consult the installer or supplier of the modem.)

Line Speeds and Clocking

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

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

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

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.

Line Addresses

Each SDLC link attached to the communications controller is identified to the network control program by one or two physical line addresses representing the physical location in the controller at which the link is attached (via line set and line interface base [LIB]).

If a single line address is used for both transmitting and receiving over the SDLC link, specify that address in the ADDRESS operand of the LINE macro representing the link. If separate line addresses are used for transmitting and receiving, specify both addresses in the ADDRESS operand of the LINE macro.

Modem New Sync Feature

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

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

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

Effect of NRZI vs. NRZ Bit Stream Encoding

The network control program transmits data over an SDLC link in either "non-return-to-zero" (NRZ) mode or "non-return-to-zero-inverted" (NRZI) mode. This program option is specified in the NRZI operand of the LINE macro that represents the SDLC link. The choice of option is determined solely by the effect on the modems serving the link, as follows.

The modems at each end of the link must maintain synchronism with each other for the entire duration of message transmission. Some modems require bit transitions (that is, 0 to 1 or 1 to 0) at intervals in the data stream in order to maintain synchronism. (Such modems are said to be sensitive to transitionless bit streams.) When operating in NRZI mode, the data terminal equipment at the ends of the link manipulate the bit stream transferred to the modems in such a way that transitions are introduced into the bit stream even when the message data being transmitted is transitionless (that is, consists of sequences of repeated binary 0's [hexadecimal 00]). (Such binary sequences are likely to occur in messages containing storage dump data or IPL data being sent to programmable controllers or terminals.) The transitions thus introduced ensure that the modems remain in synchronism. The terminal equipment that receives the altered bit stream reconverts it to its original form.

In NRZI mode, sequences of zeros (000000...) are converted to alternating ones and zeros (101010...), thus satisfying those modems sensitive to transitionless bit streams. Certain other modems, however, are sensitive to just this alternating pattern (101010...) rather than to transitionless bit streams; they will lose synchronism if subjected to 101010 patterns of sufficient length. When the SDLC link is equipped with such modems, the terminal equipment must operate in NRZ mode rather than in NRZI mode.

The rules for selecting NRZI vs. NRZ operation are as follows:

- If internal (business machine) clocking is used on an SDLC link, NRZI operation is required; specify NRZI=YES in the LINE macro representing the link (or omit the operand).
- If external (modem) clocking is used on the SDLC link, NRZI operation is required (NRZI=YES) *unless* the modems on the link are sensitive to repeated 101010 patterns, in which case specify NRZI=NO in the LINE macro. Consult your IBM representative (for IBM modems) or the modem supplier or installer (for non-IBM modems) to determine whether the modems are sensitive to repeated 101010 bit patterns.

Caution: All business machine equipment (terminal equipment) on the same SDLC link must use the same encoding scheme—that is, all use NRZI mode (NRZI=YES) or all use NRZ mode (NRZI=NO). Mixing of modes on the same SDLC link will result in total lack of communication between stations on the link. Where the stations on the SDLC link are remote 3704 or 3705 controllers, the setting of the NRZI bit in the IPL configuration data set of the remote program loader must correspond to the program-specified option (NRZI=YES or NRZI=NO).

Communications Controller Hardware Configuration

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 installed in the controller, (2) the type and number of channel adapters that join the communications controller to the host processor(s), (3) the type, number, and oscillator bit rates of the communication scanners installed, and (4) the interrupt priority to be used for each line serviced by a scanner. This information may be learned from the system designer.

Specify the storage size in the MEMSIZE operand of the BUILD macro. Also specify the type and number of channel adapters in the CA and CHANTYP operands of the BUILD macro. (Omit CHANTYP if the controller is a 3704.)

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

For each scanner, you must specify to the network control program (1) the type of scanner, (2) the machine module in which it is installed, and (3) the bit rates of the oscillators with which each scanner is equipped. This information, like the storage size and channel information, should be obtained from the system designer before you code the program generation macro instructions. Specify the details of the scanners in the TYPE, MOD, and SPEED operands of a CSB macro—one macro for each scanner in the controller.

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

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

Appendix J gives a method for determining the interrupt priority for each line in the network.

Communication Between Controller and Host Processor

Information on both the buffers within the access method and buffers within the network control program must be specified to the network control program in order for it to properly coordinate data transfers to and from the access method.

Data Transfer from Host Processor to Controller

The amount of data conveyed from the host processor to the communications controller during a single data transfer operation over the network control subchannel may vary over a wide range, depending on the number of requests and the amount of accompanying message data to be transferred. Efficient operation of the network control program requires that the program preallocate a suitable number of buffers for incoming data transfers, rather than allot buffers one at a time. Once the set of buffers is allocated, data transfer from the access method can proceed without further attention by the network control program's supervisory routine until the data transfer ends or all the preallocated buffers are filled. If the amount of data received during one transfer is insufficient to fill all of the preallocated buffers, the remaining buffers are used for subsequent data transfers until all are filled, at which point the program allocates the same number of buffers again.

The INBFRS operand of the HOST macro specifies the number of buffers the network control program is to allocate for data transfers over the network control subchannel from the access method. You should consider two factors when estimating a value for INBFRS.

If the size of a data transfer consistently exceeds the preallocated buffer space, the network control program's supervisory routine is frequently interrupted to provide more buffers for the excess data. The time the program must spend in processing the interrupts reduces the time it can devote to servicing communication lines.

On the other hand, preallocating an excessive quantity of buffers for receiving messages from the access method may deplete the buffer pool to the point that insufficient buffers are available for receiving messages over the communication lines. Buffer depletion is especially likely when (1) the buffer pool is relatively small and (2) a low message rate over the channel from the access method causes the preallocated buffers to be filled slowly, thus unduly delaying return of these buffers to the pool.

In choosing a value for INBFRS, then, strike a reasonable balance between degraded network control program efficiency due to excessive time spent processing interrupts for allocating more buffers, and unnecessary over-allocation of buffers.

Data Transfer from Controller to Host Processor

There is a limit to the amount of data the access method can receive from the network control program during a single data transfer over the network control subchannel. This limit must be specified when defining the network control program so that the program does not attempt to send more than the access method can accept. You specify this limit with the MAXBFRU and UNITSZ operands of the HOST macro. MAXBFRU designates the number of buffer units the access method allocates for a data transfer, and UNITSZ indicates the size of each unit in bytes. The total access method buffer space available is the product of the two values. (A buffer unit is the smallest amount of contiguous storage area handled as buffer space; a buffer may consist of one or more units.)

In sending a series of response (or request) blocks to the host processor, the network control program causes the access method to begin receiving each successive block 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 block sent to the host processor to be offset from the beginning of the access method buffer by enough space to allow the access method to insert the prefix. The amount of offset is specified in the BFRPAD operand of the HOST macro. OS/VS VTAM requires 28 bytes for the buffer pads; DOS/VS VTAM requires 15; OS/VS TCAM requires a minimum of 17 bytes.

Procedural Options

Several procedural options characterize the operation of SDLC links. These options include (1) the manner in which the program starts up and shuts down the teleprocessing network, (2) the amount of data to be transferred at one time between stations and the controller, and (3) the amount of data to be accumulated from a station before passing it to the access method.

Some procedural 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 network control program buffers or the maximum amount of data to be transferred at one time between the network control program and stations in the network.

Some options require relatively little forethought before you decide what to specify. Others require that you give considerable attention to the effect of your choice on message throughput and response time, among other factors.

Defining Buffer Size

The network control program contains one buffer pool of fixed-size buffers. Buffers from this pool are used for all message data transmitted over the network control subchannel and over lines operating in network control mode. In the BFRS operand of the BUILD macro you specify the size that you wish the buffers to be. (The minimum is 48 (44, if on-line testing is omitted [OLT=NO is specified in the BUILD macro]); the maximum is 248. The size is always a multiple of four bytes.) A buffer initialization process occurs immediately after the network control program is loaded into the communications controller. In this process the network control program formats into buffers all controller storage space remaining after the program is loaded. The remaining space, divided by the buffer size you have specified plus four bytes (for buffer chaining fields), yields the number of buffers in the pool.

Path Information Units

The basic unit of transmission in the teleprocessing network is the *path information unit* (PIU), which consists of network control and routing information and accompanying message text (optional). A PIU either requests a particular teleprocessing operation (request PIU) or indicates the result of an operation (response PIU). Path information units associated with SDLC physical units can originate at either the host processor or the physical unit. PIUs associated with BSC or start-stop stations always originate at the host processor.

As directed by the access method, the network control program establishes physical and logical connections between the access method and the SDLC stations in the network. While a connection is established, the network control program automatically controls the operation of each SDLC link in response to the data transfer and control operations specified in the request PIUs.

Several parameters govern the amount of data the network control program sends to a station on an SDLC link.

The size of the buffer within an SDLC physical unit must exceed by at least five bytes (for a type 1 physical unit) or nine bytes (for a type 2 physical unit) the size of a network control program buffer. In the MAXDATA operand of the PU macro representing a physical unit (type 1 or 2), you specify the maximum amount of data, in bytes, that the physical unit can hold.

Two parameters determine the maximum number of PIUs sent to an SDLC station: the *maximum outstanding* parameter and the *pass limit* parameter. Upon detecting a transmission error in a PIU it has received, the receiving unit indicates the fact in the next PIU it sends to the transmitting unit. The transmitting unit then retransmits the PIU in error and all subsequent PIUs it sent to that unit before receiving the error indication. The *maximum outstanding* parameter allows you to specify how many PIUs or PIU segments (up to seven) can be outstanding at any given time (that is, the number that can have been sent to the SDLC station before a response is received from the station). The higher the value of this parameter, the greater the degree of overlap between sending and receiving operations over the SDLC link, hence the greater the utilization of the link. However, a high maximum outstanding value also results in more PIU retransmission when an error occurs, because not only the PIU or segment in error but all succeeding ones as well are retransmitted. Thus, for links on which high error rates are experienced, you may wish to select a maximum outstanding value lower than that used for links having lower error rates. The maximum number of PIUs or PIU segments that can ever be outstanding is seven because of the internal sequence numbering scheme used to identify PIUs. Unless you specify a different value in the MAXOUT operand of the PU macro that represents the station, only one PIU is sent to each station before a response is required.

You may wish to cause some stations on an SDLC link to be serviced more frequently than others by representing them more times in the service order table for the link. By thus allowing more opportunities for contact with a particular station, relative to others, the total amount of data transferred to the station may be similarly increased. You may counter this effect, however, by using the *pass limit* parameter to restrict the amount of data (number of PIUs) exchanged with the station for any one appearance of the station in the service order table.

Only one PIU per pass is sent unless you specify a larger value in the PASSLIM operand of the PU macro representing the station.

Caution: When choosing values for MAXOUT and PASSLIM, observe any restrictions imposed on these operands by specific types of SDLC stations. See the appropriate programming publications for the types of stations used in your network for such restrictions.

Pacing

The pacing option is a means by which the network control program sends a limited number of path information units to a logical unit on an SDLC link before requiring an acknowledgment from the logical unit that it is able to receive more PIUs. Use of this option can prevent needless transmission of PIUs to a logical unit that is momentarily unable to accept them. (A similar option in VTAM [specified by the VPACING parameter] and TCAM [the OPACING parameter] can minimize the number of buffers occupied in the network control program while the data they contain awaits processing.)

The PACING operand of the LU macro allows you to specify the number of PIUs the network control program will send to the logical unit before awaiting an acknowledgment that the logical unit can receive further PIUs. This acknowledgment is called a pacing response.

Caution: When choosing a value for the PACING operand, observe any restrictions imposed on this operand by specific types of SDLC stations. See the appropriate programming publications for the types of stations used in your network for such restrictions.

Half-Duplex vs. Duplex Data Mode

A local communications controller may operate an SDLC link to a remote communications controller in either half-duplex or duplex data mode. If the SDLC link has two paths, as indicated by use of separate transmit and receive addresses, simultaneous sending and receiving on the link is possible. This is called operation in duplex data mode. If, on the other hand, the same address is used for both transmitting and receiving, these two functions must alternate (half-duplex data mode). The network control program in the local controller will operate the SDLC link to a remote communications controller in duplex data mode, if separate transmitting and receiving paths are available, unless you specify DATMODE=HALF in the PU macro representing the remote controller.

Network Slowdown

The network control program can receive message data from the host processor and from the communication lines only as long as it has buffers available into which to receive the data. The program normally receives and sends data at the same average rate, although momentary overloads can occur in which the program receives more data than it sends over a given time interval. Should the overload be protracted, however, the network control program could exhaust its supply of buffers. To prevent this condition, the network control program continuously monitors its supply of buffers and, when the supply falls to a specified level, automatically enters *slowdown mode.* The level is specified as a percentage of the total number of buffers in the program.

When in slowdown mode, the program reduces the amount of data it receives from communication lines and from the host processor but continues to send at the normal rate. Since the rate at which buffers are released, after transmission of their contents, exceeds the rate at which new buffers are obtained for receiving data, a net gain in the number of available buffers results. When the buffer supply is sufficiently replenished, the program automatically resumes normal operation.

Unless you specify a different value, the network control program enters slowdown mode when fewer than one-eighth (specified as 12 percent) of the total number of buffers are available. The SLODOWN operand of the BUILD macro allows you to specify 12, 25, or 50 percent as the minimum percentage of available buffers, below which the program enters slowdown mode.

Error Conditions and Recovery Procedures

Communication between the communications controller and stations in the teleprocessing network is subject to input/output (I/O) errors, usually caused by the transient noise conditions to which communication facilities are sometimes susceptible, or by hardware malfunctions.

If the error is of a kind that can be recovered from, the network control program makes the appropriate recovery efforts. For example, upon detecting a parity error in received data, the program signals the station to retransmit the data. Conversely, upon being informed by the station that it has received data in error, the program retransmits the data.

The maximum number of retransmissions may be specified for each SDLC station in the network. If error-free transmission is not achieved before the retransmission limit is reached, the network control program indicates the fact in its response to the access method.

If the I/O error is of the kind that inherently cannot be recovered from (such as a modem error), the network control program makes no error recovery attempt but immediately indicates in its response to the access method what kind of error occurred.

Input/output errors can occur either during transmission of message data or while the network control program is performing a control function preparatory to or following message transmission. The number of error recovery attempts for errors affecting message data is determined by the RETRIES operand of the LINE and PU macros representing the SDLC link and station. For errors occurring when receiving from a station, there can be one retry sequence. For errors occurring when transmitting to a station, there can be one or more sequences of retry attempts, with a pause between successive sequences. Specifying a pause of several seconds duration between sequences allows time for transient noise conditions on the link, which may be responsible for the repeated errors, to subside.

The number of sequences and the pause are specified in the PU macro that represents the station. The number of retries per sequence is specified in the LINE macro representing the SDLC link to which the stations are attached. You may specify up to 128 retry attempts.

Automatic Network Shutdown

The entire SDLC network attached to a communications controller (local or remote) is shut down automatically, in an orderly manner, under any of several conditions as explained below. This orderly procedure is called *automatic network shutdown* (ANS). The ANS facility is included in the program unless you specifically exclude it by coding ANS=NO in the BUILD macro. (Apart from automatic shutdown, individual links and stations can also be deactivated and reactivated by requests from the access method.)

Automatic network shutdown occurs under the following conditions.

Local controller:

- The host processor fails to respond to the network control program within a specified interval after the NCP has presented an attention signal to the channel. This interval is specified by the TIMEOUT operand of the HOST macro.
- A shutdown request is entered at the control panel of the controller.

Remote controller:

- The remote network control program detects a lapse in successful communication activity over the local-remote SDLC link currently in use for communication between the local and remote controllers. The lapse may occur either through outright failure of the link or through badly degraded performance of the link as indicated by exhaustion of error recovery procedures performed by the local network control program. The lapse interval is determined by the value you specify in the ACTIVTO operand of the GROUP macro representing the SDLC link(s) joining the local and remote controllers. This interval must be sufficiently long for the local NCP to complete its error recovery procedures for the link.

- The local network control program, upon entering automatic network shutdown mode, signals the remote NCP to shut down the network attached to the remote controller.
- A shutdown request is entered at the control panel of the controller.

Automatic network shutdown of either a local or a remote communications controller terminates any sessions in progress, logically deactivates all active SDLC physical and logical units in the network, breaks any existing switched line connections, disables any currently enabled dial-in ports, and releases any network addresses that have been assigned.

If you exclude the ANS facility, the access method will reload (IPL) the communications controller with a new copy of the network control program whenever it reactivates the controller.

Configuration Restart

Upon completion of the shutdown process, the network control program places a special "shutdown-complete" message at the end of the queue of message traffic for the channel (in the case of a local NCP) or SDLC link to the local controller (for a remote NCP). Upon reestablishing contact with the local controller (over the channel) or the remote controller (over the channel and the SDLC link), the access method receives the shutdown-complete message. The access method receives the message in the normal traffic flow from the NCP if the shutdown was initiated at the control panel.)

The access method then sends the appropriate commands to either (1) reactivate the existing NCP, restore the link, physical unit and logical unit status existing at the time of shutdown, and restart the sessions; or (2) reload (IPL) the controller with a new copy of the NCP. These two alternatives apply to either a local or a remote network control program. *Exception:* Only VTAM can reload a remote controller. The action taken by the NCP in restoring the network status and sessions is called *configuration restart.*

When reactivating a remote network control program via configuration restart, the access method can resume transmission to the remote NCP over the same SDLC link that was in use when shutdown occurred (provided that the link is operational) or over any available backup link.

If you exclude the automatic network shutdown (and configuration restart) facility from a *local* network control program by specifying ANS=NO in the BUILD macro, the NCP waits indefinitely for resumption of communications over the channel; the local controller does not enter the IPL-required state.

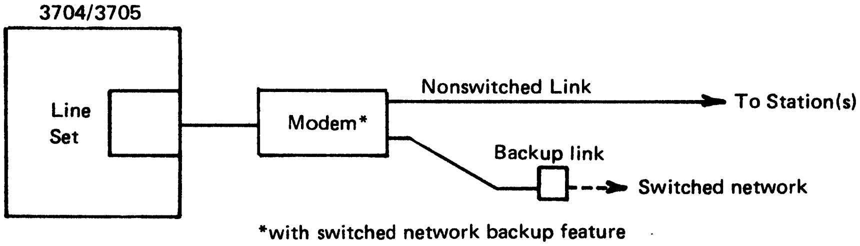
If you exclude the automatic network shutdown (and configuration restart) facility from a *remote* network control program by specifying ANS=NO, one of the following occurs. (1) If you specify an activity timeout in the ACTIVTO operand of the GROUP macro representing the local-remote SDLC link(s), the remote NCP abnormally ends; the controller enters IPL-required state and awaits reloading over the local-remote SDLC link (or over any available such link, if more than one is provided). (2) If you do not specify an activity timeout (ACTIVTO=NONE), the remote NCP does not abnormally end but instead awaits indefinitely the resumption of traffic from the local network control program.

Switched Network Backup

The switched network backup facility of the network control program permits communication between a communications controller and a SDLC station over a temporary, switched communication path provided as an alternate, or "backup", link to the usual (principal) nonswitched point-to-point or multipoint link. Provision of a backup link permits communication between controller and station to continue despite failure of the principal link.

The availability of a backup link requires the installation of appropriate equipment at the controller and stations.

The form of switched network backup used for SDLC links is called "same-port" backup. This technique requires the use of an IBM 3872, 3874, or 3875 modem equipped with the switched network backup feature to attach the controller to both the principal, nonswitched SDLC link (which may be point-to-point or multipoint) and the switched telephone network, thus:



A similar arrangement is required at each station for which the backup path is to be provided. In the event the principal nonswitched link fails, the operator at the host processor can (1) deactivate it by VTAM or TCAM operator command, (2) switch the operation of the modem at the controller to backup mode (and, by voice communication, cause the operator at the remote station to switch its modem similarly), (3) reactivate the link (as represented to the NCP and access method, not the actual failed link) with a VTAM or TCAM operator command, and (4) establish the dialed backup connection with the station.

A single LINE macro represents the principal and the backup link to the network control program. The NCP is not "aware" of, and does not participate in establishing, the switched backup connection. The program simply operates the link in the same way as it does the principal, nonswitched link once the backup connection has been manually established.

In the case of a backup facility provided for a nonswitched *multipoint* link, the backup connection can be made with only one station on the principal link at a time. A separate backup connection must be made with each station in turn if several or all stations are to be contacted using the backup facility.

Restriction: Because a switched backup connection can operate only in half-duplex data mode, and the network control program operates the principal and the backup link in exactly the same manner, the principal link also must operate only in half-duplex (*not* duplex) data mode. Specify this mode by coding one (*not* two) line addresses in the ADDRESS operand of the LINE macro representing the principal link. Further, the physical operation of the link also must be half-duplex

(DUPLEX=HALF specified in the LINE macro) to allow proper communication over the switched backup link. Line turnaround time can be minimized by internally connecting the 3872, 3874, or 3875 modem for continuous-carrier operation.

Note: The value specified in the ENABLTO operand of the BUILD macro must be carefully chosen in order to avoid timeouts during the manual dialing operation, as explained in the description of the ENABLTO operand in Chapter 3.

See the publication *Operator's Guide: VTAM Network Operating Procedures* (GC27-6997) or *Operator's Library: OS/VS TCAM* (GC30-3037) for information on the operator commands used in establishing switched network backup operation.

Backup Local-Remote SDLC Links (VTAM Users Only)

To minimize the possibility of prolonged disruption of communication between a local and a remote communications controller because of failure of the SDLC link joining them, one or more alternate, or "backup," SDLC links may be provided. The alternate links must be dedicated to backup use; they cannot be used for communication with other stations even though currently unneeded for backup use. Typically, for economic reasons, a backup link will comprise a switched line; however, the link can comprise one or two nonswitched point-to-point lines, just as the regular ("principal") SDLC link does. (A backup link serviced by a type 1 communication scanner can only be a nonswitched line.) Regardless of whether switched or nonswitched lines make up the SDLC link, nonswitched line control discipline is used on the link. (DIAL=NO must be specified in the GROUP macro for the alternate links.) In the case of a switched link, this means that the connection between the local and the remote controllers must be established by manual dialing.

Note: A backup link cannot operate via the switched backup feature of modems equipped with this feature (for example, the IBM 3872 and 3874 modems). The backup link must use a separate line interface address.

Switchover from the principal to an alternate link occurs as follows: (1) Upon discovering (by exhausting all error recovery procedures) that the principal link has failed, the local network control program notifies VTAM. The network control program also returns to VTAM, with an indication of "path error," all requests, currently pending or subsequently received, for communication with stations attached to the remote controller. (2) VTAM then informs the system operator, via console message, that the link has failed. (3) The operator must now select an alternate link and enter a command specifying that link and the remote controller to be associated with it. If the link is switched, he must also dial the telephone number of the remote controller. (4) VTAM sends the appropriate commands to the local network control program to contact the remote unit over the alternate link. (5) Once contact is established, configuration restart is initiated (provided that ANS=YES is specified in the remote NCP). Upon completion of the restart process, normal communication is resumed between the local and remote network control program over the current principal link. If ANS= NO is specified in the remote NCP, either the program abnormally ends and the controller enters IPL-required state or the program awaits indefinitely the resumption of traffic from the local NCP as explained under *Configuration Restart.*

Note: If the alternate link is switched, the enable timeout specified in the ENABLTO operand of the BUILD macro must be carefully selected to allow the appropriate interval for the dial connection to be made. See the discussion of this subject under the restriction of the ENABLTO operand (BUILD macro) in Chapter 3.

Reloading of a remote controller that is in IPL-required state can occur only after the remote program loader has been transferred into the controller storage from the diskette within the controller. The remote program loader monitors each remote-to-local SDLC link indicated, within the IPL configuration data set, as "active." (The IBM customer engineer sets this indication for each SDLC link that is to be active.) When polled on the link selected by the operator, the remote program loader transmits a request for loading (or dumping) over that link. VTAM then transmits the network control program load module to the remote controller.

An important point is that the remote network control program, unlike the local NCP, does not directly detect failure of the principal link because it performs no error recovery actions for the link. Instead, the expiration of an "activity timeout" for the principal link, resulting from link failure or shutdown of the local NCP, causes the remote NCP to perform automatic network shutdown and configuration restart (assuming that the shutdown/restart facility is included in the program) (ANS=YES).

Because the local NCP cannot successfully contact the remote controller until the latter's remote program loader begins monitoring the links, the activity timeout should be chosen to avoid an overly long wait before network shutdown is begun. On the other hand, the timeout must be long enough that shutdown does not begin prematurely, that is, before the local NCP has exhausted its error recovery efforts on the principal link. Thus, the value for the ACTIVTO operand must be carefully selected; a formula for determining this value appears in the description of the ACTIVTO operand in the GROUP macro (for NCP and PEP).

Use of the automatic network shutdown function in the remote network control program, though not essential, is highly recommended. If this option is not included, expiration of the activity timeout causes the remote program to abnormally end (Abend). If ACTIVTO=NONE is specified, the program waits indefinitely for traffic from the local controller over any available link.

After the principal SDLC link has been repaired, the system operator can initiate changeover from the alternate back to the principal link. To do so, he first enters VTAM commands to deactivate the alternate link. This action causes VTAM to (1) stop message traffic on the alternate link, (2) initiate automatic network shutdown of the remote network, and (3) deactivate the alternate link. The operator then enters VTAM commands to activate the principal link. After activating the link, VTAM initiates configuration restart of the remote network and normal communication traffic then resumes.

The remote controller can monitor up to four SDLC links (for a type 2 scanner) or two SDLC links (for a type 1 scanner). Accordingly, up to three backup links can be provided for a remote controller, in addition to the principal link. Backup links comprising switched lines need not be dedicated to a specific remote controller, where more than one remote unit is connected to the local controller. For example, two backup links might be provided, either of which could furnish an alternate path to any of three remote controllers connected to the local controller. Further, the alternate links need not have the same operating parameters as the principal link. The alternate links might, for example, operate at half the speed of the principal link.

If the backup link comprises a switched line, data transmission between local and remote controllers (in both directions) is always in half-duplex mode, even if the principal (nonswitched) line is duplex.

Both principal and alternate (backup) links are represented in the same way within network control programs: with LINE macros.

Diagnostic and Service Aids

The network control program diagnoses difficulties in SDLC network operations by means of five diagnostic and service aids: (1) on-line testing (2) address trace, (3) line trace, (4) abnormal end analysis, and (5) panel tests. These aids are useful in identifying malfunctions within the teleprocessing subsystem and the network control program. Some aids are standard (always present in the program); others are optional. Inclusion of the latter in the network control program, although optional, is recommended.

On-line Line Testing

On-line line testing is a diagnostic aid by which a terminal or console may request a variety of tests to be performed upon a communication line. The terminal operator 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 or console. This diagnostic aid, important in problem determination and on-line maintenance of communication lines, is included in a network control program unless you exclude it via the OLT operand of the BUILD macro.

The network control program recognizes each test-request message entered from a terminal and passes it to the access method, just as it does a normal message. Recognizing the message as a test request, the program sends it unchanged to the access method.

The access method detects that the message requests the on-line 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 these operations and returns responses as necessary to the access method. Upon analyzing the responses, the access method determines what further operations to perform and sends the network control program the appropriate interpretive commands.

The network control program is thus only an intermediary in on-line 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.

On-line line test operations require buffer space to hold the interpretive commands and an on-line test control block. These buffers, which the program obtains from the same pool from which it obtains buffers for normal operations, are required only for the duration of the test operation.

The network control program can execute on-line testing operations concurrently for any number of SDLC links. Some extra buffer space is needed for each additional link on which on-line testing is being conducted. Teleprocessing operations on lines not undergoing testing can continue as usual.

Address Trace Facility

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 trace data in a trace table within control storage. When the desired data has been recorded, the contents of the trace table can be displayed on the control panel of the controller. The contents of controller storage can be transferred to the host processor via the Dump program and the contents of the trace table examined in the listing of the dump.

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.

Line Trace Facility

The line trace facility is a service aid that permits detailed analysis of the operation of any communication line operating in network control mode. (All SDLC links operate only in network control mode.) This facility records operating parameters of a line each time a level two interrupt occurs for that line. (Level two is the program interrupt level at which bit service or character service for the communication line is performed.) The program places network control mode trace information in buffers obtained as required from the buffer pool, then transfers these buffers at intervals to the host processor. The host processor should accumulate these line trace records in a data set (file) from which they may be printed out for analysis.

A line trace can be initiated at any time by request from the host processor, and ended at any time by a subsequent request.

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

One line at a time can be traced; any "activate line trace" requests beyond the first one are rejected until the first is ended.

The line trace facility for SDLC links is always present in the network control program.

Abnormal End (ABEND) Facility

Programming errors detected during execution of non-supervisory portions of the network control program cause abnormal termination of program execution. Examination of abend codes within a storage dump can help in locating the error.

The optional Abend service aid extends detection of programming errors to the network control program *supervisor*, thus causing termination of the program before a supervisor error can be propagated into non-supervisory portions of the program. The Abend code appearing in the storage dump therefore gives a better indication of the location of a supervisor error, if one should occur, than a code reflecting a resultant error in the non-supervisory portion would give. Inclusion of the Abend option (by the ABEND operand of the BUILD macro) is recommended when you first begin using a network control program to control your teleprocessing network. Later, as experience demonstrates that your teleprocessing network operates routinely without abnormal termination of the network control program owing to program errors, the Abend option may be deleted from the program.

Panel Tests

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

The panel test function is always present in the network control program.

Program Generation Options and Data Sets (Files)

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

Program Generation Options

Program generation options pertain to the type of communications controller (3704 or 3705, local or remote) in which the program will be executed, complete vs. partial generation procedure, and several assembly and link editing options. All program generation options are specified in the BUILD macro.

Type of Program to be Generated

As explained in Chapter 1, when defining the control program for a local communications controller you must decide whether the program is to perform network control functions only, emulation functions only, or both. For an SDLC-only network, the only valid choice is network control functions. For a network that also includes BSC and/or start-stop lines, you may specify network control functions only, or both network control and emulation functions. Emulation functions apply only to the BSC and/or start-stop portions of the network. Specify the choice in the TYPGEN operand of the BUILD macro.

Model of Controller

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

Partial Generation

Assembling and linkage editing the many modules making up a network control program necessarily consumes substantial processing time. Once you have generated a complete network control program, however, modifications resulting from changes in network configuration and procedural options can be effected in significantly less time, via *partial generation.*

In partial generation, only selected modules are reassembled; these are then linkage edited with the object modules that require no changes to produce the modified program.

To perform a partial generation, you code PARTIAL=YES and specify, in the CONDASM operand, the names of the modules requiring reassembly. (The modules requiring reassembly for each of various changes in the program functions are listed in Appendix H.)

It is important that you retain all of the stage one and stage two assembly output listings and the object library (containing conditionally assembled modules) produced by the complete generation procedure. Saving this output facilitates subsequent partial generation.

Other Options

The remaining program generation options, and the operands of the BUILD macro by which you specify them, are:

- Whether stage two of the generation procedure is to consist of a single, multistep job or a separate job for each step, and whether a job card is required (JOBCARD).
- The region size for stage two linkage edit job steps (LESIZE) [applicable only for OS/VS].
- Whether or not the generation procedure is to produce cross-reference listings for stage two assemblies (ASMXREF).
- The value of the TIME parameter in stage two assembly EXEC statements (TIME) [applicable only for OS/VS].
- The type of device or class of devices to be used for utility data sets during stage two (UNIT) [applicable only for OS/VS].

Data Sets (Files) Used in the Generation Procedure

The names of various program data sets to be used in the generation procedure when generating under OS/VS are specified by the LOADLIB, OBJLIB, QUALIFY, UT1, UT2, UT3, and USERLIB operands of the BUILD macro. The NEWNAME operand specifies the name to be given to the generated network control program load module.

The only file name required when generating under DOS/VS is NEWNAME.

Chapter 3: NCP Generation Macro Instructions for SDLC-Only Networks

This chapter gives detailed descriptions of the macro instructions with which you define a network control program that is to control a network comprising *only* SDLC lines and stations. If you wish to define a program to accommodate BSC and/or start-stop stations (with or without SDLC stations as well), use Chapter 5 instead of this chapter. (This chapter contains the same information about defining SDLC lines and stations so does Chapter 5.)

Macro Instruction Coding Conventions

The following conventions are used in the descriptions of the 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 (default value) to be given. 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.

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. Then follow, in alphabetical order, operands that are conditional or optional. These are enclosed in brackets—[].

Data set (file) names must begin with an alphabetic character or $, @, or #.

System Definition Macro Instructions

This section contains the system definition macro instructions, PCCU, BUILD and SYSCNTRL, to be used in defining an SDLC-only network control program. (PCCU is a VTAM-only macro instruction.)

PCCU Macro Instruction (VTAM Only)

The PCCU macro instruction identifies for VTAM the 3704 or 3705 communications controller in which the network control program being defined is to be loaded and executed. This macro must appear at the beginning of the NCP generation input deck, preceding the BUILD macro, before the deck is provided to the VTAM initialization process. You may include it in the deck provided to the NCP generation procedure, but this is not required.

See the *VTAM System Programmer's Guide* for a complete description of the macro and its operands.

Name	*Operation*	*Operands*	
[symbol]	PCCU	[CUADDR=address]	(for local controller)
		[,AUTODMP={YES}] {NO }	
		[,AUTOIPL={YES}] {NO }	
		[,DUMPDS=ddname]	(OS/VS only)
		[,DUMPDS=SYS008]	(DOS/VS only)
		[,INITEST={YES}] {NO }	
		[,MAXDATA=size]	
		[,NCPLUB=SYSxxx]	(DOS/VS only)
		[,RNAME=rname]	(for remote controller only)

BUILD Macro Instruction

The first macro instruction in the program source statements (except for the VTAM-only macro, PCCU) is BUILD. This macro specifies:

- The type of controller (3704, 3705-I, or 3705-II; local or remote) that is to execute the network control program.
- The controller storage size.
- The size of buffers in the buffer pool.
- The name that is to be assigned to the network control program, resource resolution table, and block handler set resolution table load modules.
- The type and number of channel adapters in the communications controller.
- The subarea address to be assigned to the network control program being defined.
- The upper limit of the range of subarea addresses assigned to network control programs in the network.
- Certain optional facilities that may be included in the network control program.
- Certain program generation options that may be desired.
- The names of program data sets used in the generation process.
- Whether a complete or a partial program generation is to be performed.

```
Name        Operation   Operands

[symbol]    BUILD       LOADLIB=dsname,            (OS/VS only)

                        MAXSUBA=n,

                        MEMSIZE=n,

                        OBJLIB=dsname,             (OS/VS only)

                        SUBAREA=n,

                                {NCP    }
                        TYPGEN={NCP-LR}
                                {NCP-R  }

                        [,ABEND={YES}]
                                {NO }

                        [,ANS={YES}]
                              {NO }

                        [,ASMXREF={YES}]
                                  {NO }

                        [,BFRS={size} ]
                               {60  }

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

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

                        [,CONDASM={TABLE         }]
                                  {(value1,...)}

                        [,DIALTO={count} ]
                                 {60.0 }

                        [,DSABLTO={count} ]
                                  {3.0  }
```

Name *Operation* *Operands*

```
[,ENABLTO={count} ]
          {2.2   }

          {YES  }
[,JOBCARD={NO   }]
          {MULTI}

[,LESIZE=size]

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

[,NEWNAME={NCP001}  ]
          {symbol}

[,OLT={YES}]
      {NO }

[,PARTIAL={YES}]
          {NO }

[,PWROFF={YES}]
         {NO }

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

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

[,TIME=integer]

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

[,TYPSYS={OS }]
         {DOS}

[,UNIT=unit type]

[,UT1=dsname]

[,UT2=dsname]

[,UT3=dsname]
```

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

LOADLIB=dsname

(generation under OS/VS only)

Specifies the name of a partitioned OS/VS data set that will contain the network control program load module, resource resolution table module, and block handler set resolution table module 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 or $, #, or @.) This data set must be cataloged.

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

MAXSUBA=n

Specifies the upper limit of the range of subarea addresses used within the network controlled by the access method. (Each network control program within the network must be assigned a unique subarea address via the SUBAREA operands of their respective BUILD macros. A single value is required for all MAXSUBA operands in all network control programs in the network active at the same time.

VTAM Note: This value must be the same as specified in the MAXSUBA VTAM start parameter.

Note: The term *subarea* is explained in the publication *VTAM Concepts and Planning* (GC27-6998).

This upper limit must equal or exceed the highest subarea address defined in any network control program with which the program you are defining will communicate. The maximum subarea address value is always a power of two, minus one, within the range 3–255 (that is, 3,7,15,31,63,127,255). If you specify a value that is not one of those listed, the generation procedure rounds the specified value to the next higher such value. (For example, any value you specify between 16 and 30 will be rounded to 31.)

The total number of resources that can be associated with any subarea address depends on the value of MAXSUBA, as follows:

Value of MAXSUBA	*Maximum Number of Resources Possible*
3	16382
7	8190
15	4094
31	2046
63	1022
127	510
255	254

Note: Specifying an unnecessarily high value for *n* will waste NCP storage space (space is assigned for all subarea addresses whether used or not).

Example: Assume that the network includes two local communications controllers, to which subarea addresses 2 and 3 are assigned, and four remote controllers, to which addresses 4 through 7 are assigned. The highest address being 7, you would specify an upper limit of at least 7. If you wish to allow for adding more controllers to the network, however, you would specify a value greater than 7. The next higher value, 31, would allow up to 30 controllers (addresses 2 through 31) to be included in the network.

`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).

The value of *n* must be one of the following:

For 3704 (MODEL=3704): 48 or 64
For 3705-I (MODEL=3705 or 3705-1): 48, 80, 112, 144, 176, 208, or 240
For 3705-II (MODEL=3705-2): 64, 96, 128, 160, 192, 224, or 256

This operand is required.

`OBJLIB=dsname`

(generation under OS/VS only)

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

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

`SUBAREA=n`

Specifies the subarea address to be assigned to the network control program you are defining. (Each network control program active in a network [and each group of locally attached terminals served directly by the access method], and the access method itself, must have a unique subarea address.)

Note: The term *subarea* is explained in the publication *VTAM Concepts and Planning* (GC27-6998).

The minimum valid subarea address is 2—that is, SUBAREA=2. The maximum address is the value you specify in the MAXSUBA operand of this macro.

This operand is required.

```
        {NCP   }
TYPGEN= {NCP-LR}
        {NCP-R }
```

Specifies (1) whether the program is to be executed in a local or a remote communications controller; and (2) whether the program (in a local controller) can communicate with a remote controller. Select the appropriate parameter from the following:

	The program will:	
Parameter	*communicate with a remote controller*	*The controller is a:*
NCP	No	Local
NCP-LR	Yes	Local
NCP-R	No	Remote

This operand is required.

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

Specifies whether or not the optional ABEND facility is to be included in the network control program.

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

Specifies whether or not Automatic Network Shutdown is to be included in the 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.

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

Specifies whether or not you wish the generation procedure to produce cross-reference listings for the stage two program assemblies.

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

Specifies the size, in bytes, of buffers in the network control program buffer pool.

Specify *size* as a multiple of four bytes. The minimum size is 48 bytes (44 bytes if you omit the on-line testing facility from the program (BUILD: OLT=NO)). The maximum is 248. The generation procedure rounds the size you specify to the next higher multiple of four bytes, if you have not specified such a multiple.

To the value you specify (rounded, if necessary, to the next multiple) the generation procedure adds four bytes for control use. (These four bytes are never used to contain message data.)

Example: If you specify BFRS=70, the length of each buffer is 76 (70 is rounded to 72, and four bytes are added). If you specify BFRS=72, the length is also 76 (no rounding is necessary). If you omit the BFRS operand, the buffer length is 64 (60 + 4 bytes for control use).

Note: Diagnostic programs that communicate with the network control program via VTAM or TCAM (for example, TOTE) may impose restrictions on the values specified for the BFRS operand. See the appropriate manuals for such diagnostic programs for restrictions that may apply.

```
[CA=(adapter 1 [,adapter 2])]
```

(local controller only)

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

`adapter1`

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

`adapter2`

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

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

This operand specifies the type and physical location of the installed adapters. See the description of the CHANTYP operand below for the *use* of the installed adapters.

Note: If you omit both the CA and the CHANTYP operands, the default values assumed for both are as follows:

> If MODEL=3704 is specified, the values assumed are: CA=TYPE1, CHANTYP=TYPE1.
>
> If MODEL=3705 (3705-1) or MODEL=3705-2 is specified, the values assumed are: CA=TYPE2, CHANTYP=TYPE2.

If you omit the CA operand but code the CHANTYP operand, the values assumed for CA are equivalent to those specified in the CHANTYP operand.

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

If CA operand is omitted:

[1] CA=TYPE1 is assumed if MODEL=3704 is specified.
[2] CA=TYPE2 is assumed if MODEL=3705 (3705-1) or MODEL=3705-2 is specified.

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

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

Specifies the use of the channel adapters installed in the communications controller. (Accepted values for *ptype* and *stype* are TYPE1, TYPE2, TYPE3, and TYPE4.)

`ptype`

Specifies the type of the primary channel adapter to be used for initial network control program operation as type 1, type 2, type 3, or type 4. If you omit *ptype* and the controller is a 3705 (MODEL=3705 [3705-1] or MODEL=3705-2), the primary channel adapter is assumed to be type 2. If you omit *ptype* and the controller is a 3704 (MODEL=3704), the primary channel adapter is assumed to be type 1. (A 3704 can have only a type 1 channel adapter.)

`stype`

Specifies that the secondary channel adapter is installed and identifies it as type 1, type 2, type 3, or type 4. If you omit *stype,* it is assumed that the secondary channel adapter is not installed. (Omit *stype* if you code MODEL=3704; a 3704 cannot have a secondary channel adapter.)

Note: If you specify *stype,* during initialization the network control program will loop until the unused channel adapter is disabled (goes off line).

If you code *ptype* as TYPE1, *stype* cannot also be coded as TYPE1.

Figure 3-2 shows, for each of the thirteen possible 3705 channel adapter configurations, the permissible values of the CHANTYP operand. A controller can execute a network control program if there appears, opposite the adapter configuration, an adapter specification that matches the specification in the BUILD macro.

Note 1: The same network control program can be executed in controllers with dissimilar adapter configurations if the program's adapter specification appears opposite each of the adapter configurations with which you intend to use the program. If, for example, you wish to define a program that can be executed in either a controller having a single type 1 adapter or a controller having a type 1 and a type 2 adapter (with the type 1 adapter installed in the base module), you would specify CHANTYP=(TYPE1). (Before loading the program into the controller having two adapters, the unused adapter must be disabled, as note 1 directs.) The program could not, however, be executed in a controller having one or two type 2 channel adapters. (This example assumes that the program performs only network control functions.)

Note 2: See the description of the CA operand for default values assumed for this operand if the CA operand is omitted.

Adapter Configuration		*Adapter Specification*
First CA (base module)	Second CA (expansion module)	CHANTYP=
Type 1	(none)	TYPE1
Type 2	(none)	TYPE2
Type 3	(none)	TYPE3
Type 4	(none)	TYPE4
Type 1	Type 2	TYPE1[1] TYPE2 (TYPE1,TYPE2)[2] (TYPE2,TYPE1)[2]
Type 1	Type 3	TYPE1[1] TYPE3 (TYPE1,TYPE3)[2] (TYPE3.TYPE1)[2]
Type2	Type 2	TYPE2[1] (TYPE2,TYPE2)[2]
Type 2	Type 3	TYPE2[1] TYPE3[1] (TYPE2,TYPE3)[2] (TYPE3,TYPE2)[2]
Type 3	Type 2	TYPE3[1] TYPE2[1] (TYPE3,TYPE2)[2] (TYPE2,TYPE3)[2]
Type 3	Type 3	TYPE3[1] (TYPE3,TYPE3)[2]
Type 4	Type 2	TYPE4[1] TYPE2[1] (TYPE4,TYPE2)[2] (TYPE2,TYPE4)[2]
Type 4	Type 3	TYPE4 TYPE3[1] (TYPE4,TYPE3)[2] (TYPE3,TYPE4)[2]
Type 4	Type 4	TYPE4 (TYPE4,TYPE4)[2]

[1] The unused channel adapter must be disabled at the 3705 control panel before the program is loaded. Failure to do so may result in an "abend" condition.

[2] During initialization, the network control program will loop until the unused channel adapter is disabled (goes off line).

Figure 3-2. CHANTYP Operand Specifications Permissible for Each 3705 Channel Adapter Configuration (SDLC-Only Networks)

```
[CONDASM={TABLE        }  ]
         {(value1,...)}
```

Specifies, for a partial program generation, which conditionally assembled network control program modules are to be reassembled.

This operand is valid only if you specify PARTIAL=YES in this BUILD macro.

TABLE

Specifies that only the modules containing network control program tables are to be reassembled.

value1,...

Identifies specific modules to be reassembled. *value1,...* represents a sequence of two-digit numbers corresponding to the last two digits of the names of the modules to be assembled. For example, to reassemble modules SYSCG007 and SYSCG00A, you would code CONDASM=(07,0A).

The modules that may be individually reassembled, and the corresponding values to be coded in the CONDASM operand, are as follows:

SYSCG000	00	SYSCG009	09
SYSCG001	01	SYSCG00A	0A
SYSCG002	02	SYSCG00B	0B
SYSCG003	03	SYSCG00C	0C
SYSCG006	06	SYSCG00D	0D
SYSCG007	07	SYSCG00E	0E
SYSCG008	08	SYSCG010	10

The network control program tables are always assembled, regardless of which specific modules you specify in CONDASM=(value1,...).

Appendix H lists the module that must be reassembled for various changes in program functions.

```
[DIALTO={count}]
        {60.0 }
```

Specifies the timeout to be used by the network control program in detecting failure of the automatic calling unit's "abandon call and retry" signal. Specify the timeout either as an integral number of seconds (DIALTO=30) or to tenths of a second (DIALTO=40.4).

Use of the default value (60 seconds) is recommended unless the system designer recommends a different one.

The maximum timeout you may specify is 1632 seconds.

Note: See the section, *Restriction on Number of Time Intervals,* following the description of the GROUP macro.

```
[DSABLTO={count}]
         {3.0  }
```

(switched lines only)

Specifies the timeout to be used by the network control program in detecting the failure of the "data set ready" signal line of the modem to be turned off when the switched line attached to the modem is disabled. Specify the timeout either as an integral number of seconds (DSABLTO=5) or to tenths of a second (DSABLTO=7.5).

The maximum timeout you may specify is 1632 seconds.

The line remains disabled for the period specified, regardless of whether the "data set ready" signal line is turned off within the period.

Note: See the section, *Restriction on Number of Time Intervals*, following the description of the GROUP macro.

```
[ENABLTO={count}]
        {2.2  }
```

Specifies the timeout to be used by the network control program in detecting the failure of the "data set ready" signal line of the modem to be turned on when the communication line attached to the modem is enabled (for nonswitched lines) or when a dialing operation is completed—that is, the automatic calling unit has signalled connection (for switched lines). Specify the timeout either as an integral number of seconds (ENABLTO=3) or to tenths of a second (ENABLTO=3.2).

The maximum timeout you may specify is 1632 seconds.

For a nonswitched line or a switched line over which calls are made by an automatic calling unit (ACU), the default value of 2.2 seconds is usually appropriate. It is not appropriate, however, if the local network control program being defined includes any switched backup SDLC links to a remote controller. Such a link requires an enable timeout sufficiently long that it will not expire before the system operator, when manually calling the remote controller, dials the telephone number, receives an answer, and places the modem (data set) in data mode. This process may typically take from 30 seconds to more than a minute. The value you specify in ENABLTO should exceed this interval. On the other hand, the timeout value should be no greater than necessary; otherwise it will needlessly extend the time required to shut down the network control program if shutdown is initiated while a line is being activated by command from the access method. (The network control program does not shut down until all lines are deactivated.)

Note: See the section, *Restriction on Number of Time Intervals*, following the description of the GROUP macro.

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

(generation under OS/VS only)

Specifies whether or not the program generation procedure is to provide a job card for the stage two input stream, and whether the input stream will consist of more than one job.

The job card provided is in the form

```
//NCPGENnn JOB1,'NCP GENERATION',MSGLEVEL=1
```

If you specify JOBCARD=YES, or omit the operand, a single job card is provided and the program generation input stream consists of a single, multi-step job. The job card label is //NCPGEN00.

If you specify JOBCARD=NO, no job card is provided and you must therefore:

1. Specify DD DATA in the SYSIN card for the stage one assembly step.
2. Place a REPRO statement, immediately followed by an OS/VS job card, in the input stream preceding the BUILD macro.

If you specify JOBCARD=MULTI, a job card is provided for each step and the input stream therefore consists of multiple jobs. The job card labels are //NCPGENnn, where *nn* is a sequential identification number provided by the generation procedure.

Note: If you code JOBCARD=MULTI, you may specify a job card different from the one shown by using the IEBUPDTE utility program to change the job statement information in the stage 1 macro library. See Chapter 8 for information on this procedure.

```
[LESIZE=size]
```

(generation under OS/VS only)

Specifies the OS/VS region size, in K (1024) bytes, to be used by all linkage editor job steps during stage two of program generation. The number you specify is reduced by 10 and used as *value1* of the linkage editor SIZE parameter. *value2* of the SIZE parameter is always 48 (K), regardless of what you specify in the LESIZE operand.

size must exceed 10 and be less than 16384 (16, 384 K bytes).

If you omit the LESIZE operand, the EXEC card for the linkage editor job steps will have a REGION parameter of 384 K and PARM parameter values of 374 (for *value1*) and 48 (for *value2*).

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

Specifies whether the generated network control program is to be loaded into and executed by a 3705-II (MODEL=3705-2), a 3705-I (MODEL=3705), or 3704 (MODEL=3704). (A 3705-I may be specified as either MODEL=3705 or MODEL=3705-1.)

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

```
[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, specify NCP001 in this operand. If you omit the operand, the name assigned is NCP001.

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

Specifies whether or not the optional on-line terminal test and on-line line test facilities (for lines in network control mode) are to be included in the network control program. Code OLT=NO to omit the facilities; code OLT=YES (or omit the operand) to include them.

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

Specifies whether or not a partial program generation is to be performed.

If you code PARTIAL=YES, only the tables and conditionally assembled modules specified in the CONDASM operand are assembled and linkage edited with the remaining object modules. The conditionally assembled modules not specified by the CONDASM operand are obtained from the library specified by the OBJLIB operand. (You must have assembled these modules during a previous program generation for which the OBJLIB operand of the BUILD macro specified the same library name.) The modules assembled by the partial generation procedure replace the corresponding modules from the previous generation.

Appendix H lists the modules that must be reassembled for various changes in program functions.

If you code PARTIAL=NO (or omit the operand), a complete program generation is performed.

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

(local NCP only; VTAM users only)

Specifies, for a network control program to be executed in a local communications controller, whether the program will turn off the remote controller's power upon command from VTAM. PWROFF=YES is valid only if the remote controller is equipped with the remote power off feature. (Power can be turned on again only at the control panel of the remote controller.)

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

(generation under OS/VS only)

Specifies the first-level qualifier for OS/VS data sets specified by the LOADLIB, 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 alphabetic (including $, @ and #). (Omit the period [.] that separates the qualifier and the data set name; the generation procedure appends the period to the qualifier you specify.)

`NONE`

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

`SYS1`

Specifies that SYS1 is to be used as the qualifier.

```
           {12}
[SLODOWN={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 lines operating in network control mode and from the network control subchannel, but continues to send data over the lines and the subchannel. This procedure reduces the number of buffers in use.

Slowdown mode is entered when fewer than one-half (SLODOWN=50), one-quarter (SLODOWN=25) or one-eighth (SLODOWN=12 or operand is omitted) of the buffers are available.

During initialization, the NCP dynamically increases the percentage you specify if the minimum NCP buffer requirements cannot otherwise be met. The minimum number of buffers that the program must contain for each percentage value is: 80 buffers, for 12 percent; 40 buffers, for 25 percent; and 20 buffers, for 50 percent.

If the number of buffers contained in the network control program is less than 20, the program Abends.

```
[TIME=integer]
```

(applicable to OS/VS only)

Specifies the time value, in minutes, to be used as the TIME parameter in the stage two assembly EXEC statements. *integer* must be greater than 0 and less than 1441.

If you omit this operand, no TIME parameter is used for the stage two assembly EXEC statements.

```
         { NO            }
[TRACE={(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, from ten [TRACE=(YES,10)] to 256 [TRACE=(YES,256)]. If you omit the number or specify fewer than ten, the table will contain ten entries.

```
[TYPSYS={OS }]
         {DOS}
```

Specifies whether stage two of the network control program generation procedure is to be run under OS/VS or DOS/VS.

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

Specifies the type of device to be used for the assembler and linkage editor utility data sets during stage two of program generation under OS/VS. You may specify either an actual device type (for example, UNIT=2311) or the name of a class of devices (for example, UNIT=SYSDA). The maximum number of characters you may specify is eight.

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and SYSDA is assumed for the linkage editing steps.

Note: The utility data set for the linkage editor must reside on a direct-access device.

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

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

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

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

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

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

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

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

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

SYSCNTRL Macro Instruction

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

Figure 3-3 shows which options are required by VTAM and TCAM.

This macro is required and must appear directly following the BUILD macro.

Name	*Operation*	*Operands*
`[symbol]`	`SYSCNTRL`	`OPTIONS=(entry,...)`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`OPTIONS=(entry,...)`

Specifies which of the dynamic control facilities are to be included in the network control program. *entry* may be any value listed in Figure 3-3.

Entry	**Facility**
Always required by VTAM:	
BHSASSC	Modify block handler set*
ENDCALL	Physical disconnect*
MODE	Set destination mode*
RCNTRL	Request control mode reset*
RCOND	Reset conditional
RECMD	Reset at end of command*
RIMM	Reset immediate
*The VTAM initialization procedure requires these parameters even though the functions they specify are not used in an SDLC-only network.	
Always required by TCAM:	
RCOND	Reset conditional
RIMM	Reset immediate

Figure 3-3. Dynamic Control Facilities Required by VTAM and TCAM

Configuration Definition Macro Instructions

This section contains the configuration definition macro instructions (HOST, CSB, SERVICE, and LUPOOL) to be used in defining a network control program/VS for an SDLC-only network.

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 access method.
- The size of access method buffer unit used to receive data from the network control program.
- The number of buffer units the access method will allocate for receiving a data transfer.
- The number of bytes in the header prefixes used by the access method.
- The amount of time the network control program waits for a response by the host processor to an attention signal.
- The amount of time the network control program waits before presenting an attention signal on the channel after data becomes available for transfer to the host processor.

The HOST macro is required in either a local or a remote network control program. The values specified in the operands of the HOST macro coded in a remote network control program must be the same as those specified in the HOST macro of the local program with which the remote program communicates. (Omit the DELAY, STATMOD, and TIMEOUT operands from a HOST macro coded in a remote network control program.)

One HOST macro is required; it must appear among the nonpositional configuration macros that follow the two system macros (BUILD and SYSCNTRL).

Name	*Operation*	*Operands*	
[symbol]	HOST	INBFRS=count,	
		MAXBFRU=count,	
		UNITSZ=length	
		[,BFRPAD={n }] {28}	
		[,DELAY={count}] {0 }	(local NCP only)
		[,STATMOD={YES}] {NO }	(local NCP only)
		[,TIMEOUT={count}] {NONE }	(local NCP only)

[symbol]

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`INBFRS=count`

Specifies the number of controller buffers initially allocated for each data transfer to be received from the host processor. The minimum value is 1; the maximum is 255.

This operand is required.

`MAXBFRU=count`

Specifies the number of buffer units the access method will allocate for receiving data from the network control program. The minimum is 1; the maximum is 255.

This operand is required.

See the description of the UNITSZ operand for a sample calculation.

VTAM Note: The value you specify in this operand must be less than the value of the *bno* subparameter of the LFBUF (DOS/VS) or IOBUF (OS/VS) parameter of the VTAM start parameter list.

`UNITSZ=length`

Specifies the size of the access method buffer units used for data transfers from the network control program. (Specify the sum of (1) the length of the message data, (2) 13 bytes for control information, and (3) the number of pad characters specified in the BFRPAD operand.)

The access method must use one buffer unit size for all transfers from the network control program. A buffer unit is the smallest unit of contiguous storage handled as buffer space; a buffer may consist of one or more units.

The NCP generation procedure accepts values between 1 and 65535, inclusive. However, the access method imposes the following minimum and maximum values (as of the time of publication of this manual):

	Minimum:	*Maximum:*
DOS/VS VTAM	88	2024
OS/VS1 VTAM	84	2012
OS/VS2 VTAM	84	4060
OS/VS TCAM	44	255

Use the values given in the *VTAM System Programmer's Guide* or *OS/VS TCAM System Programmer's Guide* (see Preface) instead of the values above, if they differ.

VTAM Note: This operand must specify the same value as the *bsz* subparameter of the IOBUF (OS/VS only) or LFBUF (DOS/VS only) parameter of the VTAM start parameter list.

The maximum number of characters the network control program will transfer to the host processor in a single operation over the network control subchannel equals MAXBFRU times UNITSZ, minus the value specified in the BFRPAD operand. (VTAM requires that the result of this calculation must not be less than 84 (OS/VS) or 88 (DOS/VS) bytes.) *Example:* If you code MAXBFRU=5, UNITSZ=84, and BFRPAD=(28), the maximum number of characters sent to the host processor during a single channel operation is 5(84) minus 28, or 392 bytes. (This value includes the 13 bytes of control information associated with each response sent by the network control program.)

Caution: The maximum amount of data that the access method can send to the network control program should not exceed the amount that the NCP can return to the access method, as expressed by the formula above (MAXBFRU times UNITSZ, minus BFRPAD).

```
[BFRPAD={n }]
       {28}
```

Specifies the number of pad characters the network control program is to transmit to the access method immediately preceding the control information for the response associated with a teleprocessing request.

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 value you specify in the BFRPAD operand should equal the size of the prefixes that the access method uses. The required value for OS/VS VTAM is 28 bytes; for DOS/VS VTAM, 15 bytes; for OS/VS TCAM, a minimum of 17 bytes.

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

(local NCP only)

Specifies the interval, to the nearest tenth of a second, that 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 signal 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 420.0 seconds.

0

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

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

This operand is invalid for a remote network control program (TYPGEN=NCP-R).

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

Specifies whether the network control program is to use the status modifier option for sending responses to the access method in the host processor. Use of this option when the attention delay option is also used minimizes the number of asynchronous channel interrupts during data transfer between the network control program and the access method. This occurs because each time the access method sends data to the network control program, the NCP returns any accumulated response data to the access method as part of the same channel operation instead of sending a later attention interrupt to the host processor.

Use of the status modifier option is required (STATMOD=YES) if the access method (VTAM or TCAM) in the host processor is to run under OS/VS. To maximize the increase in channel performance, also specify an attention delay of at least 0.1 second in the DELAY operand of the HOST macro. If the access method is DOS/VS VTAM, the status modifier option is optional for release 33 (or higher) of DOS/VS and release 2 (or higher) of VTAM; for lower levels of VTAM or DOS/VS the option is not available and STATMOD=NO is therefore appropriate.

`[TIMEOUT={count}]`
`{NONE}`

(local NCP only)

Specifies the interval, to the nearest tenth of a second, that the network control program awaits a response to an Attention signal it has sent to the host processor before initiating automatic network shutdown, if ANS=YES is specified in (or ANS operand is omitted from) the BUILD macro. Express this interval either 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 0.2 second; the maximum is 420.0 seconds.

NONE

Specifies that the network control program is to wait indefinitely for a response from the host processor.

If the automatic network shutdown facility is included in the program (ANS=YES), and you omit the TIMEOUT operand, TIMEOUT=420.0 is assumed. If you specify ANS=NO, and you omit the TIMEOUT operand, TIMEOUT=NONE is assumed.

If you have excluded the automatic network shutdown facility (ANS=NO is specified in the BUILD macro), omit this operand.

Upon failure of the host processor to respond to the network control program, the action the NCP takes depends on how you have specified the ANS and TIMEOUT operands: (1) If you specify ANS=YES (or omit the ANS operand) and TIMEOUT=count, the NCP performs automatic network shutdown, then waits for data over the channel. (2) If you specify ANS=NO and TIMEOUT=NONE, the NCP does not perform automatic network shutdown, but instead waits indefinitely for message traffic over the channel. (The combinations ANS=YES, TIMEOUT=NONE and ANS=NO, TIMEOUT=count are invalid.)

This operand is invalid for a remote network control program (TYPGEN=NCP-R).

Note: The interval specified in the TIMEOUT operand is not counted as among the 16 possible time intervals permitted in the network control program. (The limit of 16 applies only to communication line timeouts and intervals—not to channel timeouts.)

CSB Macro Instruction

The CSB macro specifies:

- The type of communication scanner.
- The internal oscillator (business machine clock) rates for the scanner.
- The location of the scanner within the controller.

Each scanner in the controller must be represented by a CSB macro.

```
Name          Operation      Operands

[symbol]      CSB            SPEED=(rate,...)

                             [,MOD={n}]
                                   {0}

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

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`SPEED=(rate,...)`

Specifies the internal oscillator (business machine clock) rates for up to four oscillators installed in the communication scanner. (Do not confuse this SPEED operand, which specifies the oscillator bit rates, with the SPEED operand of the LINE macro, which specifies the data rate for the communication line.) The speeds must be specified in the same order that the oscillators are installed on the scanner, in ascending order according to speed. Standard oscillator bit rates are shown in Figure 3-4.

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

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

This operand is required.

Rate	*Represents:*	*Rate*	*Represents:*
45	45.5 bps	150	150.0
50	50.0	200	200.0
56	56.89	300	300.0
74	74.2	600	600.0
75	75.0	950	950.0
100	100.0	1200	1200.0
110	110.0	2000	2000.0
134	134.5	2400	2400.0

Figure 3-4. Standard Communication Scanner Oscillator Bit Rates

```
[MOD={n}]
     {0}
```

Specifies the location of the communication scanner, as shown in Figure 3-5. The line interface addresses valid for each scanner type and module location are given. (The 3704 has only one module.)

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

Figure 3-5. Location of Communication Scanners and Valid Line Interface Addresses

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

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

Valid designations for scanner type are:

If controller is a 3704 (MODEL=3704): TYPE1
TYPE2

If controller is a 3705-I (MODEL=3705 or 3705-1): TYPE1 (valid only for MOD=0)
TYPE2
TYPE3 (valid only for MOD=1, 2, or 3))

If controller is a 3705-II (MODEL=3705-2): TYPE2
TYPE3

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

SERVICE Macro Instruction

The SERVICE macro generates the service order table for a nonswitched (multipoint or point-to-point) SDLC link. (Switched SDLC links do not use a service order table.)

All nonswitched SDLC links are controlled by a multipoint discipline, regardless of whether one station or several stations are on the link.

Each nonswitched link requires a service order table in the network control program that controls the link. A nonswitched link between a local communications controller and a remote controller is always controlled by the program in the local controller. Consequently, a service order table is required only in the local (not the remote) network control program.

Within the network control program that controls an SDLC link (POLLED=YES is specified in the LINE macro), one SERVICE macro is required following the LINE macro that represents the link. Each SDLC cluster controller or communications controller attached to the link must be represented by an entry in the service order table. (Logical units associated with the SDLC cluster controller are not represented in the service order table.)

If more than one SDLC link connects a local and a remote 3704 or 3705 (that is, the principal link is augmented by one or more alternate ["backup"] links), a SERVICE macro must appear in the local NCP following each of the LINE macros representing the links.

Name	*Operation*	*Operands*
`[symbol]`	`SERVICE`	`ORDER=( entry,... )`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity. Specify a name only on the first SERVICE macro defining the service order table, if more than one macro is required.

`ORDER=( entry,... )`

Specifies the order in which the stations (physical units) on the SDLC link are to be serviced.

entry is the name of an SDLC physical unit (terminal, cluster controller, or remote communications controller) to be serviced by the network control program. These names appear on the PU (or CLUSTER or INNODE) macros associated with the line for which the service order table is being generated. A name may appear in the list one or more times. (Logical units cannot appear in the table; hence, omit names of LU macros.)

The maximum number of entries is 256.

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.

Note: The generation procedure checks the first 35 entries of the service order table to determine if the devices referred to by these entries are represented by PU, CLUSTER, or INNODE macros associated with the LINE macro for which the service order table is being defined. The converse is also true: each PU, CLUSTER, and INNODE macro associated with the LINE macro is checked to determine if it is represented in the service order table. A diagnostic message is issued if a discrepancy exists. Any service order table entries beyond 35 are not checked in this manner.

LUPOOL Macro Instruction (VTAM Users Only)

The LUPOOL macro instruction specifies a pool of logical units the network control program uses in communicating with logical units associated with physical units reached over switched SDLC links. (Logical units for physical units on nonswitched links are defined by LU macros associated with the PU (or CLUSTER) macro that represents the physical unit.)

Upon establishing a connection with a switched physical unit, the network control program allocates, from the pool, a logical unit for each logical unit contained within that physical unit. VTAM specifies to the network control program the number of logical unit control blocks to allocate and also supplies the logical unit parameters for each logical unit. Upon completion of the transmission between the network control program and the physical unit, the program releases each logical unit to the pool.

One and only one LUPOOL macro is required if the network configuration includes SDLC physical units reached over switched communication facilities—that is, the source statements include at least one GROUP macro in which LNCTL=SDLC and DIAL=YES are specified.

Name	*Operation*	*Operands*
`[symbol]`	`LUPOOL`	`NUMBER=count`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`NUMBER=count`

Specifies the number of logical units to be included in the pool. The minimum is 1. The maximum is the total number of resources that can be defined in the network control program (as determined by the value chosen for the MAXSUBA operand of the BUILD macro) minus the number of resources defined in the LINE, PU, LU, CLUSTER, and INNODE macros.

Example: If the MAXSUBA operand specifies 31 subareas, the maximum number of resources in the network controlled by VTAM is 2045, assuming that the network includes SDLC stations and BSC and/or start-stop stations. If the total number of resources (equivalent to the total number of macros listed above) is 1000, the maximum value you may specify in the NUMBER operand is 1045 (2045-1000).

To determine the highest value you need specify in this operand, assume that (1) the network control program is communicating simultaneously with SDLC physical units over all switched SDLC links, and (2) the physical units involved are those having the highest number of logical units. The total number of logical units in all connected physical units represents the greatest possible demand that can be placed on the logical unit pool. By specifying this value you can assure that the logical unit pool is never depleted.

In practice, the value of NUMBER can be less, since the assumptions above represent the most extreme conditions which may never, or only seldom, occur. Specifying too low a value, however, may result in depletion of the pool. If, after a physical connection is established with a switched physical unit, the network control program is unable to supply sufficient logical units from the pool, the program will break the physical connection before data transmission has begun. Then a new connection will have to be made and logical unit allocation attempted again. Occasional occurrences of this kind may be tolerable, but frequent occurrences demonstrates the need to increase the value specified in this operand. Thus, the circumstances unique to a given installation should be analyzed to arrive at an appropriate value.

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 hierarchical level within the arrangement. Each element is represented by a specific macro instruction within the category of macros called teleprocessing network configuration macro instructions.

The highest element is the *communication line* (or *link*). Each line in the network is represented by a LINE macro, whether the line is nonswitched multipoint, nonswitched point-to-point, or switched point-to-point.

SDLC Stations

The macro hierarchy for SDLC stations other than communications controllers (for example, IBM 3600, 3650, 3660, 3767, 3771, 3773, 3774, 3775, 3776) is LINE, PU, LU, for nonswitched links and LINE, PU for switched links. A single PU macro represents the physical unit within the SDLC station.

If the network control program communicates with the physical unit over a *nonswitched* SDLC link, one or more LU macros following the PU macro represent the logical units associated with the physical unit. (A logical unit is an application program within the physical unit together with the terminals associated with that program.)

If, for example, one physical unit having four logical units were attached to a nonswitched SDLC link, the following sequence would be required:

```
LINE
  PU
    LU
    LU
    LU
    LU
```

If the network control program communicates with the physical unit over *switched* facilities, the logical units are not represented by LU macros, and the macro sequence is simply LINE, PU. The network control program logical units are allocated dynamically from the pool of such units defined by the LUPOOL macro.

Note: In version 3 of the network control program, SDLC type 2 physical units are represented by CLUSTER macros. All SDLC stations added to an existing (version 3) NCP source program that is re-generated under version 4 should be represented by PU macros. For compatibility with the previous version (3), version 4 continues to accept CLUSTER macros; however, any subsequent changes or additions to physical unit parameters will be accommodated only in the PU macro.

Communications Controller Attached to SDLC Link

Each 3704 or 3705 communications controller connected by an SDLC link to another such controller is represented by a PU macro following the LINE macro that represents the link. Thus, when defining a network control program for a local communications controller, you would code a PU macro for the remote communications controller, and when defining a network control program for the remote communications controller you would code a PU macro to represent the local communications controller:

In NCP for the local controller:

```
LINE
    PU  (represents the remote controller)
```

In NCP for the remote controller:

```
LINE
    PU  (represents the local controller)
```

Note: In version 3 of the network control program, communications controller physical units are represented by INNODE macros. All communications controllers added to an existing (version 3) source program that is to be re-generated under version 4 should be represented by PU macros. For compatibility with the previous version (3), version 4 continues to accept INNODE macros; however, any subsequent changes or additions to physical unit parameters will be reflected only in the PU macro.

GROUP Macro

To the hierarchies of LINE, PU, and LU macros must be added one or more GROUP macros. The GROUP macro represents a grouping of lines having certain characteristics in common. The grouping is referred to as a *physical* line group because it contains lines having certain "physical" attributes in common, such as the type of stations attached to them.

Each LINE macro must be associated with a GROUP macro that precedes it. Further, although not shown above, a SERVICE macro must directly follow each LINE macro representing a nonswitched SDLC link. (The SERVICE macro defines the service order table to be used for the link.)

Exception: Omit the SERVICE macro if the program being defined is a *remote* network control program and the LINE macro represents an SDLC link to the *local* controller. (The *local* network control program controls the SDLC link to the remote controller and thus requires a SERVICE macro.)

Adding the GROUP and SERVICE macros to the LINE, PU, LU sequence shown above under *SDLC Stations* results in:

```
GROUP
    LINE
    SERVICE
        PU
            LU
            LU
            LU
            LU
```

Similarly, the SDLC link between the communications controllers would be represented as follows:

In NCP for the local controller:

```
GROUP
    LINE
    SERVICE
        PU
```

In NCP for the remote controller:

```
GROUP
    LINE
        PU
```

Notice that the SERVICE macro appears only in the local NCP, because the local communications controller is the *primary* station on the SDLC link. In the remote NCP, executed in the *secondary* station, the SERVICE macro must be omitted.

See the description of the GROUP macro later in this chapter for the attributes that communication lines must have in common in order to appear within the same line group.

Example of Teleprocessing Network Configuration Macro Instructions

Consider the SDLC network configuration shown in Figure 3-6.

The network attached to the local communications controller comprises the following communications lines:

Line 1:	Nonswitched multipoint SDLC link, two IBM 3600 systems, with four logical units each (SDLC stations)
Line 2:	Nonswitched point-to-point SDLC link, one 3704 communications controller (remote communications controller)

The network attached to the remote communications controller comprises:

Line R1:	Nonswitched multipoint SDLC link, three IBM 3790 systems, with three logical units each.

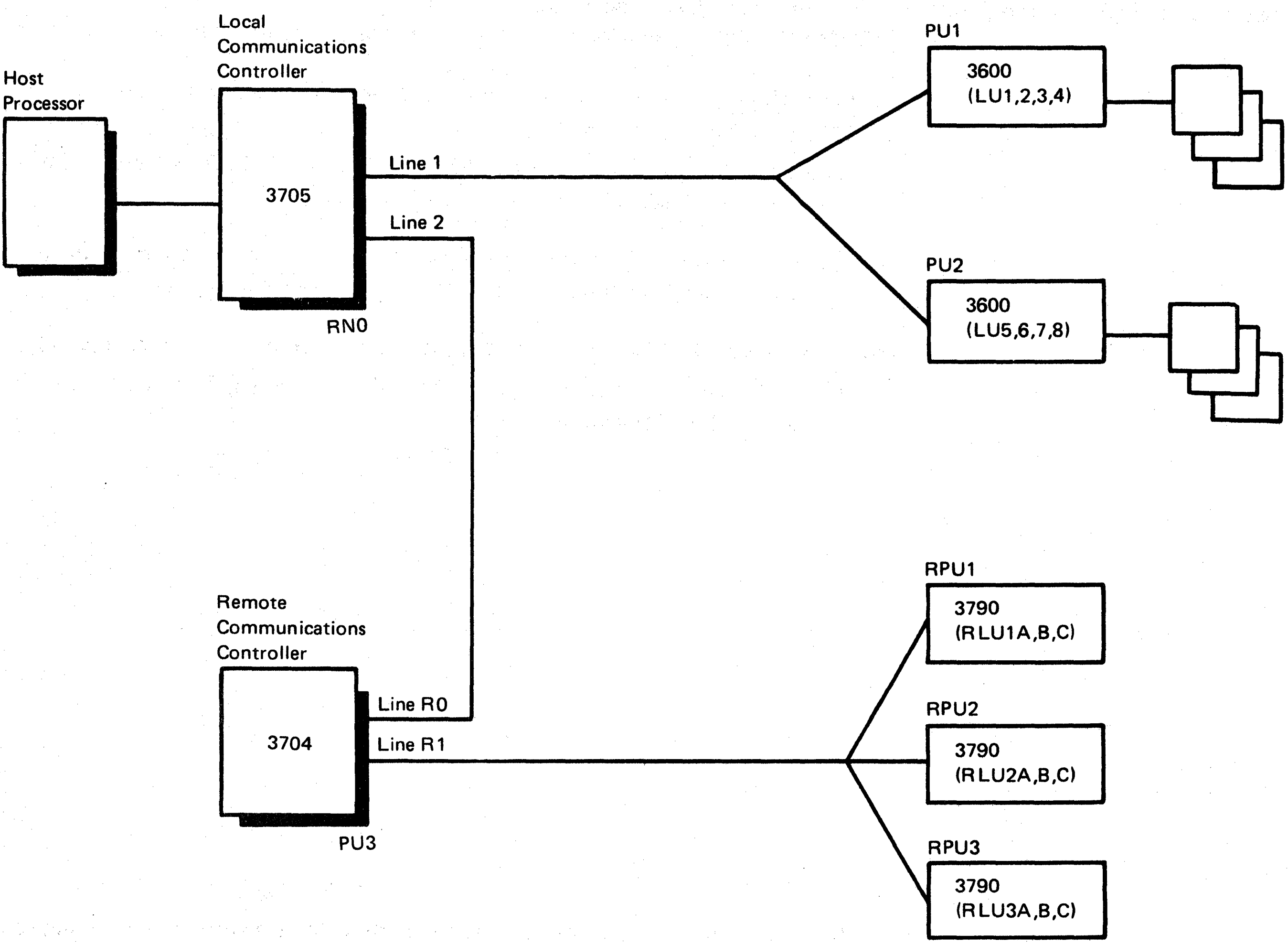


Figure 3-6. Example of SDLC Network Configuration

The teleprocessing network configuration macros required to represent the network attached to the *local* communications controller are as follows (not all required operands are shown):

```
G1     GROUP  LNCTL=SDLC
L1        LINE POLLED=YES
S1        SERVICE ORDER=(PU1,PU2)
PU1          PU  PUTYPE=2
LU1             LU
LU2             LU
LU3             LU
LU4             LU
PU2          PU  PUTYPE=2
LU5             LU
LU6             LU
LU7             LU
LU8             LU

G2     GROUP  LNCTL=SDLC
L2        LINE
S2        SERVICE ORDER=(PU3)
PU3          PU  PUTYPE=4 (represents remote controller)
```

The network control program defined for the *remote* communications controller would include:

```
RG1    GROUP  LNCTL=SDLC
RL1       LINE  POLLED=YES
RS1       SERVICE ORDER=(RPU1,RPU2, RPU3)
RPU1         PU  PUTYPE=2
RLU1A           LU
RLU1B           LU
RLU1C           LU
RPU2         PU  PUTYPE=2
RLU2A           LU
RLU2B           LU
RLU2C           LU
RPU3         PU  PUTYPE=2
RLU3A           LU
RLU3B           LU
RLU3C           LU

RG0    GROUP  LNCTL=SDLC
RL0       LINE
RN0          PU PUTYPE=4 (represents local controller)
```

See Appendix I for other sample programs that illustrate macro coding sequences.

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 SDLC stations on a link can be specified once, in the LINE macro, rather than in each of the fifteen individual PU 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 next higher level.

If, for example, only twelve of the fifteen stations 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 PU macros for the three exceptional stations.

Note: If you code only one suboperand (parameter) in an operand that can have two or more suboperands, that operand completely overrides an equivalent operand specified in a higher-level macro. The default values are assumed for the suboperands omitted at the lower level.

Summary of Macro Instructions and Operands

Figure 3-7 lists all of the operands of the network configuration macros for SDLC links. The asterisk indicates the macro under which the description of the operand appears. The bullet (•) indicates other macros in which the operand may be coded.

Note: This summary indicates only (1) the name of the macro under which the description of each operand can be found (indicated by *) and (2) other macros (indicated by ○) in which the operand can be coded. This summary does not indicate the conditions (for example, type of line control, type of station) under which use of the operand is appropriate. For this information, see the individual macro descriptions later in this chapter.

VTAM-Only Operands

Some operands convey no information to the network control program generation assembly process but must nonetheless appear in the NCP generation input deck that serves as input to the VTAM initialization procedure. (Refer to the *VTAM System Programmer's Guide* for information on this procedure.) Such operands are called *VTAM-only* operands. The macro assembly step of the NCP generation procedure permits each VTAM-only operand to appear in the macros indicated by a V in the table of Figure 3-7. The assembly process does not check such operands for proper syntax, however, nor does it verify that any related operands are present or absent. For the meanings of VTAM-only operands, see the *VTAM System Programmer's Guide.*

	Macro Instruction					
	GROUP	LINE	PU	INNODE	CLUSTER	LU
ACTIVTO	*					
ADDR			*	*	*	
ADDRESS		*				
ANSWER	V	V				
ANSTONE	●	*				
AUTO		*				
BATCH	●	●	●			*
BNNSUP	●	●	*			
BUFLIM	V	V	V			V
CALL	V	V				
CLOCKNG	●	*				
CONFIG	●	*				
DATMODE	●	●	*	*		
DATRATE	●	*				
DIAL	*					
DISCNT	V	V	V			
DUPLEX	●	*				
FEATUR2	V	V				
INTPRI	●	*				
IRETRY	●	●	*	*	*	
ISTATUS	V	V	V		V	V
LNCTL	*					
LOCADDR						*
LOGAPPL	V	V	V		V	V
LOGTAB	V	V	V		V	V
MAXDATA	●	●	*		*	
MAXLU	●	●	*			
MAXOUT	●	●	*	*	*	
MODETAB	V	V	V			V
NEW SYNC	●	*				
NRZI	●	*				
PACING	●	●	●		●	*

Figure 3-7. Summary of Operands for SDLC-Only Configuration Macro Instructions (Part 1 of 2)

	Macro Instruction					
	GROUP	LINE	PU	INNODE	CLUSTER	LU
PASSLIM	●	●	*	*	*	
PAUSE	●	*				
POLLED	●	*				
PUTYPE	●	●	*			
REPLYTO	*					
RETRIES	●	*	*	*	*	
RING	●	*				
SERVLIM	●	*				
SPDSEL	●	*				
SPEED	●	*				
SSCPFM	V	V	V			V
SUBAREA			*	*		
TADDR	●	*				
TRANSFR	●	*				
USSTAB	V	V	V			V
VIDSEQ	V	V				
VPACING	V	V			V	V

Figure 3-7. Summary of Operands for SDLC-Only Configuration Macro Instructions (Part 2 of 2)

Macro Instructions for SDLC Operations

This section contains the teleprocessing network configuration macro instructions (GROUP, LINE, PU, CLUSTER, LU and INNODE) to be used in specifying an SDLC-only network control program.

GROUP Macro Instruction

An SDLC line group consists of lines that have the following characteristics in common:

- All lines in the group are nonswitched point-to-point, or nonswitched multipoint, or switched point-to-point.
- All lines in the group are polled, or all are nonpolled.

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

For each group of SDLC links, one GROUP macro is required.

All GROUP macros for SDLC line groups must appear in the source program following any GROUP macros for BSC and/or start-stop line groups.

The GROUP macro indicates the beginning of a sequence of LINE, (SERVICE,) PU, and LU (or LINE, [SERVICE,] CLUSTER, and LU; or LINE, [SERVICE,] and INNODE) 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.

| (VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Name	*Operation*	*Operands*	
symbol	GROUP	LNCTL=SDLC	
		[,ACTIVTO={count}] {NONE }	(remote NCP only)
		[,DIAL={YES}] {NO }	
		[,REPLYTO={count}] {NONE }	
		VTAM-only operands:	
		ANSWER= BUFLIM= CALL= DISCNT= FEATUR2= ISTATUS= LOGAPPL= LOGTAB= MODETAB= SSCPFM= USSTAB= VIDSEQ= VPACING=	

`symbol`

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

`LNCTL=SDLC`

Specifies that the line group consists of SDLC links.

This operand is required.

```
[ACTIVTO={count}]
        {NONE }
```

(applicable only in a remote network control program)

Specifies the interval, to the nearest tenth of a second, that the remote NCP will allow to elapse since the most recent communication from the local NCP over the SDLC link before the remote NCP (1) begins automatic network shutdown and configuration restart or (2) abnormally ends, resulting in re-loading (IPL) of the controller. The action taken depends on the combination of ACTIVTO and ANS operand options you select, as shown in Figure 3-8.

`count`

Specifies the timeout value, to the nearest tenth of a second, that the remote NCP will wait for a communication from the local NCP before initiating shutdown or abnormally ending. The minimum value is 60.0 seconds; the maximum is 420.0 seconds.

NONE

Specifies that the remote NCP will wait indefinitely for communication from the local NCP; that is, automatic network shutdown and configuration restart will not occur.

This operand is valid only if (1) this program is a remote network control program (TYPGEN=NCP-R is specified in the BUILD macro), (2) this GROUP macro represents the SDLC link to the local communications controller (LNCTL=SDLC is specified in this GROUP macro), and (3) PUTYPE= 4 is specified in the PU macro.

If you omit this operand and specify ANS=YES (or you omit the ANS operand), ACTIVTO=420.0 is assumed. If you omit this operand and specify ANS=NO, ACTIVTO=NONE is assumed.

The value specified for this operand must be carefully chosen (see the discussion of the effects of the activity timeout under *Backup Local-Remote SDLC Links*, in Chapter 2).

The following expression may be used as a guide in determining the appropriate ACTIVTO value:

$$\text{MO} - \text{IPL} \geq \text{ACTIVTO} \geq t{*}n \geq \text{SDO}$$

(≥ means "greater than or equal to")

MO is the maximum time the remote controller may be out of service owing to link failure.

IPL is the interval normally required to load (IPL) the remote controller.

ACTIVTO is the value specified in the ACTIVTO operand.

t and *n* are the values specified in the RETRIES operand of the PU (or INNODE) macro (within the *local* network control program) that represents the remote controller.

SDO is the amount of time normally consumed by "short-duration outages" of the local-remote link (outages caused by momentary loss of modem carrier, excessive noise, power surges, or other disruptive line conditions).

Action taken by remote NCP upon failure of SDLC link to local controller:

ACTIVTO=	*ANS=*	*Number of SDLC local-remote links*	*Action*
count	YES	One	ANS and configuration restart occur; remote NCP then awaits message traffic from local controller over the SDLC link.
count	YES	Two or more	ANS and configuration restart occur; remote NCP then awaits message traffic from local controller over all available SDLC links.
count	NO	One	Expiration of timeout causes abnormal end (Abend) of remote NCP; reloading (IPL) occurs over link (after repair).
count	NO	Two or more	Expiration of timeout causes abnormal end (Abend) of remote NCP; reloading (IPL) occurs over any available link.
NONE	NO	One, two, or more	Remote NCP waits indefinitely for traffic from local controller.
NONE	YES	One, two, or more	(Invalid combination)

Figure 3-8. Effect of ACTIVTO and ANS Operand Options on Recovery Action of Remote NCP upon Failure of Link to Local Controller

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

Specifies whether or not the lines in the group require switched line control procedures. If they do, code DIAL=YES. If they do not, code DIAL=NO or omit the operand.

Note: SDLC local-remote backup links, whether they comprise switched or nonswitched lines, use nonswitched line control procedures; for these links only DIAL=NO is valid.

```
[REPLYTO={count}]
         {NONE }
```

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

The REPLYTO operand is valid only if the lines in the group are polled (POLLED=YES is specified in the LINE macros [or in the GROUP macro]).

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

If you specify REPLYTO=NONE, no timeout occurs regardless of the time that elapses between sending to the station and receiving a response. The maximum value is 1632 seconds.

If you omit the REPLYTO operand, the network control program uses a timeout of 1.0 second. For most teleprocessing networks, this standard value is appropriate unless the system designer specifies a different value.

Restriction on Number of Time Intervals Specified

A maximum of 16 different time intervals may be specified in the network control program. This includes (1) seven standard intervals (including default values) that are common to all network control programs, and (2) intervals that you explicitly specify in the REPLYTO operand of the GROUP macro. If the total number of different time intervals is 16, the default values are used for all subsequent time specifications the generation procedure encounters in processing the source statements.

The standard time intervals for all network control programs are:

0.0 (immediate action required)
1.0 seconds
2.2 seconds
3.0 seconds
23.5 seconds
60.0 seconds
NONE (infinite time delay)

Specifying Lower-Level Operands in the GROUP Macro

In addition to the preceding operands, most operands of the LINE, PU, CLUSTER, INNODE, and LU macros can be specified in the GROUP macro instead of the individual macros mentioned. Figure 3-7 shows which of the lower-level operands you may specify in the GROUP macro.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the GROUP macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this GROUP macro description means only that the NCP generation assembly procedure accepts them as valid operands of the GROUP macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

LINE Macro Instruction

The LINE macro represents one SDLC link (line) attached to the communications controller.

The LINE macro specifies:

- Whether the communications facility of which the link is a part is half-duplex or duplex.
- The line interface address, within the communications controller, to which the link is attached.
- The line interface address, within the communications controller, to which the related automatic calling unit (ACU), if any, is attached.
- The speed of the line—that is, the rate (in bits per second) at which the controller and stations on the line transmit data.
- The interrupt priority of the line.
- Whether clocking of the communication line is provided by the modem (modem clocking) or by the communication scanner (business machine clocking).
- Whether the controller that will execute the program you are defining is a tributary station on the line represented by this macro, and if so, the addressing character to which the controller will respond.
- Whether the modem by which this line is attached to the controller, if a dual data rate modem, is to operate at the higher or the lower rate.
- Whether the modem to which this line is attached has the "new sync" feature.
- Whether the modem to which this line is attached is sensitive to transitionless bit streams or to repeated digit (binary 10) bit patterns.
- Whether (for a switched line) the modem operates in ring indicator mode (not applicable in U.S. and Canada).
- Certain procedural options the network control program is to use for the line.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only for the VTAM initialization process and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each LINE macro coded causes a resource name to be generated.

One LINE macro must be coded for each SDLC link connected to the controller. *Exception:* A single LINE macro instruction represents either one or two communication lines comprising an SDLC link. All LINE macros representing lines in a physical line group must appear between the GROUP macro representing that group and the next GROUP macro.

A remote communications controller may have one principal SDLC link to the local controller, and from one to three alternate ("backup") SDLC links to that controller. Each link is represented within the remote network control program by a LINE macro; therefore, up to four LINE macros representing SDLC links to the local controller may appear in the generation input deck for the remote program.

```
Name     Operation   Operands

symbol   LINE        ADDRESS={line addr          }
                             {(xmt addr,rcv addr)}

                     SPEED=(rate[,rate])

                     [,ANSTONE={YES}]
                               {NO }

                     [,AUTO=address]

                     [,CLOCKNG=({INT}[,{INT}])]
                               {EXT}  {EXT}

                     [,CONFIG={SW    }]
                              {NONSW}

                     [,DATRATE=({HIGH}[,{HIGH}])]
                                {LOW }  {LOW }

                     [,DUPLEX=({HALF}[,{HALF}])]
                               {FULL}  {FULL}

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

                     [,NEWSYNC=({YES}[,{YES}])]
                                {NO }  {NO }

                     [,NRZI={YES}]
                            {NO }

                     [,PAUSE={t  }]
                             {0.1}

                     [,POLLED={YES}]
                              {NO }

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

                     [,RING={YES}]
                            {NO }

                     [,SERVLIM=count]

                     [,SPDSEL={YES}]
                              {NO }

                     [,TADDR={char}]
                             {NONE}

                     [,TRANSFR=count]
```

Name	*Operation*	*Operands*
		VTAM-only operands:
		ANSWER= BUFLIM= CALL= DISCNT= FEATUR2= ISTATUS= LOGAPPL= LOGTAB= MODETAB= SSCPFM= USSTAB= VIDSEQ= VPACING=

```
symbol
```

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

```
ADDRESS={line addr          }
        {(xmt addr,rcv addr)}
```

Specifies the one or two line interface addresses for an SDLC link. Determine line addresses used in your communications controller from your IBM representative.

If the communications controller is to transmit and receive over the *same* path (regardless of whether the communication facility constituting the path is half-duplex or duplex, specify a single line interface address.
Example: ADDRESS=020.

If the communications controller is to transmit and receive over *separate* paths (regardless of whether the communications facility is half-duplex or duplex, specify *two* line interface addresses. *Example:* ADDRESS=(020,021).

Figure 3-9 gives the range of valid line addresses for this operand.

This operand is required.

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

Figure 3-9. Valid Line and Auto Call Interface Addresses for LINE Macro Instruction

SPEED=(rate[,rate])

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

If the modem at the controller is a dual-rate modem (whether the rate used is program selected or selected by a manual switch on the modem), specify the higher of the two rates.

If one line interface address is specified in the ADDRESS operand of this macro, specify SPEED=rate. If two line interface addresses (xmt addr, rcv addr) are specified in the ADDRESS operand, code SPEED=rate, if the speed for both sides of the link are the same. Code SPEED=(rate,rate) for the transmit and receive addresses, respectively, if the rates differ.

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

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

The maximum speed you may specify if the line is connected 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.

[ANSTONE={YES}]
{NO }

(call-in switched links only)

Specifies whether the network control program, upon answering a call over a call-in switched link, is to transmit an "answer tone" to the calling station to signify completion of the line connection.

It is necessary for the program to send the tone only when the modem that attaches the link to the communications controller does not itself send an answer tone. Most modems in the United States and Canada do provide the tone, therefore ANSTONE=NO is normally appropriate. If in doubt, consult the supplier or installer of the modem.

This operand is valid only for switched links used for incoming calls (DIAL=YES [or DIAL=NO and CONFIG=SW] is specified).

[AUTO=address]

Specifies that the auto call facility is present for the link represented by this LINE macro and gives the automatic calling unit (ACU) interface address. This may be determined from the system designer.

A link that the network control program is to use to call a station (that is, you code CALL=OUT or CALL=INOUT in this macro) may or may not be equipped with the ACU facility.

If the link is so equipped, code the ACU interface address in the operand (in hexadecimal, without framing [‘ ’] characters). For example, AUTO=020.

If the link does not have the ACU facility, omit the operand. Calls to stations over this line must in this case be made manually.

Figure 3-9 gives the range of valid addresses.

```
[CLOCKNG=( {INT} [,{INT}] )]
           {EXT}   {EXT}
```

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

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

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—CLOCKNG=INT or CLOCKNG=EXT.

If the ADDRESS operand specifies two line interface addresses specify only the first suboperand, if both sides of the link use internal clocking or both use external clocking. If one uses internal clocking and the other uses external, specify CLOCKNG=(INT,EXT) or CLOCKNG=(EXT,INT), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

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

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

```
[CONFIG={SW   }]
        {NONSW}
```

(remote NCP only)

Specifies, within a remote network control program (TYPGEN=NCP-R), whether the physical communication path making up the local-remote SDLC link (as specified by LNCTL=SDLC and POLLED=NO) is switched or nonswitched. (Each local-remote SDLC link is operated by the NCP as a nonswitched link, regardless of whether the physical path is nonswitched or switched.) If the path is switched (CONFIG=SW is specified), the remote NCP can monitor the link for ring indicator signals (if you specify RING=YES) and present an answer tone (if you specify ANSTONE=YES) when the remote controller is called by the local NCP.

This operand is valid only in a remote network control program (TYPGEN=NCP-R) and the link to the local controller is specified as LNCTL= SDLC, DIAL=NO, and POLLED=NO.

```
[DATRATE=( {HIGH} [, {HIGH} ] )]
           {LOW }    {LOW }
```

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

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—DATRATE=HIGH or DATRATE=LOW. 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.

If the ADDRESS operand in this LINE macro specifies two line interface addresses, specify only the first suboperand, if both sides of the link use the high rate or both use the low rate. If the rates differ, specify DATRATE=(HIGH,LOW) or DATRATE=(LOW,HIGH), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

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

If you omit this operand, DATRATE=LOW or DATRATE=(LOW,LOW) is assumed. If you omit the second subparameter, and the ADDRESS operand specifies two addresses, the same value is assumed for this subparameter as you specified for the first subparameter.

```
[DUPLEX=( {HALF} [, {HALF} ] )]
          {FULL}    {FULL}
```

Specifies whether the SDLC link and modem constitute a half-duplex or (full-)duplex facility. Determine from the system designer the appropriate value to code.

Note: This should not be confused with half-duplex or duplex data transfer. This operand specifies only the physical characteristic of the communications facility (lines and modems).

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—DUPLEX=HALF or DUPLEX=FULL.

If the ADDRESS operand specifies two line interface addresses, specify only the first suboperand, if both sides of the link are half-duplex or both are duplex; if one is half-duplex and the other is duplex, specify DUPLEX=(HALF,FULL) or DUPLEX=(FULL,HALF), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

Note: For SDLC links, this operand has no effect on the activation of the request-to-send signal; this signal is always active if the ADDRESS operand specifies two addresses and is active only when sending data if the ADDRESS operand specifies a single address.

```
         {0}   {0}
[INTPRI=( {1} [,{1}] )]
         {2}   {2}
         {3}   {3}
```

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

If the ADDRESS operand of this LINE macro specifies one line interface address, specify only the first suboperand.

If the ADDRESS operand of this LINE macro specifies two line interface addresses, specify only the first suboperand, if the priority is the same for both sides of the link. Specify both suboperands if the priority differs. (The first and second suboperands are the priorities for the first and second addresses, respectively.)

If this line is serviced by a type 2 or type 3 communication scanner, the valid range for INTPRI is 0 through 3. If this line is serviced by a type 1 communication scanner, the only valid values are 0 and 1, with 1 being the higher priority.

Appendix J gives a method for determining the interrupt priorities for each line in the network.

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

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

NEWSYNC=YES is valid only if (1) the modem (at the controller) serving the line has the new sync feature, *and* (2) if the communications controller is the multipoint master (not tributary) station for a duplex (*not* half-duplex) line on which multipoint line control is used, *and* (3) the modem at the remote station does not continuously send carrier signal to the modem at the controller. (The modem at the remote station sends carrier continuously if it is wired internally to do so [consult the installer or supplier of the modem if in doubt] or the remote station continuously sends the "request-to-send" signal to the modem [consult your IBM representative to determine if this is the case].)

NEWSYNC=NO is required if there is only one remote station on the communication line.

If you omit the NEWSYNC operand, NEWSYNC=YES is assumed if you specify POLLED=YES, DUPLEX=FULL, CLOCKNG=EXT, and DIAL=NO. (All four operands must be so specified.) If you omit the operand and any of the four remaining operands is not as shown, NEWSYNC=NO is assumed.

Determine from your IBM representative or the installer or supplier of the modem (if other than an IBM modem) whether the appropriate conditions above prevail. If they do not, the new sync function cannot be used.

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—NEWSYNC=YES or NEWSYNC=NO.

If the ADDRESS operand specifies two line interface addresses, specify only the first suboperand, if the new sync option is to be used for both sides of the link (NEWSYNC=YES) or for neither (NEWSYNC=NO). If the option is required for only one of the two, specify NEWSYNC=(YES,NO) or NEWSYNC=(NO,YES), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

Specifies whether the data terminal equipment (controller and remote station) at the ends of the SDLC link must operate in "non-return-to-zero-inverted" (NRZI) mode (NRZI=YES) or in "non-return-to-zero" (NRZ) mode (NRZI=NO).

If internal (business machine) clocking is used on the link, specify NRZI=YES or omit the operand.

If external (modem) clocking is used on the link, specify NRZI=YES or omit the operand *unless* the modem is sensitive to repeated binary 10 bit patterns (that is, 10101010...). (Sensitivity to this pattern can cause the modem to lose synchronism when it encounters sufficiently long sequences of this pattern in message data.) If the modem is sensitive to this bit pattern, specify NRZI=NO. To determine whether your modems are sensitive to this pattern, consult your IBM representative (for IBM modems) or the supplier or installer of the modem (for non-IBM modems).

Note: NRZI=NO is the proper choice for most non-IBM modems; NRZI=YES is appropriate for IBM modems.

Caution: All business machine equipment (terminal equipment) on the same SDLC link must use the same encoding scheme—that is, all use NRZI mode (NRZI=YES) or all use NRZ mode (NRZI=NO). Mixing of modes on the same SDLC link will result in total lack of communication between stations on the link. In the case of an SDLC link between a local and a remote communications controller, the equivalent choice (NRZI=YES or NRZI=NO) is required for the LINE macros in each program (local and remote) that represent the SDLC link, and must correspond to the setting of the NRZI bit in the IPL configuration data set of the remote program loader. (The IBM customer engineer sets the NRZI bit.)

This operand is valid only if LNCTL=SDLC is specified in the GROUP macro.

```
[PAUSE={t  }]
       {0.1}
```

Specifies the average duration of the polling cycle in seconds or seconds and tenths of seconds. The polling cycle extends from the moment polling begins with the first active entry in the service order table to the moment polling next begins at the same entry. If the time expended in a complete polling cycle–that is, in servicing all active entries in the service order table–equals or exceeds the time you specify as *t*, the next polling cycle begins immediately. If, on the other hand, the time expended in a complete polling cycle is less than *t*, the beginning of the next polling cycle is deferred until *t* seconds have elapsed since the beginning of the cycle just completed. During the pause, the SDLC link is in "poll-wait" state; any outgoing data ready for transmission to the SDLC stations on the link is sent during this pause.

Allowing a pause to elapse when activity on the link is relatively low can reduce the amount of processing time consumed by unproductive polling.

Note: The larger the number of active entries in the service order table, the greater the likelihood will be that polling cycles will proceed continuously, without intervening pauses.

t may be from 0 to 25.5; if you omit this operand, a pause of 0.1 seconds is assumed.

This operand is not valid if the network control program communicates with a remote communications controller over this line.

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

Specifies whether or not the stations attached to the SDLC link must be polled and addressed by the controller in which this network control program will be executed.

Note: All lines in an SDLC line group must be specified as POLLED=YES, or all must be specified as POLLED=NO. Both options cannot be included in the same line group.

Code POLLED=YES if the controller in which this network control program is executed is the primary station on the SDLC link.

Code POLLED=NO if the controller in which the program is executed is the *secondary* station on the link. (POLLED=NO is valid only if the program is a remote NCP [TYPGEN=NCP-R], the link represented by this LINE macro is a nonswitched line [DIAL=NO], and the physical unit on the link is a type 4 [PU macro, PUTYPE=4, or an INNODE macro represents the unit].)

If you omit this operand, the line is a nonswitched SDLC link (LNCTL=SDLC, DIAL=NO), and:

- This is the first LINE macro following the GROUP macro, then POLLED=NO is assumed if the program is to be executed in a remote controller (TYPGEN=NCP-R). Otherwise, POLLED=YES is assumed.
- This is not the first LINE macro following the GROUP macro, then the value assumed is the same as that specified in the first LINE macro.

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

(applicable to network control mode only)

Specifies the number of attempts, via retransmission, to recover from text errors in message data transmitted over the link specified by this LINE macro.

Text-write errors: When an error occurs while sending to the station (text-write errors), the network control program retransmits the block on which the error occurred. If the error recurs, retransmission is repeated. The network control program persists in retransmitting until it successfully sends the block or until a maximum number of retries (retransmissions) has occurred. This maximum is specified as the *m* parameter of the RETRIES operand. *m* may be from 0 to 128. RETRIES=0 specifies no retry attempts, and is equivalent to specifying RETRIES=NONE.

Optionally, you may also specify that the network control program pause after completing the retry sequence, then begin a new retry sequence. This second sequence continues until the block is successfully transmitted or the maximum, *m*, is again reached. Alternation of retry sequence and pause continues until the error is cleared or the maximum number of retry sequences is reached. The pause, specified by the *t* parameter, may be from 1 to 255 seconds. The maximum number of retry sequences, specified by *n*, may be from 1 to 127.

The maximum number of retries—$(n+1)*m$—is 128. The maximum number of retries per sequence, *m*, is always specified in the RETRIES operand of the LINE macro.

You may specify the *t* and *n* parameters individually for each such station connected to the SDLC link represented by this LINE macro. (Specify these values in the RETRIES operand of the PU [or CLUSTER or INNODE] macro.)

The *t* and *n* parameters can be specified also in a PU (or INNODE) macro associated with a LINE macro for a backup SDLC local-remote link (that is, a PU or INNODE macro from which the SUBAREA operand is omitted).

Text-read errors: For an error occurring when receiving from a station (text-read error), the network control program sends a negative response to the station, causing the station to retransmit the block in error. If the error recurs, the network control program again sends a negative response and receives the block again. The network control program persists in this way until (1) the block is successfully received, or (2) the station sends an EOT character or sequence instead of retransmitting the block, or (3) the network control program has sent a maximum number of negative responses. This maximum is specified as the *m* parameter of the RETRIES operand. *m* may be from 0 to 255. RETRIES=0 specifies no retry attempts, and is equivalent to RETRIES=NONE. RETRIES=255 specifies unlimited attempts; that is, no limit to the number of times the program sends a negative response to the station. Values from 1 through 254 indicate a specific number of retries.

The parameters *t* and *n* do not apply to text-read errors.

Maximum, Minimum, and Omitted Values:

The maximum number of retries (retransmissions) for text-write errors equals $(n+1)m$; the maximum number for text-read errors is *m*. (If *m* is 255, however, there is no maximum.)

If *m* is 0 or 255 (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 *n* is omitted and *t* is *not* 0: $n=1$ is assumed if $(n+1)m \leq 128$, and $n=0$ is assumed if $(n+1)m > 128$.

If *n* is omitted and *t* is 0, $n=0$ is assumed.

If you code RETRIES=NONE, no error recovery is attempted for read or write text errors.

If you omit this operand entirely, the assumed values will be $m=7$, $t=0$, $n=0$.

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

(switched links only; not applicable to U.S. and Canada)

Specifies whether or not the ring indicator mode of automatic answer operation is to be used for the switched link represented by this LINE macro. This depends solely upon the type of modem (data set) that connects the link to the controller. Determine from the modem supplier or installer whether it has a "ring indicator interface" lead.

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

The RING operand is valid only for a switched line (DIAL=YES is specified or [in a remote NCP only] DIAL=NO and CONFIG=SW are specified).

[SERVLIM=count]

(multipoint links only)

Specifies the maximum number of regular scans of the service order table that the network control program will make for normal servicing of physical and logical units on the link before it makes a special scan of the table. The regular scan of the table accommodates normal transmission of path information units between the access method or host application programs and physical or logical units on the link. In the special scan of the service order table, the network control program determines whether there are any outstanding commands from the access method to interrogate or alter the on-line status of any physical units on the link. If so, the program fulfills the first such command, then resumes regular scans of the table to perform normal servicing. If no status commands are outstanding, the program immediately resumes regular scans unless, in the previous regular scan, the program found no stations to be in the "contacted" state–that is no station is presently active. In this case, resumption of regular scans occurs after a delay of 2.2 seconds. If more than one status command is outstanding, only one is honored each time the special scan is made; the remaining status commands are fulfilled one at a time, in turn, during subsequent special scans of the table.

Upon completing a regular scan, the program begins the special scan when (1) in the regular scan just completed the program found no active stations, or (2) the maximum number of regular scans specified by SERVLIM has been reached–whichever occurs first.

Specifying a low value in SERVLIM gives the network control program more frequent opportunities to fulfill accumulated status commands than does specifying a higher value. Such status commands can be fulfilled more promptly, but at the cost of frequent interruptions to normal servicing. Conversely, specifying a higher value in SERVLIM causes fewer interruptions to normal servicing of stations than does a lower value, but delayed fulfillment of the status commands is more likely to result. The relative number of status commands the access method will issue for the link served by the service order table, the relative importance of the alternatives described above, and experience should influence your selection of a value for the SERVLIM operand.

Caution: The network control program will perform a timeout for any status command issued for the physical unit of an SDLC station whose power is off. The duration of this timeout interval is as specified in the REPLYTO operand of the GROUP macro.

Normal servicing of physical and logical units is interrupted during the timeout interval because regular scanning of the service order table is suspended. Only when the specified interval expires does regular scanning (and therefore normal servicing) resume. A low value for SERVLIM (which causes relatively frequent special scans) and/or a high value for REPLYTO (resulting in long timeout delays) will result in serious degradation of message throughput on the SDLC link if status commands are received for SDLC stations whose power is off. (The timeout interval recurs for each successive special scan.)

You may minimize the possibility of throughput degradation by specifying a high value in the SERVLIM operand and/or a low value in the REPLYTO operand. Alternatively, the risk may be averted by arranging network operating procedures to avoid sending commands to physical units whose power is off.

If you omit this operand, a value of 4 (four regular scans of the table) is assumed.

This operand is valid only for a nonswitched link for which DIAL=NO and POLLED=YES are specified in the GROUP and LINE macros, respectively.

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

Specifies, for a link connected to a modem capable of transmitting at either of two data rates, whether the data rate may be changed by request from the access method.

Specify SPDSEL=YES if you wish to allow the program to change the data rate of the modem upon request from the access method. SPDSEL=YES is valid only if external clocking is used on the link represented by this LINE macro (CLOCKNG=EXT is specified). The data rate specified in the SPEED operand must be the higher of the two data rates.

Do not specify SPDSEL=YES if the modem used on this link has only a single data rate.

Specify SPDSEL=NO (or omit the operand) if the modem has one data rate or if it has two data rates but the network control program is not to change the rate on request from the access method.

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

(remote controller only)

Specifies, if this program is to be executed in a remote controller on an SDLC link, the one-character address you wish to assign to this controller.

Code TADDR=char, where *char* is the two-digit hexadecimal representation of a single, EBCDIC SDLC station address (the address of the remote controller). You may assign as the address any bit configuration except hexadecimal 00 or FF.

However, both the address you specify in the TADDR operand and the address you specify in the ADDR operand of the PU (or INNODE) macro (within the local NCP) representing the remote controller must be identical to the physical address entered into the IPL configuration data set in the diskette contained within the remote controller. (This address is entered by the customer engineer.)

This operand is valid only if POLLED=NO is specified in this LINE macro.

```
[TRANSFR=count]
```

Specifies the maximum number of buffers that the network control program will obtain and fill with message data from an SDLC station before transferring them to the host processor. If no ending characters have been received from the station by the time the specified number of buffers are filled with message data, the network control program discards all the data received and sends a negative acknowledgment to the station. (The discarded data is *not* sent to the host processor.)

The minimum value of *count* is 1. The maximum value is the smaller of (1) 255 or (2) the result of multiplying the values of the MAXBFRU and UNITSZ operands of the HOST macro, subtracting the value of the BFRPAD operand of the HOST macro, and dividing the result by the buffer size specified in the BFRS operand of the BUILD macro. Expressed as a formula:

$$\text{count} \leq \frac{(\text{MAXBFRU})(\text{UNITSZ})-\text{BFRPAD}}{\text{BFRS}}$$

If you omit this operand, the network control program uses the lesser of (1) 255 or (2) the result of the calculation expressed by the preceding formula.

Specifying Lower-Level Operands in a Higher-Level Macro

In addition to the preceding operands, most operands of the PU, CLUSTER, and INNODE macros can be specified in the LINE macro (or the GROUP macro) instead of the individual macros mentioned. Figure 3-7 shows which of the lower-level operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the LINE macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this LINE macro description means only that the NCP generation assembly procedure accepts them as valid operands of the LINE macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

CLUSTER Macro Instruction

The CLUSTER macro represents an SDLC station of the clustered type.

Note: In version 3 of the network control program, cluster controller physical units are represented by CLUSTER macros. For compatibility with version 3, version 4 and 5 accepts CLUSTER macros and for this reason, operands pertaining to SDLC stations are shown in this CLUSTER macro description. However, when defining a new source program to be generated under NCP version 4 or 5, or when adding new SDLC stations to an existing version 3 source program to be re-generated under version 4 or 5, use PU macros, not CLUSTER macros, to represent the cluster controllers.

The CLUSTER macro specifies:

- The address of the SDLC station.
- Whether immediate polling retry is required.
- The maximum number of PIUs or PIU segments the network control program will send to the SDLC station (physical unit) before awaiting a response from the station.
- The maximum number of contiguous path information units the network control program will send at one time to the station.
- The number of error recovery attempts, via retransmission, the network control program will make when transmission errors occur.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Name *Operation* *Operands*

```
symbol  CLUSTER     ADDR=chars

                    [,IRETRY={YES}]
                             {NO }

                    [,MAXDATA=size]

                    [,MAXOUT={n}]
                             {1}

                    [,PASSLIM={n}]
                              {1}

                    [,RETRIES=( [t[,n]] )]
```

VTAM-only operands:

```
                    FEATUR2=
                    ISTATUS=
                    LOGAPPL=
                    LOGTAB=
                    VPACING=
```

`symbol`

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

`ADDR=chars`

Specifies the hexadecimal representation of the eight-bit address of the SDLC clustered station represented by this CLUSTER macro.

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

Specifies whether or not the network control program, when an “idle detect timeout” condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program re-polls the control unit; otherwise, the program bypasses the unsuccessfully polled control unit and services the station represented by the next entry in the service order table.

```
[MAXDATA=size]
```

Specifies the maximum amount of data, in bytes (including the transmission header and request/response header), that the physical unit can receive in one data transfer—that is, one entire PIU or a PIU segment.

To determine the amount of data that the physical unit can receive, consult the publications for the specific type of SDLC station represented by this CLUSTER macro.

The maximum amount of user data the network control program sends to the physical unit in one data transfer is the value of MAXDATA minus nine bytes (three bytes for the request/response header, and six bytes for the transmission header).

If you omit this operand, the program sends the station a maximum of one network control program buffer, plus six bytes.

```
[MAXOUT={n}]
        {1}
```

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the SDLC cluster control unit represented by this CLUSTER macro before requesting a response from the control unit. The minimum is one; the maximum is seven. The network control program will send one PIU or segment if you omit this operand.

```
[PASSLIM={n}]
         {1}
```

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the SDLC cluster control unit represented by this CLUSTER macro for any one appearance of the control unit in the service order table. The minimum is one PIU or segment; the maximum is 254.

If you omit this operand, the program sends one PIU or segment at a time.

```
[RETRIES=( [,t[,n]] )]
```

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts—via retransmission—to recover from text errors in message data sent to the SDLC cluster control unit specified by this CLUSTER macro.

The meanings of the *t* and *n* parameters are the same as for those parameters in the RETRIES operand of the LINE macro.

Specifying these parameters in the CLUSTER macro allows you to designate different values for each cluster control unit attached to the line.

If you specify RETRIES=NONE in the LINE macro for the SDLC link to which the SDLC cluster control units are connected, no error recovery is attempted for read-text or write-text errors.

Specifying Lower-Level Operands in a Higher-Level Macro

In addition to the preceding operands, the PACING operand of the LU macro can be specified in the PU, CLUSTER, LINE, or GROUP macro, instead of the LU macro.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the CLUSTER macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this CLUSTER macro description means only that the NCP generation assembly procedure accepts them as valid operands of the CLUSTER macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

PU Macro Instruction

The PU macro represents a physical unit of any type (1, 2, or 4) with which the network control program communicates over a nonswitched or switched SDLC link.

The PU macro specifies:

- The station address of the physical unit.
- The physical unit type (1, 2, or 4).
- The maximum amount of data the physical unit can receive from the network control program in one segment.
- The maximum number of consecutive path information units (PIU) or PIU segments the network control program will send to the physical unit before servicing other physical units on the link.
- Whether immediate polling retry is required.
- The number of error recovery attempts the network control program will make when transmission errors occur.
- The maximum number of path information units or PIU segments the network control program will send to the physical unit before requesting a response from the unit.
- The subarea address of the physical unit (type 4 only).

The sequence of PU macros (with intervening LU macros, if required) must appear directly following the SERVICE macro that defines the service order table for the SDLC link:

```
LINE POLLED=YES
SERVICE
PU
LU
LU
PU
LU
LU
 .
 .
 .
```

If the local and remote controllers (physical unit type 4) can communicate over one or more backup (alternate) SDLC links in lieu of the principal link, a PU macro must appear after each LINE (or SERVICE) macro corresponding to an alternate link. Omit the SUBAREA operand from each of these alternate PU macros. Also, if the alternate PU macro is coded in a *local* network control program (and thus represents the remote controller), omit all other PU macro operands except IRETRY and RETRIES; or, if the alternate PU macro is coded in a *remote* network control program (and thus represents the local controller), only the MAXOUT and PASSLIM operands are valid; they should specify the same values as these operands specify in the *principal* PU macro.

Exception: If the PU macro appears in the network control program executed in the controller that is the secondary station on the SDLC link (and the PU macro thus represents the primary station on the link), no SERVICE macro is used. The sequence then is simply:

```
LINE
PU
```

Example:

Local communications controller: Remote communications controller:

```
        Local NCP (subarea 4)             Remote NCP (subarea 5)

L1      LINE    POLLED=YES,       LA      LINE    POLLED=NO
                RETRIES=5

                                  LC1     PU      SUBAREA=4,
                                                  PUTYPE=4,
        SERVICE ORDER=(RC1)                       DATMODE=FULL,
                                                  MAXOUT=7,
RC1     PU      SUBAREA=5,                        PASSLIM=200
                PUTYPE=4,
                DATMODE=FULL,     LAALT   LINE    POLLED=NO
                MAXOUT=7,
                PASSLIM=200,      LC1ALT  PU      PUTYPE=4,
                RETRIES=(10,5)                    MAXOUT=7,
                                                  PASSLIM=200

L1ALT   LINE    POLLED=YES,
                RETRIES=5

        SERVICE ORDER=(RC1ALT)

RC1ALT  PU      PUTYPE=4,
                RETRIES=(10,5)
```

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only for the VTAM initialization procedure and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each PU macro coded causes a resource name to be generated.

```
Name        Operation    Operands

symbol      PU           [ADDR=chars]

                         [,BNNSUP=3270]

                         [,DATMODE={HALF}]
                                   {FULL}

                         [,IRETRY={YES}]
                                  {NO}

                         [,MAXDATA=size]

                         [,MAXLU=count]

                         [,MAXOUT={n}]
                                  {1}

                         [,PASSLIM={n}]
                                   {1}

                                  {1  }
                         [,PUTYPE={2  }]
                                  {4  }
                                  {1,2}

                         [,RETRIES=(,[t][,n])]

                         [,SUBAREA=n]

                         VTAM-only   operands:
                         BUFLIM=
                         DISCNT=
                         ISTATUS=
                         LOGAPPL=
                         LOGTAB=
                         MODETAB=
                         SSCPFM=
                         USSTAB=
                         VPACING=
```

`symbol`

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

`[ADDR=chars]`

Specifies the hexadecimal representation of the eight-bit address of the physical unit represented by this PU macro. Any bit configuration except hexadecimal 00 or FF is valid. This operand is invalid if (1) POLLED=NO is specified in the LINE macro; (2) POLLED=YES and PUTYPE=4 are specified but the SUBAREA operand is omitted; or (3) DIAL=YES is specified in the GROUP macro.

This operand is required if the communication facility is nonswitched (DIAL=NO), the physical units on the link must be polled (POLLED=YES), and the physical unit type is 1, or 2, or 4 (and the SUBAREA operand is specified).

Note: For a type 4 physical unit, the address specified in the ADDR operand in the *local* network control program and the address specified in the TADDR operand of the LINE macro within the remote network control program must be identical to the physical address of the remote controller. (The customer engineer enters the physical address into the IPL configuration data set in the diskette contained within the remote controller.)

[BNNSUP=3270]

Specifies that the physical unit represented by this PU macro is a 3270 (operating in SDLC mode).

This operand is invalid if the line is switched (DIAL=YES) or if the physical unit type is not 1.

```
[DATMODE={HALF} ]
         {FULL}
```

Specifies, for a communications controller (type 4 physical unit) on a nonswitched SDLC link that allows simultaneous transmission in both directions, whether the network control program is to communicate with the distant controller in half-duplex or duplex data mode. This operand is valid only if the ADDRESS operand of the corresponding LINE macro specifies both a receive and a transmit address and the LINE macro represents the principal (not backup) SDLC link.

If you (1) omit this operand, (2) code the SUBAREA operand, and (3) specify, in the LINE macro, both a receive and a transmit address, then DATMODE=FULL is assumed.

This operand is invalid if (1) the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R), or (2) the program is to be executed in a local controller (TYPGEN=NCP-LR) and the SUBAREA operand is omitted from this macro, or (3) the physical unit type is not 4.

Minimizing Line Turnaround Delay:

Each reversal in direction of message transmission over an SDLC link causes a line turnaround delay of several milliseconds if the carrier signal transmitted by the modem is interrupted. Efficiency of the link is thus lower than if the carrier signal is transmitted continuously, because no message data can be transmitted during these intervals.

Interruption of the carrier signal, and hence line turnaround delays, are inherent and unavoidable if the communications facility (including lines and modems) is half-duplex. They may also occur, however, in a duplex facility and in this case they may be prevented by ensuring that the carrier signal is transmitted continuously. Continuous carrier transmission results from either (1) continuous activation of the "request-to-send" signal sent from the communications controller to the modem, or (2) "strapping" (internally connecting) the modem to transmit carrier continuously independent of whether the "request-to-send" signal is activated.

The "request-to-send" signal (and hence the carrier sent by the modem) is activated continuously if: (1) the communication facility is duplex, (2) the line set to which the SDLC link is attached is duplex, (3) separate receive and transmit addresses are specified in the ADDRESS operand of the LINE macro, and (4) DATMODE=FULL is specified in the LINE macro.

The "request-to-send" signal is not activated continuously if the line set is half-duplex *or* if DATMODE=HALF is specified in the LINE macro. The line turnaround delay can still be minimized or eliminated, however, if the modem is strapped for continuous carrier operation. Consult the supplier or installer of the modem to determine if it is capable of continuous carrier operation and, if so, have the modem so connected. (The modem cannot be strapped for continuous carrier if the communication facility is half-duplex.)

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

Specifies whether or not the network control program, when an 'idle detect timeout' condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program re-polls the controller; otherwise, the program services the station represented by the next entry in the service order table.

This operand is valid only if POLLED=YES is specified in the LINE macro. If you omit this operand and the LINE macro specifies POLLED=YES, IRETRY=NO is assumed.

Note: You may specify IRETRY in a PU macro associated with either the principal or the backup SDLC link.

```
[MAXDATA=size]
```

(type 1 and 2 physical units only)

Specifies the maximum amount of data, in bytes (including the transmission header and request/response header), that the physical unit can receive in one data transfer—that is, one entire PIU or a PIU segment.

To determine the amount of data that the physical unit can receive, consult the publications for the specific type of SDLC station represented by this PU macro.

The maximum amount of user data the network control program sends to the physical unit in one data transfer is the value of MAXDATA minus five bytes (for a type 1 physical unit) or minus nine bytes (for a type 2 physical unit). (These values represent the lengths of the request/response header (three bytes) and the transmission header (two bytes for a type 1 physical unit; six bytes for a type 2 physical unit.)

The value you specify for MAXDATA, divided by the NCP buffer size (specified in the BFRS operand of the BUILD macro), determines the number of NCP buffers sent in one data transfer to the physical unit represented by this PU macro.

This operand is valid only for a nonswitched SDLC station of physical unit type 1 or 2.

If you omit this operand, BFRS+2 is assumed as the maximum amount of data for PUTYPE=1 and BFRS+6 is assumed for PUTYPE=2.

```
[MAXLU=count]
```

(switched link only)

Specifies, for a physical unit on a switched link (DIAL=YES), the maximum number of logical units associated with any physical unit that can communicate with the network control program over the link. The minimum is 1; the maximum is 255.

This operand is required if the physical unit is on a switched link (DIAL=YES); otherwise it is invalid.

```
[MAXOUT={n}]
        {1}
```

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the physical unit represented by this PU macro before requesting a response from the physical unit. The minimum is one; the maximum is seven.

If you omit this operand and specify DIAL=NO, (1) MAXOUT=1 is assumed if you specify POLLED=YES or (2) MAXOUT=7 is assumed if you specify POLLED=NO.

```
[PASSLIM={n}]
         {1}
```

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the physical unit represented by this PU macro.

If this physical unit is associated with a LINE macro in which POLLED=YES is specified, the program services the station represented by the next entry in the service order table when the pass limit value is reached.

If this PU macro is coded in a remote network control program and represents a local communications controller, the remote program stops sending to the local controller when the pass limit is reached. Transmission to the local controller resumes when the local network control program again polls the remote controller.

The minimum is one PIU or segment. The maximum is 254. The default value assumed if you omit this operand is 1 (PASSLIM=1).

If this PU macro appears in a remote network control program and represents a local communications controller, the maximum value is 254. This is also the recommended value and the default value.

```
         {1    }
[PUTYPE={2    }]
         {4    }
         {(1,2)}
```

Specifies the physical unit type constituting the SDLC station represented by this PU macro. PUTYPE=(1,2) specifies, only for a group of switched SDLC links, that physical units of either type 1 or type 2 can communicate with the communications controller over the same link.

PUTYPE=1 and PUTYPE=2 are valid only for polled links (POLLED=YES).

PUTYPE=1 and PUTYPE=2 are valid for either nonswitched or switched links (DIAL=NO or DIAL=YES); PUTYPE=4 is valid only for nonswitched links (DIAL=NO).

`[RETRIES=( [,t][,n] )]`

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts—via retransmission—to recover from text errors in message data sent to the physical unit represented by this PU macro.

The meanings of the *t* and *n* parameters are the same as for these parameters in the RETRIES operand of the LINE macro.

`[SUBAREA=n]`

(nonswitched SDLC link only; specified in local NCP only)

Specifies the subarea address assigned to the network control program in the remote communications controller (physical unit type 4) represented by this PU macro. (The network control program in each controller in the network must have a unique subarea address.) The value of *n* specified in this SUBAREA operand and in the SUBAREA operand of the BUILD macro for the remote NCP must be identical.

This operand is invalid for a switched SDLC link or a remote network control program.

The minimum valid subarea address is 1—that is, SUBAREA=1. The maximum address is the value you specify in the MAXSUBA operand of the BUILD macro.

If this PU macro and its associated LINE macro represent the principal SDLC link to a physical unit type 4 (remote communications controller), SUBAREA must be specified.

If this PU macro and its associated LINE macro represent a *backup* SDLC link to a remote controller, omit the SUBAREA operand and all other operands except PUTYPE, IRETRY, and RETRIES. (When communicating with the remote controller over the backup link, the network control program uses the same values as specified in the PU macro for the principal link except for the values of PUTYPE, IRETRY, and RETRIES.)

(At least one PU [or INNODE] macro in the source program must specify the SUBAREA operand if TYPGEN=NCP-LR is specified.)

Note: A backup SDLC link can be defined only if the line constituting the link is serviced by a type 2 (not type 1) communication scanner (TYPE=TYPE2 is specified in the CSB macro).

Specifying Lower-Level Operands in a Higher-Level Macro

In addition to the preceding operands, some operands of the LU macro can be specified in the PU (or LINE or GROUP) macro instead of the LU macro. Similarly, most operands of the PU macro can be specified instead in the LINE or GROUP macro. Figure 3-7 shows which of the lower-level operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the PU macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

LU Macro Instruction

The LU macro instruction represents a logical unit associated with an SDLC station (type 1 or 2 physical unit) attached to a nonswitched SDLC link, and specifies:

- The local address of the logical unit.
- Use of the pacing option.
- Whether data transfer from VTAM to the logical unit is in batch mode (VTAM users only).

Each logical unit associated with a type 1 or 2 physical unit on a nonswitched SDLC link must be represented by a separate LU macro instruction. The sequence of LU macros must immediately follow the PU (or CLUSTER) macro representing the physical unit. The sequence must be in ascending order of local addresses assigned to logical units (as specified in the LOCADDR operand of this macro). That is, the LU macro specifying the lowest address must appear first, following the PU (or CLUSTER) macro, and the LU macro specifying the highest address must appear last. (LU macros are not required for any local addresses with which no logical units are associated; however, the generation procedure generates logical units for each local address not defined, up to the highest address for which there is an LU macro.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information for the VTAM initialization procedure and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each LU macro coded causes a resource name to be generated.

Name	*Operation*	*Operands*
symbol	LU	LOCADDR=n

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

                            [,PACING={(n[,m])}]
                                     {(1,1)  }
```

VTAM-only operands:

```
                            BUFLIM=
                            ISTATUS=
                            LOGAPPL=
                            LOGTAB=
                            MODETAB=
                            SSCPFM=
                            USSTAB=
                            VPACING=
```

symbol

Provides a name for the logical unit and is required. *symbol* may be any valid assembler-language symbol; the first character may not be $. (This name must not be specified in the service order table.)

`LOCADDR=n`

Specifies the local address (in decimal, without leading zeros) of the logical unit. For type 1 physical units the minimum is 0; the maximum is 255. For type 2 physical units, the minimum is 1; the maximum is 255.

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

(VTAM users only)

Specifies whether data transfer from VTAM to the logical unit is in batch mode. (Lower data transfer priority is provided for transfers in batch mode than for those in interactive mode.)

```
[,PACING={(n[,m])}]
         {(1,1)  }
```

Specifies whether the network control program is to pace the logical unit; that is, whether while sending to the logical unit the program is to require the logical unit to acknowledge, at intervals, receipt of the message data and its ability to accept more data.

n

Specifies the number of requests the network control program is to send to the logical unit before stopping transmission to await a pacing response from that unit. The minimum is 1; the maximum is 255.

m

Specifies in which of the *n* requests the network control program is to turn on the pacing bit. The minimum is 1 (that is, the first request); the maximum is the value specified for *n*.

If you omit *m*, the network control program turns on the pacing bit in the last (*n* th) request sent.

0

Specifies that the network control program is not to pace the logical unit.

If the logical unit is associated with a type 1 physical unit (PUTYPE=1), PACING=(1,1) must be specified (or the PACING operand omitted).

Note: Pacing applies only to normal (synchronous) requests.

Specifying LU Operands in a Higher-Level Macro

Some operands of the LU macro can be specified in the PU, LINE, or GROUP macro instead of in the LU macro. Figure 3-7 shows which of the operands you may code at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the LU macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization procedure. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

INNODE Macro Instruction

The INNODE macro instruction represents an IBM 3704 or 3705 Communications Controller attached to an SDLC link.

In version 3 of the network control program, communications controllers with which the program communicates over an SDLC link are represented by INNODE macros. For compatibility with version 3, versions 4 and 5 accept INNODE macros to represent communications controllers, and for this reason, this INNODE macro is included in this publication. However, when defining a new source program to be generated under NCP version 4 or 5, or when adding new communications controllers to an existing version 3 program to be re-generated under version 4 or 5, use PU macros, not INNODE macros, to represent the controllers.

The INNODE macro specifies:

- The address of the communications controller.
- Whether immediate polling retry is required.
- The maximum number of path information units (PIU) or PIU segments the network control program will send to the communications controller before awaiting a response from the controller.
- The maximum number of consecutive path information units or PIU segments the network control program will send at one time to the communications controller.
- The number of error recovery attempts the network control program will make when transmission errors occur.
- The subarea address of the communications controller.

The INNODE macro is valid only for:

- A network control program that is to operate in a local communications controller and that is capable of communicating with a remote communications controller (TYPGEN=NCP-LR is specified in the BUILD macro) over a nonswitched link (DIAL=NO); or
- A network control program that is to operate in a remote controller (TYPGEN=NCP-R is specified in the BUILD macro) that communicates with the local controller over a nonswitched link (DIAL=NO).

The INNODE macro for the communications controller must appear directly following the SERVICE macro defining the service order table for the SDLC link to which the controller is attached, unless this network control program is to operate in a remote controller, in which case the INNODE macro must immediately follow the LINE macro.

If the local and the remote controller can communicate over one or more alternate (backup) SDLC links in lieu of the principal link, an INNODE macro must appear after each LINE (or SERVICE) macro corresponding to an alternate link. Omit the SUBAREA operand from each of these alternate INNODE macros. Also (1) omit all other INNODE macro operands except IRETRY and RETRIES, if the alternate INNODE macro is coded in a *local* network control program (and thus represents the *remote* controller); or (2) if the alternate INNODE macro is coded in a *remote* network control program (and thus represents the *local* controller), only the MAXOUT and PASSLIM operands are valid and must specify the same values as these operands specify in the *principal* INNODE macro.

Note: Communications controllers and cluster controllers cannot both be attached to the same SDLC link. Therefore, CLUSTER macros and INNODE macros cannot both appear following a SERVICE (or LINE) macro for an SDLC link.

```
Name        Operation   Operands

symbol      INNODE      [ADDR=chars]

                        [,DATMODE={HALF}]
                                  {FULL}

                        [,IRETRY={YES}]
                                 {NO }

                        [,MAXOUT=n]

                        [,PASSLIM={n}]
                                  {1}

                        [,RETRIES=(,t[,n])]

                        [,SUBAREA=n]
```

symbol

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

[ADDR=chars]

(valid in local NCP only)

Specifies the hexadecimal representation of the eight-bit address of the controller represented by this INNODE macro. This address and the address specified in the TADDR operand of the LINE macro within the remote network control program must be identical to the physical address of the remote controller. (The customer engineer enters the physical address into the IPL configuration data set in the diskette contained within the remote controller.)

This operand is required if the program you are defining is to be executed in a local communications controller and can communicate with a remote controller (TYPGEN=NCP-LR is specified in the BUILD macro) and if you specify the SUBAREA operand in this macro; otherwise, this operand is invalid. 00 and FF are invalid address values.

```
[DATMODE={HALF}]
         {FULL}
```

(valid in local NCP only)

Specifies, for a communications controller on an SDLC link allowing simultaneous transmission in both directions, whether the network control program is to communicate with the distant controller in half-duplex or duplex data mode. This operand is valid only if the ADDRESS operand of the corresponding LINE macro specifies both a receive and a transmit address, and the LINE macro represents the principal (not backup) SDLC link.

If you (1) omit this operand, (2) code the SUBAREA operand, (3) specify, in the LINE macro, both a receive and a transmit address, and (4) specify, in the BUILD macro, TYPGEN=NCP-LR, then DATMODE=FULL is assumed.

This operand is invalid if (1) the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R is specified in the BUILD macro) or (2) the program is to be executed in a local controller (TYPGEN=NCP-LR) and the SUBAREA operand is omitted from this macro.

Minimizing Line Turnaround Delay:

Each reversal in direction of message transmission over an SDLC link causes a line turnaround delay of several milliseconds if the carrier signal transmitted by the modem is interrupted. Efficiency of the link is thus lower than if the carrier signal is transmitted continuously, because no message data can be transmitted during these intervals.

Interruption of the carrier signal, and hence line turnaround delays, are inherent and unavoidable if the communications facility (including lines and modems) is half-duplex. They may also occur, however, in a duplex facility and in this case they may be prevented by ensuring that the carrier signal is transmitted continuously. Continuous carrier transmission results from either (1) continuous activation of the "request-to-send" signal sent from the communications controller to the modem, or (2) "strapping" (internally connecting) the modem to transmit carrier continuously independent of whether the "request-to-send" signal is activated.

The "request-to-send" signal (and hence the carrier sent by the modem) is activated continuously if: (1) the communication facility is duplex, (2) the line set to which the SDLC link is attached is duplex, (3) separate receive and transmit addresses are specified in the ADDRESS operand of the LINE macro, and (4) DATMODE=FULL is specified in the LINE macro.

The "request-to-send" signal is not activated continuously if the line set is half-duplex *or* if DATMODE=HALF is specified in the LINE macro. The line turnaround delay can still be minimized or eliminated, however, if the modem is strapped for continuous carrier operation. Consult the supplier or installer of the modem to determine if it is capable of continuous carrier operation and, if so, have the modem so connected. (The modem cannot be strapped for continuous carrier if the communication facility is half-duplex.)

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

(valid in local NCP only)

Specifies whether or not the network control program, when an "idle timeout detect" condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program repolls the controller; otherwise, the program bypasses the unsuccessfully polled controller and services the station represented by the next entry in the service order table.

This operand is invalid if the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R is specified in the BUILD macro).

If you omit this operand and the program is executed in a local communications controller (TYPGEN: NCP-LR), IRETRY=NO is assumed.

Note: You may specify IRETRY in an INNODE macro associated with either the principal or the backup SDLC link.

```
[MAXOUT=n]
```

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the controller represented by this INNODE macro before requesting a response from the controller. The minimum is one; the maximum is seven.

If you omit this operand, a *local* network control program will send one PIU or segment to the remote controller represented by this INNODE macro; a *remote* network control program will send up to seven PIUs or segments to the local controller represented by this INNODE macro.

This operand is invalid in an INNODE macro associated with a backup SDLC link (SUBAREA operand is omitted).

```
[PASSLIM={n} ]
        {1}
```

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the communications controller represented by this INNODE macro.

If this operand appears in a *local* network control program (TYPGEN=NCP-LR), the program services the station represented by the next entry in the service order table when the pass limit value is reached.

If this operand appears in a *remote* network control program (TYPGEN=NCP-R), the program stops sending to the local controller when the pass limit value is reached, and waits until polled again by the local network control program before resuming transmission.

The minimum is one PIU or segment. The maximum is 254.

The default value assumed if you omit this operand is 1 (PASSLIM=1).

If this INNODE macro appears in a remote network control program (and thus represents a local communications controller), the maximum value is 254. This is also the recommended value and the default value assumed if you omit this operand.

```
[RETRIES=( [,t[,n] )]
```

(valid in local NCP only)

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts–via retransmission–to recover from text errors in message data sent to the communications controller represented by this INNODE macro.

The meanings of the *t* and *n* parameters are the same as for these parameters in the RETRIES operand of the LINE macro.

Specifying these parameters in the INNODE macro allows you to designate different values for each communications controller attached to the line. (You may instead specify the *t* and *n* parameters in the RETRIES operand of the LINE macro.)

Note: In order to avoid loss of contact with the remote controller when that controller must access its disk (as, for example, when the remote controller storage is being dumped), specify the number of retries and the value of REPLYTO (see the GROUP macro) such that a minimum of 30 seconds total retry time is assured.

```
[SUBAREA=n]
```

(valid in local NCP only)

Specifies the subarea address assigned to the network control program in the remote communications controller represented by this INNODE macro. (The network control program in each controller in the network must have a unique subarea address.) The value of *n* specified in this SUBAREA operand and in the SUBAREA operand of the BUILD macro for the remote NCP must be identical.

The minimum valid subarea address is 1—that is, SUBAREA=1. The maximum address is the value you specify in the MAXSUBA operand of the BUILD macro.

Code this operand only in the INNODE macro associated with the LINE macro representing the *principal* SDLC link to the remote controller. Omit it from any INNODE macro associated with a LINE macro representing a *backup* (alternate) link.

This operand is invalid for a remote network control program (TYPGEN=NCP-R).

If program is to be executed in a local communications controller, the following rules apply.

If this INNODE macro and its associated LINE macro represent the *principal* SDLC link to a remote controller, SUBAREA must be specified.

If this INNODE macro and its associated LINE macro represent a *backup* SDLC link to a remote controller, omit the SUBAREA operand and all other operands except IRETRY and RETRIES. (When communicating with the remote controller over the backup link, the network control program uses the same values as specified in the INNODE macro for the principal link except for the values of IRETRY and RETRIES.)

(At least one INNODE (or PU) macro in the source program must specify the SUBAREA operand.)

Note: A backup SDLC link can be defined only if the line constituting the link is serviced by a type 2 (not type 1) communication scanner (TYPE=TYPE2 is specified in the CSB macro).

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.

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

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

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

[symbol]

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

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

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

mask1 . . . mask4

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

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

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

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

The bit settings you specify should correspond to the high speed lines requiring increased scanning. For each such line interface installed in the controller, a high speed select feature is present that blocks the attachment of lines to all but the lowest address in the corresponding Lib portion.

Example: Assume that a 3705 having three modules is equipped with type 3 scanners in the first and second expansion modules, but not in the base module. If high speed select features are present in the second scanner for LIB portions 3 and 7 (thus allowing high speed lines to be attached to addresses 0A6 and 0AE), you would specify HSPDSEL=(,00010001,00000000). The first comma signifies that no type 3 scanner is installed in the base module; the first eight-bit mask indicates that increased scanning frequency is required for addresses 0A6 and 0AE in LIB portions 3 and 7, respectively; and the second mask indicates that no addresses in the second expansion module (scanner position 3) require increased scanning frequency.

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

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

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

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

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

`limit1...limit4`

Specifies the scan limits for each installed type 2 or type 3 scanner. Each limit can be from 0 to 3; these values have the meanings shown below. *Limit1* specifies the scan limit for the first scanner position (base module), *limit2* for the second position (first expansion module), etc. All addresses associated with a scanner are scanned if the scan limit for that scanner is 0. Scan limits of 1, 2, and 3 reduce the number of addresses scanned to 8, 48, and 16, respectively. If a scanner position does not contain a type 2 or type 3 scanner, code a comma for the corresponding limit [for example, SCANCTL=(limit1,,limit3,,asmask)]. If a type 2 or type 3 scanner is installed but you specify no limit, the generation procedure assigns the appropriate limit based on the range of actual installed addresses and line speeds as specified in the LINE macros.

The scan limits have the following meanings:

Scan Limit	*Addresses Scanned*	*Addresses Not Scanned*	*Maximum Line Speed*
For IBM 3705:			
0	020-05F 0A0-0FF 120-17F 1A0-1FF	(all addresses scanned)	4,800 bps
1	020-027 0A0-0A7 120-127 1A0-1A7	028-05F 0A8-0FF 128-17F 1A8-1FF	56,000 bps
2	020-04F 0A0-0CF 120-14F 1A0-1CF	050-05F 0D0-0FF 150-17F 1D0-1FF	9.600 bps
3	020-02F 0A0-0AF 120-12F 1A0-1AF	030-05F 0B0-0FF 130-17F 1B0-1FF	19,200 bps
For IBM 3704:			
0	020-03F	(all addresses scanned)	4,800 bps
1	020-027	028-03F	50,000 bps
2	020-03F	(all addresses scanned)	9,600 bps
3*	020-02F	030-03F	19,200 bps

*If 3704 is equipped with two LIBs and the speed of any line(s) is 19,200 bps, specify a scan limit of 2 and do not use address substitution.

`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 substitution; all addresses enabled.
3	1	Address substitution is to be performed for address 6 in LIB position 1. Addresses 8 and 9 in all LIB positions are disabled.
3	0	No address substitution; all addresses enabled.

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

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.

Part III.
Defining Network Control (and Emulation) Functions: BSC and/or Start-Stop Networks

This Part (Chapters 4 and 5) is of interest only to users needing to define network control programs (with or without partitioned emulation feature) for BSC and/or start-stop networks. You may wish to remove and file one or both of these chapters if unneeded—see *How to Use this Book* and the associated chart *To Define a Program* at the front of this book.

Chapter 4: Network Control and Emulation Functions for BSC and/or Start-Stop Networks

Described in this chapter are the many aspects of a teleprocessing subsystem that you must identify to the network control program to tailor it to your particular teleprocessing installation.

You should read this chapter if you wish to define a program capable of operating BSC and/or start-stop lines in network control mode only, or in both network control mode and emulation mode. (Upon request from the access method, the network control program can change the operation of a line from network control mode to emulation mode, and vice versa, if start-stop of BSC stations are attached to the line.)

If you wish to define a network control program that includes SDLC lines as well as BSC and/or start-stop lines, read Chapter 2 for information on SDLC functions. (Read Chapter 2 *instead* of this chapter if you intend to define a program for a network comprising *only* SDLC links.)

Many of the aspects of the teleprocessing subsystem covered in this chapter apply only to operation in network control mode. Others apply to both network control and emulation modes. In most cases, you specify the aspect in exactly the same way for both modes. For example, the type of line control—start-stop or BSC—is specified in the same operand (LNCTL) of the same macro (GROUP) whether the lines in the group are to be operated in network control mode or in emulation mode, or in both modes alternately.

If the program you are defining is to operate all lines always in network control mode, you may ignore any references to emulation mode in this chapter.

If the program you are defining is to operate some lines in network control mode and others in emulation mode or the same lines in both modes, alternately, observe the references to emulation mode as well as those to network control mode.

If you wish to define a program that will always operate all lines in emulation mode only, you may skip this chapter entirely and read Chapter 6 instead. Chapter 6 includes those aspects of the teleprocessing subsystem that apply only to operation in emulation mode.

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

- The stations and lines of the teleprocessing network
- The communications controller hardware configuration
- Data transfer between the communications controller and the host processor
- Procedural options governing message traffic between the controller and the network in network control mode
- Procedural options governing message traffic between the controller and the network in emulation mode
- Optional message processing within the controller (for start-stop and BSC stations in network control mode only)

The remaining two major sections explain the optional diagnostic and service aid facilities that may be included in the network control program, and the program generation options and data sets (files) that the generation procedure will use in creating a network control program load module. The description of each characteristic and option is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the program generation macro instructions given in Chapter 5.

For many characteristics, especially those relating to the equipment configuration, the decisions about what to code in the macro instructions 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 buffers of message data to be accumulated from a start-stop or BSC station before forwarding them to the host processor. Such characteristics, which affect the message-handling capacity and throughput of the teleprocessing subsystem, require careful consideration before specifying the corresponding parameters in the program generation macro instructions.

Once you are familiar with those characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions that define the network control program. At that point you should go on to Chapter 5, which provides detailed information on coding the macro instructions.

BSC and Start-Stop Network Characteristics

This section applies only to start-stop and BSC network operation. (See Chapter 2 for information on SDLC network operation.) The descriptions of the network characteristics give the names of the applicable macro instructions and operands. Unless otherwise indicated, the macros and operands named apply to both network control mode and emulation mode. For your convenience, the operand mentioned can often be specified in a macro instruction different from the one named, as explained in Chapter 5. The description of the operand always appears under the macro named, however.

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 communications controller. For line operations in network control mode, this definition includes (1) computers, (2) transmission control units such as the IBM 2701 and 2703, (3) other 3704 and 3705 controllers, (4) the input/output units (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as *terminals*, and (5) control units (such as IBM 3271 cluster control units) to which input/output units are attached. (See Chapter 2 for information on SDLC network operation.)

Each start-stop or BSC station that communicates with the 3704 or 3705 communications controller in network control mode is represented by a TERMINAL macro instruction. Usually, a TERMINAL macro represents a single station. An exception occurs when stations of the same type, or a limited number of types, communicate with the controller over the switched telephone network. A single TERMINAL macro can sometimes be used to represent whichever of these stations is currently communicating with the controller. (Stations with which the controller communicates only in emulation mode are not represented by TERMINAL or CLUSTER macros.)

Type of Station

Type of station means the numerical designation by which the station is known, or an abbreviation thereof (for example, 1050, 2780, SYS3 [System/3]). Appendix A lists the types of stations with which the communications controller, executing a network control program, can communicate in network control mode and in emulation mode. For network control mode, 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. For operation in emulation mode, type of station is specified in the TERM or CUTYPE operand of the LINE macro.

Terminal Features

For some types of terminals and control units, the presence or absence of certain features with which the terminal or control unit may be equipped must be known to the network control program. The features that must be specified differ for network control mode and emulation mode. If the communications controller is to communicate with a terminal in network control mode, specify the appropriate features from the list of features for network control mode, and similarly, for emulation mode. If the controller is to communicate with the terminals in both modes, specify the appropriate features from both lists.

Features for Operation in Network Control Mode

The presence of some of the features below is specified in the FEATURE operand of the corresponding TERMINAL macro. For these, the suboperand that specifies the presence or absence of the feature appears in parentheses after the description. Other features are specified in the macros and operands indicated.

Transmit Interrupt (IBM 1050, 2741, 3767 in 2741 mode): If the terminal has this feature, the communications controller can interrupt a transmission from the terminal by sending the break signal. (FEATURE: BREAK or NOBREAK)

Buffered Receive (IBM 2740 Model 2, 2770, 2972 Models 8 and 11, 3270, 3780): If the terminal has this function, 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 multipoint line. The presence of the feature and the interval are specified by the BFRDLAY operand of the TERMINAL macro. See also the discussion of buffered terminals under *Teleprocessing Subsystem Operation—Start-Stop and BSC* in this chapter.

Conversational Mode (IBM 1050, 2740 Models 1 and 2 with Record Checking feature, 2770 with Conversational Mode feature, and all IBM BSC stations except 2715 and 2780): A station equipped with this feature can receive message data, instead of the usual positive acknowledgment, in response to a message block sent by the station. The message block the station receives in reply 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. Conversational mode is specified in the CONV operand of the TERMINAL macro.

If you specify the conversational mode feature, the network control program automatically replies to a message block from the station with the next block it currently holds for sending to that station. If the program has no data to send, it replies with a positive acknowledgment.

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 (NL) character occurs in message data. (FEATURE: ACR or NOACR)

Record 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 thereof. (FEATURE: CHECK or NOCHECK, SCTL or NOSCTL, XCTL or NOXCTL)

Interrupt (IBM 2741, 3767 in 2741 mode), 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. (FEATURE: ATTN or NOATTN)

Features for Operation in Emulation Mode

The presence of the features below is specified in the FEATURE operand of the LINE macro representing the line over which the controller communicates with the terminal.

Record Checking: Some start-stop stations have the record checking capability (also called longitudinal redundancy checking), and others do not. For each line to be operated in emulation mode you must specify to the network control program whether the terminals with which the program communicates over that line have the record checking capability. If the terminal is an IBM 1050, 1060, 2260, 2265, 2845, 2848, or System/7, all of which do have this capability, specify LRC in the FEATURE operand of the LINE macro. Also specify LRC for an IBM 2740 (Model 1 or 2), if it is equipped with the Record Checking feature. For other types of start-stop terminals, specify NOLRC.

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

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

Dual Code: Either of two transmission codes (EBCDIC and USASCII) can be transmitted on a binary synchronous communication line attached to an IBM 2701 Data Adapter Unit equipped with the Dual Code feature for that line. The code used is changed from one to the other by command from the access method. The same function can be performed when the IBM 3704 or 3705 is installed in place of the 2701. Specify DUALCODE in the FEATURE operand of the LINE macro representing the line, if the Dual Code feature was used for that line when the line was attached to the 2701. Otherwise, specify NODUALCD or omit the parameter. (In addition to EBCDIC and USASCII, transparent USASCII is supported as a dual code option for a line serviced by a type 3 scanner.)

End of Transmission Character

You may specify that the EOB character, instead of the EOT character, is to signify end of transmission for messages from terminals equipped to send EOB EOT ending sequences. If you so specify, the terminal operator can signal the end of each transmission by pressing only the EOB key, rather than both the EOB and EOT keys. (In this case, each transmission from the terminal consists of a single block.) This option, which applies only to operation in network control mode, is specified in the ENDTRNS operand of the TERMINAL or COMP macro.

Printer Line Length and Carriage Return Rate

The network control program recognizes each carriage return (CR) character and horizontal tab (HT) character in text being sent to a nonbuffered start-stop terminal in network control mode. Upon detecting either character, the program sends a sequence of idle characters immediately following the CR or HT character. A sufficient number of idle characters delays further printing on a terminal printer until the movable printing mechanism (carriage, or carrier) has had sufficient time to reach the next printing position. The next position is one of the tab locations, in the case of the HT character, or the left margin on the next printing line, in the case of the CR character.

The appropriate number of idle characters to send is determined from (1) the maximum length of the line of print—that is, the number of character positions between the left margin and the rightmost printing position; and (2) the rate at which the printing mechanism moves, expressed as the number of character positions traversed by the mechanism for each idle character. From these two values, which you specify for each start-stop line in the network, by the LINESIZ and CRRATE operands of the LINE macro, the generation procedure calculates the required number of idle characters. The values should be carefully selected to suit the type of terminal connected to the line. Too few idle characters sent following each CR or HT character will allow insufficient time for the mechanism to reach the next printing position, resulting in random printing of text characters on the print line. Too many idle characters, on the other hand, will cause excessive delay in the resumption of printing, resulting in wasted time on the communication line.

Thus, for example, if the terminals attached to a given line have a printer line length of 60, and a carrier return rate of 10 printing positions per idle character, you would specify the values 60 and 10, respectively, in the LINESIZ and CRRATE operands of the LINE macro. If you do not specify the line size or return rate, the network control program uses the default values given under the description of these operands.

If the printers attached to the line have differing line lengths, specify in LINESIZ the maximum length used by any terminal on the line.

Communication Line Characteristics

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

Line characteristics refer to (1) the functional attributes of the transmission path, for example, whether the line is half-duplex or duplex; (2) logical characteristics, such as the transmission code and line control scheme employed; and (3) related aspects of the line such as the address by which it is known to the network control program.

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

Nonswitched Multipoint Line

A multipoint line control discipline is typically used for a nonswitched line to which several stations are attached, that is, a nonswitched multipoint line. The controller contacts a specific station by sending a polling character or addressing character assigned to and recognized only by that station. The one station recognizing that character responds appropriately; the other stations ignore the character.

A multipoint discipline must also be used for a line to which only one station is attached, if that station must be polled or addressed by the controller before sending or receiving data. A multipoint line is therefore one on which a multipoint discipline must be used, regardless of the number of stations—several or only one—with which the controller communicates over that line.

To specify a line as requiring a multipoint discipline, specify POLLED=YES in the LINE macro, if the line is to be operated in network control mode. If it is to be operated only in emulation mode, you need not specify it as a multipoint line. (The access method is responsible for properly controlling multipoint lines in emulation mode.)

The network control program requires a service order table for each nonswitched start-stop or BSC communication line that requires a multipoint discipline and that is to be operated in network control mode. This table contains one or more entries representing each station and each component of a station with which the program can establish a session upon request from the access method. (Sessions are described later in this chapter under *BSC and Start-Stop Subsystem Operation.*) The program attempts to establish sessions with stations and components in the same sequence as their respective entries appear in the service order table.

Directly following a LINE macro for a start-stop or BSC multipoint line (or a nonswitched point-to-point line that requires a multipoint discipline), code a SERVICE macro that defines the service order table to be used for that line.

Nonswitched Point-to-Point Line

To designate the communications controller as the secondary station on a BSC point-to-point line to be operated in network control mode, 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 (that is, 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. A nonswitched point-to-point line is not identified as such if it is to be operated only in emulation mode.

Switched Point-to-Point Line

For each switched point-to-point line connection, or "port," 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 LINE macro, indicates that the lines are switched lines. If any lines within the group are switched, all must be.) This operand is applicable whether the line is to be operated in network control mode or in emulation mode, or both.

A switched line port can receive calls from either BSC stations or start-stop terminals, but not from both. Any type of BSC station can call the controller over a line designated for use by BSC stations (provided that all use the same transmission code), as the line control discipline for all such stations is similar. On the other hand, a line designated for use by start-stop terminals can receive calls from only a single type of terminal, except when multiple-terminal-access operation is specified for that line. (See *The Multiple Terminal Access Facility* later in this chapter.)

If the line is to be operated in network control mode, you may designate, in the CALL operand of the LINE macro, whether the line is to be used for receiving calls from stations (CALL=IN), for making calls to stations (CALL=OUT), or both (CALL=INOUT). Each line used for outgoing calls must be included in a dial set by means of the DIALSET macro. Dial sets are explained in *Switched Network Operation*, under *BSC and Start-Stop Subsystem Operation.*

Half-Duplex vs. Duplex Lines

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

Line Speeds and Clocking

The following characteristics must be specified as indicated whether the line is operated in network control mode or in emulation mode. In the SPEED operand of each LINE macro, specify the data rate at which the communication line is to operate. This is the rate at which the station, controller, and modems are designed to transmit data over the communications facility that links the station and the controller.

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

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

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 representing the physical location in the controller at which the line is attached (via line set and line interface base [LIB]).

Transmission Codes

The transmission code to be used for communicating with each station must be identified to the network control program. The program translates outgoing data characters from its internal processing code, EBCDIC, to the specified transmission code, and vice versa, for incoming data characters. (The transmission code used on a multipoint line must be the same for all stations attached to that line.)

Specify the required transmission code in the CODE operand of the LINE macro representing the communication line. (For BSC stations, the code you specify in the LINE macro also informs the network control program which line control scheme is to be used; the transmission code and line control scheme are related.)

Line and Subchannel Addresses

Each communication line attached to the communications controller is identified to the network control program by a line address. Whether the line is to be operated in network control mode or in emulation mode, specify this address in the ADDRESS operand of the corresponding LINE macro. If the line is to be operated in emulation mode, also specify the CPU subchannel address(es) corresponding to the line address. (Each line operated in emulation mode requires its own CPU subchannel address[es] in the host processor. The multi-subchannel line access [MSLA] facility of the network control program with the PEP extension permits two or more emulation subchannels to communicate, alternately, with the same communication line. The address of each subchannel to be associated with a line must be specified in the ADDRESS operand.)

Modems and Automatic Calling Units

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

New Sync Feature

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

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

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

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

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

Automatic Calling Units

Any switched network line that the network control program is to use for calling stations may be equipped with an automatic calling unit. If a line is equipped with an ACU (whether the line is to be operated in network control mode or in emulation mode), specify the ACU address in the AUTO operand of the corresponding LINE macro. Determine the automatic calling unit addresses from the system designer.

Communications Controller Hardware Configuration

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 installed in the controller, (2) the type and number of channel adapters that join the communications controller to the host processor(s), (3) the type, number, and oscillator bit rates of the communication scanners installed, and (4) the interrupt priority to be used for each line serviced by a scanner. This information may be learned from the system designer.

If you specify that the program is to perform only emulation functions (TYPGEN=EP in the BUILD macro), you need not specify the size of storage. However, when you specify TYPGEN=NCP or TYPGEN=PEP you must specify the storage size in the MEMSIZE operand of the BUILD macro.

If you specify that the program is to perform only emulation functions (TYPGEN=EP), specify in the CA operand of the BUILD macro the type(s) of channel adapter installed in the controller. If you specify that the program is to perform network control functions (with or without emulation functions as well), specify the channel adapter type(s) in both the CA and CHANTYP operands of the BUILD macro.

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

For each scanner, you must specify to the network control program (1) the type of scanner, (2) the machine module in which it is installed, and (3) the bit rates of the oscillators with which it is equipped. This is true whether operation of the attached lines is in network control mode or emulation mode.

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

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

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

Appendix J gives a method for determining the interrupt priority for each line in the network.

Communication Between Controller and Host Processor

For the network control program to operate one or more lines in network control mode, information on both the buffers within the access method and the buffers within the network control program must be specified to the network control program, as explained in the two sections below on data transfer. Buffers for network control operation are allocated from a single pool of buffers used for all line and channel data transfers in network control mode.

Associated with each communication line specified as operable in emulation mode, and serviced by a type 3 scanner, is a pair of buffers contained within the control blocks related to the line. The size of each buffer in the pair is user specified as 4, 8, 16, 32, 64, 96, 128, 160, 192, or 224 bytes.

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than do smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

The size of the emulation mode buffers for a line serviced by a type 3 scanner is specified in the BUFSIZE operand of the LINE macro for the line. If you do not specify a size, 32-byte buffers are provided for lines operating at speeds of 9600 bps or less, and 64-byte buffers are provided for lines operating at higher speeds (as specified in the SPEED operand of the LINE macros).

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

Data Transfer from Host Processor to Controller

The amount of data conveyed from the host processor to the communications controller during a single data transfer operation over the network control subchannel may vary over a wide range, depending on the number of requests and the amount of accompanying message data to be transferred. Efficient operation of the network control program requires that the program preallocate a suitable number of buffers for incoming data transfers, rather than allot buffers one at a time. Once the set of buffers is allocated, data transfer from the access method can proceed without further attention by the network control program's supervisory routine until the data transfer ends or all the preallocated buffers are filled. If the amount of data received during one transfer is insufficient to fill all of the preallocated buffers, the remaining buffers are used for subsequent data transfers until all are filled, at which point the program allocates the same number of buffers again.

The INBFRS operand of the HOST macro specifies the number of buffers the network control program is to allocate for data transfers over the network control subchannel from the access method. You should consider two factors when estimating a value for INBFRS.

If the size of a data transfer consistently exceeds the preallocated buffer space, the network control program's supervisory routine is frequently interrupted to provide more buffers for the excess data. The time the program must spend in processing the interrupts reduces the time it can devote to servicing communication lines.

On the other hand, preallocating an excessive quantity of buffers for receiving messages from the access method may deplete the buffer pool to the point that insufficient buffers are available for receiving messages over the communication lines. Buffer depletion is especially likely when (1) the buffer pool is relatively small and (2) a low message rate over the channel from the access method causes the preallocated buffers to be filled slowly, thus unduly delaying return of these buffers to the pool.

In choosing a value for INBFRS, then, strike a reasonable balance between degraded network control program efficiency due to excessive time spent processing interrupts for allocating more buffers, and unnecessary over-allocation of buffers.

Data Transfer from Controller to Host Processor

There is a limit to the amount of data the access method can receive from the network control program during a single data transfer over the network control subchannel. This limit must be specified when defining the network control program so that the program does not attempt to send more than the access method can accept. You specify this limit with the MAXBFRU and UNITSZ operands of the HOST macro. MAXBFRU designates the number of buffer units the access method allocates for a data transfer, and UNITSZ indicates the size of each unit in bytes. The total access method buffer space available is the product of the two values. (A buffer unit is the smallest amount of contiguous storage area handled as buffer space; a buffer may consist of one or more units.)

In sending a series of responses (or requests) to the host processor, the network control program causes the access method to begin receiving each successive block 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 block sent to the host processor to be offset from the beginning of the access method buffer by enough space to allow the access method to insert the prefix. The amount of offset is specified in the BFRPAD operand of the HOST macro. OS/VS VTAM requires 28 bytes for the buffer pads; DOS/VS VTAM requires 15; OS/VS TCAM requires a minimum of 17.

Procedural Options for Operation in Network Control Mode

A number of procedural options characterize the operation of lines in network control mode. These options include (1) the manner in which the program starts up and shuts down the teleprocessing network, (2) the amount of data to be transferred at one time between stations and the controller, (3) the amount of data to be accumulated from a station before passing it to the access method, and (4) the number of sessions to be conducted concurrently on a start-stop or BSC multipoint line. (Network control program sessions are described later in this chapter, under *BSC and Start-Stop Subsystem Operation.*) By careful selection of these options you can "customize" a network control program to best meet the requirements of your teleprocessing applications.

Some procedural 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 network control program buffers or the maximum amount of data to be transferred at one time between the network control program and stations in the network.

Some options require relatively little forethought before you decide what to specify. Others require that you give considerable attention to the effect of your choice on message throughput and response time, among other factors.

Defining Buffer Size

The network control program contains one buffer pool of fixed-size buffers. Buffers from this pool are used for all message data transmitted over the network control subchannel and over lines operating in network control mode. (The buffer pool is not used for message data transmitted over emulation subchannels or over lines in emulation mode.) In the BFRS operand of the BUILD macro you specify the size that you wish the buffers to be. (The minimum is 48 (44, if on-line testing is omitted [OLT=NO is specified in the BUILD macro]); the maximum is 248. The size is always a multiple of four bytes.) A buffer initialization process occurs immediately after the network control program is loaded into the communications controller. In this process the network control program formats into buffers all controller storage space remaining after the program is loaded. The remaining space, divided by the buffer size you have specified plus four bytes (for buffer chaining fields), yields the number of buffers in the pool.

Logical Connection Stations

Each start-stop and BSC station connected to a communication line operated in network control mode is generally represented within the network control program by its own set of control blocks, each defined by a separate TERMINAL macro. This is not true, however, for a switched line over which the controller receives calls from stations. (This is known as a *call-in* switched line.) In this case the program maintains one set of control blocks for any station that calls in over that line, regardless of the number of terminals that may do so. The control blocks represent a "dummy" station called a *call-in logical-connection* station. The control blocks represent whichever station has called over that line at any given moment. A single TERMINAL macro in which CTERM=YES is specified represents a call-in logical-connection station.

BSC and Start-Stop Subsystem Operation

This section describes those procedural options that apply in general to communication lines operated in network control mode, and not uniquely to binary synchronous stations or start-stop (asynchronous) terminals. Options specific to one or the other, but not both, of these categories of stations appear in subsequent sections. (Teleprocessing subsystem operation for SDLC links is covered in Chapter 2.)

The Session

The ability of the network control program to conduct multiple sessions on the same start-stop or BSC multipoint line in network control mode depends on the fact that data transfer does not occur continuously for the duration of the session. For example, in interactive applications such as inquiry-response, the elapsed time between receiving a response from the host processor and entering the next inquiry typically exceeds the time required for transmission of the inquiry and response. The terminal operator typically needs five seconds or more "think time" after seeing the response to prepare his next inquiry, whereas transmission time for the inquiry and response together often consumes but one or two seconds. The interval during which the terminal is not using the line can profitably be used to service other terminals on the same line.

Buffered terminals are another example. With such terminals the operator keys his message data into a buffer instead of directly on the communication line; the line is not needed for transmission until the entire message or block of a message has been accumulated in the buffer. The IBM 2740 Model 2 is an example of such a terminal. A terminal may also receive data from the line into a buffer rather than sending it directly to the printer or other output device. The 2740 Model 2 with the buffered receive feature works in this manner.

While its operator is keying message data into the buffer, on the one hand, and while the terminal is printing the contents of the buffer, on the other, the terminal has no need of the communication line. Since data transmission to and from a buffered terminal usually is much faster than the data entry or printing operations, the terminal requires the line for a relatively small proportion of the session period. Again, the line can be used for servicing other terminals in the interim.

Interleaving transmissions with several stations maximizes the utilization of a multipoint communication line, thus permitting more stations to share the same line than if only one session were possible. A direct result is reduced cost of communication lines and line attachment hardware within the communications controller.

The number of concurrent sessions to be conducted on a line depends on several factors. Among these are (1) the relative amount of time when a terminal is in use that it does not need the communication line, and (2) the permissible delay between readiness to use the terminal and availability of the communication line.

The number of concurrent sessions is called the *session limit* and is specified in the SESSION operand of the LINE macro. The network control program does not only limit the number of sessions to this value, it also tries to maintain that many sessions in order to maximize utilization of the line. The number of sessions in progress will be less than the session limit whenever the network control program has teleprocessing requests for fewer devices than allowed by the session limit, or when fewer devices are ready to communicate with the host processor.

(In the case of clustered BSC stations such as the IBM 3270, the session limit is not applicable if general polling is used to solicit input from the attached terminals. The network control program cannot control the number of terminals that may respond to a general poll of the clustered station, and a separate session is established with each terminal that responds.)

The Service Order Table

The sequence in which the network control program attempts to establish sessions on a multipoint line operated in network control mode is determined by a *service order table* associated with the line. This table is defined by the SERVICE macro you code directly following the LINE macro representing the nonswitched multipoint line. Each station with which the host processor may request a session must be represented by at least one entry in the table. If the station consists of a control unit with one or more individually pollable or addressable components (for example, the IBM 3270 and 1050 terminals), each terminal or component must be represented in the service order table. This is true whether the terminal or component is polled, addressed, or both. In addition, a cluster control unit must be represented in the table if general polling is to be used.

The same device may be represented by more than one entry in the service order table. Multiple entries are of value if you wish the program to attempt to begin sessions more frequently with some devices than with others.

The contents of the service order table for a BSC or start-stop line can be changed during program execution by a control request from the host processor. Control requests can cause the program to add or delete devices or change the order or frequency with which the devices are serviced. Thus the network control program can be kept responsive to application requirements, if these should change from time to time.

Logical Connections

A session is said to be *active* when the network control program is communicating with, or is ready to communicate with, the associated device. The rest of the time the session is *suspended* (or *inactive*). The active and inactive portions of a session may be represented thus:

```
____            ________                    ____  (end)
    ------------        --------------------
time-->
```

The solid lines represent the session in its active state, the broken lines represent the suspended state.

Three concurrent sessions may be represented thus:

```
1 ___                _____                   _____
      ---------------      ------------------      ---------

2     _____
            ----------------     ------------------     ----

3           _____                _____                  ____
                 ----------------     ----------------------
```

time-->

In most applications it is necessary to limit the amount of time a session is permitted to be active, to prevent a device, once in session, from monopolizing the communication line to the exclusion of other devices. The period during which a session is active is called a *logical connection.* A logical connection exists when the network control program and the device in session are engaged in data transfer from one to the other. The length of a logical connection is limited by the *transmission limit*, which is the maximum number of transmissions that may be transferred in either direction between the network control program and the device during the logical connection. When the limit is reached, the network control program breaks the logical connection, thus suspending the session for the moment. The program is then free to service the next session, or attempt to establish a new session, if the session limit has not been reached.

Although the intent of the transmission limit is to restrict the time a session may be active, it does not represent a fixed amount of time or number of data characters. The actual length of a logical connection, in time, is determined by the number of transmissions, the number of data characters in each, and the speed of transmission.

The transmission limit will not always be reached, as the device in session may run out of data to send or become unable to receive, or the network control program may run out of data transfer requests for the device before the transmission limit is reached.

The transmission limit can be individually specified for each device with which the host processor will establish sessions. This allows you to base the limit on the application requirements for the device. The limit is specified in the XMITLIM operand of the TERMINAL or COMP macro representing the device. If the application warrants, you may specify that unlimited transmissions are to be allowed during an active session. This choice should be carefully considered, however, as it can result in monopolization of the communication line by a single device for lengthy periods, thus negating the benefits of having multiple sessions.

Once a session has been established, the network control program re-polls the device for each subsequent transmission to be solicited from the device. The logical connection is maintained during the polling operation. Unless you specify otherwise, the program polls a device only once to solicit the next transmission. If ready to transmit, the device will respond positively to this poll.

You may have the program repeat the polling operation one or more times, if you wish to allow the device more time in which to respond with its next transmission. Specify the required number of polls in the POLIMIT operand of the LINE macro for the line. The network control program will then accept an equivalent number of negative responses to polling before breaking the logical connection. The value specified in the POLIMIT operand is accordingly called the *negative response limit.*

Allowing the program to poll the device more than once is often appropriate for interactive applications in which the terminal operator needs several seconds of "think time" in which to prepare his next transmission. Consider, however, that (1) no message data is communicated during the polling operations, and (2) because the network control program is maintaining the logical connection, no other session can be serviced until the negative polling limit is reached, even if the terminal has no more data to transmit.

Once the polling limit is reached, the network control program can proceed in one of three ways. Unless you specify otherwise, the program breaks the logical connection and cancels the read request that caused polling. The program then goes on to service the next suspended session (or to resume service seeking).

There are two alternatives to this course of action, however. You may specify that the network control program (1) break the logical connection, or (2) maintain the connection and notify the host processor that the negative polling limit was reached. In the latter case, the program also flags all subsequent requests that may be on the queue for the device to prevent them from being executed. It is then the responsibility of the host processor either to signal the network control program to go ahead with execution of the remaining requests, or to cancel those requests and send other ones in their place. These options, like the negative polling limit, are specified in the POLIMIT operand of the LINE macro.

Sessions may be suspended in one other way. Most types of input/output errors that occur during an active session cause suspension of that session.

Session Servicing and Service Seeking

The activity of attempting to establish a new session on a start-stop or BSC line in network control mode is called *service seeking.* The network control program performs service seeking whenever the number of existing (suspended) sessions is less than the session limit established for the line. Thus, the program always tries to conduct the specified number of sessions.

The activity of servicing existing sessions is called *session servicing.* Servicing a session consists of establishing a logical connection, then sending or receiving data, or both, until the logical connection ends because (1) a request from the host processor ends the session, (2) the transmission limit is reached, or (3) the negative polling limit is reached, or (4) an input/output error occurs.

Session servicing and service seeking alternate in a sequence of operations called a *service cycle.*

A service cycle consists of both session servicing and service seeking whenever there exists at least one session, but the total number of sessions is less than the session limit. If no sessions exist at the moment, session servicing does not take place—the service cycle consists only of service seeking. Conversely, if the number of existing sessions equals the session limit, there is no need for service seeking—the service cycle consists exclusively of session servicing.

When session servicing, the network control program conducts a single logical connection in turn for each existing session, in the same sequence as the devices in session appear in the service order table.

When service seeking, the program attempts to establish a new session with one or more of the devices for which no session currently exists, and for which the network control program currently contains a request to begin a session.

As in session servicing, the sequence in which the network control program attempts to establish new sessions corresponds to the order in which the device entries appear in the service order table. Each service-seeking operation begins with the entry following the last entry handled in the previous service-seeking operation. The service order table is a "wraparound" table; that is, service seeking does not stop at the end of the table but resumes automatically with the first entry in the table.

The maximum number of devices with which the program attempts to establish a session during each service seeking operation is called the *service limit.* If the program always contained a teleprocessing request for every device, and if every device were always ready to engage in a session, only one service order table entry would require checking in each service-seeking operation because a new session would always be established with the device. This is not normally the case, however, and the network control program usually must make an attempt for each of several devices before successfully establishing a session.

Unless you specify otherwise, the network control program uses as the service limit one-half of the devices represented in the service order table.

You may instead designate in the SERVLIM operand of the LINE macro a specific maximum number of inactive devices for which the program is to attempt service seeking.

Several factors influence the distribution of the service cycle between session servicing and service seeking.

One major factor is the amount of data transferred between the communications controller and devices during logical connections. The longer the transmissions, the more time is spent in session servicing.

A second major factor is the value selected for the service limit. In periods when the network control program has teleprocessing requests for few of the devices represented in the service order table, a large service limit can result in much service-seeking activity because the program will have to make numerous attempts before establishing a new session. On the other hand, in periods when the program does have teleprocessing requests for most of the devices, it will be able to establish sessions much sooner. The value of the service limit would have less influence in this case, since most of the time service seeking would end with establishment of a new session before the service limit was approached.

Another factor affecting the relative time spent in session servicing and service seeking is the *service priority.* This factor is effective only when the session limit exceeds the number of existing sessions by more than one. Unless you specify differently, the network control program returns to servicing existing sessions after one service-seeking operation—that is, after one attempt, successful or not, to establish a new session. This is referred to as giving priority to "old" sessions.

The alternative is to have the program perform the service seeking operation more than once, the total number of operations equalling the difference between the number of existing sessions and the session limit. This is called giving priority to "new" sessions, and is specified by coding SERVPRI=NEW in the LINE macro.

Assume, for example, that two sessions currently exist and that the session limit is five. If priority is given to old sessions (SERVPRI=OLD), the network control program will perform a single service-seeking operation after servicing the two existing sessions. Then it will return to servicing the two sessions once again. But if new sessions have priority (SERVPRI=NEW), the program performs three service seeking operations in succession, the value of three being the difference between the session limit and the number of existing sessions.

It can be seen that the larger the difference between the number of old sessions and the session limit, the more your choice of service priority affects the relative time spent in session servicing and service seeking. When the session limit exceeds the number of existing sessions by only one, the value in SERVPRI has no effect, since in either case only one service-seeking operation will be performed.

A final factor influencing the distribution of the service cycle between session servicing and service seeking is the negative polling limit described above, and specified in the POLIMIT operand of the LINE macro. The higher the limit, and the more often devices fail to respond promptly to polling, once a logical connection has been established, the more time will be spent in session servicing.

During periods of low teleprocessing activity, there may be intervals when no sessions currently exist on a BSC or start-stop line operating in network control mode. The service cycle accordingly consists only of service seeking (provided that the network control program currently contains at least one request to begin a session). Non-productive polling, and the resultant processing "overhead," can be minimized by specifying a *service-seeking pause* of from several seconds to many minutes. This pause, which you specify in the PAUSE operand of the LINE macro, is in effect at the end of each service seeking operation. When at least one new session is established, the pause is inoperative, since to observe it would delay session servicing as well as minimize non-productive polling. Since it is not in effect when the service cycle consists of both session servicing and service seeking, the service-seeking pause is not a factor in the distribution of the service cycle between these two activities.

During periods when no sessions currently exist and the program currently has no requests to begin a session, no service cycle exists; the line is idle. The service cycle resumes when the program receives from the host processor a new request to establish a session.

As indicated by the foregoing discussion, numerous factors influence the handling of sessions on a multipoint communication line. All of these factors should be considered in terms of their effect on your teleprocessing applications, when specifying operations over start-stop and BSC lines.

Sessions on Point-to-Point Lines

The concept of a session as a defined sequence of data interchanges between host processor and device is valid for point-to-point lines as for multipoint lines.

However, the advantage of multiple concurrent sessions is not available for a point-to-point line; since the network control program communicates with only one station over the line, only one session can exist.

As mentioned earlier, the access method requests that the communications controller establish sessions on lines in network control mode without regard for the way in which the network control program will conduct them. The type of communication line—point-to-point or multipoint—over which the network control program will conduct the session therefore does not concern the host processor.

Because only one session at a time can exist, the session limit, service limit, service priority, service-seeking pause, transmission limit, and polling limit parameters are not applicable for a point-to-point line.

Switched Network Operation

The switched network facilities of the network control program are designed to permit a high degree of utilization of the switched network connections, or "ports," available to the communications controller. Maximum utilization of these ports reduces the number required to support a given number of terminals, with attendant savings in line and controller hardware costs.

The network control program's switched network facilities (applicable only to communication in network control mode with BSC and start-stop stations) accommodate both *call-out* operation, in which the controller calls remote stations upon request from the host processor, and *call-in* operation, in which the controller answers calls from stations. Switched network connections (ports) may be designated for use in fulfilling call-in requests, call-out requests, or both.

Call-Out Operation

The network control program maximizes the utilization of call-out lines by dynamically allocating them to handle the call-out requests with a minimum of delay. To enable the network control program to fulfill call-out requests, you must (1) define a dial set consisting of switched lines having similar characteristics (that is, same line control scheme [start-stop or BSC], same terminal types and features [start-stop] or compatible types and features [BSC]), each line being designated as a call-out line, and (2) specify the stations with which the network control program can communicate using lines in the dial set. (See *The Multiple-Terminal-Access Facility* later in this chapter for another method of increasing switched-line utilization.) Because the program allocates the lines dynamically, any line in the set may be used to communicate with any of the designated stations. You might, for example, establish a dial set of three lines to accommodate call-out requests for 20 stations. The program fulfills call-out requests in the order they are received from the host processor. If at the moment a request is received there are no other requests received before it, and a line is available, the program fulfills the request immediately. Otherwise, the program places the request on the queue for the dial set, to be handled in its turn.

Lines to be used for handling call-out requests must be designated as CALL=OUT in the corresponding LINE macros; alternatively, they may be designated as CALL=INOUT if they are to be used interchangeably for both kinds of requests.

A station associated with the dial set is represented by its own TERMINAL macro, which represents that station and no other; the telephone number by which the program calls the station is contained within the program. The number is specified when defining the program via the DIALNO operand of the TERMINAL macro. It can be changed later, during program execution, by means of the dynamic control facility.

You may improve line utilization by assigning an *alternate dial set* to the original dial set, which is accordingly called the *primary dial set.* The alternate dial set in effect "helps" to handle the load of call-out requests for the primary set, when the alternate set is not fully occupied with its own call-out requests. The alternate dial set is in fact a primary dial set for its own group of stations.

Thus, for example, the switched network facilities for an installation might consist of three dial sets, *A, B,* and *C,* each of which services call-out requests for a group of 20 stations. Each is the primary dial set for its designated group. If *B* were also defined as the alternate dial set for *A*, and *C* as the alternate for *B*, *B* could help with *A*'s overloads, while *C* could help with *B*'s overloads.

Dial sets are defined with the DIALSET macro. In addition to specifying the list of communication lines to make up the set, you may, in the DIALALT operand, specify the name of an alternate dial set.

Two other DIALSET macro operands—QLIMIT and QLOAD—determine the network control program's action when the program cannot immediately service a call-out request because all lines in the dial set are busy. Unless you specify a different value in QLIMIT, the program will place only one call-out request on the queue for the dial set. When this *queue limit*—whether it is one or a greater value that you specify—is reached, the program rejects any further call-out requests for that dial set. Or you may specify a queue limit of zero, in which case the program rejects *any* call-out request it cannot immediately service.

The other operand, QLOAD, specifies the number of unfulfilled requests the program will permit to accumulate on the queue for the dial set before using a line from the *alternate* dial set (if any), to service requests. The number must not exceed that specified in QLIMIT, for then the number of unfulfilled requests cannot reach the value that would cause the program to use the alternate dial set.

Call-In Operation

A switched line used for call-in (but not call-out) operation is not included in a dial set.

Since the host processor does not "know" which station will be calling from moment to moment, it directs its call-in requests to a logical-connection station associated with each line. Once a station has called, the logical-connection station represents that station until the logical connection ends.

To specify that a line is available for call-in requests, specify CALL=IN in (or omit the CALL operand from) the LINE macro. Alternatively, you may designate the line as available for servicing either call-in or call-out requests (CALL=INOUT).

Using Lines in a Dial Set for Both Call-Out and Call-In Operations

You may wish to allow lines in a dial set to service both call-out and call-in requests. This is another way to maximize line utilization, because the lines that might otherwise be idle, in periods when the program receives few call-out requests, can instead be used to service call-in requests.

To specify a line in a dial set as available for handling either incoming or outgoing calls, code CALL=INOUT in the corresponding LINE macro. A dial set may include any combination of lines specified as available for call-out operation, or both call-out and call-in operation.

If all lines in a dial set are designated as available for both incoming and outgoing calls, the possibility exists that all may become occupied with outgoing calls, thus preventing any stations from being able to call the controller. To prevent this, you may specify, in the RESERVE operand of the DIALSET macro, a minimum number of lines the program must hold in reserve for accommodating incoming calls.

Switched Network Backup

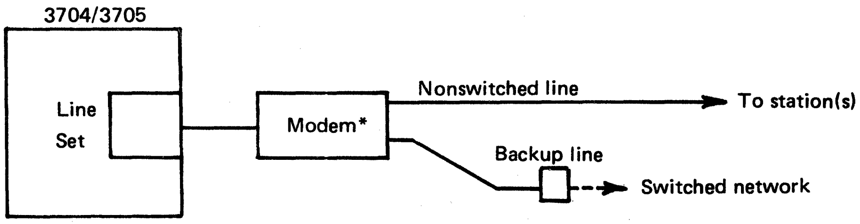
The switched network backup facility of the network control program permits communication between a communications controller and a SDLC station over a temporary, switched communication path provided as an alternate, or "backup", line to the usual (principal) nonswitched point-to-point or multipoint line. Provision of a backup line permits communication between controller and station to continue despite failure of the principal line.

The availability of a backup line requires the installation of appropriate equipment at the controller and stations.

Two forms of switched network backup are available: "same-port" backup and "alternate-port" backup. The same-port backup technique can be used for either SDLC or BSC stations. Alternate-port backup is available only for BSC stations (except 3270s).

Same-Port Backup

The same-port backup technique requires the use of an IBM 3872, 3874, or 3875 modem equipped with the switched network backup feature to attach the controller to both the principal, nonswitched SDLC or BSC line (which may be point-to-point or multipoint) and the switched telephone network, thus:



*with switched network backup feature

A similar arrangement is required at each station for which the backup path is to be provided. In the event the principal nonswitched line fails, the operator at the host processor can (1) deactivate it by a TCAM or VTAM operator command, (2) switch the operation of the modem at the controller to backup mode (and, by voice communication, cause the operator at the remote station to switch its modem similarly), (3) reactivate the line (as represented to the NCP and the access method, not the actual failed line) with a TCAM or VTAM operator command, and (4) establish the dialed backup connection with the station.

A single LINE macro represents the principal and the backup line to the network control program. The NCP is not "aware" of, and does not participate in establishing, the switched backup connection. The program simply operates the line in the same way as it does the principal, nonswitched line once the backup connection has been manually established.

In the case of a backup facility provided for a nonswitched *multipoint* line, the backup connection can be made with only one station on the principal line at a time. A separate backup connection must be made with each station in turn if several or all stations are to be contacted using the backup facility.

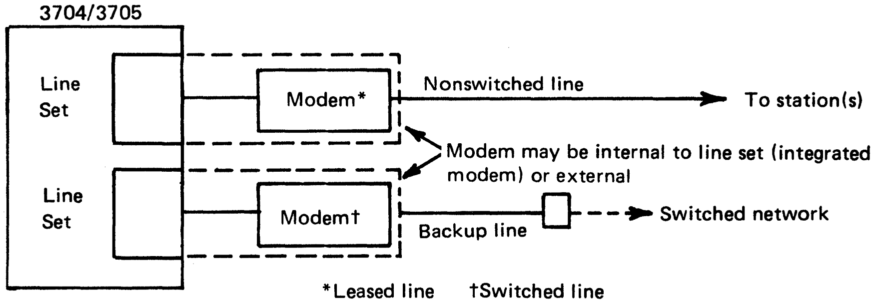
Restriction: Because a switched backup connection can operate only in half-duplex data mode, and the network control program operates the principal and the backup line in exactly the same manner, the principal and the backup line also must operate only in half-duplex (*not* duplex) data mode. Specify this mode by coding one (*not* two) line addresses in the ADDRESS operand of the LINE macro representing the principal line. Further, the physical operation of the line also must be half-duplex (DUPLEX=HALF specified in the LINE macro) to allow proper communication over the switched backup line. Line turnaround time can be minimized by internally connecting the 3872, 3874, or 3875 modem for continuous-carrier operation.

Note: The value specified in the ENABLTO operand of the BUILD macro must be carefully chosen in order to avoid timeouts during the manual dialing operation, as explained in the description of the ENABLTO operand in Chapter 5.

See the publication *Operator's Guide: VTAM Network Operating Procedures* (GC27-6997) or *Operator's Library: OS/VS TCAM* (GC30-3037) for information on the operator commands used in establishing switched network backup operation.

Alternate-Port Backup

The alternate-port backup technique employs a switched-network connection (port), equipped with its own, separate, switched network modem, thus:



The principal and backup lines are represented to the network control program by separate LINE macros. The modem (3872, 3874, or 3875) at the remote station must be equipped with the switched network backup feature (or a separate switched network modem provided in addition to the regular modem, with provision for switching the station from one to the other).

If more than one station is attached to a modem equipped with the switched network backup and fanout features, all attached stations (up to three) can operate concurrently over the backup connection, provided that all are switched to backup operation.

Only terminal components represented to the network control program and the access method by TERMINAL macros can participate in backup operation; components represented by COMP macros cannot participate.

In the case of a multipoint principal line, the network control program may continue to communicate normally with any stations on that line not affected by the line failure while concurrently communicating in backup mode with another station over the backup connection. Further, the program can communicate in backup mode with several stations affected by the failure of the principal line, provided that sufficient backup ports are available.

Establishing the switched backup connection using the alternate-port technique requires that the operator at the host processor enter the appropriate access method commands. Either the operator or the NCP can select the particular switched port to be used. The operator at the remote station to be contacted must switch the station's modem to switched network operation (and back to normal operation when the principal line is restored to service); thus, voice contact is required between the host processor and the remote station when the backup connection is made or terminated.

Two alternatives are available for dialing the backup connection to a station: automatic calling, and program-assisted manual dialing. (A backup connection can be made only from the host processor, not from a remote station.)

Automatic calling—that is, automatic dialing of the telephone number—is possible if the backup port is equipped with an automatic calling unit (ACU), the line address of the ACU is specified in the AUTO operand of the LINE macro for the backup port, and the dial digits are specified in the DIALNO operand of the TERMINAL macro representing the station.

Manual dialing is required if the backup port is not equipped with an ACU. The operator at the host processor must manually dial the telephone number of the station to be reached. If the number is specified in the DIALNO operand of the TERMINAL macro for the station, the access method will inform the operator of the telephone number and the specific line on which to make the call, when he enters the command to switch to the backup line. If the number is not specified in the DIALNO operand, the access method tells him only which line is to be used for the call; he must consult a list of stations and their backup telephone numbers to determine which number to dial.

When communication with the remote station(s) is to be restored to the principal line, the operator at the host processor enters a command to switch the station(s) to the principal line. The network control program breaks the backup connection when all stations (if more than one are attached to the line via a fanout modem) have been switched back to the principal line.

Note: If the telephone number of a remote station is changed after the network control program is generated, and the automatic calling technique is to be used, the new number *must* be specified in the DIALNO operand and the NCP tables reassembled via partial program generation in which CONDASM=TABLE is specified (or via complete program generation).

See Appendix D for an example of how alternate-port backup operation is specified in the network control program. Refer to the TCAM Programmer's Guide for the commands required for backup operation.

Beginning Network Operation

After the network control program is loaded into the communications controller and begins execution, the teleprocessing access method must send control requests that logically activate the communication lines. After a line is activated, the network control program accepts teleprocessing requests for any devices connected to that line.

Type of Line Control

All types of stations with which the communications controllers can communicate use one of three line control schemes: binary synchronous (BSC), start-stop (or asynchronous), and synchronous data link control (SDLC). Each communication line attached to the controller uses one of these schemes; the same line never uses more than one.

When defining the network control program you specify which type of line control the program is to use for each line. All lines in a line group use the same line control; therefore the type is specified in the GROUP macro. The operand is LNCTL.

Network Slowdown

The network control program can receive message data from the access method (via the network control subchannel) and from lines in network control mode only as long as it has buffers available into which to receive the data. The program normally receives and sends data at the same average rate, although momentary overloads can occur in which the program receives more data than it sends over a given time interval. Should the overload be protracted, however, the network control program could exhaust its supply of buffers. To prevent this condition, the network control program continuously monitors its supply of buffers and, when the supply falls to a specified level, automatically enters *slowdown mode.* The level is specified as a percentage of the total number of buffers in the program.

When in slowdown mode, the program reduces the amount of data it receives from lines in network control mode and from the network control subchannel, but continues to send at the normal rate. Since the rate at which buffers are released, after transmission of their contents, exceeds the rate at which new buffers are obtained for receiving data, a net gain in the number of available buffers results. When the buffer supply is sufficiently replenished, the program automatically resumes normal operation.

Unless you specify a different value, the network control program enters slowdown mode when fewer than one-eighth (specified as 12 per cent) of the total number of buffers are available. The SLODOWN operand of the BUILD macro allows you to specify 12, 25, or 50 per cent as the minimum percentage of available buffers, below which the program enters slowdown mode.

Terminal Timeouts

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

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

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the network control program uses the timeout intervals indicated in the descriptions of these two operands for all lines in the group represented by that macro. Some applications may justify allowing unlimited intervals, that is, no timeout at all. This also may be specified in the REPLYTO or TEXTTO operands.

Conversational Response

Some BSC and start-stop stations equipped with parity checking accept message data as a positive response to a block of text the station has transmitted. The IBM 1050, 2740 Models 1 and 2 with Record Checking, 2770, and 2972 without the Batched Message Input feature, and all BSC stations except the 2715 and 2780, are such devices.

Transmitting message data instead of the normal positive response eliminates the line turnaround time incurred when sending a positive response character followed by an addressing character, then receiving a response to addressing before sending message data to the station. The benefit of this "conversational write" operation is improved line utilization.

For each station or component capable of accepting message data as a positive response to text, you may specify conversational operation in the CONV operand of the TERMINAL or COMP macro representing the device. The network control program then withholds sending a positive response after executing a read (or invite) operation if the next request directed to the same device is a write request. Instead, the program sends the message data conveyed by the write request. (For BSC stations, the network control program sends message data as a response only if the data received in the preceding read or invite operation was a complete message [ended by an ETX character].) This option applies only to stations on lines operating in network control mode.

Polling and Addressing Characters

Certain types of start-stop and BSC stations must be polled or addressed by the communications controller in order to transmit to or receive from the controller. To receive data from the station, the controller sends a polling character (or sequence) assigned to and recognized by that station. Receipt of the polling character causes the station to unlock the keyboard, allowing the operator to enter data, or to activate an input device such as a tape reader, if he has previously readied the device to transmit data. Similarly, the controller sends a specific addressing character (or sequence) to signal the station to be ready to accept data from the controller.

If the station is a terminal having more than one input component, such as a card reader and/or tape reader, as well as a keyboard, a polling sequence may be assigned to each component. This allows the controller to solicit data from individual components. Or the polling sequence may specify activation of any input component that the operator has made ready to transmit data.

If the terminal has more than one output component, such as a display, a card punch and/or tape punch, as well as a printer, each may have its own addressing sequence, which allows the communications controller to send data to a specific unit.

Some BSC stations, such as the IBM 2770, recognize component selection characters within message data the station receives over the line. The output component to which data is sent may thus be changed as receipt of the message progresses.

Stations activated by polling and addressing characters are used in network configurations requiring attachment of more than one station to a communication line. Such a line is called a multipoint line, and each station is assigned polling and addressing characters different from those assigned to any other stations on that line. This allows a particular station to be activated, while all others remain idle. (In addition to the individual characters, a group addressing character (or sequence) may be assigned to certain stations to permit simultaneous transmission of data to all stations in the group. Or a broadcast address character may be assigned to all stations on the line to permit addressing all stations simultaneously.)

Stations using polling and addressing are not limited to use on multipoint lines. The IBM 1050, for example, always must be polled and addressed, even if only one 1050 is attached to a line, and regardless of whether that line is nonswitched or switched.

The network control program performs the polling and addressing functions for all stations connected to lines operating in network control mode. The access method performs these functions for all stations connected to lines operated in emulation mode. Therefore, in defining the network control program, you must specify the polling and addressing characters for each station with which the program communicates over a line operating in network control mode, but not for those stations on lines that will operate only in emulation mode. Specify the polling and addressing characters for lines in network control mode as follows.

In the POLL operand of each TERMINAL macro representing a station that must be polled, specify the polling character or characters recognized by that station. Similarly, in the ADDR operand of the same macro, specify the addressing character or characters recognized by that station. If a station is used for output only, you would specify only the ADDR operand; conversely, you would specify only the POLL operand for a station to be used for input only.

If a station has more than one individually addressable output component, the access method may establish individual sessions with each, provided that all components with which sessions are to be established are represented by a TERMINAL or COMP macro the ADDR operands of which specify the addressing characters required. Likewise, if you wish to establish independent sessions with each of two or more input components, each must be represented by a TERMINAL or COMP macro the POLL operands of which specify the appropriate polling characters. If a station has but one input and one output component, only the TERMINAL macro is required. Each additional component requires a COMP macro. However, the same COMP macro can represent both one input component and one output component.

A terminal having multiple input and output devices requires only a TERMINAL macro if you wish to allow only one session at a time with the terminal, provided that the terminal has the common polling and addressing capability. (An example is the IBM 1050, for which the common polling character is 0 and the common addressing character is 9.) The common polling and addressing characters must be specified in the POLL and ADDR operands of the TERMINAL macro.

If your network includes start-stop terminals that permit group or broadcast addressing, and you wish to use the facility (when operating in network control mode) code, in addition to the TERMINAL macros representing the individual terminals, an extra COMP or TERMINAL macro that specifies—in the ADDR operand—the assigned broadcast address or group address.

If your network includes multi-component BSC terminals connected to point-to-point lines, you may wish to specify selection sequences for each of the output components. Selection characters for one output component may be specified in the ADDR operand of the TERMINAL macro. Specify the selection sequences for any additional output components in separate COMP macros.

Station Telephone Numbers

In order to call a station (in network control mode) that is connected to the switched telephone network, the network control program must be given the telephone number of that station. This is done by specifying the dial digits in the DIALNO operand of the TERMINAL macro representing the station.

If necessary, you may change the number to a different number during execution of the network control program, via the dynamic control facility.

When specifying a telephone number in the DIALNO operand, you may include a dialing pause, in multiples of one second, between successive digits. A dialing pause can allow time for receiving a secondary dial tone, as when an outside-line code or direct-distance-dialing network access code must be dialed before dialing the station's telephone number. End-of-number and separator characters may be included in the sequence of dial digits specified in the DIALNO operand, if the modem is designed to use these characters. (The use of these characters is explained in the description of the DIALNO operand.)

Number of Attempts to Dial a Station

Unless you specify otherwise, the network control program, upon receiving a request to call a station in network control mode, automatically dials the station's telephone number up to four times in succession. If the last attempt is unsuccessful, the network control program returns to the host processor a response indicating the failure. To specify a different number of attempts to dial a 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.)

Preventing a Monopoly of Network Control Program Buffers

The network control program fills all requests for buffers from a single buffer pool and no BSC or start-stop station should monopolize the supply of buffers to the extent that other stations are prevented from communicating with the controller in network control mode. Such excessive buffer monopolization could occur if the network control program were to accumulate too much data from a station before forwarding the data to the host processor.

You can prevent monopolization of buffers with two options. The first prevents excessive accumulation within the controller 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 passing the data to the host processor. This is desirable because message processing routines within the host processor can then examine an entire block at once.

If the station were to send an excessively long block of data, as could happen if a terminal operator entered thousands of characters without sending an end-of-block character, an unreasonably large number of network control program buffers could be filled by the arriving data. To prevent all of these buffers from being tied up until the block is complete, the network control program restricts the number of buffers that can be filled with arriving data before passing their contents to the host processor. Reception of data from the station is not interrupted when this happens; the network control program continues to allocate buffers for the remaining data.

Each partial block the program passes to the access method is called a *sub-block,* and the network control program is said to be operating in *sub-blocking mode* with respect to the line over which the data is being received. The response header that precedes each partial block indicates to the access method that the data that follows is a sub-block, not a complete block.

Aside from the consideration of buffer monopolization, there is a limit to the amount of data the access method can receive from the network control program during a single data transfer operation. The program accordingly restricts a sub-block to only as much data as it can deliver to the host processor in a single data transfer operation over the network subchannel.

In the TRANSFR operand of the LINE macro you may specify the size of a sub-block, in terms of the number of buffers to be filled before forwarding to the access method. Specifying this parameter in the LINE macro allows you to set a different limit for each of various lines.

If you omit the TRANSFR operand, or if you specify a number of buffers greater than the network control program can pass at one time to the access method, the network control program uses as the sub-block size the maximum number of buffers it can deliver to the access method.

Limiting the size of a sub-block 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 sub-block buffers to the access method and then releases the buffers to the buffer pool. If for any reason the network control program cannot promptly transfer the sub-blocks to the access method (as when the host processor channel is slow to respond to the controller's signal for service), the data accumulating from the station can still cause monopolization.

For this reason, you may also set a limit on the number of sub-blocks the network control program will accept from a start-stop or BSC station during a single logical connection. If the station sends enough data to cause the limit to be reached, the program breaks the logical connection for that station and is then free to service another station on the same line, if a multipoint line. Limiting the number of sub-blocks also serves to prevent a station from monopolizing a multipoint line so as to delay servicing of other stations on the same line. In the CUTOFF operand of the LINE macro you specify the number of sub-blocks to be accepted from a station. If you omit the CUTOFF operand, the network control program continues to accept sub-blocks from a station until it has received an entire block.
It may be appropriate for the application program in the host processor to defer processing of any sub-block data until the entire block is received. If any sub-block contains an error (indicated by a bit in the response header for the sub-block), the application program should normally discard all of the sub-blocks in the block, not only the one in error. There is no provision for requesting re-transmission of sub-blocks in error.

Use of Buffer Delay for 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 has multiple data blocks to send to the same terminal, it must wait after sending each block for the terminal to print the contents of its buffer before it is able to send the next block. If the line is a multipoint line operating in network control mode, the network control program can use the time the line would otherwise be idle for communicating with other terminals. That is, at any given moment the program can be sending to one of several terminals while the others are printing data received earlier.

For each start-stop or BSC terminal attached to a line operating in network control mode, specify in the BFRDLAY operand of the TERMINAL macro the delay in seconds that the network control program should wait after sending each block before sending the next one. The value you specify should equal the length of time the terminal needs to print or otherwise display the contents of its buffer.

The terminals for which specifying a buffer delay is appropriate are:

IBM 2740 Model 2 with Buffered Receive feature
IBM 2770
IBM 2980
IBM 3275
IBM 3277
IBM 3284
IBM 3286
IBM 3780

Error Conditions and Recovery Procedures

Communication between the communications controller and stations in the teleprocessing network is subject to *input/output* (I/O) errors, usually caused by the transient noise conditions to which communication lines are sometimes susceptible, or by hardware malfunctions. The network control program takes no action for errors on lines operating in emulation mode. Any error recovery procedures desired must be executed by the access method.

For lines operating in network control mode, if the error is of a kind that can be recovered from, the network control program makes the appropriate recovery efforts. For example, upon detecting a parity error in received data, the program signals the station to retransmit the data, if the station is of the kind that can automatically retransmit, as from a buffer, or of the kind that can visually inform its operator to re-enter the same data. Conversely, upon being informed by the station that it has received data in error, the program retransmits the data.

The maximum number of retransmissions may be specified for each line operated in network control mode. If error-free transmission is not achieved before the retransmission limit is reached, the network control program indicates the fact in its response to the access method.

If the I/O error is of the kind that inherently cannot be recovered from (such as a modem error), the network control program makes no error recovery attempt but immediately indicates in its response to the access method what kind of error occurred.

Once the network control program notifies the access method that it is unable to clear the error condition, it makes no further attempts at error recovery. Nor does the program attempt to execute any further requests for the device affected. Instead, it places the device in an *error lock* status. The access method can then take appropriate action. For example, it can send a control request that causes the network control program to remove all unfulfilled requests from its queue for the device and return them to the access method, which can then modify the sequence of requests or build a new sequence and send it to the network control program. Or the access method can ignore the errror condition and send a control request that reinstates the queued requests, thus removing the device from error lock status.

Input/output errors can occur either during transmission of message data or while the network control program is performing a control function preparatory to or following message transmission. I/O errors are consequently divided into *text-mode* errors and *control-mode* errors.

The number of error recovery attempts for text-mode errors is determined by the RETRIES operand of the LINE macro. The number you specify applies to all stations with which the network control program communicates over the line represented by the LINE macro. The number of error recovery attempts for control-mode errors is set by the CRETRY operand of the GROUP macro. Thus, the same maximum number of recovery attempts applies to all stations on all lines making up the line group.

Text-mode errors may be further classified as *text-read* errors and *text-write* errors. Text-read errors are those occurring when receiving from a station; text-write errors occur when sending to a station.

For text-read errors you may specify a single sequence of retransmissions, from one to 254. Or you may specify unlimited retransmissions.

For text-write errors you may specify a single retransmission sequence or multiple sequences. When you specify multiple sequences, the program executes the first sequence immediately after detecting the error condition. If transmission is still unsuccessful at the end of the sequence, the program pauses, then executes the next sequence. This activity continues until successful transmission is accomplished or the limit on retransmissions is reached. Specifying a pause between retransmission sequences allows time for transient noise conditions on the line, which may be responsible for the repeated errors, to subside.

Unless you specify otherwise in the RETRIES or CRETRY operands, the network control program will perform a maximum of two recovery attempts, for start-stop stations, and seven attempts, for BSC stations, for text-read, text-write, and control-mode errors. As stated above, error recovery is attempted only for lines operating in network control mode.

Protecting Against Failure of Modem or Automatic Calling Unit

Various malfunctions occurring within a modem or automatic calling unit (ACU), if not protected against, could cause the communication line attached to the modem or ACU to be tied up indefinitely, without the knowledge of the network control program, thus effectively rendering the line unusable. To prevent such an occurrence, the network control program has three timeout values, *enable timeout, disable timeout,* and *dial timeout,* for lines operating in network control mode.

If a switched line is not enabled (that is, the data set ready signal line in the modem is not turned on) within the enable timeout period, the enable operation is terminated abnormally. Similarly, if the switched line is not disabled (data set ready signal line is not turned off) within the disable timeout period, the disable operation terminates abnormally.

Unless you specify different values, the network control program uses an enable timeout of 2.2 seconds and a disable timeout of 3.0 seconds. These values are generally appropriate, but you may specify different ones in the ENABLTO and DISABLTO operands of the BUILD macro. The timeouts specified apply to all communication lines operating in network control mode.

The third timeout, dial timeout, protects against the failure of the automatic calling unit for a switched line to reach a called station within a reasonable period. This condition may result from failure of the called station to answer, from failure of the common-carrier equipment to complete the connection path, or from a malfunction within the ACU. Expiration of the dial timeout for any of these reasons causes abnormal termination of the dialing operation.

A dial timeout of 60 seconds is normally appropriate, and this is the value the network control program uses for lines in network control mode unless you specify a different one in the DIALTO operand of the BUILD macro.

Direction of Transmission

In most network configurations the stations are capable of both sending and receiving message data. However, for each BSC or start-stop station attached to a line operating in network control mode, you may specify that the station be used for input only, or for output only, via the DIRECTN operand of the TERMINAL macro representing the station. If you specify input only, the network control program will reject any requests from the host processor to send to the station. Conversely, specifying output only causes the program to reject any requests to accept data from the station.

Erasing Critical Data in Buffers

Normally, the network control program releases buffers to the buffer pool after use without first clearing the buffer contents. As buffers are reallocated for subsequent operations, the old contents are overlaid with new message data.

For maximum assurance that security-sensitive data remaining in buffers returned to the buffer pool is not subsequently transmitted to an unintended destination, you may specify that the network control program clear buffers before returning them to the pool as follows. Specifying ERASE=YES in the BUILD macro causes the network control program to: (1) erase all data buffers received from the host processor but rejected because the network control program has entered slowdown mode; (2) erase all buffers containing data blocks, received from the host processor, in which the network control program detected a transmission error; and (3) erase data buffers containing PIUs sent to the host processor, after the host processor indicates successful receipt of the data.

In addition, you may specify CDATA=YES in individual CLUSTER, TERMINAL, and COMP macros to cause the program to erase buffers that have been used for receiving from or sending to the corresponding start-stop or BSC stations; this function is not applicable for data received from or sent to SDLC stations.

Automatic Network Shutdown

All lines attached to a communications controller (local or remote) and currently operating in network control mode are shut down automatically, in an orderly manner, under any of several conditions as explained below. (Any lines currently operating in emulation mode are unaffected by shutdown of lines in network control mode.)

This orderly procedure is called *automatic network shutdown* (ANS). The ANS facility is included in the program unless you specifically exclude it by coding ANS=NO in the BUILD macro. (Apart from automatic shutdown, individual lines and stations can also be deactivated and reactivated by requests from the access method.)

Automatic network shutdown occurs under the following conditions.

Local controller:

- The host processor fails to respond to the network control program within a specified interval after the NCP has presented an attention signal to the channel. This interval is specified by the TIMEOUT operand of the HOST macro.
- A shutdown request is entered at the control panel of the controller.

Remote controller:

- The remote network control program detects a lapse in successful communication activity over the local-remote SDLC link currently in use for communication between the local and remote controllers. The lapse may occur either through outright failure of the link or through badly degraded performance of the link as indicated by exhaustion of error recovery procedures performed by the local network control program. The lapse interval is determined by the value you specify in the ACTIVTO operand of the GROUP macro representing the SDLC link(s) joining the local and remote controllers. This interval must be sufficiently long for the local NCP to complete its error recovery procedures for the link.
- The local network control program, upon entering automatic network shutdown mode, signals the remote NCP to shut down the network attached to the remote controller.
- A shutdown request is entered at the control panel of the controller.

Automatic network shutdown of either a local or a remote communications controller terminates any sessions in progress, logically deactivates all active stations and SDLC physical and logical units in the network, breaks any existing switched line connections, disables any currently enabled dial-in ports, and (for SDLC stations) releases any network addresses that have been assigned.

(If the network control program is currently communicating with any BSC stations over a switched network backup connection, the program reestablishes the principal, nonswitched connection, then breaks the switched backup connection. Upon reactivation of the network, contact with such stations will be initiated over the principal line.)

If you exclude the ANS facility, the access method will reload (IPL) the communications controller with a new copy of the network control program whenever it reactivates the controller.

Configuration Restart

Upon completion of the shutdown process, the network control program places a special "shutdown-complete" message at the end of the queue of message traffic for the channel (in the case of a local NCP) or SDLC link to the local controller (for a remote NCP). Upon reestablishing contact with the local controller (over the channel) or the remote controller (over the channel and the SDLC link), the access method receives the shutdown-complete message. The access method receives the message in the normal traffic flow from the NCP if the shutdown was initiated at the control panel.)

The access method then sends the appropriate commands to either (1) reactivate the exist ing NCP, restore the link, station, physical unit and logical unit status existing at the time of shutdown, and restart the sessions; or (2) reload (IPL) the controller with a new copy of the NCP. These two alternatives apply to either a local or a remote network control program. The action taken by the NCP in restoring the network status and sessions is called *configuration restart.*

When reactivating a remote network control program via configuration restart, the access method can resume transmission to the remote NCP over the same SDLC link that was in use when shutdown occurred (provided that link is operational) or over any available backup link.

If you exclude the automatic network shutdown (and configuration restart) facility from a *local* network control program by specifying ANS=NO in the BUILD macro, the NCP waits indefinitely for resumption of communications over the channel; the local controller does not enter the IPL-required state.

If you exclude the automatic network shutdown (and configuration restart) facility from a *remote* network control program by specifying ANS=NO, one of the following occurs. (1) If you specify an activity timeout in the ACTIVTO operand of the GROUP macro representing the local-remote SDLC link(s), the remote NCP abnormally ends; the controller enters IPL-required state and awaits reloading over the local-remote SDLC link (or over any available such link, if more than one is provided). (2) If you do not specify an activity timeout (ACTIVTO=NONE), the remote NCP does not abnormally end but instead awaits indefinitely the resumption of traffic from the local network control program.

Critical Situation Notification

During automatic network shutdown of lines in network control mode, the program can automatically send a predefined message to each active start-stop and BSC station that alerts the station to the impending shutdown. This "critical situation" message is sent to each active station in whose TERMINAL macro you have specified CRITSIT=YES, provided that the line to that station is currently operating in network control mode. Define the content of the message in the CSMSG and CSMSGC operands of the BUILD macro. A message header, if required for IBM 3271 or 3275 terminals, may be specified in the CSMHDR and CSMHDRC operands of the BUILD macro.

The critical situation message always begins with the date and time (the latter is in 24-hour format) and ends with the text you specify in the CSMSG operand.

Example:

```
05/14/75 19.27.05 NETWORK SHUTDOWN IN PROGRESS-NO FURTHER
TRANSMISSIONS UNTIL NOTIFIED.
```

The network control program does not automatically send a message to notify stations when the network is again operational; this is the responsibility of the access method.

Other Options

The options described up to this point apply to network operations in general, and not specifically to binary synchronous or start-stop stations. Other procedural options do concern one or the other of these categories of stations. These are described next.

BSC Subsystem Operation

The options described in this section apply only to binary synchronous (BSC) stations. If your teleprocessing network does not include any BSC stations, you may omit reading this section. Options common to both BSC and start-stop stations are described under the heading, *BSC and Start-Stop Subsystem Operation.*

Transmission in Transparent Mode

The network control program is capable of sending and receiving data over a BSC line in *transparent mode.* Transparent mode allows transmission of message data containing any bit patterns, including those that the sending and receiving stations normally recognize and act upon as line control characters.

For transmission to a station on a line operating in network control mode, the teleprocessing request specifies whether the network control program is to send the message data in normal or in transparent mode. The program accordingly transmits, at the appropriate times, either the nontransparent control characters—STX, ETB, and ETX—or the transparent control sequences—DLE STX, DLE ETB, DLE ETX.

For transmission in network control mode from a station to the communications controller, the network control program automatically deletes the line control characters it detects in the received message data, before forwarding the data to the host processor. The control information accompanying the data indicates to the host processor whether the network control program received the data in nontransparent mode or in transparent mode. If a station is on a line operating in emulation mode, the network control program does not insert and delete the required transparent control sequences. The access method must include all required control sequences in message data it delivers to the network control program for forwarding to a station. Conversely, the network control program delivers to the access method, unchanged, transparent message data it receives from lines in emulation mode.

Intermediate Block Checking Mode

When receiving from a station in network control mode, the network control program automatically examines the block check characters (BCC) that follow each intermediate-transmission-block (ITB) character, if any, in the received data. In the ITBMODE operand of the TERMINAL macro for the station you specify whether the network control program is to send error information bytes (EIB) to the host processor following each ITB character. If you specify insertion of EIBs, the application program in the host processor can scan the received data for ITB characters, and by analyzing the first byte following the ITB—the error information byte—determine whether an error occurred in the intermediate block.

When receiving from the host processor message data to be sent to a station in network control mode, the network control program can automatically remove EIBs, if any, before transmitting the message data. If you specify deletion, the network control program deletes the first character following each ITB it detects within the data to be sent. You should specify deletion only if each first following character is in fact an EIB; otherwise the receiving station will not receive the first data character of each intermediate block.

Intermediate Block Checking Mode for Transparent Text

If you specify XITB=YES in the BUILD macro, the network control program is capable of inserting ITB sequences and error information blocks in transparent text as well as in non-transparent text, for BSC stations on lines operating in network control mode. Insertion of ITB sequences and error information blocks occurs only for those BSC stations whose TERMINAL macros specify, in the ITBMODE operand, use of the intermediate block checking facility, as follows:

If the BUILD macro specifies XITB=YES and the *first* parameter of the ITBMODE operand specifies intermediate block checking, the network control program substitutes an error information block for each DLE ITB sequence in transparent text received from the station.

If the BUILD macro specifies XITB=YES and the *second* parameter of the ITBMODE operand specifies intermediate block checking, the network control program inserts DLE ITB sequences into transparent text being sent to the station. Special two-byte fields within the text received from the host processor determine the intervals at which the DLE ITB sequences are inserted.

If the BUILD macro specifies XITB=NO (or you omit the XITB operand), no insertion of ITBs and EIBs is performed for any BSC station.

This function is performed only if the communication line to a station is currently operating in network control mode.

ID Exchange and Verification

The network control program can receive an identification (ID) sequence from any BSC station that calls the communications controller over a switched line operating in network control mode, and can either check that sequence against a list of valid sequences within the program or pass the sequence to the access method for checking. (If the access method is operating under DOS/VS, sequences *must* be passed to the access method for checking.) Upon failing to recognize the sequence as valid, the network control program does not proceed with message transmission. Instead it either breaks the line connection, or maintains the connection but forwards the unrecognized sequence to the host processor for checking against a list kept there. In the latter case, the host processor can signal the network control program to proceed with message transmission or break the line connection.

After either the network control program or the access method checks the received sequence, the network control program can send an ID sequence in reply.

Advantages of NCP Verification vs. Access Method Verification

If the network control program verifies a received ID sequence, message transmission can begin sooner than if the ID must be forwarded to the access method for checking. But the storage space needed within the network control program to maintain a list of ID sequences can be considerable, if there are many sequences. Conversely, if the access method maintains the list, storage requirements within the controller are minimized, but ID sequence checking by the access method may take longer.

A compromise, if OS/VS VTAM or TCAM is being used, is to keep within the network control program the ID sequences for the stations that call most often, and keep within the host processor those sequences representing stations that call less frequently. (If DOS/VS VTAM is the access method used, all sequences received from BSC stations *must* be passed to VTAM.)

The various ID verification options for checking by the network control program are specified with the IDSEQ operand of the TERMINAL macro and the NOMATCH operand of the IDLIST macro. ID verification by VTAM is specified by the (VTAM-only) VIDLIST macro. See the *OS/VS TCAM System Programmer's Guide* for information on ID verification by TCAM.

Defining a Controller ID Sequence

If you choose to define an ID sequence within the controller to be sent in response to ID sequences received from BSC stations, specify the sequence in the CUID operand of the BUILD macro. The maximum length is 20 characters. However, because different types of stations may expect ID sequences of different lengths, you must specify the required length in the CUIDLEN operand of the TERMINAL macro for each station to which the sequence is to be sent. If you omit the CUIDLEN operand, no controller ID sequence is transmitted.

Controller ID sequences are not sent to stations on lines currently operating in emulation mode.

Sending and Receiving WACK Sequences

When receiving message data from a BSC station in network control mode, the network control program may need to temporarily defer further input from that station. This can happen, for example, when the network control program has no further read requests from the host processor for receiving additional message data. When this occurs, the program responds to the block just received with a WACK (wait-acknowledgment) sequence 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 the station. Upon receiving the WACK sequence, the station replies with an ENQ character.

Exchanging of WACK and ENQ sequences can continue for as long as the network control program needs to defer input from the station, or until the station breaks the connection. When the program is ready to resume receiving from the station, it sends the deferred positive acknowledgment (ACK-0 or ACK-1).

There is no limit to the number of WACK sequences the network control program will send. The program does, however, limit the number of WACK sequences it will accept from the station, unless you wish to allow the program to receive them without limit.

Unless you specify otherwise, the network control program will accept up to 15 consecutive WACK sequences from a station. You may specify a different number, or specify unlimited acceptance of WACK sequences, in the WACKCNT operand of the GROUP macro.

Sending and Receiving Temporary Text Delay Sequences

When the network control program must temporarily suspend *sending* to a station in network control mode, it need not break the logical connection. Instead it can transmit a *temporary text delay* (TTD) sequence in lieu of the next message block. The TTD sequence informs the receiving station that the communications controller will continue sending after a short pause. The station replies to the TTD sequence with a NAK character. Exchange of TTD and NAK can continue as long as the program needs to defer transmission, or until the station breaks the logical connection.

There is no limit on the number of TTD sequences the network control program will send. The program does, however, limit the number of TTD sequences it will accept from a station, unless you wish to allow unlimited acceptance. When the limit is reached, the program breaks the logical connection.

Unless you specify otherwise, the program will accept up to 15 consecutive TTD sequences from a station. You may specify a different limit, or specify that the program is to accept them without limit, in the TTDCNT operand of the GROUP macro.

Frequency of Transmission of Synchronous Idle (SYN) Characters

In binary synchronous communications, a *synchronous idle* (SYN) character must be transmitted on the communication line at regular intervals to maintain the sending and receiving stations in synchronism. Binary synchronous stations transmit these characters periodically when sending message data. (A sequence of SYN characters is also transmitted when the line is otherwise idle.)

The network control program conforms to normal BSC practice by sending the SYN characters once each second. In rare circumstances it may be appropriate to change this interval; this can be done with the SYNDLAY operand of the GROUP macro. (The change will be effective only for lines serviced by a type 1 or type 2 scanner. The type 3 scanner hardware sends SYN characters at one-second intervals.) This function applies only to lines operating in network control mode.

SDLC/BSC Path Function

The SDLC/BSC path function is an option by which the network control program can transmit data originating at an SDLC station directly to a specified BSC station without first sending the data to the host processor. (Normally, all data transmitted in the network passes through the access method in the host processor.) Only path information units (PIU) containing message data pass directly between the SDLC station and the BSC station; error and control messages associated with the SDLC station are sent to the access method.

To establish the logical path between the SDLC and the BSC station, you specify, in the DATASW operand of the LU macro defining the SDLC logical unit, the name of the TERMINAL macro representing the BSC station with which the logical unit is to communicate.

Associated with each BSC station that is to participate in the SDLC/BSC path function there must be a block-handling routine that converts the BSC message data to the format required by the SDLC logical unit. (Format conversion for the message data transmitted in the opposite direction [that is, SDLC to BSC format] is always present in the network control program.) This block-handling routine must be specified as executable at point 3 (that is, executable after the data is received from the BSC station) and must be active whenever the SDLC–BSC path is to be available for data transfer.

An IBM-supplied block-handling routine that makes the proper conversion may be invoked by specifying an SPAFPT3 macro in the point 3 block handler associated with the BSC station. (See the description of the SPAFPT3 macro in Chapter 5.) Or you may code a user block-handling routine to convert the PIU format, provided that it is functionally equivalent to the IBM-supplied routine. In this case you would include a UBHR macro calling the user-written routine, rather than an SPAFPT3 macro, in the block-handler.

For more information about the SDLC/BSC path function, including requirements and restrictions about its use, see *SDLC/BSC Path Function System Programmer's Guide* (GC30-3029).

Start-Stop (Asynchronous) Subsystem Operation

The options described in this section apply only to start-stop terminals (also called asynchronous terminals). If your teleprocessing network does not include any start-stop terminals, you may omit reading this section.

Options common to both start-stop and BSC stations are described earlier in this chapter under the heading, *BSC and Start-Stop Subsystem Operation.*

The Multiple Terminal Access Facility

A major feature of the network control program is its ability to communicate in network control mode with a variety of dissimilar, commonly used start-stop terminals over the same switched network connection, or "port." This feature, called the *multiple terminal access* facility, makes it unnecessary to reserve a separate port for each type of terminal, as has often been the case in teleprocessing networks. This facility therefore serves to minimize the number of communication lines and their attendant modems and line-attachment hardware, resulting in lowered communications costs. At the same time, greater utilization of the remaining lines is achieved.

These types of terminals are accommodated by the multiple terminal access facility:

IBM 1050
IBM 2740 (basic)
IBM 2740 with Record Checking
IBM 2740 with Transmit Control
IBM 2740 with Transmit Control and Checking
IBM 2741
Western Union TWX

The multiple terminal access facility allows the network control program either to call any MTA terminal over a line defined as a call-out line, or to receive calls from MTA terminals over a line defined as a call-in line. The same line may be used for both call-out and call-in MTA operation. Lines used for call-out operation are included in dial sets, as for non-MTA operation.

When answering calls over a line defined as an MTA line (specified by the MTALIST operand of the LINE macro), the program automatically determines the type of terminal in terms of its line control discipline and transmission code employed. Once these have been determined, the network control program carries on sessions and logical connections in the usual way.

In addition, by analyzing a code entered by the terminal operator when calling the controller, the network control program can distinguish among terminals that, while of the same type, require differing terminal or line operating parameters or procedural options.

These parameters and options are:

- The carriage return rate.
- The presence or absence of the accelerated carrier return (ACR) feature for 1050 terminals.
- The length of the print line used by the terminal printer.
- The line speed, interrupt priority, modem data rate, and bit clocking options.
- The maximum number of sub-blocks to be accepted from the terminal during a single logical connection.
- The maximum number of attempts to recover from text-mode errors.

To use the multiple terminal access facility, you (1) define the types of terminals to be handled as MTA terminals (these may be any combination of the terminals listed above), and (2) specify the lines with which the network control program will communicate with each of the types of terminals. All lines used for MTA operation must operate in network control mode.

Your requirements for the MTA facility are specified in the MTALCST, MTALIST, MTATABL, and MTAPOLL macros; the MTALIST operand of the LINE macro, and the LCST operands of the TERMINAL macro; and the MTARTO and MTARTRY operands of the BUILD macro. An example of the use of MTA facilities appears in Appendix D.

Transmission of Attention Signals

The network control program, when transmitting to a terminal in network control mode, can respond to "attention" signals received from the terminal in either of two ways. (1) The program can interrupt its transmission to the terminal and immediately notify the host processor that the terminal has sent the attention signal. The program halts any remaining requests for that terminal. It is then up to the host processor to determine the next teleprocessing operation for the terminal. (2) The network control program can ignore the interrupt, and continue sending to the terminal.

In the FEATURE operand of the TERMINAL macro for each terminal equipped to send attention signals (IBM 1050, 2741; AT & T 83B3; WU 115A, TWX), specify the ATTN parameter. If you wish the network control program to interrupt its transmission upon receiving the attention signal, specify ATTN=ENABLED in the TERMINAL macro. If you wish the program to ignore the signal, specify ATTN=DISABLED (or omit the ATTN operand). The foregoing applies to attention signals received while the network control program is transmitting to a terminal. The program can also monitor a communication line operating in network control mode for an attention signal or a disconnect condition detected while the program is momentarily executing no read or write commands for the line, provided that an active session is in progress with that terminal. The program notifies the access method that it has detected an attention signal or a disconnect condition.

Specify whether this function is required in the MONITOR operand of the LINE macro. The option is required if terminals on the line will communicate with TSO applications. (The network control program performs the monitoring function only if directed to do so by command from the access method. TSO causes the access method to send the required command.)

Logical Keyboard Lock for TWX Terminals

The keyboard of a TWX terminal, unlike those of other start-stop terminals, cannot be locked by the network control program when no read or write operation is in effect for the terminal. (Locking the keyboard prevents data from being entered when the network control program is not ready to receive from the terminal.)

Therefore, the network control program sends TWX terminals a character sequence that "jogs" the printing mechanism when the program is not ready to receive. This serves as a signal to the terminal operator not to enter data.

The character used to jog the TWX printer mechanism is a null character, unless you specify a different character in the KBDLOCK operand of the GROUP macro. Any character specified should be a non-printing, non-spacing character that jogs the printer mechanism.

Carriage Return Delay

A terminal operator may press the return (carriage return) key of the terminal at the end of a message block he is entering from the keyboard. If the network control program sends message data to the terminal immediately after receiving the block the terminal has just sent, the first several characters of the data the program sends may be printed randomly during the return motion of the terminal's printing mechanism. To prevent this from happening, you may specify that the program pause momentarily after completing a read operation before starting the next write operation. This allows time for the printing mechanism to return to the left margin. Specify this function in the CRDLAY operand of the TERMINAL or COMP macro that represents the terminal printer. The program pauses between the read and write operations only if the message block received from the terminal ended with a carriage return (new line) character, or, for an IBM 1050 equipped with the automatic EOB feature, an EOB character.

This carriage return delay function is performed only for IBM 1050, 2740 Model 1, and 2741 terminals and Western Union TWX terminals (as specified in the TERM operand of the TERMINAL or COMP macro) and any multiple-terminal-access terminal (TERM=MTA).

Downshifting on Space Characters

Some AT & T 83B3, Western Union 115A, and World Trade teletypewriter terminals, upon sending or receiving a space character, automatically down-shift so that subsequent message text is in lower-case mode. Automatic downshifting avoids the need to send a LTRS character to effect downshifting. In the LINE macro for each teletypewriter line, indicate whether the terminals are equipped with the downshift function. Specify this function in the SPSHIFT operand if the line is to operate in network control mode, and in the FEATURE operand (SPACE parameter) if the line is to operate in emulation mode.

Deleting FIGS and LTRS Characters

Message data received from 83B3, 115A, and World Trade teletypewriter terminals contains the two case-shifting characters, FIGS and LTRS. If the lines to which such terminals are attached operate in network control mode, the network control program removes FIGS and LTRS characters from the data it transfers to the host processor, unless you specify, in the FGSLTRS operand of the LINE macro for such terminals, that the program is to leave these characters in the data. (The characters are not deleted from message data received over lines operating in emulation mode.)

TWX ID Exchange and Verification

The network control program can recognize an identification (ID) sequence from any TWX terminal that calls or is called by the communications controller over the switched telephone network, and can either check that sequence against a list of valid sequences within the program or pass the sequence to the access method for checking. (If the access method is DOS/VS VTAM, sequences must be passed to VTAM for checking.) Upon failing to recognize a sequence as valid, the network control program does not proceed with message transmission. Instead it either breaks the line connection, or it maintains the connection but forwards the unrecognized sequence to the access method for checking. In the latter case, the access method can signal the network control program to proceed with message transmission or to break the line connection.

After either the network control program or the access method checks the received sequence, the network control program can send an ID answerback sequence to the terminal before receiving text from the terminal.

Defining an Answerback Sequence

In the TWXID operand of the BUILD macro you may specify the answerback sequence to be sent to TWX terminals. Two different sequences may be specified: one to be sent when a terminal calls the controller, and the other to be sent when the controller calls a terminal. The maximum length of either sequence is 20 characters. In the CUIDLEN operand of the TERMINAL macro for each TWX terminal you must specify the length of the answerback sequence to be sent. If you omit the CUIDLEN operand, the program does not send the answerback sequence to that TWX terminal.

Answerback sequences are not sent to TWX terminals on lines currently operating in emulation mode.

Options for World Trade Teletypewriter Terminals

In addition to the downshift-on-space-character and FIGS/LTRS options mentioned above, there are two other procedural options for World Trade teletypewriter terminals.

Pad Characters

Some World Trade teletypewriter terminals have a motor that runs continuously whether or not the terminal is sending or receiving data. Others have motors that run only during data transmission; the motor stops automatically after about ten seconds have elapsed since the terminal sent or received a character. Terminals of the latter type must receive several pad, or idle, characters before receiving message characters, to allow sufficient time for the motor to reach operating speed. The number of characters required depends on the data rate on the communication line.

For lines running in emulation mode, the access method is responsible for including the appropriate number of idle characters in message data it sends to the terminal.

For lines running in network control mode, the network control program automatically sends the idle (pad) characters, if you specify the required number in the PADCNT operand of the GROUP macro for the group of lines to which such terminals are attached.

EOB and EOT Sequences

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

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

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

Specify the required EOB and EOT sequences in the EOB and EOT operands of the GROUP macro if any of the lines in the group are to operate in emulation mode, and in the WTTYEOB and WTTYEOT operands of the GROUP macro if any of the lines are to operate in network control mode.

Procedural Options for Operation in Emulation Mode

When defining a line that is always to operate in emulation mode, there are only three procedural options. These are (1) the type of line control discipline to be used for each line; (2) the terminal timeouts required, and (3)—for World Trade teletypewriters only—the end-of-block and end-of-transmission sequences to be recognized by the program.

Type of Line Control

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

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

Terminal Timeouts

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

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

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the program uses the timeout intervals indicated in the descriptions of these two operands for all lines in the group represented by that macro. Some applications may justify unlimited intervals, that is, no timeout at all. This also may be specified in the REPLYTO or TEXTTO operands.

EOB and EOT Sequences for World Trade Teletypewriter Terminals

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

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

Multi-Subchannel Line Access Facility

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

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

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

The MSLA facility can be used in the following ways:

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

Block Handling Options

Block handling refers to the optional message processing of message data within the communications controller. The network control program can process either message data from the host processor before sending it to a start-stop or BSC station, or message data received from a station before sending it to the access method. Processing is possible only when the message data between controller and station is transmitted in network control mode.

The IBM-supplied network control program modules provide two standard message processing functions. Each is performed by a block-handling routine invoked by a program generation macro instruction. In addition, user-coded block handling routines may be added to the network control program during the generation procedure. A 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 be included in a user block-handling routine.)

Two optional IBM-supplied block-handling routines allow for insertion of time-of-day and date into messages and automatic correction of incorrect message text. An additional IBM-supplied routine may be used to convert message data from BSC to SDLC format, when the SDLC/BSC data path function is used. (See *SDLC/BSC Data Path Function* under *BSC Subsystem Operation* in this chapter.)

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 the access method over the network control subchannel, or from a station over a line in network control mode.

The date may be in any of four formats: (1) month/day/year, for example, 10/21/75; (2) year followed by day of year, for example, 75.294 (October 21, 1975); (3) year/month/day (75/10/21): or (4) day/month/year (21/10/75).

The time of day 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

Automatic text correction is an editing function by which the network control program replaces text incorrectly entered from a terminal keyboard with the corrected characters the terminal operator subsequently sends. The 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 the program finds a sequence of three canceling characters, it deletes from the block those three characters plus the three immediately preceding characters.

A keyboard operator may, for instance, enter COMMUNCIATE and seeing that 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 specified the text correction option and designated *backspace* as the text canceling character, the text-correction block-handling routine deletes the five backspace character and C I A T E. The remaining characters 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 message text. For example if / is the character chosen, and a keyboard operator enters ATLANITC///TIC, the text correction block-handling routine corrects the word to ATLANTIC.

The EDIT macro specifies the text correction function.

SDLC/BSC Path Function Block Handling Routine

Use of the SDLC/BSC path function for message transmission between an SDLC logical unit and a designated BSC station (specified by the DATASW operand of the LU macro) requires that the network control program convert path information units (PIU) transmitted between the logical unit and the BSC station from the SDLC format to the BSC format, and vice versa. You may invoke an IBM-supplied conversion routine by specifying an SPAFPT3 macro in a block-handling routine that you assign to the BSC station. The block handler must be specified as executable at point 3 (that is, processing [conversion] of the data is done after receipt from the BSC station). (The code necessary to convert from the SDLC to the BSC format is always present in the network control program, so no macro is provided to call this routine.) Alternatively, you may code a user block-handling routine to do the BSC-to-SDLC format conversion and include it in a point 3 block handler via a UBHR macro.

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 communications controller assembler language (similar to the operating system assembler language), assemble it using the controller assembler, and then place the routine in a data set available to the network control program generation procedure. Then you include in the program generation source statements a UBHR macro instruction that specifies the name of your routine and the point at which the network control program is to execute it.

Guidelines for Writing User Block Handling Routines

User block handling routines permit you to add certain data handling functions to the network control program. These routines typically examine and manipulate incoming or outgoing data contained in NCP buffers. If you have a good general understanding of the network control program and the access method, you may add such routines to the program with little likelihood of disrupting the NCP code. On the other hand, routines that perform more complex functions, such as leasing and releasing buffers or scheduling input/output operations, require that you have an intimate understanding of the internal operation of the network control program and the access method. Adding such routines must be approached with caution to avoid disrupting the network control program logic.

Coding user block handling routines requires knowledge of the information in these publications: *IBM 3704 and 3705 Communications Controllers Principles of Operation* (GC30-3004), *IBM 3704 and 3705 Communications Controllers Assembler Language* (GC30-3003), and *IBM 3704 and 3705 Communications Controllers Network Control Program/VS Program Logic Manual* (GC30-3013).

The assembled object modules containing the user block handling routines must be placed in the data set specified by the USERLIB operand of the BUILD macro. During stage two of the program generation procedure, user block handling routines that you have specified in the appropriate UBHR macros are included in the generated NCP load module.

Specific rules and guidelines to be observed in coding user block handling routines are as follows:

- All registers may be used in a user block handling routine. Before passing control to a user routine, the network control program saves all registers; when receiving control from the routine, the NCP restores all registers.
- At entry to a user block handling routine, register two points to the queue on which the block being handled is enqueued.
- A POINT (2), (3) macro instruction causes the address of the block at the head of the queue to be returned in register three.
- You may use a SCAN macro instruction to scan the text in chained NCP buffers containing the block being processed.
- You may use the DEQUE, ENQUE, and INSERT macros to dequeue, enqueue, or insert the block whose address was returned by the POINT macro instruction.
- You may use the LEASE macro to obtain NCP buffers; you must release any buffers thus obtained with a RELEASE macro.
- Use a SYSXIT macro to return control from the user block handling routine to the network control program.
- If a user block handling routine is to be executed for more than one BSC or start-stop station or line, code it such that it is serially reusable.
- The network control program/VS program logic manual mentioned above describes the macro operand formats and gives details for use of supervisory macros.
- If the user routine changes the amount of message text accompanying a BTU (basic transmission unit), the routine must accordingly update the BCUTLEN field of the BTU and the data count fields of the buffer prefix areas.
- User block handling routines should not modify any part of the first 34 bytes (the BTU) of a header buffer or the first four bytes of any other buffer.
- Logic errors encountered in user block handling routines can cause the network control program to end abnormally (Abend). For many kinds of logic errors, a dump listing of the NCP will reveal the following: (1) The level 5 instruction address register (IAR–register 0) will point within the user block handling routine. (2) Bit 4 of storage location X'685' (indicating block handlers in execution) will be on. (3) The Abend code will appear at location X'760'. (Abend codes are described in the *Program Reference Handbook*.)

Associating Block Handling Routines with Stations

The requirements of the application determine how the network control program is to process messages before sending them to the teleprocessing network or the access method. The requirements may vary for different stations, or for different components of a station. You may wish, for example, to provide the text correction function for messages entered from a terminal keyboard but not for messages received from a paper 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 message data through the network control program. 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 has obtained the use of a communication line for transmitting the message to the station. Another block handler in the same set can include routines that process message data from the host processor *after* the program has obtained use of a line. This block handler may also include routines that process message data from a station before the network control program releases the line over which it received the data.

A third block handler in the set may be assigned to process message data received from a station after the program has released the line for use in communicating with another station.

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 TERMINAL, COMP, or CLUSTER 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 difficulties in network operations by means of seven diagnostic and service aids: (1) on-line testing (for lines and terminals), (2) address trace, (3) line trace for lines in network control mode, (4) line trace for lines in emulation mode, (5) abnormal end analysis, (6) panel tests, and (7) dynamic dumping of controller storage. These aids are useful in identifying malfunctions within the teleprocessing subsystem and the network control program. Some aids are standard (always present in the program); others are optional. Inclusion of the latter in the network control program, although optional, is recommended.

On-line Testing

On-line terminal testing is an important 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. A similar aid, on-line line testing, allows communication line problems to be diagnosed. These diagnostic aids, important in problem determination and on-line maintenance of lines in network control mode, are included in a network control program unless you exclude them via the OLT operand of the BUILD macro. (The network control program does not participate in on-line testing operations for lines in emulation mode. This is solely the responsibility of the access method.)

The network control program recognizes each test-request message entered from a terminal and passes it to the access method, just as it does a normal message. Recognizing the message as a test request, the program does not perform any block processing but sends it unchanged to the access method.

The access method detects that the message requests the on-line 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 these operations and returns responses as necessary to the access method. Upon analyzing the responses, the access method determines what further operations to perform and sends the network control program the appropriate interpretive commands.

The network control program is thus only an intermediary in on-line 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.

On-line terminal test operations require buffer space to hold the interpretive commands and an on-line test control block. These buffers, which the program obtains from the same pool from which it obtains buffers for normal operations on lines in network control mode, are required only for the duration of the test operation.

Note: Diagnostic programs that communicate with the network control program via VTAM or TCAM (for example, TOTE may impose restrictions on the NCP buffer size (specified in the BFRS operand of the BUILD macro). See the appropriate manuals for such diagnostic programs for restrictions that may apply.

The network control program can execute on-line testing operations concurrently for any number of communication lines operating in network control mode. Some extra buffer space is needed for each additional line on which on-line testing is being conducted. Teleprocessing operations on lines not undergoing testing can continue as usual.

Address Trace Facility

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 trace data in a trace table within control storage. When the desired data has been recorded, the contents of the trace table can be displayed on the control panel of the controller. The contents of controller storage can be transferred to the host processor via the Dump program and the contents of the trace table examined in the listing of the dump.

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.

The address trace function is performed only for lines operating in network control mode.

Line Trace Facility for Network Control Mode

The network control mode line trace facility is a service aid that permits detailed analysis of the operation of any communication line currently operating in network control mode. This facility records operating parameters of a line each time a level two interrupt occurs for that line. (Level two is the program interrupt level at which bit service or character service for the communication line is performed.) The program places network control mode trace information in buffers obtained as required from the buffer pool, then transfers these buffers at intervals to the host processor. The host processor should accumulate these line trace records in a data set (file) from which they may be printed out for analysis.

A line trace can be initiated at any time by request from the host processor, and ended at any time by a subsequent request, provided that the line is currently operating in network control mode.

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

One line at a time can be traced; any "activate line trace" requests beyond the first one are rejected until the first is ended.

The line trace facility for network control mode is always present in the network control program.

If the network control program includes emulation functions as well as network control functions (that is, TYPGEN=PEP is specified in the BUILD macro), the emulation mode line trace facility (see below) can be used to trace lines operating in emulation mode.

Line Trace Facility for Emulation Mode

The emulation mode line trace facility of the program is a service aid that permits detailed analysis of the operation of any communication line currently operating in emulation mode. This facility records operating parameters of a line each time a level two interrupt (except bit-service interrupt) or level three interrupt occurs for that line. (Level two is the program level at which bit service or character service for the communication line is performed. Level three is the program level at which the servicing of channel interrupts is performed.) The program accumulates this information within controller storage. (The emulation mode line trace, unlike the line trace for network control mode, does not accumulate the trace information in buffers and does not automatically transfer the buffer contents to the host processor. The contents of the controller storage must be dumped to make the line trace records available or dynamically dumped using the dynamic dump utility.)

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

Line traces of lines in emulation mode are initiated at the control panel of the communications controller or via the dynamic dump utility. Any number of lines may be traced concurrently.

The line trace facility for emulation mode is always present in a network control program with the PEP extension. The number of lines to be traced and the size of the trace table are specified in the LINETRC operand of the BUILD macro.

Abnormal End (ABEND) Facility

Programming errors detected during execution of non-supervisory portions of the network control program cause abnormal termination of program execution. Examination of abend codes within a storage dump can help in locating the error.

The optional Abend service aid extends detection of programming errors to the network control program *supervisor*, thus causing termination of the program before a supervisor error can be propagated into non-supervisory portions of the program. The Abend code appearing in the storage dump therefore gives a better indication of the location of a supervisor error, if one should occur, than a code reflecting a resultant error in the non-supervisory portion would give. Inclusion of the Abend option (by the ABEND operand of the BUILD macro) is recommended when you first begin using a network control program to control your teleprocessing network. Later, as experience demonstrates that your teleprocessing network operates routinely without abnormal termination of the network control program owing to program errors, the Abend option may be deleted from the program. The Abend facility is not available if the network control program executes only emulation functions (that is, if you specify TYPGEN=EP in the BUILD macro).

Panel Tests

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

The panel test function is always present in the network control program.

Dynamic Dump Facility (Emulation Mode Only)

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

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

The dynamic dump facility can be used only with a network control program that includes emulation functions.

Program Generation Options and Data Sets (Files)

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

Program Generation Options

Program generation options pertain to the kind of functions (network control or emulation) that the program is to perform, the type of communications controller (3704 or 3705, local or remote) in which the program will be executed, complete vs. partial generation procedure, and several assembly and link editing options. All program generation options are specified in the BUILD macro.

Type of Program to be Generated

As explained in Chapter 1, when defining the control program for a local communications controller you must decide whether the program is to perform network control functions only, emulation functions only, or both. Specify the choice in the TYPGEN operand of the BUILD macro.

Model of Controller

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

Partial Generation

Assembling and linkage editing the many modules making up a network control program necessarily consumes substantial processing time. Once you have generated a complete network control program, however, modifications resulting from changes in network configuration and procedural options can be effected in significantly less time, via *partial generation.*

In partial generation, only selected modules are reassembled; these are then linkage edited with the object modules that require no changes to produce the modified program.

To perform a partial generation, you code PARTIAL=YES and specify, in the CONDASM operand, the names of the modules requiring reassembly. (The modules requiring reassembly for each of various changes in the program functions are listed in Appendix H.) A control program that performs only emulation functions (TYPGEN=EP) consists of a single object module, so that reassembly and link-editing of selected modules is not possible. This is not a disadvantage, however, because complete generation of an emulation program requires much less time than a complete generation of a program that includes network control functions.

It is important that you retain all of the stage one and stage two assembly output listings and the object library (containing conditionally assembled modules) produced by the complete generation procedure. Saving this output facilitates subsequent partial generation.

Other Options

The remaining program generation options, and the operands of the BUILD macro by which you specify them, are:

- Whether stage two of the generation procedure is to consist of a single, multi-step job or a separate job for each step, and whether a job card is required (JOBCARD).
- The region size for stage two linkage edit job steps (LESIZE) [applicable only for OS/VS].
- Whether or not the generation procedure is to produce cross-reference listings for stage two assemblies (ASMXREF) [not applicable for TYPGEN=EP].
- The value of the TIME parameter in stage two assembly EXEC statements (TIME) [applicable only for OS/VS; not applicable for TYPGEN=EP].
- The type of device or class of devices to be used for utility data sets during stage two (UNIT) [applicable only for OS/VS].

Data Sets (Files) Used in the Generation Procedure

The names of various program data sets to be used in the generation procedure when generating under OS/VS are specified by the LOADLIB, OBJLIB, QUALIFY, UT1, UT2, UT3, and USERLIB operands of the BUILD macro. (USERLIB is not applicable for TYPGEN=EP.) The NEWNAME operand specifies the name to be given to the generated network control program load module.

The only file name required when generating under DOS/VS is NEWNAME.

Chapter 5: NCP Generation Macro Instructions for BSC—Start-Stop—(SDLC)

This chapter gives detailed descriptions of the macro instructions with which you define a network control program that includes network control functions, with or without emulation functions as well.

Macro Instruction Coding Conventions

The following conventions are used in the descriptions of the 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 (default value) to be given. 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.

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. Then follow, in alphabetical order, operands that are conditional or optional. These are enclosed in brackets—[].

Data set (file) names must begin with an alphabetic character or $, @, or #.

System Definition Macro Instructions

This section contains the system definition macro instructions, PCCU, BUILD and SYSCNTRL, to be used in defining a network control program that performs only network control functions or both network control and emulation functions. A program capable of performing both categories of functions is a network control program with the partitioned emulation programming (PEP) extension. (PCCU is a VTAM-only macro instruction.)

The section, *System Definition Macro Instructions* in Chapter 7 contains the single system definition macro, BUILD, to be used in defining a program that performs only emulation functions.

PCCU Macro Instruction (VTAM Only)

The PCCU macro instruction identifies for VTAM the 3704 or 3705 communications controller in which the network control program being defined is to be loaded and executed. This macro must appear at the beginning of the NCP generation input deck, preceding the BUILD macro, before the deck is provided to the VTAM initialization process. You may include it in the deck provided to the NCP generation procedure, but this is not required.

See the *VTAM System Programmer's Guide* for a complete description of the macro and its operands.

Name	*Operation*	*Operands*	
[symbol]	PCCU	[CUADDR=address]	(for local controller)
		[,AUTODMP={YES}] {NO }	
		[,AUTOIPL={YES}] {NO }	
		[,DUMPDS=ddname]	(OS/VS only)
		[,DUMPDS=SYS008]	(DOS/VS only)
		[,INITEST={YES}] {NO }	
		[,MAXDATA=size]	
		[,NCPLUB=SYSxxx]	(DOS/VS only)
		[,RNAME=rname]	(for remote controller only)

BUILD Macro Instruction

The first macro instruction in the program source statements (except for the VTAM-only macro, PCCU) is BUILD. This macro specifies:

- The type of program to be generated: network control program with PEP extension, or network control program without PEP extension.
- The type of controller (3704, 3705-I, or 3705-II; local or remote) that is to execute the network control program.
- The controller storage size.
- The size of buffers in the buffer pool.
- The name that is to be assigned to the network control program, resource resolution table, and block handler set resolution table load modules.
- The type and number of channel adapters in the communications controller.
- The range of CPU subchannel addresses used for communication between the network control program and the access method in emulation mode.
- The subarea address to be assigned to the network control program being defined.
- The upper limit of the range of subarea addresses assigned to network control programs in the network.
- Certain optional facilities that may be included in the network control program.
- Certain program generation options that may be desired.
- The names of program data sets used in the generation process.
- Whether a complete or a partial program generation is to be performed.

All operands whose descriptions appear below are applicable when defining a program that performs both network control and emulation functions (that is, TYPGEN=PEP or TYPGEN=PEP-LR is specified in this macro). If you are defining a program that performs only network control functions, these operands are not applicable and should be ignored:

DYNADMP	LINETRC
HICHAN	LOCHAN

Name *Operation* *Operands*

```
[symbol]    BUILD      LOADLIB=dsname,           (OS/VS only)

                       MAXSUBA=n,

                       MEMSIZE=n,

                       OBJLIB=dsname,            (OS/VS only)

                       SUBAREA=n,

                              {NCP   }
                              {NCP-LR}
                       TYPGEN={NCP-R }
                              {PEP   }
                              {PEP-LR}

                       [,ABEND={YES}]
                               {NO }

                       [,ANS={YES}]
                             {NO }

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

Name *Operation* *Operands*

```
[,BFRS= size  ]
        60

[,CA=(adapterl[,adapter 2])]

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

[,CONDASM={TABLE        }]
          {(value1,...)}

[,CSMHDR=chars]

[,CSMHDRC=chars]

[,CSMSG=chars]

[,CSMSGC=chars]

[,CUID=chars]

[,DIALTO={count} ]
         {60.0 }

[,DSABLTO={count} ]
          {3.0   }

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

[,ENABLTO={count} ]
          {2.2   }

[,ERASE={YES}]
        {NO }

[,HICHAN=(addrl[,addr2])]

[,ITEXTTO={count} ]
          {NONE  }

          {YES  }
[,JOBCARD={NO   }]
          {MULTI}

[,LESIZE=size]

[,LINETRC=( [YES][[,lines][,entries]])]

[,LOCHAN=(addrl[,addr2])]

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

[,MTARTO={count}]
         {1.0  }

[,MTARTRY={count}]
          {0     }
```

Name	*Operation*	*Operands*

```
                 {NCP001}
     [,NEWNAME={PEP001}]
                 {symbol}

     [,OLT={YES}]
           {NO }

     [,OPCSB2={YES}]
              {NO }

     [,PARTIAL={YES}]
               {NO }

     [,PWROFF={YES}]
              {NO }

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

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

     [,TIME=integer]

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

     [,TYPSYS={OS }]
              {DOS}

     [,TWXID=([inchars][,outchars])]

     [,UNIT=unit type]

     [,USERLIB=dsname]

     [,UT1=dsname]

     [,UT2=dsname]

     [,UT3=dsname]

     [,XBREAK={integer}]
              {NONE   }

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

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

LOADLIB=dsname

(generation under OS/VS only)

Specifies the name of a partitioned OS/VS data set that will contain the network control program load module, resource resolution table module, and block handler set resolution table module 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 or $, #, or @.) This data set must be cataloged.

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

MAXSUBA=n

Specifies the upper limit of the range of subarea addresses used within the network controlled by the access method. (Each network control program within the network must be assigned a unique subarea address via the SUBAREA operands of their respective BUILD macros. A single value is required for all MAXSUBA operands in all network control programs in the network active at the same time.

| **VTAM Note:** This value must be the same as specified in the MAXSUBA VTAM start parameter.)

Note: The term *subarea* is explained in the publication *VTAM Concepts and Planning* (GC27-6998).

This upper limit must equal or exceed the highest subarea address defined in any network control program with which the program you are defining will communicate. The maximum subarea address value is always a power of two, minus one, within the range 3–255 (that is, 3,7,15,31,63,127,255). If you specify a value that is not one of those listed, the generation procedure rounds the specified value to the next higher such value. (For example, any value you specify between 16 and 30 will be rounded to 31.)

The total number of resources that can be associated with any subarea address depends on the value of MAXSUBA, as follows:

Value of MAXSUBA	*Maximum Number of Resources Possible*
3	16383-x
7	8191-x
15	4095-x
31	2047-x
63	1023-x
127	511-x
255	255-x

The value of x is 1, if the network control program includes (1) SDLC stations only, or (2) start-stop and/or BSC stations only. The value of x is 2 if the program includes both (1) SDLC stations and (2) start-stop and/or BSC stations.

Note: Specifying an unnecessarily high value for *n* will waste NCP storage space (space is assigned for all subarea addresses whether used or not).

Example: Assume that the network includes two local communications controllers, to which subarea addresses 2 and 3 are assigned, and four remote controllers, to which addresses 4 through 7 are assigned. The highest address being 7, you would specify an upper limit of at least 7. If you wish to allow for adding more controllers to the network, however, you would specify a value greater than 7. The next higher value, 31, would allow up to 30 controllers (addresses 2 through 31) to be included in the network.

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).

The value of *n* must be one of the following:

For 3704 (MODEL=3704): 48 or 64
For 3705-I (MODEL=3705 or 3705-1): 48, 80, 112, 144, 176, 208, or 240
For 3705-II (MODEL=3705-2): 64, 96, 128, 160, 192, 224, or 256

This operand is required.

OBJLIB=dsname

(generation under OS/VS only)

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

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

SUBAREA=n

Specifies the subarea address to be assigned to the network control program you are defining. (Each network control program active in a network, each group of locally attached terminals served directly by the access method, and the access method itself must have a unique subarea address.)

Note: The term *subarea* is explained in the publication *VTAM Concepts and Planning* (GC27-6998).

The minimum valid subarea address is 2—that is, SUBAREA=2. The maximum address is the value you specify in the MAXSUBA operand of this macro.

This operand is required.

```
        {NCP   }
        {NCP-LR}
TYPGEN= {NCP-R }
        {PEP   }
        {PEP-LR}
```

Specifies (1) whether the program is to be executed in a local or a remote communications controller; (2) whether the program includes emulation functions in addition to network control functions: and (3) whether the program (in a local controller) can communicate with a remote controller. Select the appropriate parameter from the following:

	The program will:		
Parameter	*include emulation functions*	*communicate with a remote controller*	*The controller is a:*
NCP	No	No	Local
NCP-LR	No	Yes	Local
NCP-R	No	No	Remote
PEP	Yes	No	Local
PEP-LR	Yes	Yes	Local

(If you wish to generate a program that performs only emulation functions, do not use the macro instructions in this chapter—see Chapter 7 instead.)

This operand is required.

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

Specifies whether or not the optional ABEND facility is to be included in the network control program.

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

Specifies whether or not Automatic Network Shutdown is to be included in the 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.

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

Specifies whether or not you wish the generation procedure to produce cross-reference listings for the stage two program assemblies.

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

Specifies the size, in bytes, of buffers in the network control program buffer pool. (Buffers in the pool are used only for data received over the network control subchannel and over communication lines operating in network control mode.)

Specify *size* as a multiple of four bytes. The minimum size is 48 bytes (44 bytes if you omit the on-line testing facility from the program (BUILD: OLT=NO)). The maximum is 248. The generation procedure rounds the size you specify to the next higher multiple of four bytes, if you have not specified such a multiple.

To the value you specify (rounded, if necessary, to the next multiple) the generation procedure adds four bytes for control use. (These four bytes are never used to contain message data.)

Example: If you specify BFRS=70, the length of each buffer is 76 (70 is rounded to 72, and four bytes are added). If you specify BFRS=72, the length is also 76 (no rounding is necessary). If you omit the BFRS operand, the buffer length is 64 (60 + 4 bytes for control use).

Note: Diagnostic programs that communicate with the network control program via VTAM or TCAM (for example, TOTE, may impose restrictions on the values specified for the BFRS operand. See the appropriate manuals for such diagnostic programs for restrictions that may apply.

`[CA=(adapter1[,adapter2])]` *(local controller only)*

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

`adapter1`

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

`adapter2`

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

See Figure 5-1 for valid specifications of the CA operand.

This operand specifies the type and physical location of the installed adapters. See the description of the CHANTYP operand below for the *use* of the installed adapters.

Note: If you omit both the CA and the CHANTYP operands, the default values assumed for both are as follows:

For TYPGEN=NCP or TYPGEN=NCP-LR:

If MODEL=3704 is specified, the values assumed are: CA=TYPE1, CHANTYP=TYPE1.
If MODEL=3705 (3705-1) or MODEL=3705-2 is specified, the values assumed are:
CA=TYPE2, CHANTYP=TYPE2.

For TYPGEN=PEP or TYPGEN=PEP-LR:

If MODEL=3704 or MODEL=3705 (3705-1) is specified, the values assumed are:
CA=TYPE1, CHANTYP=TYPE1.
If MODEL=3705-2 is specified, the values assumed are: CA=TYPE4, CHANTYP=TYPE4.

If you omit the CA operand but code the CHANTYP operand, the values assumed for CA are equivalent to those specified in the CHANTYP operand.

Adapter Configuration		*Adapter Specification*
First CA (base module)	Second CA (expansion module)	CA=
Type 1	(none)	TYPE1[3,5]
Type 2	(none)	TYPE2
Type 3	(none)	TYPE3
Type 4	(none)	TYPE4[4,6]
Type 1	Type 2	(TYPE1,TYPE2)[1]
Type 1	Type 3	(TYPE1,TYPE3)[2]
Type 2	Type 2	(TYPE2,TYPE2)
Type 2	Type 3	(TYPE2,TYPE3)
Type 3	Type 2	(TYPE3,TYPE2)
Type 3	Type 3	(TYPE3,TYPE3)
Type 4	Type 2	(TYPE4,TYPE2)[1]
Type 4	Type 3	(TYPE4,TYPE3)[2]
Type 4	Type 4	(TYPE4,TYPE4)

[1] Valid for TYPGEN=EP only if Type 2 CA is disabled.
[2] Valid for TYPGEN=EP only if Type 3 CA is disabled.

If CA operand is omitted:

CA=		MODEL=		TYPGEN=	
[3] TYPE1	is assumed if	3704 or 3705	and	EP, PEP, or PEP-LR	are specified
[4] TYPE4		3705-2		EP, PEP, or PEP-LR	
[5] TYPE1		3704		NCP or NCP-LR	
[6] TYPE2		3705 or 3705-2		NCP or NCP-LR	

Figure 5-1. CA Operand Specifications Valid for Each 3704 and 3705 Channel Adapter Configuration

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

Specifies the use of the channel adapters installed in the communications controller. (Accepted values for *ptype* and *stype* are TYPE1, TYPE2, TYPE3, and TYPE4)

`ptype`

Specifies the type of the primary channel adapter to be used for initial network control program operation as type 1, type 2, type 3, or type 4. If you omit *ptype* and the controller is a 3705 (MODEL=3705 or MODEL=3705-2), the primary channel adapter is assumed to be type 2. If you omit *ptype* and the controller is a 3704 (MODEL=3704), the primary channel adapter is assumed to be type 1. (A 3704 can have only a type 1 channel adapter.)

`stype`

Specifies that the secondary channel adapter is installed and identifies it as type 1, type 2, type 3, or type 4. If you omit *stype*, it is assumed that the secondary channel adapter is not installed. (Omit *stype* if you code MODEL=3704; a 3704 cannot have a secondary channel adapter.)

If you code *ptype* as TYPE1, *stype* cannot also be coded as TYPE1.

Figure 5-2 shows, for each of the thirteen possible 3705 channel adapter configurations, the permissible values of the CHANTYP operand. A controller can execute a network control program if there appears, opposite the adapter configuration, an adapter specification that matches the specification in the BUILD macro.

Note 1: The same network control program can be executed in controllers with dissimilar adapter configurations if the program's adapter specification appears opposite each of the adapter configurations with which you intend to use the program. If, for example, you wish to define a program that can be executed in either a controller having a single type 1 adapter or a controller having a type 1 and a type 2 adapter (with the type 1 adapter installed in the base module), you would specify CHANTYP=(TYPE1). (Before loading the program into the controller having two adapters, the unused adapter must be disabled, as note 1 directs.) The program could not, however, be executed in a controller having one or two type 2 channel adapters. (This example assumes that the program performs only network control functions.)

Note 2: See the description of the CA operand for default values assumed for this operand if the CA operand is omitted.

Adapter	*Configuration*	*Adapter Specification*
First CA (base module)	Second CA (expansion module)	CHANTYP=
Type 1	(none)	TYPE1
Type 2	(none)	TYPE2
Type 3	(none)	TYPE3
Type 4	(none)	TYPE4
Type 1	Type 2	TYPE1[1] TYPE2[2] (TYPE1,TYPE2)[5] (TYPE2,TYPE1)[3]

Figure 5-2. Channel Adapter Specifications Permissible for Each 3705 Channel Adapter Configuration (Part 1 of 2)

Adapter	*Configuration*	*Adapter Specification*
Type 1	Type 3	TYPE1[1]
		TYPE3[2]
		(TYPE1,TYPE3)[5]
		(TYPE3,TYPE1)[4]
Type 2	Type 2	TYPE2[1,2]
		TYPE2,TYPE2)[2,5]
Type 2	Type 3	TYPE2[1,2]
		TYPE3[1,2]
		(TYPE2,TYPE3)[2,5]
		(TYPE3,TYPE2)[2,5]
Type 3	Type 2	TYPE3[1,2]
		TYPE2[1,2]
		(TYPE3,TYPE2)[2,5]
		(TYPE2,TYPE3)[2,5]
Type 3	Type 3	TYPE3[1,2]
		(TYPE3,TYPE3)[2,5]
Type 4	Type 2	TYPE4[1]
		TYPE2[1]
		(TYPE4,TYPE2)[5]
		(TYPE2,TYPE4)[5]
Type 4	Type 3	TYPE4[1],
		TYPE3[1,2]
		(TYPE4,TYPE3)[5]
		(TYPE3,TYPE4)[5]
Type 4	Type 4	TYPE 4
		(TYPE4,TYPE4)[5]

[1] The unused channel adapter must be disabled at the 3705 control panel before the program is loaded. Failure to do so may result in an "abend" condition.

[2] Invalid if TYPGEN=PEP or TYPGEN=PEP-LR is specified.

[3] This is the only valid choice if you wish to have network control mode operations conducted over the type 2 channel adapter and emulation mode operations over the type 1 or type 4 channel adapter.

[4] This is the only valid choice if you wish to have network control mode operations conducted over the type 3 channel adapter and emulation mode operations over the type 1 or type 4 channel adapter.

[5] During initialization, the network control program will loop until the unused channel adapter is disabled (goes off line).

Figure 5-2. Channel Adapter Specifications Permissible for Each 3705 Channel Adapter Configuration (Part 2 of 2)

```
[CONDASM={TABLE          }  ]
         {(value1,...)}
```

Specifies, for a partial program generation, which conditionally assembled network control program modules are to be reassembled.

This operand is valid only if you specify PARTIAL=YES in this BUILD macro.

TABLE

Specifies that only the modules containing network control program tables are to be reassembled.

`value1,...`

Identifies specific modules to be reassembled. *value1,...* represents a sequence of two-digit numbers corresponding to the last two digits of the names of the modules to be assembled. For example, to reassemble modules SYSCG007 and SYSCG00A, you would code CONDASM=(07,0A).

The modules that may be individually reassembled, and the corresponding values to be coded in the CONDASM operand, are as follows:

SYSCG000	00	SYSCG009	09
SYSCG001	01	SYSCG00A	0A
SYSCG002	02	SYSCG00B	0B
SYSCG003	03	SYSCG00C	0C
SYSCG006	06	SYSCG00D	0D
SYSCG007	07	SYSCG00E	0E
SYSCG008	08	SYSCG010	10

The network control program tables are always assembled, regardless of which specific modules you specify in CONDASM=(value1,...).

Appendix H lists the module that must be reassembled for various changes in program functions.

[CSMHDR=chars]

(IBM 3271, 3275 only)

Specifies the header of the critical situation message to be sent to any IBM 3271 or 3275 terminals in the network controlled by the network control program being defined. A header must be specified for these terminals if the program is to send critical situation messages to them (see the CSMSG operand of this macro). The header must contain the appropriate device control characters and may also include any other valid characters desired. (See the *IBM 3270 Component Description* manual, GA27-2749, for the required device control characters.) The header specified by this operand is sent only to 3271 and 3275 terminals; you need not code this operand if your network does not include terminals of these types.

Code *chars* as the hexadecimal representation of the EBCDIC characters to be sent.

You may specify up to 255 hexadecimal digits in this operand. If the header required exceeds this length, code a CSMHDRC operand for the remaining characters, up to a combined total of 476 hexadecimal digits. Code an even number of hexadecimal digits, since each two hexadecimal digits represent one EBCDIC character.

Note: Although CSMHDR and CSMHDRC may together specify up to 476 hexadecimal digits of header and CSMSG and CSMSGC may together specify up to 476 hexadecimal digits of text, the combined total of header and text specified in these four operands may not exceed 476 hexadecimal digits.

`[CSMHDRC=chars]`

(IBM 3271, 3275 only)

Specifies up to 221 additional hexadecimal digits of header for the critical situation message specified by the CSMHDR operand. This operand is valid only if you also code CSMHDR. See the restriction on total header length under the CSMHDR operand.

`[CSMSG=chars]`

(start-stop and BSC stations only)

Specifies the text of the "critical situation" message to be sent to the active start-stop and BSC on lines in network control mode 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. Code *chars* as the hexadecimal representation of the EBCDIC characters desired.

If this operand is omitted, the network control program will not notify stations before automatic network shutdown occurs. The maximum number of hexadecimal digits is 255.

If the message required exceeds this length, code a CSMSGC operand for the remaining characters, up to a combined total of 476 hexadecimal digits. Code an even number of hexadecimal digits, since each two digits represent one EBCDIC character.

Note: Although CSMSG and CSGMSGC may together specify up to 476 hexadecimal digits of text and CSMHDR and CSMHDRC may together specify up to 476 hexadecimal digits of header, the combined total of header and text specified in these four operands may not exceed 476 hexadecimal digits.

`[CSMSGC=chars]`

(start-stop and BSC stations only)

Specifies up to 221 additional hexadecimal digits of text for the critical situation message specified by the CSMSGC operand. This operand is valid only if you also code CSMSG. See the restriction on total text length under the CSMSG operand.

If the network to be controlled by the program being defined includes any IBM 3271 or 3275 terminals, the CSMHDR operand is also required.

`[CUID=chars]`

(BSC stations only)

Specifies the characters in the controller identification sequence that the network control program may send to BSC stations on switched lines operating in network control mode. Code *chars* as the hexadecimal representation of the EBCDIC characters to be sent. You may specify a maximum of 40 hexadecimal digits (equivalent to 20 EBCDIC 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 (or higher-level macro), the CUIDLEN operand. 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 (see the IDLIST macro description), but it will not send the controller ID sequence in return.

```
[DIALTO={count}]
        {60.0 }
```

Specifies the timeout to be used by the network control program in detecting failure of the automatic calling unit's "abandon call and retry" signal. Specify the timeout either as an integral number of seconds (DIALTO=30) or to tenths of a second (DIALTO=40.4).

Use of the default value (60 seconds) is recommended unless the system designer recommends a different one.

The maximum timeout you may specify is 1632 seconds.

Note: See the section, *Restriction on Number of Time Intervals*, following the description of the GROUP macro.

```
[DSABLTO={count}]
         {3.0  }
```

(switched lines only)

Specifies the timeout to be used by the network control program in detecting the failure of the "data set ready" signal line of the modem to be turned off when the switched line attached to the modem is disabled. This operand applies only to line operation in network control mode. Specify the timeout either as an integral number of seconds (DSABLTO=5) or to tenths of a second (DSABLTO=7.5).

The maximum timeout you may specify is 1632 seconds.

The line remains disabled for the period specified, regardless of whether the "data set ready" signal line is turned off within the period.

Note: See the section, *Restriction on Number of Time Intervals*, following the description of the GROUP macro.

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

(applicable to emulation mode only)

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

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

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

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

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

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

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

For Type 1 channel adapter: (1) If the native subchannel of the adapter is used for network control operations, an emulation subchannel of this adapter can be used for dynamic dump data transfer; the native subchannel cannot be used for this purpose. (2) If the native subchannel of the adapter is *not* used for network control operations, the native subchannel (but not an emulation subchannel) can be used for dynamic dump data transfer.

For Type 2 and Type 3 channel adapter: These types of channel adapter cannot be used for dynamic dump data transfer.

For Type 4 channel adapter: (1) An emulation subchannel can be used for dynamic dump data transfer regardless of whether the native subchannel is used for network control operations. (2) The native subchannel can be used for dynamic dump data transfer if this subchannel is not used for network control operations.

Figure 5-3 shows, for each combination of CA and CHANTYP operand specifications, the valid ways the DYNADMP operand can be used to specify subchannels to be used for transfer of dynamic dump data to the host processor.

If CA=	*and CHANTYP=*	*Then to allow dynamic dump data transfer over these subchannels*:*	*Code DYNADMP=*
TYPE1	TYPE1	ESC$_1$	(YES,*addr1*)
(TYPE1,TYPE2)	TYPE1	ESC$_1$	(YES,*addr1*)
	(TYPE1,TYPE2)	ESC$_1$	(YES,*addr1*)
	(TYPE2,TYPE1)	NSC$_1$	(YES,NSC)
(TYPE1,TYPE3)	TYPE1	ESC$_1$	(YES,*addr1*)
	(TYPE1,TYPE3)	ESC$_1$	(YES,*addr1*)
	(TYPE3,TYPE1)	NSC$_1$	(YES,NSC)
TYPE4	TYPE4	ESC$_1$	(YES,*addr1*)
(TYPE4,TYPE4)	TYPE4	ESC$_1$	(YES,*addr1*)
		ESC$_2$	(YES,NONE,*addr2*)
		ESC$_1$ and ESC$_2$	(YES,*addr1,addr2*)
(TYPE4,TYPE2)	TYPE2	ESC$_1$	(YES,*addr1*)
		NSC$_1$	(YES,NSC)
(TYPE4,TYPE3)	TYPE3	ESC$_1$	(YES,*addr1*)
		NSC$_1$	(YES,NSC)

*ESC – emulation subchannel
NSC – native subchannel

Subscripts indicate channel adapter
$_1$first (base) adapter
$_2$second (expansion) adapter

Figure 5-3. Valid Subchannel Address Specifications for Dynamic Dump Data Transfer

The dynamic dump facility cannot be included in a program that does not perform emulation functions.

```
[ENABLTO={count}]
        {2.2  }
```

Specifies the timeout to be used by the network control program in detecting the failure of the "data set ready" signal line of the modem to be turned on when the communication line attached to the modem is enabled (for nonswitched lines) or when a dialing operation is completed–that is, the automatic calling unit has signalled connection (for switched lines). This operand applies only to line operation in network control mode. Specify the timeout either as an integral number of seconds (ENABLTO=3) or to tenths of a second (ENABLTO=3.2).

The maximum timeout you may specify is 1632 seconds.

For a nonswitched line or a switched line over which calls are made by an automatic calling unit (ACU), the default value of 2.2 seconds is usually appropriate. It is not appropriate, however, if the network control program being defined includes any switched backup links to SDLC stations. Such a link requires an enable timeout sufficiently long that it will not expire before the system operator, when manually calling the station, dials the telephone number, receives an answer, and places the modem (data set) in data mode. This process may typically take from 30 seconds to more than a minute. The value you specify in ENABLTO should exceed this interval. On the other hand, the timeout value should be no greater than necessary; otherwise it will needlessly extend the time required to shut down the network control program if shutdown is initiated while a line is being activated by command from the access method. (The network control program does not shut down until all lines are deactivated.)

Note: See the section, *Restriction on Number of Time Intervals*, following the description of the GROUP macro.

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

Specifies whether or not the network control program is to include the buffer erase function.

ERASE=YES is required if you wish to specify CDATA=YES in any CLUSTER, TERMINAL, or COMP macro within the program.

```
[HICHAN=(addr1[,addr2])]
```

(applicable to emulation functions only)

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

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

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

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

For a program to be executed in a 3705, the address must equal 4*n*-1, where *n* is an integer equal to or exceeding 1. Choose the appropriate subchannel address from the list below:

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

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

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

This operand is required if the program includes emulation functions.

Note: Also see description of LOCHAN operand.

```
[ITEXTTO={count}]
        {NONE }
```

(start-stop and BSC stations only)

Specifies the text timeout interval to be used by the network control program for any terminal or component for which INHIBIT=TEXTTO is specified in the TERMINAL or COMP macro.

The maximum timeout you may specify is 1632 seconds.

ITEXTTO=NONE specifies that no timeout is to occur.

Note: See the section, *Restriction on Number of Time Intervals* following the description of the GROUP macro.

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

(generation under OS/VS only)

Specifies whether or not the program generation procedure is to provide a job card for the stage two input stream, and whether the input stream will consist of more than one job.

The job card provided is in the form

```
//NCPGENnn JOB1,'NCPGENERATION',MSGLEVEL=1
```

If you specify JOBCARD=YES, or omit the operand, a single job card is provided and the program generation input stream consists of a single, multi-step job. The job card label is //NCPGEN00.

If you specify JOBCARD=NO, no job card is provided and you must therefore:

1. Specify DD DATA in the SYSIN card for the stage one assembly step.
2. Place a REPRO statement, immediately followed by an OS/VS job card, in the input stream preceding the BUILD macro.

If you specify JOBCARD=MULTI, a job card is provided for each step and the input stream therefore consists of multiple jobs. The job card labels are //NCPGENnn, where *nn* is a sequential identification number provided by the generation procedure.

Note: If you code JOBCARD=MULTI, you may specify a job card different from the one shown by using the IEBUPDTE utility program to change the job statement information in the stage 1 macro library. See Chapter 8 for information on this procedure.

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

Specifies the OS/VS region size, in K (1024) bytes, to be used by all linkage editor job steps during stage two of program generation. The number you specify is reduced by 10 and used as *value1* of the linkage editor SIZE parameter. *Value2* of the SIZE parameter is always 48 (K), regardless of what you specify in the LESIZE operand. *size* must exceed 10 and be less than 16384 (16,384 K bytes).

If you omit the LESIZE operand, the EXEC card used for the linkage editor job steps will have a REGION parameter of 384K and PARM parameter values of 374 (for *value1*) and 48 (for *value2*).

`[LINETRC=([YES][[,lines][,entries]])]` *(applicable to emulation functions only)*

Specifies the maximum number of lines in emulation mode that can be traced concurrently and the number of trace table entries provided. The line trace functions may be initiated from the control panel of the communications controller. (The *Control Panel Guide* (see Preface) explains the use of the line trace option.)

`lines`

Specifies the maximum number of lines that are to be traced concurrently. If you omit this parameter, all lines currently operating in emulation mode can be traced at once. The minimum value of *lines* is 1; the maximum is 352.

`entries`

Specifies the number of eight-byte entries in the trace table. The minimum is 12 entries; the maximum is 23,680.

If you omit this parameter, the trace table will contain 200 eight-byte entries.

`[LOCHAN=(addr1[,addr2])]` *(applicable only to emulation functions)*

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

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

This operand defines for each channel adapter the lowest emulation subchannel address associated with that adapter, regardless of whether this address is used for an emulation-mode line or for dynamic dump data transfer.

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

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

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

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

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

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

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

The address range specified by the LOCHAN and HICHAN operands applies only to emulation subchannels. The network control (native) subchannel address may, but need not, lie within this range. *Exception:* The network control subchannel address must not be within the LOCHAN-HICHAN range if the communications controller is equipped with two type 4 channel adapters.

It is recommended that values specified by the LOCHAN and HICHAN operands correspond to the low and high subchannel addresses actually installed within the controller. However, the program generation procedure does not verify that this is the case. Failure to follow this recommendation may cause the following:

- All commands (except Sense, Test I/O, and I/O No-op) issued to unassigned subchannels within the LOCHAN–HICHAN range will be rejected.
- All commands issued to subchannels outside the LOCHAN–HICHAN range will result in a permanently busy ("hung") subchannel.

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

Specifies whether the generated network control program is to be loaded into and executed by a 3705-II (MODEL=3705-2), a 3705-I (MODEL=3705), or 3704 (MODEL=3704). (A 3705-I may be specified as either MODEL=3705 or MODEL=3705-1.)

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

```
[MTARTO={count}]
        {1.0  }
```

(multiple-terminal-access lines only)

Specifies the reply timeout to be used when a terminal on a multiple-terminal-access line calls the controller. Specify the timeout either as an integral number of seconds (MTARTO=5) or to tenths of a second (MTARTO=5.5).

This operand is valid only if the teleprocessing network includes lines defined as multiple-terminal-access lines (see the MTALIST operand of the LINE macro).

Note: See the section, *Restriction on Number of Time Intervals,* following the description of the GROUP macro.

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

(multiple-terminal-access lines only)

Specifies the number of times the network control program is to retry the multiple-terminal-access sign-on procedure after identifying the type of MTA terminal that has called this controller.

The maximum number of retries is 255.

This operand is valid only if the teleprocessing network includes lines defined as multiple-terminal-access lines (see the MTALIST operand of the LINE macro).

```
          {NCP001}
[NEWNAME={PEP001}]
          {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. If a block handler set resolution table load module is generated, its name is the name you specify as *symbol*, followed by the letter B.)

Alternatively, specify NCP001 or PEP001 in this operand, whichever is appropriate. If you omit the operand, the name assigned is NCP001 if in the TYPGEN operand you specify NCP, NCP-LR, or NCP-R; the name is PEP001 if you specify PEP or PEP-LR.

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

Specifies whether or not the optional on-line terminal test and on-line line test facilities (for lines in network control mode) are to be included in the network control program. Code OLT=NO to omit the facilities; code OLT=YES (or omit the operand) to include them.

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

(applicable to emulation mode only)

Specifies that a 20-byte data buffer is to be provided for communication lines (1) that are serviced by a type 2 communication scanner, (2) which you have specified as operable in emulation mode, and (3) for which you have specified CHNPRI=HIGH in the LINE macro. The buffers, which are permanently assigned to the line (that is, they are not obtained from the NCP buffer pool), provide extra protection against overruns that can result from temporary slowdowns in channel operation or temporary peaks in data traffic in the network. Lines serviced by a type 2 scanner for which you do not specify OPCSB2=YES have two four-byte buffers.

The 20-byte buffer is used only when the line is operating in emulation mode.

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

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

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

Specifies whether or not a partial program generation is to be performed.

If you code PARTIAL=YES, only the tables and conditionally assembled modules specified in the CONDASM operand are assembled and linkage edited with the remaining object modules. The conditionally assembled modules not specified by the CONDASM operand are obtained from the library specified by the OBJLIB operand. (You must have assembled these modules during a previous program generation for which the OBJLIB operand of the BUILD macro specified the same library name.) The modules assembled by the partial generation procedure replace the corresponding modules from the previous generation.

Appendix H lists the modules that must be reassembled for various changes in program functions.

If you code PARTIAL=NO (or omit the operand), a complete program generation is performed.

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

(local NCP only; VTAM users only)

Specifies, for a network control program to be executed in a local communications controller, whether the program will turn off the remote controller's power upon command from VTAM. PWROFF=YES is valid only if the remote controller is equipped with the remote power off feature. (Power can be turned on again only at the control panel of the remote controller.)

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

(generation under OS/VS only)

Specifies the first-level qualifier for OS/VS data sets specified by the LOADLIB, 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 alphabetic (including $, @ and #). (Omit the period [.] that separates the qualifier and the data set name; the generation procedure appends the period to the qualifier you specify.)

`NONE`

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

`SYS1`

Specifies that SYS1 is to be used as the qualifier.

```
          {12}
[SLODOWN={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 lines operating in network control mode and from the network control subchannel, but continues to send data over the lines and the subchannel. This procedure reduces the number of buffers in use.

Slowdown mode is entered when fewer than one-half (SLODOWN=50), one-quarter (SLODOWN=25) or one-eighth (SLODOWN=12 or operand is omitted) of the buffers are available.

During initialization, the NCP dynamically increases the percentage you specify if the minimum NCP buffer requirements cannot otherwise be met. The minimum number of buffers that the program must contain for each percentage value is: 80 buffers, for 12 percent; 40 buffers, for 25 percent; and 20 buffers, for 50 percent.

If the number of buffers contained in the network control program is less than 20, the program Abends.

`[TIME=integer]` *(applicable to OS/VS only)*

Specifies the time value, in minutes, to be used as the TIME parameter in the stage two assembly EXEC statements. *integer* must be greater than 0 and less than 1441.

If you omit this operand, no TIME parameter is used for the stage two assembly EXEC statements.

```
         { NO              }
[TRACE={( 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, from ten [TRACE=(YES,10)] to 256 [TRACE=(YES,256)]. If you omit the number or specify fewer than ten, the table will contain ten entries.

`[TWXID=( [inchars] [,outchars] )]` *(TWX terminals on lines in network control mode only)*

Specifies the EBCDIC ID answerback sequence, in hexadecimal representation, for all TWX 33/35 terminals with which the program communicates in network control mode.

inchars specifies the sequence to be sent to TWX terminals when the terminal calls the controller.

outchars specifies the sequence to be sent to TWX terminals when the controller calls the terminal.

inchars and *outchars* may each contain up to 40 hexadecimal digits (equivalent to 20 EBCDIC characters).

Note: The network control program sends the answerback sequence not only when calling or answering the TWX terminal, but also at the end of each outgoing message it sends to the terminal. When defining the answerback sequence you should therefore consider the time needed to send and print the sequence. If, for instance, each session with the terminal includes a series of short messages sent by the program (each message followed by the answerback sequence), you may wish to limit the sequence to a very few characters to avoid an excessive proportion of session time being expended in transmitting and printing the answerback sequences.

The answerback sequence may comprise any valid TWX characters, including non-printing characters such as carriage return, line feed, LTRS, and FIGS.

This operand has no meaning if no TWX terminals are included in the network, or if communication with TWX terminals is only in emulation mode.

```
[TYPSYS={OS }]
        {DOS}
```

Specifies whether stage two of the network control program generation procedure is to be run under OS/VS or DOS/VS.

`[UNIT=unit type]`

(generation under OS/VS only)

Specifies the type of device to be used for the assembler and linkage editor utility data sets during stage two of program generation under OS/VS. You may specify either an actual device type (for example, UNIT=2311) or the name of a class of devices (for example, UNIT=SYSDA). The maximum number of characters you may specify is eight.

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and SYSDA is assumed for the linkage editing steps.

Note: The utility data set for the linkage editor must reside on a direct-access device.

`[USERLIB=dsname]`

(generation under OS/VS only)

Specifies the name of a partitioned OS/VS data set that contains the user-written block handlers. (The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set name may be up to eight characters long; the first must be alphabetic or $, #, or @.) This data set must be cataloged.

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

`[UT1=dsname]`

(generation under OS/VS only)

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

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

`[UT2=dsname]`

(generation under OS/VS only)

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

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

`[UT3=dsname]`

(generation under OS/VS only)

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

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

```
[XBREAK={integer}]
        {NONE   }
```

(lines in network control mode only)

`integer`

Specifies the number of break characters the network control program is to place on the line to interrupt transmission from the terminal. This is applicable only for lines in network control mode and terminals for which the LINE and TERMINAL macros specify DUPLEX=FULL and FEATURE=BREAK, respectively.

The minimum value is 3; the maximum is 255.

NONE

Specifies that the network control program will not send break characters.

If you omit the XBREAK operand, a value of 3 is assumed if the network includes any duplex lines (DUPLEX=FULL in the LINE [or GROUP] macro) to which terminals having the break function are attached (FEATURE=BREAK in the TERMINAL [or higher-level] macro).

If the network does not include such lines and terminals, XBREAK=NONE is assumed if you omit the XBREAK operand.

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

(BSC stations on lines in network control mode only)

Specifies whether the network control program is to insert transparent ITB sequences and error-information blocks in transparent text sent to stations for which intermediate block checking is specified, as follows:

If you specify XITB=YES, and the *first* parameter of the ITBMODE operand of the TERMINAL macro for the station specifies intermediate block checking, the program substitutes an error information block for each DLE ITB sequence in transparent text received from the station.

If you specify XITB=YES and the *second* parameter of the ITBMODE operand specifies intermediate block checking, the program inserts DLE ITB sequences into transparent text being sent to the station.

If you specify XITB=NO (or omit the XITB operand), no insertion of DLE ITB sequences and EIBs is performed for any BSC station.

SYSCNTRL Macro Instruction

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

Figure 5-4 shows which options are required by VTAM and TCAM.

This macro is required and must appear directly following the BUILD macro.

Name	*Operation*	*Operands*
`[symbol]`	`SYSCNTRL`	`OPTIONS=(entry,...)`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`OPTIONS=(entry,...)`

Specifies which of the dynamic control facilities are to be included in the network control program. *entry* may be any value listed in Figure 5-4.

Entry	Facility
Always required by VTAM:	
BHSASSC	Modify block handler set
ENDCALL	Physical disconnect
MODE	Set destination mode
RCNTRL	Request control mode reset
RCOND	Reset conditional
RECMD	Reset at end of command
RIMM	Reset immediate
Required by VTAM for certain operator control functions:	
NAKLIM	Change line negative polling response limit
SESSION	Change session limit
SSPAUSE	Change service-seeking pause
XMTLMT	Change device transmission limit

(See *VTAM System Programmer's Guide* for operator control functions requiring these options.)

Entry	Facility
Always required by TCAM:	
RCOND	Reset conditional
RIMM	Reset immediate
Required by TCAM for certain operator control functions:	
BACKUP	Switched network backup
BHSASSC	Modify block handler set
DVSINIT	Change device session initiation information
LNSTAT	Display line status
MODE	Set destination mode
RECMD	Reset at end of command
SESINIT	Change line session initiation information

(See *TCAM Programmer's Guide* for operator control functions requiring these options.)

Figure 5-4. Dynamic Control Facilities Required by VTAM and TCAM

Configuration Definition Macro Instructions

This section contains the configuration definition macro instructions (HOST, CSB, IDLIST, SERVICE, LUPOOL, MTALCST, MTALIST, MTAPOLL, MTATABL, and DIALSET) to be used in defining a network control program/VS that performs network control functions only, or both network control and emulation functions. Also appearing here (following the IDLIST macro) is a VTAM-only macro, VIDLIST, that may be included in an NCP generation input deck but has meaning only to the VTAM initialization procedure.

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 access method.
- The size of the access method buffer unit used to receive data from the network control program.
- The number of buffer units the access method will allocate for receiving a data transfer.
- The number of bytes in the header prefixes used by the access method.
- The amount of time the network control program waits for a response by the host processor to an attention signal.
- The amount of time the network control program waits before presenting an attention signal on the channel after data becomes available for transfer to the host processor.

The HOST macro is required in either a local or a remote network control program. The values specified in the operands of the HOST macro coded in a remote network control program must be the same as those specified in the HOST macro of the local program with which the remote program communicates. (Omit the DELAY, STATMOD, and TIMEOUT operands from a HOST macro coded in a remote network control program.)

One HOST macro is required; it must appear among the nonpositional configuration macros that follow the two system macros (BUILD and SYSCNTRL) and must precede the first MTALCST macro, if any.

```
Name        Operation   Operands

[symbol]    HOST     INBFRS=count,
                     MAXBFRU=count,
                     UNITSZ=length
                     [,BFRPAD={n  }]
                              {28 }
                     [,DELAY={count}]      (local NCP only)
                             {0    }
                     [,STATMOD={YES}]     (local NCP only)
                               {NO }
                     [,TIMEOUT={count}]  (local NCP only)
                               {NONE }
```

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`INBFRS=count`

Specifies the number of controller buffers initially allocated for each data transfer to be received from the host processor over the network control subchannel. The minimum value is 1; the maximum is 255.

This operand is required.

`MAXBFRU=count`

Specifies the number of buffer units the access method will allocate for receiving data from the controller over the network control subchannel. The minimum is 1; the maximum is 255.

This operand is required.

See the description of the UNITSZ operand for a sample calculation.

VTAM Note: The value you specify in this operand must be less than the value of the *bno* subparameter of the LFBUF (DOS/VS only) or IOBUF (OS/VS only) parameter of the VTAM start parameter list.

`UNITSZ=length`

Specifies the size of the access method buffer units used for data transfers over the network control subchannel. (Specify the sum of (1) the length of the message data, (2) 13 bytes for control information, and (3) the number of pad characters specified in the BFRPAD operand.)

The access method must use one buffer unit size for all transfers over the network control subchannel. A buffer unit is the smallest unit of contiguous storage handled as buffer space; a buffer may consist of one or more units.

The NCP generation procedure accepts values between 1 and 65535, inclusive. However, the access method imposes the following minimum and maximum values (as of the time of publication of this manual):

	Minimum:	*Maximum:*
DOS/VS VTAM	88	2024
OS/VS1 VTAM	84	2012
OS/VS2 VTAM	84	4060
OS/VS TCAM	44	255

Use the values given in the *VTAM System Programmer's Guide* or *OS/VS TCAM System Programmer's Guide* (see Preface) instead of the values above, if they differ.

VTAM Note: This operand must specify the same value as the *bsz* subparameter of the IOBUF (OS/VS only) or LFBUF (DOS/VS only) parameter of the VTAM start parameter list.

The maximum number of characters the network control program will transfer to the host processor in a single operation over the network control subchannel equals MAXBFRU times UNITSZ, minus the value specified in the BFRPAD

operand. (VTAM requires that the result of this calculation must not be less than 84 (OS/VS) or 88 (DOS/VS) bytes.) *Example:* If you code MAXBFRU=5, UNITSZ=84, and BFRPAD=(28), the maximum number of characters sent to the host processor during a single channel operation is 5(84) minus 28, or 392 bytes. (This value includes the 13 bytes of control information associated with each response sent by the network control program.)

Caution: The maximum amount of data that the access method can send to the network control program should not exceed the amount that the NCP can return to the access method, as expressed by the formula above (MAXBFRU times UNITSZ, minus BFRPAD).

```
[BFRPAD={28} ]
        {15}
```

Specifies the number of pad characters the network control program is to transmit over the network control subchannel to the access method immediately preceding the control information for the response associated with a teleprocessing request.

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 value you specify in the BFRPAD operand should equal the size of the prefixes that the access method uses. The required value for OS/VS VTAM is 28 bytes; for DOS/VS VTAM, 15 bytes; for OS/VS TCAM, a minimum of 17 bytes.

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

(local NCP only)

Specifies the interval, to the nearest tenth of a second, that 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 signal to the host processor over the network control subchannel.

`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.

`0`

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

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

This operand is invalid for a remote network control program (TYPGEN= NCP-R).

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

Specifies whether the network control program is to use the status modifier option for sending responses to the access method in the host processor. Use of this option when the attention delay option is also used minimizes the number of asynchronous channel interrupts during data transfer between the network control program and the access method. This occurs because each time the access method sends data to the network control program, the NCP returns any accumulated response data to the access method as part of the same channel operation instead of sending a later attention interrupt to the host processor.

Use of the status modifier option is required (STATMOD=YES) if the access method in the host processor is to run under OS/VS. To maximize the increase in channel performance, also specify an attention delay of at lease 0.1 second in the DELAY operand of the HOST macro. If the access method is DOS/VS VTAM, the status modifier option is optional for release 33 (or higher) of DOS/VS and release 2 (or higher) of VTAM; for lower levels of VTAM or DOS/VS the option is not available and STATMOD=NO is therefore appropriate.

```
[TIMEOUT={count} ]
        {NONE  }
```

(local NCP only)

Specifies the interval in seconds that the network control program awaits a response to an Attention signal it has sent to the host processor over the network control subchannel before initiating automatic network shutdown, if ANS=YES is specified in the BUILD macro. Express this interval either 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 0.2 second; the maximum is 420.0 seconds.

NONE

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

If the automatic network shutdown facility is included in the program (ANS=YES), and you omit the TIMEOUT operand, TIMEOUT=420.0 is assumed. If you specify ANS=NO, and you omit the TIMEOUT operand, TIMEOUT=NONE is assumed.

If you have excluded the automatic network shutdown facility (ANS=NO is specified in the BUILD macro), omit this operand.

Upon failure of the host processor to respond to the network control program, the action the NCP takes depends on how you have specified the ANS and TIMEOUT operands: (1) If you specify ANS=YES (or omit the ANS operand) and TIMEOUT=count, the NCP performs automatic network shutdown, then waits for data over the channel. (2) If you specify ANS=NO and TIMEOUT=NONE, the NCP does not perform automatic network shutdown, but instead waits indefinitely for message traffic over the channel. (The combinations ANS=YES, TIMEOUT=NONE and ANS=NO, TIMEOUT=count are invalid.)

This operand is invalid for a remote network control program (TYPGEN=NCP-R).

Note: The interval specified in the TIMEOUT operand is not counted as among the 16 possible time intervals permitted in the network control program. (The limit of 16 applies only to communication line timeouts and intervals—not to channel timeouts.)

CSB Macro Instruction

The CSB macro specifies:

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

Each scanner in the controller must be represented by a CSB macro, regardless of whether the communication lines with which the scanner is associated are to operate in network control mode or in emulation mode, or a combination of the two.

```
Name          Operation      Operands

[symbol]      CSB            SPEED=(rate,...)

                             [,MOD={n}]
                                   {0}

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

                             [,WRAPLN=line addr]
```

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`SPEED=(rate,...)`

Specifies the internal oscillator (business machine clock) rates for up to four oscillators installed in the communication scanner. (Do not confuse this SPEED operand, which specifies the oscillator bit rates, with the SPEED operand of the LINE macro, which specifies the data rate for the communication line.) The speeds must be specified in the same order that the oscillators are installed on the scanner, in ascending order according to speed. Standard oscillator bit rates are shown in Figure 5-5.

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

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

This operand is required.

Rate	*Represents:*	*Rate*	*Represents:*
45	45.5 bps	150	150.0
50	50.0	200	200.0
56	56.89	300	300.0
74	74.2	600	600.0
75	75.0	950	950.0
100	100.0	1200	1200.0
110	110.0	2000	2000.0
134	134.5	2400	2400.0

Figure 5-5. Standard Communication Scanner Oscillator Bit Rates

```
[MOD={n} ]
     {0}
```

Specifies the location of the communication scanner, as shown in Figure 5-6. The line interface addresses valid for each scanner type and module location are given. (The 3704 has only one module.)

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

Figure 5-6. Location of Communication Scanners and Valid Line Interface Addresses

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

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

Valid designations for scanner type are:

If controller is a 3704 (MODEL=3704): TYPE1
TYPE2

If controller is a 3705-I (MODEL=3705): TYPE1 (valid only for MOD=0)
TYPE2
TYPE3 (valid only for MOD=1, 2, or 3))

If controller is a 3705-II (MODEL=3705-2): TYPE2
TYPE3

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

`[WRAPLN=line addr]`

(applicable only to emulation functions)

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

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

Specify the hexadecimal line address without framing quote characters, for example, WRAPLN=02F. The address specified must be within the range shown in Figure 5-6, and must appear in the ADDRESS operand of one of the LINE macros.

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

This operand is required if the program includes emulation functions (TYPGEN=PEP is specified).

IDLIST Macro Instruction

The IDLIST macro specifies:

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

Note: The IDLIST macro can be used only in a network control program that communicates with OS/VS VTAM or OS/VS TCAM. It is invalid for use with DOS/VS VTAM.

An identification list is optional for each switched line over which BSC or TWX stations will call or be called by the controller, provided that the line is operated in network control mode. 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. (The same list may be used for more than one line.)

Depending on how you code the IDSEQ operand, the network control program will only recognize and accept a received ID sequence (for either call-in or call-out operation), or (for call-in operation only) it may also determine which particular station has called.

Note: You may specify that ID sequences received from stations that call the controller be checked by the access method instead of (or in addition to) the NCP checking. Refer to descriptions of the VIDLIST (VTAM-only) macro (see *VTAM System Programmer's Guide)* and IDSEQ operand of the TERMINAL macro (in the present chapter). See the *OS/VS TCAM System Programmer's Guide* for information about ID verification by TCAM. ID sequences received during call-out operations are never passed to the access method.

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

[symbol]

Provides a name for the ID list and is required except as indicated in the description of the IDSEQ operand. *symbol* may be any valid assembler-language symbol. If *symbol* is required, the first character may not be $.

IDSEQ={(chars,...) }
{((chars,term name),...)}

Specifies the identification sequence for each of the stations that may call or be called by the controller, and optionally, the name of the TERMINAL macro for the station from which the sequence is expected. (Either (chars,...) or ((chars, term name),...) may be specified in this operand; the two cannot be intermixed.)

chars

Specifies one identification sequence that the network control program is to recognize as valid. Code *chars* as the hexadecimal representation of the

EBCDIC characters to be recognized. You may specify a maximum of 40 hexadecimal digits (equivalent to 20 EBCDIC characters) in one sequence.

Note: Any EOT, ENQ, or ACK characters sent by a TWX terminal as part of its ID sequence are deleted by the network control program as it receives the sequence into a buffer. It is therefore necessary to omit any of these three characters when specifying the ID sequence in the IDSEQ operand. Failure to omit them will cause the program never to recognize the received sequence.

`[termname]`

Specifies the name of the TERMINAL macro representing the station with which the sequence coded as *chars* is to be associated. (Do not specify a TERMINAL macro in which CTERM=YES is coded.)

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

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.)

Calculate the value for MAXLEN by adding:

For the list header: 4 bytes

For each entry in the list:

The number of bytes required to contain the ID sequence, plus either:

2 bytes (if you omit *termname* in the IDSEQ operand); or

4 bytes (if you specify *termname* in the IDSEQ operand.)

Round the size of each entry to the next higher fullword.

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 upon receiving an ID sequence it does not recognize 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 unrecognized ID sequences, code NOMATCH=PASS (or omit the operand). (If the network control program receives no ID sequence, this fact 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. Then, upon receiving an ID sequence it does not recognize, the network control program breaks the line connection.

VIDLIST Macro Instruction (VTAM Only)

The VIDLIST macro instruction specifies, for VTAM, a list of identification sequences for BSC or TWX stations that call the communications controller over a switched line operated in network control mode; VTAM checks the sequences specified in this macro. Either IDLIST or VIDLIST macros, or both, may be coded within the NCP generation deck, for the same communication line or for different lines. Use VIDLIST if you wish VTAM to check the received identification sequence; use IDLIST if you wish the network control program to check the sequences; and use both macros if you wish both programs to check ID sequences.

This macro, if used, must appear between the SYSCNTRL macro and the first GROUP macro. It may appear anywhere within the configuration definition macro instruction section of the network control program generation deck. All of the following are valid examples:

```
PCCU        PCCU        PCCU
BUILD       BUILD       BUILD
SYSCNTRL    SYSCNTRL    SYSCNTRL
CSB         CSB         VIDLIST
HOST        HOST        CSB
IDLIST      VIDLIST     HOST
IDLIST      VIDLIST     IDLIST
VIDLIST     GROUP       VIDLIST
GROUP           .       IDLIST
    .           .       GROUP
    .           .           .
    .                       .
                            .
```

See the *VTAM System Programmer's Guide* for a complete description of the macro and its operands.

Name	*Operation*	*Operands*
[symbol]	VIDLIST	VIDSEQ=((chars,term name),...)

SERVICE Macro Instruction

The SERVICE macro generates the service order table for a nonswitched start-stop or BSC line controlled by a multipoint discipline, or for a nonswitched (multipoint or point-to-point) SDLC link. (Switched lines, whether SDLC, BSC, or start-stop, never require a service order table.)

A nonswitched line is controlled by a multipoint discipline, regardless of whether one station or several stations are on the line, if the station(s) must be polled and addressed. A switched line used for communicating with IBM 1050s, although controlled by a multipoint discipline, does not require a service order table.

Each nonswitched line requires a service order table in the network control program that controls the line. A nonswitched link between a local and a remote communications controller is always controlled by the program in the local controller. Consequently, a service order table is required only in the local (not the remote) network control program.

For a nonswitched start-stop or BSC line, one SERVICE macro is required directly following the LINE macro, if POLLED=YES is specified. Each individually polled and addressed device represented by a TERMINAL or COMP macro (or by a CLUSTER macro in which the GPOLL operand is specified) must be represented by at least one entry in the service order table.

Within the network control program that controls an SDLC link (POLLED=YES is specified in the LINE macro), one SERVICE macro is required following the LINE macro that represents the link. Each SDLC cluster controller or communications controller attached to the link must be represented by an entry in the service order table. (Logical units associated with the SDLC cluster controller are not represented in the service order table.)

If more than one SDLC link connects a local and a remote 3704 or 3705 (that is, the principal link is augmented by one or more alternate ["backup"] links), a SERVICE macro must appear in the local NCP following each of the LINE macros representing the links.

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

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity. Specify a name only on the first SERVICE macro defining the service order table, if more than one macro is required.

`ORDER=(entry,...)`

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

entry is the name of a station, clustered terminal, component, or remote communications controller to be serviced by the network control program. These names appear on the TERMINAL, CLUSTER, COMP, PU, or INNODE macros associated with the line for which the service order table is being generated. A station or component name may appear in the list one or more times. (Logical units cannot appear in the table; hence, omit names of LU macros.)

The maximum number of entries is 256.

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. Omit the MAXLIST operand in each of the additional macros.

Note: The generation procedure checks the first 35 entries of the service order table to determine if the devices referred to by these entries are represented by CLUSTER, TERMINAL, COMP, PU, or INNODE macros associated with the LINE macro for which the service order table is being defined. The converse is also true: each CLUSTER, TERMINAL, COMP, PU, and INNODE macro associated with the LINE macro is checked to determine if it is represented in the service order table. A diagnostic message is issued if a discrepancy exists. Any service order table entries beyond 35 are not checked in this manner.

`[MAXLIST=n]`

(start-stop and BSC lines only)

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 access method sends a control request to add entries.

If you omit this operand, *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.)

This operand is invalid if the service order table is for an SDLC link, that is, if LNCTL=SDLC is specified in the GROUP macro for the associated link.

LUPOOL Macro Instruction (VTAM Users Only)

The LUPOOL macro instruction specifies a pool of logical units the network control program uses in communicating with logical units associated with physical units reached over switched SDLC links. (Logical units for physical units on nonswitched links are defined by LU macros associated with the PU (or CLUSTER) macro that represents the physical unit.

Upon establishing a connection with a switched physical unit, the network control program allocates, from the pool, a logical unit for each logical unit contained within that physical unit. VTAM specifies to the network control program the number of logical unit control blocks to allocate and also supplies the logical unit parameters for each logical unit. Upon completion of the transmission between the network control program and the physical unit, the program releases each logical unit to the pool.

One and only one LUPOOL macro is required if the network configuration includes SDLC physical units reached over switched communication facilities—that is, the source statements include at least one GROUP macro in which LNCTL=SDLC and DIAL=YES are specifed.

Name	*Operation*	*Operands*
`[symbol]`	`LUPOOL`	`NUMBER=count`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`NUMBER=count`

Specifies the number of logical units to be included in the pool. The minimum is 1. The maximum is the total number of resources that can be defined in the network control program (as determined by the value chosen for the MAXSUBA operand of the BUILD macro) minus the number of resources defined in the LINE, PU, LU, TERMINAL, COMP, CLUSTER, and INNODE macros.

Example: If the MAXSUBA operand specifies 31 subareas, the maximum number of resources in the network controlled by VTAM is 2045, assuming that the network includes SDLC stations and BSC and/or start-stop stations. If the total number of resources (equivalent to the total number of macros listed above) is 1000, the maximum value you may specify in the NUMBER operand is 1045 (2045-1000).

To determine the highest value you need specify in this operand, assume that (1) the network control program is communicating simultaneously with SDLC physical units over all switched SDLC links, and (2) the physical units involved are those having the highest number of logical units. The total number of logical units in all connected physical units represents the greatest possible demand that can be placed on the logical unit pool. By specifying this value you can assure that the logical unit pool is never depleted.

In practice, the value of NUMBER can be less, since the assumptions above represent the most extreme conditions which may never, or only seldom, occur. Specifying too low a value, however, may result in depletion of the pool. If, after a physical connection is established with a switched physical unit, the network control program is unable to supply sufficient logical units from the pool, the program will break the physical connection before data transmission has begun. Then a new connection will have to be made and logical unit allocation attempted again. Occasional occurrences of this kind may be tolerable, but frequent occurrences demonstrate the need to increase the value specified in this operand. Thus, the circumstances unique to a given installation should be analyzed to arrive at an appropriate value.

MTALCST Macro Instruction

If one or more switched lines in the network are to be used as multiple-terminal-access lines for start-stop terminals, a line control selection table must be generated within the network control program. (Only one table is generated regardless of the number of multiple-terminal-access lines.) The table provides information the program needs to identify the type of terminal that calls the controller over a multiple-terminal-access line.

The MTALCST macro defines one entry in the table. Each entry represents a particular combination of operating parameters, such as the transmission code, type of line control, length of print line, and carriage return rate.

Code one MTALCST macro for each distinct combination of parameters. The maximum number of MTALCST macros you may code is 63.

All MTALCST macros must be grouped together in a single sequence. See Appendix D for an example of how this and other MTA macros are used to establish multiple-terminal-access operations.

```
Name    Operation   Operands

symbol  MTALCST     GROUP=entry,

                    SPEED=rate

                    [,ACR={YES}]
                          {NO }

                    [,CLOCKNG={INT}]
                              {EXT}

                           {BCD }
                           {EBCD}
                           {COR }
                    [,CODE={DIC1}]
                           {BCD2}
                           {COR2}
                           {DIC3}

                    [,CRRATE=integer]

                    [,DATRATE={HIGH}]
                              {LOW }

                    [,INTPRI={0}]
                             {1}

                             {1050 }
                             {2740A}
                    [,LCTYPE={2740D}]
                             {2740E}
                             {2740F}
                             {2741 }
                             {TWX  }
```

```
[,LINESIZ=integer]

             {m    }
[,RETRIES={2    }]
             {NONE}

[,TRANSFR=count]
```

symbol

Provides a name for the line control selection table entry defined by this macro and is required. *symbol* may be any valid assembler-language symbol; the first character may not be $.

symbol is referred to by the LCST operand of the MTATABL macro.

GROUP=entry

Specifies the name of the GROUP macro for any line group whose terminal characteristics are the same as the characteristics of the terminal that will call the controller over the multiple-terminal-access line. *entry* must not be the name of a GROUP macro that represents a group of multiple-terminal-access lines.

This operand is required.

SPEED=rate

Specifies the data rate (in bits per second) for the multiple-terminal-access lines associated with this entry.

If you specify CLOCKNG=INT (or omit the CLOCKNG operand), this rate must be one of the four oscillator rates specified for the communication scanner to which the lines are attached (SPEED operand of the CSB macro). Specify the line speed in bits per second, omitting a fractional part, if any. For example, specify a line speed of 110 bps as SPEED=110; specify a line speed of 134.5 bps as SPEED=134 (omitting the decimal point and fraction).

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

This operand is required.

[ACR={YES}]
{NO }

Specifies whether or not the 1050 terminals that may call the controller on this line are equipped with the Accelerated Carrier Return feature. Specify ACR=YES only if all 1050 terminals that may call the controller over this line are equipped with the feature.

[CLOCKNG={INT}]
{EXT}

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

If the scanner provides clocking, code CLOCKNG=INT. If the modem (whether external to or contained within the controller) provides clocking, code CLOCKNG=EXT.

```
        {BCD }
        {EBCD}
        {COR }
[CODE={DIC1}]
        {BCD2}
        {COR2}
        {DIC3}
```

Specifies the transmission code with which the network control program is to communicate with the type of terminal represented by this macro. Associated with each transmission code is a translation table in the network control program. The contents of the translation tables are in the *Program Reference Handbook* in the section entitled "Line Character Codes."

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

```
              CODE=

IBM 1050:     EBCD     (Extended BCD code)
              BCD      (BCD code 1)
              BCD2     (BCD code 2)

IBM 2740:     EBCD
              BCD
              COR      (Correspondence code 1)

IBM 2741:     COR
              COR2     (Correspondence Code 2)
              BCD
              BCD2
              EBCD

TWX:          DIC1     (Data interchange code 1)
              DIC3     (Data interchange code 3)
```

```
[CRRATE=integer]
```

Specifies the number of print positions that the carriage printers connected to the line will return for each idle character the network control program sends. The minimum is 1; the maximum is 255.

If you omit this operand, a rate of 40 print positions per idle character is assumed for TWX terminals; 13 is assumed for IBM 1050 terminals equipped with the Accelerated Carrier Return feature. Ten positions per idle character is assumed for IBM 2740 and 2741 terminals and for 1050 terminals without the ACR feature.

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

Specifies which data rate is to be used on the modem (data set) that attaches the multiple-terminal-access line to the controller. This operand is valid only if the modem has a dual data rate. (Determine this from the system designer.)

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

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

If the modem has only one data rate, specify DATRATE=LOW or omit the operand.

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

Specifies the interrupt priority for the multiple-terminal-access MTA lines used for communicating with the terminals represented by this entry.

This operand is valid only if the scanner to which the lines are attached is a type 1 scanner (TYPE=1 in the CSB macro representing the scanner). For a type 2 scanner, the interrupt priority is specified in the LINE macro for the MTA line.

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

Specifies the type of terminal and 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
TWX	Western Union TWX

```
[LINESIZ=integer]
```

Specifies the length of the print line, in number of print positions, for printer-type devices connected to the line represented by this MTALCST macro. The minimum value for *integer* is 1; the maximum is 255.

If you omit this operand, a line length of 72 print positions is assumed for TWX terminals; 130 positions are assumed for 1050, 2740, and 2741 terminals.

```
          {m   }
[RETRIES={2   }]
          {NONE}
```

Specifies the number of attempts, via retransmission, to recover from text errors in message data sent to or received from an IBM 1050 or an IBM 2740 with record checking. (Other types of multiple-terminal-access terminals are not capable of retransmission.) *m* is the number of attempts, from 1 to 255. (Any value less than 255 specifies the exact number of attempts; 255 specifies unlimited attempts.)

If you code RETRIES=NONE, no error recovery is attempted for read text or write text errors.

If you omit the operand, a maximum of two recovery attempts will be made for terminals capable of retransmission. If the terminal is not capable of retransmission, *m* is assumed to be 0, that is, no recovery is attempted.

```
[TRANSFR=count]
```

Specifies a limit on the number of buffers that the network control program will obtain to receive message text from a terminal before transferring 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 or end-of-transmission character (or until the sub-block limit specified in the CUTOFF operand of the LINE macro is reached).

The minimum value of *count* is 1. The maximum is the smaller of (1) 255 or (2) the result of multiplying the values of the MAXBFRU and UNITSZ operands of the HOST macro, subtracting the value of the BFRPAD operand of the HOST macro, and dividing the result by the buffer size specified in the BFRS operand of the BUILD macro. Expressed as a formula:

$$\text{count} \leq \frac{(\text{MAXBFRU})(\text{UNITSZ})-\text{BFRPAD}}{\text{BFRS}}$$

If you omit this operand, the network control program uses the smaller of (1) 255 or (2) the result of the calculation expressed by the preceding formula.

MTALIST Macro Instruction

The MTALIST macro defines a list the network control program uses in determining the type of terminal that calls the controller over a call-in multiple-terminal-access line. MTALIST macros must be coded immediately after the MTALCST macros.

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

See Appendix D for an example of how this and other MTA macros are used to establish multiple-terminal-access operations.

Name	*Operation*	*Operands*
`symbol`	`MTALIST`	`LCTYPE=( type,... )`

`symbol`

Provides a name for the multiple terminal access list and is required; it is referred to by the MTALIST operand of the LINE macro. *symbol* may be any valid assembler-language symbol; the first character may not be $.

`LCTYPE=( type,... )`

Specifies the types of terminals and associated line controls to be associated with the line whose LINE macro refers to this MTALIST macro.

Any of the following types may be specified, singly or in combination:

LCTYPE=	*Type of Terminal*
1050	IBM 1050
2740A	IBM 2740 (basic)
2740D	IBM 2740 with Transmit Control feature
2740E	IBM 2740 with Transmit Control and Record Checking features
2740F	IBM 2740 with Record Checking feature
2741	IBM 2741
TWX	Western Union TWX

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 call-in 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 call-in multiple-terminal-access line need be specified.

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

Name	*Operation*	*Operands*
`[symbol]`	`MTAPOLL`	`POLL=(chars,...)`

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`POLL=(chars,...)`

Specifies the list of polling characters. Code the hexadecimal representation of the EBCDIC equivalent of the polling characters.

Example: Assume that the network includes four IBM 1050 terminals, each having a keyboard and tape reader as input devices and that the polling characters for the eight devices are A5, A6, B5, B6, C5, C6, D5, D6. You would therefore code POLL=(C1F5,C1F6,C2F5,C2F6,C3F5,C3F6, C4F5,C4F6). (Each of the eight hexadecimal sequences represents one of the EBCDIC polling sequences.)

MTATABL Macro Instruction

For each unique combination of line control discipline and transmission code specified by an MTALCST macro, an MTATABL macro is required. In this macro you specify the line control and code used, and the name of each MTALCST macro that defines a set of operating parameters for that particular line control and code.

For example, if you have coded a single MTALCST macro to represent a 2741 using BCD code, thus:

```
MTA1     MTALCST LCTYPE=2741,CODE=BCD,...
```

you would code one MTATABL macro that specifies the same line control and code:

```
         MTATABL     LCST=(MTA1),LCTYPE=2741,CODE=BCD
```

If, on the other hand, you have coded two MTALCST macros, both representing a 2741 using BCD code but each defining a different set of operating parameters, thus:

```
MTA1     MTALCST LCTYPE=2741,CODE=BCD,LINESIZ=72,...
MTA2     MTALCST LCTYPE=2741,CODE=BCD,LINESIZ=90,...
```

you would code an MTATABL macro that specifies the same line control and transmission code, and that names both MTALCST macros:

```
         MTATABL     LCST=(MTA1,MTA2),LCTYPE=2741,
                     CODE=BCD
```

Code only one MTATABL macro for a given combination of line control type and transmission code. In one MTATABL macro you may specify the names of up to ten MTALCST macros.

See Appendix D for an example of how this and other MTA macros are used to establish multiple-terminal access operations.

Name	*Operation*	*Operands*
[symbol]	MTATABL	LCST=(mtalcst name,...)

```
                                    {BCD }
                           [,CODE={EBCD}]
                                    {COR }
                                    {DIC1}

                                      {1050 }
                                      {2740A}
                                      {2740D}
                           [,LCTYPE={2740E}]
                                      {2740F}
                                      {2741 }
                                      {TWX  }
```

```
[symbol]
```

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check it for validity.

```
LCST=(mtalcst name,...)
```

Specifies the name of one or more MTALCST macros representing the combination of line control and transmission code specified by this MTATABL macro.

You may specify from one to ten MTALCST macro names.

When there are two translation tables for one transmission code (as for the binary coded decimal, Correspondence, and data interchange codes), you might specify one translation table for some terminals of a given line control type (LCTYPE operand) and code (CODE operand) and the other translation table for other terminals of the same type and code. In such cases, code the *mtalcst names* for both variants in the LCST operand of the same MTATABL macro.

Assume, for example, that your network includes (1) a 2741 using Correspondence code 1 (specified as LCTYPE=2741 and CODE=COR in one MTALCST macro); and (2) a 2741 using Correspondence code 2 (specified as LCTYPE=2741 and CODE=COR2 in another MTALCST macro). In the LCST operand of a MTATABL macro you would name both MTALCST macros.

```
       {BCD }
[CODE={EBCD}]
       {COR }
       {DIC1}
```

Specifies the transmission code used by the type of terminal this MTATABL macro represents. Specify the same type of code as you specified in the MTALCST macro named in the LCST operand of this macro. (For example, specify CODE=BCD in this macro if you specify CODE=BCD or CODE=BCD2 in the MTALCST macro.) (CODE=COR is invalid for a 1050 terminal.)

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

Specifies the type of line control used by the terminal this MTATABL macro represents. Specify the same line control type as you have specified in the MTALCST macros named in the LCST operand of this MTATABL macro.

```
[symbol]
```

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check it for validity.

```
LCST=(mtalcst name,...)
```

Specifies the name of one or more MTALCST macros representing the combination of line control and transmission code specified by this MTATABL macro.

You may specify from one to ten MTALCST macro names.

When there are two translation tables for one transmission code (as for the binary coded decimal, Correspondence, and data interchange codes), you might specify one translation table for some terminals of a given line control type (LCTYPE operand) and code (CODE operand) and the other translation table for other terminals of the same type and code. In such cases, code the *mtalcst names* for both variants in the LCST operand of the same MTATABL macro.

Assume, for example, that your network includes (1) a 2741 using Correspondence code 1 (specified as LCTYPE=2741 and CODE=COR in one MTALCST macro); and (2) a 2741 using Correspondence code 2 (specified as LCTYPE=2741 and CODE=COR2 in another MTALCST macro). In the LCST operand of a MTATABL macro you would name both MTALCST macros.

```
       {BCD }
[CODE={EBCD}]
       {COR }
       {DIC1}
```

Specifies the transmission code used by the type of terminal this MTATABL macro represents. Specify the same type of code as you specified in the MTALCST macro named in the LCST operand of this macro. (For example, specify CODE=BCD in this macro if you specify CODE=BCD or CODE=BCD2 in the MTALCST macro.) (CODE=COR is invalid for a 1050 terminal.)

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

Specifies the type of line control used by the terminal this MTATABL macro represents. Specify the same line control type as you have specified in the MTALCST macros named in the LCST operand of this MTATABL macro.

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 and all must operate in network control mode. This allows the network control program to use any of the lines to call a station of a specific type. See Appendix D for an example of how this macro is used to establish operations over lines to be used for switched and multiple-terminal-access facilities.

```
Name        Operation   Operand

[symbol]    DIALSET     LINES=(line name,...)

                        [,DIALALT=dialset name]

                        [,QLIMIT={count}]
                                 {1    }

                        [,QLOAD={count}]
                                {0    }

                        [,RESERVE={count}]
                                  {0    }
```

`[symbol]`

Provides a name for the dial set and is required except as indicated in the description of the LINES operand. *symbol* may be any valid assembler-language symbol; the first character may not be $. 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.

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. No other operands may be specified in the additional macros.

`[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.

The DIALSET macro specified by the DIALALT operand 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, if the program is unable to fulfill the requests because all lines in the dial set (or alternate dial set, if any), are occupied. 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 zero; the maximum is 255. A value of 0 specifies that the network control program will reject *any* request for which the program does not immediately have a line available.

```
[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.

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 hierarchical level within the arrangement. Each element is represented by a specific macro instruction within the category of macros called teleprocessing network configuration macro instructions.

The highest element is the *communication line.* Each line in the network is represented by a LINE macro. This is true whether start-stop, BSC, or SDLC stations are connected to the line, and whether the line is nonswitched multipoint, nonswitched point-to-point, or switched point-to-point.

If the line is to operate only in emulation mode, no other macros are required to represent the stations connected to the line. Operation in network control mode, however, requires that the LINE macro be followed by a macro for each of the remaining elements in the hierarchy.

Start-Stop Terminals and Non-Clustered BSC Stations

Each start-stop terminal and each BSC station (exclusive of cluster-type stations) is generally represented by a TERMINAL macro. Each pollable or addressable component of a terminal may be represented by a COMP macro.

Consider a network consisting of three lines, six terminals, and seventeen components, arranged as shown in Figure 5-7.

If all the lines are to operate only in emulation mode, only LINE macros are required to represent the network:

Macro:	*Represents:*
LINE	Line 1
LINE	Line 2
LINE	Line 3

On the other hand, if all lines are to operate in network control mode, whether they may sometimes operate instead in emulation mode, TERMINAL macros are required as well:

```
LINE          Line 1
  TERMINAL      Terminal A, all components
  TERMINAL      Terminal B, components B1, B2
  TERMINAL      Terminal C, components C1, C2
LINE          Line 2
  TERMINAL      Terminal D, components D1, D2
LINE          Line 3
  TERMINAL      Terminal E, all components
  TERMINAL      Terminal F, components F1, F2
```

The foregoing macro sequence sufficiently represents the network if only one polling sequence and one addressing sequence are required for each station. This is the case if the station is a computer, or a transmission control unit (such as an IBM 2701), or a terminal having only one input component and one output component (typically a keyboard and a printer [for instance, IBM 2741] or a keyboard and a display screen [for instance, IBM 2260]). One TERMINAL macro also suffices to represent a terminal having multiple input and/or output components, if only a general polling and a general addressing sequence are required.

If, however, more than one polling sequence or addressing sequence, or both, are required, each additional pair of polling and addressing sequences must be specified in a COMP macro. The conditions under which COMP macros may be required are explained further in the description of the COMP macro in this chapter.

Assume now that individual polling and addressing sequences are needed for each terminal in the configuration of Figure 5-7. Three COMP macros must be added to the preceding macro sequence, resulting in this sequence:

```
LINE            Line 1
  TERMINAL        Terminal A, components A1, A2
    COMP            Components A3, A4
  TERMINAL        Terminal B, components B1, B2
  TERMINAL        Terminal C, components C1, C2
LINE            Line 2
  TERMINAL        Terminal D, components D1, D2
LINE            Line 3
  TERMINAL        Terminal E, components E1, E2
    COMP            Components E3, E4
    COMP            Component E5
  TERMINAL        Terminal F, components F1, F2
```

Each COMP macro can represent one input component and one output component. Therefore, assuming that components A3 and E3 are input (polled) components and A4 and E4 are output (addressed) components, one COMP macro for each TERMINAL macro is required to accommodate these added components. The third additional component of terminal E (component E5) requires another COMP macro.

COMP macros should be coded only if needed to specify polling and addressing sequences beyond the first polling and addressing sequences specified in the TERMINAL macro. Avoiding COMP macros where possible conserves storage space within the communications controllers.

Clustered BSC Stations

The preceding arrangement differs for clustered BSC stations such as the IBM 3270 and 2972. For such stations the macro sequence is LINE, CLUSTER, and TERMINAL, rather than LINE, TERMINAL, and COMP.

Consider, for example, a line connected to two terminal control units (for example, 2972), to each of which are attached three terminals (for example 2980).

You would represent this configuration by the sequence:

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

The lowest level in the hierarchy is represented by the TERMINAL macros, the next level (cluster controller) by CLUSTER macros, and the highest (communication lines) by the LINE macro.

SDLC Stations

The macro hierarchy for SDLC stations other than communications controllers (for example, IBM 3600, 3650, 3660, 3767, 3771, 3773, 3774, 3775, 3776) is LINE, PU, LU, for nonswitched links and LINE, PU for switched links. A single PU macro represents the physical unit within the SDLC station.

If the network control program communicates with the physical unit over a *nonswitched* SDLC link, one or more LU macros following the PU macro represent the logical units associated with the physical unit. (A logical unit is an application program within the physical unit together with the terminals associated with that program.)

If, for example, one physical unit having four logical units were attached to a nonswitched SDLC link, the following sequence would be required:

```
LINE
  PU
    LU
    LU
    LU
    LU
```

If the network control program communicates with the physical unit over *switched* facilities, the logical units are not represented by LU macros, and the macro sequence is simply LINE, PU. The network control program control blocks representing logical units are allocated dynamically from the pool of such control blocks defined by the LUPOOL macro.

Note: In version 3 of the network control program, SDLC station physical units are represented by CLUSTER macros. All SDLC stations added to an existing (version 3) NCP source program that is re-generated under version 4 or 5 should be represented by PU macros. For compatibility with the previous version (3), version 4 and 5 continue to support CLUSTER macros; however, any subsequent changes or additions to physical unit parameters will be accommodated only in the PU macro.

Communications Controller Attached to SDLC Link

Each 3704 or 3705 communications controller connected by an SDLC link to another such controller is represented by a PU macro following the LINE macro that represents the link. Thus, when defining a network control program for a local communications controller, you would code a PU macro for the remote communications controller, and when defining a network control program for the remote communications controller you would code a PU macro to represent the local communications controller:

In NCP for the local controller:

```
LINE
    PU (represents the remote controller)
```

In NCP for the remote controller:

```
LINE
    PU (represents the local controller)
```

Note: In version 3 of the network control program, communications controller physical units are represented by INNODE macros. All communications controllers added to an existing (version 3) source program that is to be re-generated under version 4 or 5 should be represented by PU macros. For compatibility with the previous version, version 4 and 5 continue to accept INNODE macros; however, any subsequent changes or additions to physical unit parameters will be reflected only in the PU macro.

GROUP Macro

To the hierarchies of LINE, PU (or CLUSTER or INNODE), LU, TERMINAL, and COMP macros must be added one or more GROUP macros. The GROUP macro, unlike the others, represents not a physical element of the network but a grouping of lines having certain characteristics in common. The grouping is referred to as a *physical* line group because it contains lines having certain "physical" attributes in common, such as the type of stations attached to them.

Each LINE macro in the foregoing macro sequences must be associated with a GROUP macro that precedes it. Assume for example that in the configuration shown in Figure 5-7, the first two lines have similar attributes allowing them to be in the same line group, but that the third line has different characteristics which require it to be in a different line group. (A single line can constitute a line group.) Further, although not shown above, a SERVICE macro must directly follow each LINE macro representing a line on which polling and addressing are required. Since this is true of all three lines in the example, each requires a SERVICE macro.

The complete macro sequence would then be:

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

If all of the lines were dissimilar, each of the three LINE macros would have to be preceded by a GROUP macro. If all were alike, all could be in the same line group, with only one GROUP macro immediately preceding the first LINE macro.

For the clustered BSC stations, adding the GROUP and SERVICE macros gives the sequence:

```
GROUP
    LINE
    SERVICE
        CLUSTER
            TERMINAL
            TERMINAL
            TERMINAL
        CLUSTER
            TERMINAL
            TERMINAL
            TERMINAL
```

Adding the GROUP and SERVICE macros to the LINE, PU, LU sequence shown above under *SDLC Stations* results in:

```
GROUP
    LINE
    SERVICE
        PU
            LU
            LU
            LU
            LU
```

Similarly, the SDLC link between the communications controllers would be represented as follows:

In NCP for the local controller:

```
GROUP
    LINE
    SERVICE
        PU
```

In NCP for the remote controller:

```
GROUP
    LINE
        PU
```

Notice that the SERVICE macro appears only in the local NCP, because the local communications controller in the *primary* station on the SDLC link. In the remote NCP, executed in the *secondary* station, the SERVICE macro must be omitted.

See the description of the GROUP macro later in this chapter for the attributes that communication lines must have in common in order to appear within the same line group.

Composite Example of Teleprocessing Network Configuration Macro Instructions

Consider the teleprocessing network configuration shown in Figure 5-6. The network attached to the local communications controller comprises the following communication lines:

Line 1: Nonswitched multipoint, five IBM 1050 terminals—each having keyboard/printer and one 1056 card reader (start-stop terminals)

Lines 2 and 3: Nonswitched point-to-point lines, each communicating with an IBM 2701 (nonclustered BSC station)

Line 4: Switched point-to-point line over which the local communications controller can call two IBM 1130s and one IBM 2780 (nonclustered BSC stations)

Line 5: Nonswitched multipoint SDLC link, two IBM 3600 systems, with two logical units each (SDLC stations)

Line 6: Nonswitched point-to-point SDLC link, one 3704 communications controller (remote communications controller)

The network attached to the remote communications controller comprises:

Line R1: Nonswitched multipoint line, two IBM 3270 systems, each comprising a 3271 controller, four 3277 displays, and a 3284 printer.

Line R2: Switched point-to-point line over which any number of IBM 2741 terminals can call the remote controller.

Line R3: Nonswitched multipoint SDLC link, three IBM 3790 systems, with three logical units each.

The teleprocessing network configuration macros required to represent the network attached to the *local* communications controller are as follows (not all required operands are shown):

```
G1      GROUP  LNCTL=SS
L1         LINE   POLLED=YES,TERM=1050
S1         SERVICE ORDER=(T1,C1,T2,C2,T3,C3,T4,C4,T5,C5)
T1            TERMINAL
C1               COMP
T2            TERMINAL
C2               COMP
T3            TERMINAL
C3               COMP
T4            TERMINAL
C4               COMP
T5            TERMINAL
C5               COMP

G2      GROUP  LNCTL=BSC
L2         LINE   TERM=2701
T6            TERMINAL
L3         LINE   TERM=2701
T7            TERMINAL

G3      GROUP3 LNCTL=BSC,DIAL=YES
L4         LINE
T8            TERMINAL TERM=1130
T9            TERMINAL TERM=1130
T10           TERMINAL TERM=2780

G4      GROUP  LNCTL=SDLC
L5         LINE
           SERVICE ORDER=(PU1,PU2)
PU1           PU PUTYPE=2
LU1              LU
LU2              LU
LU3              LU
LU4              LU
PU2           PU PUTYPE=2
LU5              LU
LU6              LU
LU7              LU
LU8              LU

G5      GROUP  LNCTL=SDLC
L6         LINE
S2         SERVICE ORDER=(PU3)
PU3           PU PUTYPE=4 (represents remote controller)
```

The network control program defined for the *remote* communications controller would include:

```
RG1     GROUP  LNCTL=BSC
RL1         LINE   POLLED=YES,CUTYPE=3271
RS1         SERVICE ORDER=(RCL1,RT1,RT2,RT3,RT4,RT5,RCL2,RT6,     X
                   RT7,RT8,RT9,RT10)
RCL1            CLUSTER TERM=3277
RT1                TERMINAL
RT2                TERMINAL
RT3                TERMINAL
RT4                TERMINAL
RT5                TERMINAL  TERM=3284
RCL2            CLUSTER TERM=3277
RT6                TERMINAL
RT7                TERMINAL
RT8                TERMINAL
RT9                TERMINAL
RT10               TERMINAL  TERM=3284

RG2     GROUP  LNCTL=SS
RL2         LINE
RT11            TERMINAL TERM=2741,CTERM=YES

RG3     GROUP  LNCTL=SDLC
RL3         LINE   POLLED=YES
RS3         SERVICE ORDER=(RPU1,RPU2,RPU3)
RPU1            PU PUTYPE=2
RLU1A              LU
RLU1B              LU
RLU1C              LU
RPU2            PU PUTYPE=2
RLU2A              LU
RLU2B              LU
RLU2C              LU
RPU3            PU PUTYPE=2
RLU3A              LU
RLU3B              LU
RLU3C              LU

RG0     GROUP  LNCTL=SDLC
RL0         LINE
RN0             PU PUTYPE=4  (represents local controller)
```

These network examples include each of the types of hierarchical arrangements described earlier, as well as the three types of line connections available: nonswitched point-to-point, nonswitched multipoint, and switched point-to-point.

Shown in the above macros are operands that specify the type of line connection, type of line control, and type of cluster or terminal, as appropriate.

See Appendix I for other sample programs that illustrate macro coding sequences.

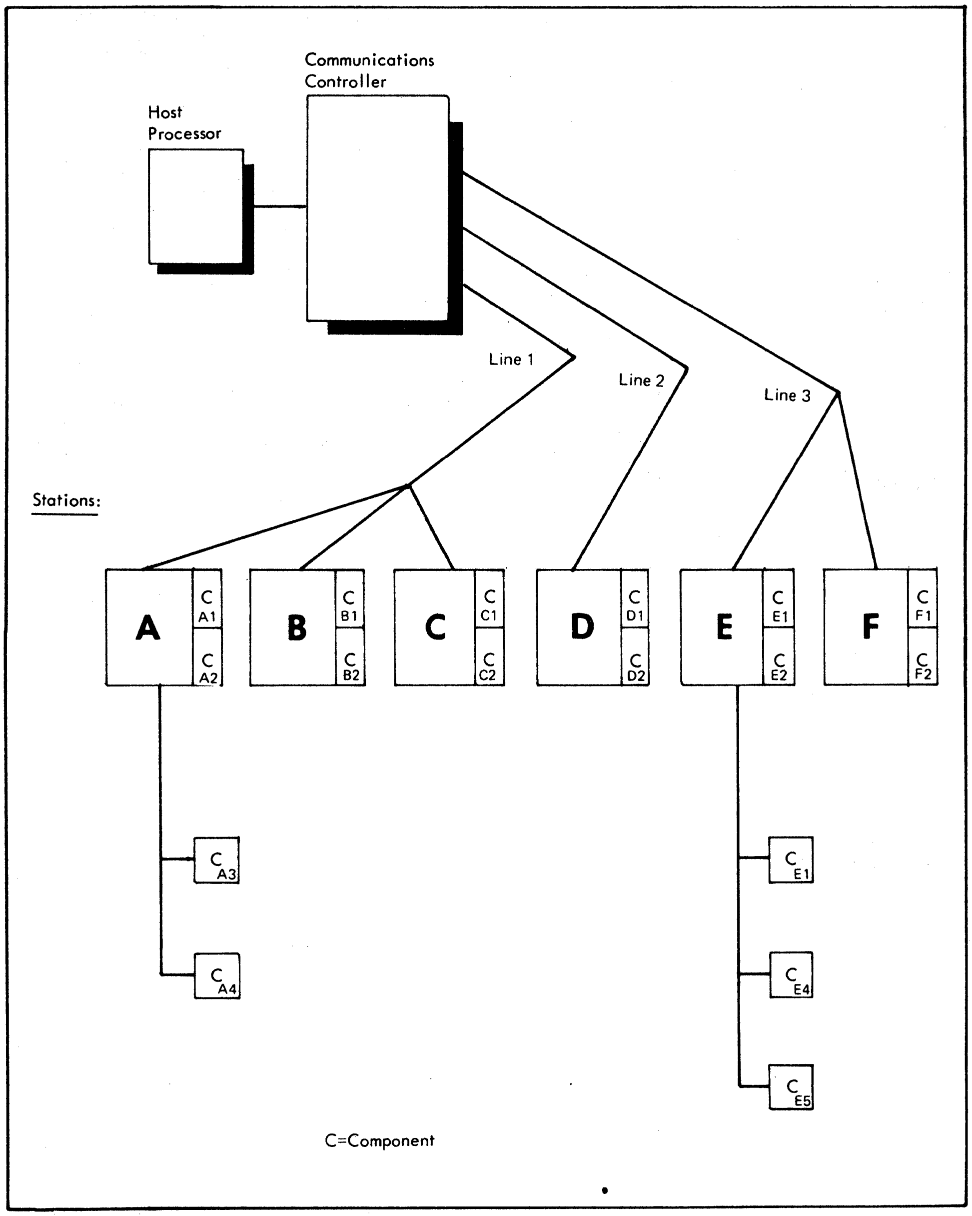


Figure 5-7. Example of Network Configuration

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.

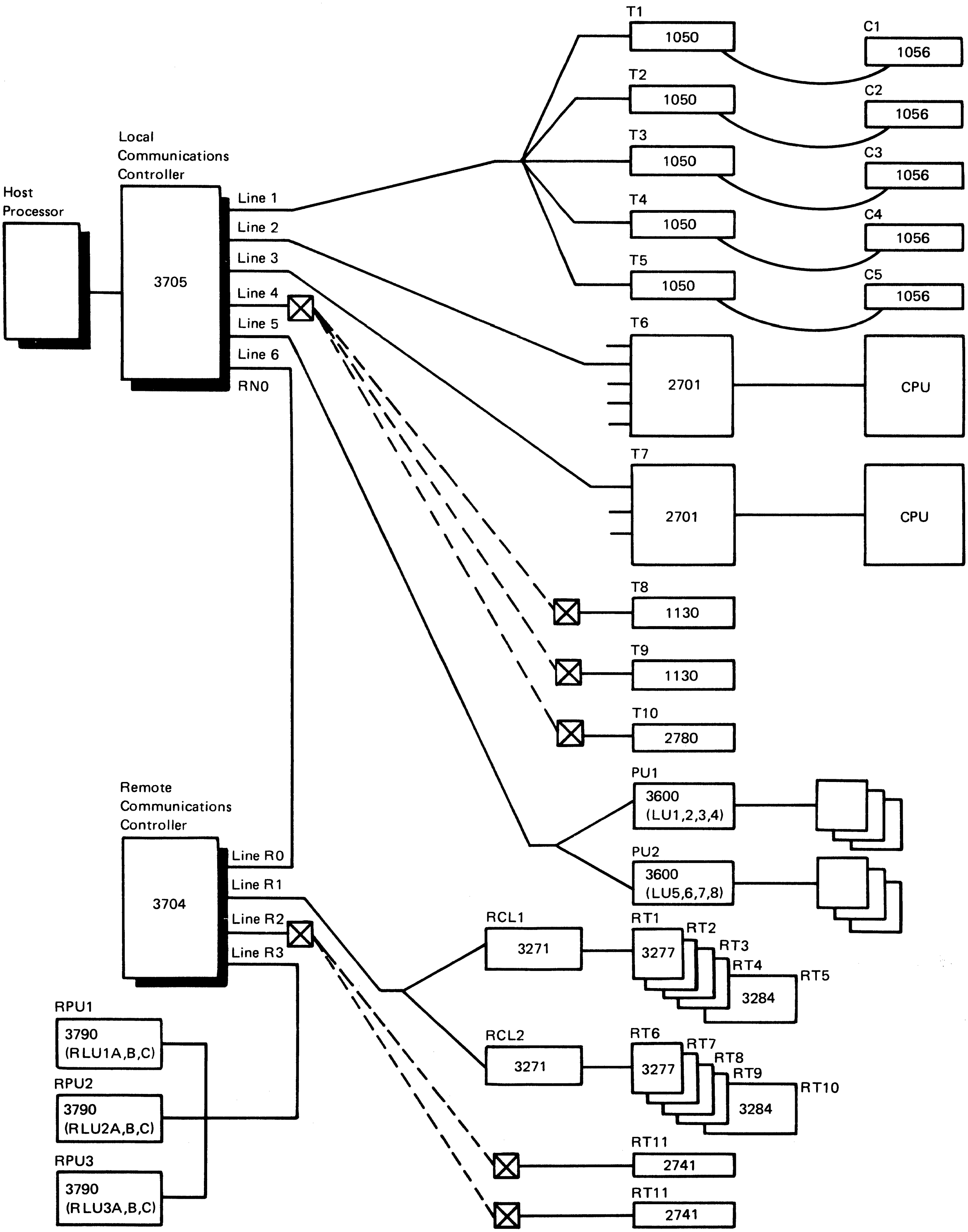


Figure 5-8. Composite Example of Network Configuration

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.

Note: If you code only one suboperand (parameter) in an operand that can have two or more suboperands, that operand completely overrides an equivalent operand specified in a higher-level macro. The default values are assumed for the suboperands omitted at the lower level.

Summary of Macro Instructions and Operands

Figure 5-9 lists all of the operands of the teleprocessing network configuration macros for lines to be operated in network control mode and for those to be operated in emulation mode. The asterisk indicates the macro under which the description of the operand appears. The bullet (•) indicates other macros in which the operand may be coded.

Note: This summary indicates only (1) the name of the macro under which the description of each operand can be found (indicated by *) and (2) other macros (indicated by •) in which the operand can be coded. This summary does *not* indicate the conditions (for example, type of line control, type of station) under which use of the operand is appropriate. For this information, see the individual macro descriptions later in this chapter.

VTAM-Only Operands

Some operands convey no information to the network control program generation assembly process but must nonetheless appear in the NCP generation input deck that serves as input to the VTAM initialization procedure. (Refer to the *VTAM System Programmer's Guide* for information on this procedure.) Such operands are called *VTAM-only* operands. The macro assembly step of the NCP generation procedure permits each VTAM-only operand to appear in the macros indicated by a V in the table of Figure 5-9. The assembly process does not check such operands for proper syntax, however, nor does it verify that any related operands are present or absent. For the meanings of VTAM-only operands, see the *VTAM System Programmer's Guide.*

	Macro Instruction									
	For Network Control Mode								*For Emulation Mode*	
	GROUP	LINE	INNODE	CLUSTER	PU	LU	TERMINAL	COMP	GROUP	LINE
ACTIVTO	*									
ADDR			*	*	*		*	●		
ADDRESS		*								*
ANSTONE	●	*								
ANSWER	V	V								
ATTN	●	●					*			
AUTO		*								*
BATCH	●	●				*				
BFRDLAY	●	●		●			*			
BHEXEC	●	●		*			*	●		
BHSET	●	●		*			*	●		
BNNSUP	●	●			*					
BUFLIM	V	V			V	V	V	V		
BUFSIZE									●	*
CALL	●	*								
CDATA	●	●		*			*	●		
CHAREC	*								*	
CHECK									●	*
CHNPRI									●	*
CLOCKNG	●	*							●	*
CODE	●	*							●	*
CONV	●	●					*	●		
CRDLAY	●	●					*	●		
CRETRY	*									
CRITSIT	●	●		●			*			
CRRATE	●	*								
CTERM							*			
CU									●	*
CUIDLEN	●	●					*			
CUTOFF	●	*								
CUTYPE	●	●		*					●	*
DATASW						*				
DATMODE	●	●	*		*					
DATRATE	●	*							●	*
DELAY									*	
DEVICE							V	V		
DIAL	*								*	
DIALALT	●	*								
DIALNO							*			
DIALSET	●	*					*			
DIRECTN	●	●					*			
DISABLE									●	*
DISCNT	V	V			V					
DUALCOM										*
DUPLEX	●	*							●	*
ENDTRNS	●	●					*	●		
EOB									*	
EOT									*	
EXEC	●	●		●			*	●		

V indicates the macros in which the VTAM-only operand at the left may be coded. See the *VTAM System Programmer's Guide* for a description of this operand.

Figure 5-9. Summary of Operands for Configuration Macro Instructions (Part 1 of 3)

Macro Instruction

	For Network Control Mode								*For Emulation Mode*	
	GROUP	LINE	INNODE	CLUSTER	PU	LU	TERMINAL	COMP	GROUP	LINE
FANOUT							*			
FEATURE[1]	●	●		*			*		●	*
FEATUR2	V	V		V			V			
FGSLTRS	●	*								
GPOLL				*						
IDSEQ	●	●					*			
INHIBIT	●	●		●			*	●		
INTPRI	●	*							●	*
IRETRY	●	●	*	*	*					
ISTATUS	V	V		V	V	V	V	V		
ITBMODE	●	●		●			*	●		
KBDLOCK	*									
LCST	●	●					*			
LGRAPHS	●	●		*			*	●		
LINESIZ	●	*								
LNCTL	*								*	
LNQTCNT	●	*								
LOCADDR						*				
LOGAPPL	V	V		V	V	V	V	V		
LOGTAB	V	V		V	V	V	V	V		
MAXDATA	●	●		*	*					
MAXLU	●	●			*					
MAXOUT	●	●	*	*	*					
MODEM									●	*
MODETAB	V	V			V	V				
MONITOR	●	*								
MPTALT	●	*								
MTALIST	●	*								
MULTI									●	*
NEGPOLP	●	*								
NEWSYNC	●	*							●	*
NRZI	●	*								
PACING	●	●		●	●	*				
PAD									●	*
PADCNT	*									
PARCHK	●	*								
PASSLIM	●	●	*	*	*					
PAUSE	●	*								
PECHAR	*									
POLIMIT	●	*								
POLL							*	●		
POLLED	●	*								
POLLTO	●	*								
PT3EXEC	●	●		*			*	●		
PUTYPE	●	●			*					
QUIET									●	*
QUIETCT									*	

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

V indicates the macros in which the VTAM-only operand at the left may be coded. See the *VTAM System Programmer's Guide* for a description of this operand.

Figure 5-9. Summary of Operands for Configuration Macro Instructions (Part 2 of 3)

	Macro Instruction									
	For Network Control Mode								*For Emulation Mode*	
	GROUP	LINE	INNODE	CLUSTER	PU	LU	TERMINAL	COMP	GROUP	LINE
REDIAL	●	*								
REPLYTO	*								*	
RETRIES	●	*	*	*	*					
RING	●	*							●	*
SCLSET	●	*								
SERVLIM	●	*								
SERVPRI	●	*								
SESSION	●	*								
SPDSEL	●	*								
SPEED	●	*							●	*
SPSHIFT	●	*								
SSCPFM	V	V			V	V				
SUBAREA			*		*					
SYNDLAY	*									
TADDR	●	*							●	*
TERM	●	*		●			*		●	*
TEXTTO	*								*	
TRANSFR	●	*								
TTDCNT	*									
TYPE	*	*								
UNITXC									●	*
USE	●	*								
USSTAB	V	V			V	V				
UTERM							V			
VIDSEQ	V	V								
VPACING	V	V		V	V					
WACKCNT	*									
WAKDLAY	*									
WTTYEOB	*									
WTTYEOT	*									
XMITLIM	●	●		*			*	●		
YIELD	●	*								

V indicates the macros in which the VTAM-only operand at the left may be coded. See the *VTAM System Programmer's Guide* for a description of this operand.

Figure 5-9. Summary of Operands for Configuration Macro Instructions (Part 3 of 3)

Teleprocessing Network Configuration Macro Instructions

This section contains the teleprocessing network configuration macro instructions (GROUP, LINE, CLUSTER, PU, LU, TERMINAL, COMP, and INNODE) to be used in specifying communication lines that are to operate in network control mode, or in either network control mode or emulation mode. (The VTAM macro instruction, VTERM, also appears in this section.) Refer to Chapter 7 for a line group or a line that is to operate only in emulation mode.

GROUP Macro Instruction

A communication line group consists of lines that have the following characteristics in common:

- All lines in the group are nonswitched point-to-point, or nonswitched multipoint, or switched point-to-point.
- All lines in the group are polled, or all are nonpolled.
- All stations connected to lines in the group are start-stop stations, or all are binary synchronous stations, or all are SDLC stations.
- If the stations are binary synchronous, they may be of different types, in any combination—for example, IBM 2770, IBM 2780, and IBM 1130—but all must use the same transmission code. All BSC stations use a uniform line control scheme.
- If the stations are SDLC, they may be of different types, in any combination. All SDLC stations use a uniform transmission code and line control scheme.
- If the stations are start-stop, all must be of the same type—for example, they may be IBM 1050 or IBM 2741, but not both—and all must use the same transmission code. If they are IBM 2740 terminals, they must have certain features in common. For example, a line group cannot include both 2740s with the record checking feature and 2740s without this feature. An exception is a line group consisting of multiple-terminal-access lines. These lines can accommodate IBM 1050, 2740, and 2741 terminals, using the same or different transmission codes, and Western Union TWX terminals, in any combination. (See *Defining a Stand-Alone Line Group for MTA*, below.)

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

To define a BSC or start-stop line group that is to operate only in emulation mode, see the description of the GROUP macro in Chapter 7.

For each line group, one GROUP macro is required.

All GROUP macros for SDLC line groups must appear in the source program following any GROUP macros for BSC and/or start-stop line groups.

The GROUP macro indicates the beginning of a sequence of LINE, [SERVICE,] TERMINAL, and COMP (or LINE, [SERVICE,] CLUSTER, and TERMINAL macros; or LINE, [SERVICE,] PU, and LU macros; or LINE,[SERVICE,] and PU macros) for lines and devices within the group, and specifies:

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

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Defining a Stand-Alone Line Group for MTA

If, for multiple terminal access support, it is necessary to define a stand-alone line group (a GROUP macro followed by no LINE and TERMINAL macros), specify the following operands in the GROUP macro:

DIAL=YES
LNCTL=SS
TERM=1050 (or 2740-1, or 2741, or TWX)
POLLED=YES (only if you specify TERM=1050)
FEATURE=(XCTL,CHECK) (only if you specify TERM=2740-1)

All other LINE and TERMINAL operands are invalid for a stand-alone line group and if coded are ignored.

See Appendix D for an example of the use of stand-alone line groups in establishing multiple-terminal-access operations.

```
Name      Operation     Operands

symbol    GROUP         [ACTIVTO={count}]          (remote NCP only)
                                 {NONE }

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

                        [,CRETRY=count]

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

                        [,DIAL={YES}]
                               {NO }

                        [,EOB=(char[,F])]

                        [,EOT=(char[,F])]

                        [,KBDLOCK=chars]

                                {SS  }
                        [,LNCTL={BSC }]
                                {SDLC}

                        [,PADCNT={count}]
                                 {0    }

                        [,PECHAR={chars}]
                                 {EO   }

                        [,QUIETCT= {count}]
                                   {0    }

                        [,REPLYTO={count}]
                                  {NONE }

                        [,SYNDLAY={count}]
                                  {1.0  }
```

Name	*Operation*	*Operands*

```
[,TEXTTO={count}]
         {NONE }

[,TTDCNT={count}]
         {15   }

       {EP }
[,TYPE={NCP}]
       {PEP}

[,WACKCNT={count}]
          {15   }

          {count}
[,WAKDLAY={2.2  }]
          {NONE }

[,WTTYEOB=chars]

[,WTTYEOT=chars]
```

VTAM-only operands:

```
ANSWER=
BUFLIM=
CALL=
DISCNT=
FEATUR2=
ISTATUS=
LOGAPPL=
LOGTAB=
MODETAB=
SSCPFM=
USSTAB=
VIDSEQ=
VPACING=
```

`symbol`

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

```
[ACTIVTO={count}]
         {NONE }
```

(applicable only for an SDLC line group in a remote network control program)

Specifies the interval, to the nearest tenth of a second, that the remote NCP will allow to elapse since the most recent communication from the local NCP over the SDLC link before the remote NCP (1) begins automatic network shutdown and configuration restart or (2) abnormally ends, resulting in re-loading (IPL) of the controller. The action taken depends on the combination of ACTIVTO and ANS operands you select, as shown in Figure 5-10.

`count`

Specifies the timeout value, to the nearest tenth of a second, that the remote NCP will wait for a communication from the local NCP before initiating shutdown. The minimum value is 60.0 seconds; the maximum is 420.0 seconds.

count is valid only if you specify ANS=YES in the BUILD macro for this remote NCP.

NONE

Specifies that the remote NCP will wait indefinitely for communication from the local NCP; that is, automatic network shutdown and configuration restart will not occur.

This operand is valid only if (1) this program is a remote network control program (TYPGEN=NCP-R is specified in the BUILD macro), (2) this GROUP macro represents the SDLC link to the local communications controller (LNCTL=SDLC is specified in this GROUP macro) and (3) PUTYPE=4 is specified in the PU macro.

If you omit this operand and specify ANS=YES (or you omit the ANS operand), ACTIVTO=420.0 is assumed. If you omit this operand and specify ANS=NO, ACTIVTO=NONE is assumed.

The value specified for this operand must be carefully chosen (see the discussion of the effects of the activity timeout under *Backup Local Remote SDLC Links*, in Chapter 2).

The following expression may be used as a guide in determining the appropriate ACTIVTO value:

$$\text{MO} - \text{IPL} \geq \text{ACTIVTO} \geq t*n \geq \text{SDO}$$

($\geq$ means "greater than or equal to")

MO is the maximum time the remote controller may be out of service owing to link failure.

IPL is the interval normally required to load (IPL) the remote controller.

ACTIVTO is the value specified in the ACTIVTO operand.

t and n are the values specified in the RETRIES operand of the PU (or INNODE) macro (within the *local* network control program) that represents the remote controller.

SDO is the amount of time normally consumed by "short-duration outages" of the local-remote link (outages caused by momentary loss of modem carrier, excessive noise, power surges, or other disruptive line conditions).

Action taken by remote NCP upon failure of SDLC link to local controller:

ACTIVTO=	*ANS=*	*Number of SDLC local-remote links*	*Action*
count	YES	One	ANS and configuration restart occur; remote NCP then awaits message traffic from local controller over the SDLC link.
count	YES	Two or more	ANS and configuration restart occur; remote NCP then awaits message traffic from local controller over all available SDLC links.
count	NO	One	Expiration of timeout causes abnormal end (Abend) of remote NCP; reloading (IPL) occurs over link (after repair).
count	NO	Two or more	Expiration of timeout causes abnormal end (Abend) of remote NCP; reloading (IPL) occurs over any available link.
NONE	NO	One, two, or more	Remote NCP waits indefinitely for traffic from local controller.
NONE	YES	One, two, or more	(Invalid combination)

Figure 5-10. Effect of ACTIVTO and ANS Operand Options on Recovery Action of Remote NCP Upon Failure of Link to Local Controller

```
           ⎧[XONOFF] [,chars] ⎫
[CHAREC=(  ⎨XON,chars         ⎬ )]
           ⎪XOFF,chars        ⎪
           ⎩NO,chars          ⎭
```

(83B3, 115A lines operating in emulation mode only), (TWX lines in emulation or network control mode)

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

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

Specify this operand as follows:

For TWX terminals:

If you wish the program to recognize, as ending characters:

- Either XON (transmit-on) or XOFF (transmit-off), but no other character, specify CHAREC=XONOFF or omit the operand. (On some TWX terminals XON is called DC1 (device control 1) and XOFF is called DC2.)

- Either XON, or XOFF, or one or two user-selected characters, specify CHAREC=(XONOFF,chars). Specify *chars* as the hexadecimal representation of the characters that the terminal sends.
- Only XON or one or two user-selected characters, specify CHAREC=(XON,chars).
- Only XOFF or one or two user-selected characters, specify CHAREC=(XOFF,chars).
- Only one or two characters that you designate, but not XON or XOFF, specify CHAREC=(NO,chars).

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

Note: The program will not recognize ENQ (enquiry) characters, or any other characters other than WRU, XON, and XOFF, unless you specify the required characters in this operand.

For 83B3 and 115A terminals (emulation mode only):

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

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

```
[CRETRY=count]
```

(start-stop and BSC lines only)

Specifies the maximum number of attempts at error recovery the network control program is to make when recoverable errors occur on any line in the group, when the line is in control mode (as opposed to text mode). (Errors in control mode are usually those that occur during polling or addressing.) The minimum number is 0 (no retries). The highest maximum you can specify is 254 (CRETRY=254). If you code CRETRY=255, the network control program retries the operation indefinitely. (The network control program makes error recovery attempts only when the line is operating in network control mode.)

If you omit this operand, the network control program attempts error recovery twice for start-stop lines, or seven times for BSC lines.

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

(World Trade teletypewriters in emulation mode only)

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

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

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

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

Specifies whether or not the lines in the group require switched line control procedures. If they do, code DIAL=YES. If they do not, code DIAL=NO or omit the operand.

Note: SDLC local-remote backup links, whether they comprise switched or nonswitched lines, use nonswitched line control procedures; for these links only DIAL=NO is valid.

```
[EOB=(char[,F])]
```

(World Trade teletypewriter lines operating in emulation mode only)]

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

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

```
[EOT=(char[,F])]
```

(World Trade teletypewriter lines operating in emulation mode only)

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

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

`[KBDLOCK=chars]`

(TWX terminal lines in network control mode only)

Specifies the non-printing, non-space character the network control program is to send to TWX terminals in the line group when necessary to signal the terminal operator to avoid using the keyboard momentarily.

If you omit this operand, the program sends the Null character as the signal, which is normally appropriate.

The signal is sent only when monitor mode is active for the line (monitor mode is activated from TSO/TCAM).

This operand is valid only if you specify DIAL=YES and LNCTL=SS in the GROUP macro, MONITOR=YES and POLLED=NO in the LINE (or GROUP) macro, and TERM=TWX in the TERMINAL (or higher-level) macro.

```
        {SS  }
[LNCTL={BSC }]
        {SDLC}
```

Specifies whether the line group contains start-stop lines (LNCTL=SS), BSC lines (LNCTL=BSC), or synchronous data link control lines (LNCTL=SDLC).

A line group may contain lines of only one type. Start-stop and BSC line groups may appear in the program generation deck in any sequence; SDLC line groups, if any, must appear after all start-stop and BSC line groups.

Example:

```
G1  GROUP   LNCTL=SS,...
    *
G2  GROUP   LNCTL=BSC,...
    *
G3  GROUP   LNCTL=BSC,...
    *
G4  GROUP   LNCTL=SS,...
    *
G5  GROUP   LNCTL=SDLC,...
    *
G6  GROUP   LNCTL=SDLC,...
```

(* represents intervening lower-level macros)

If you omit the LNCTL operand from a GROUP macro, a line group is assumed to be (1) a start-stop line group, if no GROUP macros in which SDLC is specified precede the macro; or (2) an SDLC line group, if a GROUP macro in which LNCTL=SDLC is specified does precede the macro.

LNCTL=SDLC is valid only for a line that is operated only in network control mode (TYPE=NCP is specified in the GROUP macro).

All lines in an SDLC line group must be specified as POLLED=YES, or all must be specified as POLLED=NO. Both options cannot be included in the same line group.

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

(World Trade teletypewriter lines in network control mode only)

Specifies the number of idle characters to be sent to a World Trade teletypewriter terminal to permit its motor to reach full speed before receiving data. This is required only for a line on which such terminals are not equipped with a continuously running motor. Sufficient characters must be specified to create a 1.5 second delay on the line. The characters are sent only when the line is in network control mode.

The minimum value of *count* is 0; the maximum is 255.

```
[PECHAR={chars} ]
        {E0    }
```

(TWX lines in network control mode only)

Specifies the hexadecimal representation of the data character that the network control program is to insert into data received from a TWX terminal when the program checks for and detects incorrect parity. If you omit this operand, the program inserts a backward slash (\) character (X'E0').

This operand is valid only if PARCHK=ODD or PARCHK=EVEN is specified in one or more LINE macros associated with this GROUP macro (or is specified in this GROUP macro).

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

(start-stop lines in emulation mode only)

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

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

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

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

```
[REPLYTO={count} ]
         {NONE  }
```

Specifies the reply timeout value, in seconds, for the lines in the line group. If at the expiration of this interval the network control program has not received from the station a response to polling or selection, or to message text, it makes no further attempt to communicate with the station. Instead, it indicates that a timeout error has occurred. This action occurs whether the line is operating in network control mode or in emulation mode. *Exception:* The REPLYTO operand is ignored for BSC lines operating in emulation mode.

If this GROUP macro represents an SDLC line group, the REPLYTO operand is valid only if the lines in the group are polled (POLLED=YES is specified in the LINE macro). It is invalid in a remote program (TYPGEN=NCP-R is specified in the BUILD macro).

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

If you specify REPLYTO=NONE, no timeout occurs regardless of the time that elapses between sending to the station and receiving a response.

Reply timeouts for start-stop lines 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 lines operating in network control mode, and 51.1 seconds, for lines operating in emulation mode.

Note: If any IBM 1050 in the line group includes a paper tape punch, specify a value of at least 23.5 seconds in this operand. If you specify a lesser time, or omit the operand, the network control program may time out after sending data to the paper tape punch.

If you omit the REPLYTO operand, the network control program uses a timeout of 23.5 seconds for TWX terminals (network control mode only), 1.0 second for SDLC stations, and 3.0 seconds for all other types of stations. *Exception:* No reply timeout is provided for (1) IBM 2740 Model 1 terminals (TERM=2740-1) without at least one of these features: checking (FEATURE=CHECK), transmit control (FEATURE=XCTL), or station control (FEATURE=SCTL); or (2) multiple-terminal-access terminals (TERM=MTA); or (3) World Trade teletypewriter terminals (TERM=WTTY). (If you specify REPLYTO for any of these types, it is ignored.)

For most teleprocessing networks, the standard value of 3.0 seconds (23.5 seconds for TWX terminals, 1.0 second for SDLC stations) is appropriate (except for IBM 1050s with paper tape punches) unless the system designer specifies different values.

```
[SYNDLAY={count}]
        {1.0  }
```

(BSC lines in network control mode only)

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

Note: This operand determines the interval only for lines serviced by a type 1 or type 2 communication scanner. The interval for lines serviced by a type 3 scanner is determined directly by the scanner.

```
[TEXTTO={count}]
       {NONE }
```

(BSC and start-stop lines only)

Specifies the text timeout value, in seconds, for the lines in the line group. If the interval between any two successive message characters received from a station exceeds this value, the network control program ends the Read or Invite operation with a text timeout error indication. This action occurs whether the line is operating in network control mode or in emulation mode. *Exception:* The TEXTTO operand is ignored for BSC lines operating in emulation mode.

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

If you specify TEXTTO=NONE, no timeout occurs regardless of the time interval that elapses between receipt of successive characters.

Text timeouts are resolved to the nearest half-second. *Example*: If you specify a value of 20.2, the text timeout value will be 20 seconds (nearest half second to 20.2).

Note: The value you specify for count is a nominal value. The actual elapsed interval may vary anywhere between the nominal value and twice the nominal value. A nominal value of 30 seconds, for example, will result in an actual interval of 30 to 60 seconds.

If TERM=2741 is specified in the TERMINAL (or higher-level) macros in this line group, and you omit the TEXTTO operand, a value of NONE is assumed. For any other type of terminal specified in the TERM operand, a value of 23.5 seconds is assumed, for lines operating in network control mode, and 25.6 seconds, for lines operating in emulation mode. (If TERM=MTA is specified, the TEXTTO operand has no meaning and should be omitted.)

The maximum value is 1632 seconds, for lines operating in network control mode, and 51.1 seconds, for lines operating in emulation mode.

For most teleprocessing networks the standard value of 23.5 (or 25.6) seconds is appropriate. Use of this value is recommended unless the system designer specifies a different one.

```
[TTDCNT={count}]
        {15   }
```

(BSC lines in network control mode only)

Specifies the maximum number of times the BSC temporary text delay (TTD) sequence is to be accepted from a station before the operation is aborted. The TTD sequence notifies the controller that the station is temporarily unable to send the next block of data. The maximum count is 255. Any value less than 255 specifies the maximum number of times the sequence is to be accepted; 255 specifies that the sequence is to be accepted without limit. This operand applies only to line operation in network control mode.

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

```
      {NCP}
[TYPE={EP }]
      {PEP}
```

Specifies, for the line group represented by this GROUP macro, whether all lines operate in network control mode (TYPGEN=NCP), or all lines operate in emulation mode (TYPGEN=EP), or some lines operate in one mode and some operate in another, or alternately in either mode (TYPGEN=PEP).

Note: A line group consisting of SDLC lines can operate only in network control mode.

Specify TYPE=NCP if all lines in the group are always to operate in *network control* mode.

Specify TYPE=PEP if some lines in the group are to operate in *network control* mode and others in *emulation* mode, or if any lines are to operate alternately in both modes.

Specify TYPE=EP if all lines in the group are always to operate in *emulation* mode.

The valid choices for this operand depend on whether the program can perform only network control functions, or only emulation functions, or both, as specified in the TYPGEN operand of the BUILD macro:

If...	*Then valid choices are...*
BUILD: **TYPGEN=**	**GROUP:** **TYPE=**
NCP or NCP-LR or NCP-R	NCP
EP	EP[1,2]
PEP or PEP-LR	PEP[1] NCP EP[1,2]

[1] Not valid if GROUP macro represents an SDLC line group.

[2] If this group is always to operate in emulation mode, refer to the next section, *Macro Instructions for Operation in Emulation Mode Only*, rather than to the present section.

(See the TYPE operand under the LINE macro for the corresponding table showing valid parameters for the TYPE operand.)

Default values: If you omit the TYPE operand from the GROUP macro, the values assumed are as follows:

BUILD: **TYPGEN=**	**Default for** **GROUP:** **TYPE=**
NCP or NCP-LR or NCP-R	NCP
EP	EP
PEP or PEP-LR	PEP if line group is start-stop or BSC (GROUP: LNCTL=SS or BSC) NCP if line group is SDLC (GROUP: LNCTL=SDLC)

Note: If you specify TYPE=PEP, all valid operands for operation in network control mode or in emulation mode may be specified in this GROUP macro. If you specify TYPE=NCP, only valid operands for operation in network control mode should be coded; operands applicable to emulation mode will be ignored if coded. If you specify TYPE=EP, only valid operands for operation in emulation mode should be coded; operands applicable to network control mode will be ignored if coded.

```
[WACKCNT={count}]
        {15   }
```

(BSC lines in network control mode only)

Specifies the maximum number of times the BSC wait-before-transmit (WACK) sequence is to be accepted from a station before the operation is to be aborted. The WACK sequence notifies the controller that the station is temporarily not ready to receive. Any value less than 255 specifies the maximum number of times the sequence is to be accepted; 255 specifies that the sequence is to be accepted without limit. This operand applies only to line operation in network control mode.

The maximum count is 225; the minimum is 1.

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.

```
         {count}
[WAKDLAY={2.2  }]
         {NONE }
```

(BSC lines in network control mode only)

Specifies the maximum interval that is to elapse before the network control program responds to message text received from a station on any line in the line group operating in network control mode. 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.

If you code WAKDLAY=NONE, the network control program will not send a WACK sequence when unable to send a positive acknowledgment.

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.

`[WTTYEOB=chars]`

(World Trade teletypewriter lines in network control mode only)

Specifies, in hexadecimal representation, the EBCDIC end-of-block sequence required for World Trade teletypewriters (teleprinters) connected to lines in the group that are operating in network control mode. Specify up to eight hexadecimal characters (four EBCDIC characters). Any valid character may be included in the sequence, but the FIGS and LTRS characters may each be included only once. (If the terminal is equipped to send who-are-you [WRU] sequences, avoid using the letter D in the sequence.) A typical EOB sequence is FIGS x, where *x* is any valid character (except FIGS).

`[WTTYEOT=chars]`

(World Trade teletypewriter lines in network control mode only)

Specifies, in hexadecimal representation, the EBCDIC end-of-transmission sequence required for World Trade teletypewriters connected to lines in the group that are operating in network control mode. Any valid character may be included in the sequence, but the FIGS and LTRS characters may each be included only once. (If the terminal is equipped to send who-are-you [WRU] sequences, avoid using the letter D in the sequence.) A typical EOT sequence is FIGS y LTRS, where *y* is any valid character (except FIGS or LTRS).

This operand is required if World Trade teletypewriters are connected to lines in the group represented by this GROUP macro.

Restriction on Number of Time Intervals Specified

A maximum of 16 different time intervals may be specified in the network control program. This includes (1) seven standard intervals (including default values) 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. If the total number of different time intervals is 16, the default values are used for all subsequent time specifications the generation procedure encounters in processing the source statements. 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 (immediate action required)
1.0 seconds
2.2 seconds
3.0 seconds
23.5 seconds
60.0 seconds
NONE (infinite time delay)

Specifying Lower-Level Operands in the GROUP Macro

In addition to the preceding operands, most operands of the LINE, CLUSTER, INNODE, PU, LU, TERMINAL, and COMP macros can be specified in the GROUP macro instead of the individual macros mentioned. Figure 5-9 shows which of the lower-level operands you may specify in the GROUP macro.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the GROUP macro instruction convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this GROUP macro description means only that the NCP generation assembly procedure accepts them as valid operands of the GROUP macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

LINE Macro Instruction

The LINE macro represents one start-stop or BSC communication line or one SDLC link attached to the communications controller. The explanations and operands given below are applicable when defining a line that is to be operable alternately in network control mode and emulation mode. When specifying a line that is to operate only in network control mode, these operands are not applicable and should be ignored:

BUFSIZE
CHECK
CHNPRI
CU
DISABLE
DUALCOM
FEATURE*
MODEM
PAD
QUIET
TERM*
UNITXC

*This operand is valid for network control mode and can be coded in the LINE macro, but see the operand description under the TERMINAL macro rather than under the LINE macro.

To define a start-stop or BSC line that is to operate only in emulation mode, see the description of the LINE macro in Chapter 7.

The LINE macro specifies:

- Whether the communications facility of which the line is a part is half-duplex or duplex.
- Whether or not the controller that will execute the program you are defining is to be the primary station on a nonswitched point-to-point BSC contention line.
- The line interface address, within the communications controller, to which the line is attached.
- The line interface address, within the communications controller, to which the related automatic calling unit (ACU), if any, is attached.
- The speed of the line—that is, the rate (in bits per second) at which the controller and stations on the line transmit data.
- The interrupt priority of the line.
- Whether clocking of the communication line is provided by the modem (modem clocking) or by the communication scanner (business machine clocking).
- Whether the controller that will execute the program you are defining is a tributary station on the line represented by this macro, and if so, the addressing character to which the controller will respond.
- Whether the modem by which this line is attached to the controller, if a dual data rate modem, is to operate at the higher or the lower rate.
- Whether the modem to which this line is attached has the "new sync" feature.
- Whether the modem to which this line is attached is sensitive to transitionless bit streams or to repeated dibit (binary 10) bit patterns (SDLC links only).
- Whether (for a switched line) the modem operates in ring indicator mode (not applicable in U.S. and Canada).
- Whether or not the stations on the line are polled.
- The transmission code used by stations on the line.
- Certain procedural options the network control program is to use for the line.

If the start-stop or BSC line represented by this LINE macro is to operate in emulation mode, as well as in network control mode (that is, alternately in either mode), the LINE macro also specifies:

- The emulation subchannel address associated with the line.

- The priority of the emulation subchannel associated with the line, with respect to other emulation subchannels.
- The type of transmission control unit (IBM 2701, 2702, 2703) that the network control program is to emulate when communicating over the line.
- The dual communications interface, if any, associated with the line (applicable only if the transmission control unit [2701] being emulated is equipped with the dual communications interface feature.)
- The type of start-stop or BSC station connected to the line.
- Certain features with which the stations connected to the line are equipped.
- Certain procedural options the network control program is to use when operating the line in emulation mode.

Some of the items in the preceding lists apply only to certain types of lines or stations. The descriptions of the individual operands that follow indicate the conditions under which you code the parameters they specify.

If following a LINE macro you code no TERMINAL macros (as may be the case for a switched line [DIAL=YES in the GROUP macro]), you must code CALL=OUT in the LINE macro (or in the GROUP macro) and specify the type of device in the TERM operand of the same macro. In the LINE macro you may also specify the FEATURE operand; all other TERMINAL operands, if coded in the same LINE macro, are ignored. (See Appendix D for an example of switched network operation.)

| *(VTAM Users Only)* Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only for the VTAM initialization process and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each LINE macro coded causes a resource name to be generated.

One LINE macro must be coded for each communication line connected to the controller, whether it is to operate only in network control mode or alternately in either network control or emulation mode. *Exception:* A single LINE macro instruction represents either one or two communication lines comprising an SDLC link. All LINE macros representing lines in a physical line group must appear between the GROUP macro representing that group and the next GROUP macro.

Note: Any LINE macros representing SDLC links must appear in the program generation input deck following LINE macros representing BSC and start-stop lines. (An example appears in the description of the LNCTL operand of the GROUP macro, earlier in this chapter.)

A remote communications controller may have one principal SDLC link to the local controller, and from one to three alternate ("backup") SDLC links to that controller. Each link is represented within the remote network control program by a LINE macro; therefore, up to four LINE macros representing SDLC links to the local controller may appear in the generation input deck for the remote program.

```
Name      Operation      Operands

symbol    LINE                   {line addr                            }
                         ADDRESS={ (line addr,subchan addrl[,subchan}
                                       addr2,...,subchan addrn])
                                 { (xmt addr,rcv addr)                 }

                         SPEED=(rate[,rate])

                         [,ANSTONE={YES}]
                                   {NO }

                         [,AUTO=address]

                         [,BUFSIZE=n]

                                {IN   }
                         [,CALL={OUT  }
                                {INOUT}

                         [,CHECK={DCD  }]
                                 {NODCD}

                         [,CHNPRI={NORMAL}]
                                  {HIGH  }

                         [,CLOCKNG=({INT}[,{INT}])]
                                   {EXT}  {EXT}

                                {EBCDIC  }
                                {USASCII }
                                {BCD     }
                                {EBCD    }
                                {COR     }
                         [,CODE={KATAKANA}]
                                {ITA2    }
                                {ZSC3    }
                                {DIC1    }
                                {BCD2    }
                                {COR2    }
                                {DIC3    }

                         [,CONFIG={SW    } ]
                                  {NONSW}

                         [,CRRATE=integer]

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

                         [,CUTOFF={count}]
                                  {NO   }

                                  {SDLC1}
                                  {2972 }
                         [,CUTYPE={3271 }]
                                  {3275 }

                         [,DATRATE=({HIGH}[,{HIGH}])]
                                   {LOW }  {LOW }

                         [,DIALALT={dial set name}]
                                   {NONE         }
```

```
Name      Operation      Operands

                         [,DIALSET={dial set name} ]
                                   {NONE         }

                         [,DISABLE={YES}]
                                   {NO }

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

Name *Operation* *Operands*

```
[,DUPLEX=( {HALF} [,{HALF}] )]
           {FULL}    {FULL}

[,FEATURE=( [{DUALCODE}]
             {NODUALCD}

           [,{IMEND   }]
             {NOIMEND}

           [,{LRC   }]
             {NOLRC}

           [,{SPACE   }] )]
             {NOSPACE}

[,FGSLTRS={IN }]
          {OUT}

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

[,LINESIZ=integer]

[,LNQTCNT={count }]
          {1     }

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

[,MONITOR={YES}]
          {NO }

[,MPTALT={YES}]
         {NO }

[,MTALIST=entry]

[,NEGPOLP={n   }]
          {NONE}

[,NEWSYNC=({YES}[,{YES}])]
           {NO }  {NO }

[,NRZI={YES}]
       {NO }

[,PAD={YES}]
      {NO }

         {ODD     }
[,PARCHK={EVEN    }]
         {NOCHECK }

[,PAUSE={t}]
        {0}
                  {WAIT   }
[,POLIMIT=([{n}][,{NOWAIT}])  ]
            {1}   {QUEUE  }

[,POLLED={YES}]
         {NO }

[,POLLTO={ERROR  }]
         {NEGRESP}
```

Name *Operation* *Operands*

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

[,REDIAL={count} ]
         {3    }

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

[,RING={YES} ]
       {NO }

[,SCLSET={YES} ]
         {NO }

[,SERVLIM=count]

[,SERVPRI={OLD} ]
          {NEW}

[,SESSION={count} ]
          {1    }

[,SPDSEL={YES} ]
         {NO }

[,SPSHIFT={YES} ]
          {NO }

[,TADDR={char} ]
        {NONE}

[,TERM=type]

[,TRANSFR=count]

       {EP }
[,TYPE={NCP} ]
       {PEP}

[,UNITXC={YES} ]
         {NO }

[,USE={NCP} ]
      {EP }

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

VTAM-only operands:

```
ANSWER=
BUFLIM=
CALL=
DISCNT=
FEATUR2=
ISTATUS=
LOGAPPL=
LOGTAB=
MODETAB=
SSCPFM=
USSTAB=
VIDSEQ=
VPACING=
```

`symbol`

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

```
        {line addr                                                    }
ADDRESS={(line addr, subchan addr1[,subchan addr2,...,subchan addrn])}
        {(xmt addr,rcv addr)                                          }
```

Specifies the start-stop or BSC line interface address for the line represented by this LINE macro, and specifies the corresponding emulation subchannel address, if the line is to be operable in emulation mode; or specifies the one or two line interface addresses for an SDLC link. Determine line addresses used in your communications controller from your IBM representative.

For start-stop and BSC lines:

Code ADDRESS=line addr, where *line addr* is the three-hexadecimal-digit controller address to which the line is attached. *Example:* ADDRESS=02F.

Note: Lines on which the data rate is 19,200 bps or more require two consecutive controller addresses. Specify in the ADDRESS operand only the even address of the even-odd pair. *Example:* If the addresses assigned to a 19,200 bps line are 022 and 023, you would code ADDRESS=022.

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

If the line represented by this LINE macro is to operate sometimes in emulation mode, specify both the line address and the corresponding emulation subchannel address(es) as ADDRESS=(line addr,subchan addr*1* [,subchan addr*2*, . . . ,subchan addr*n*]). Specify *three* hexadecimal digits for the line address. Specify *two* hexadecimal digits, each followed by -0 or -1, for each subchannel address: -0 to indicate that the associated channel adapter is installed in the *base* module, -1 to indicate that the associated adapter is installed in the first *expansion* module (3705 only). (-0 may be omitted, in which case the base module is assumed as the adapter location. *Examples:* (1) ADDRESS=(02F,2E-0): line address 02F associated with subchannel address 2E via channel adapter in base module; (2) ADDRESS=(02F,2E-0,17-1,1C-0): line address 02F associated with subchannel addresses 1C and 2E via channel adapter in base module and with subchannel address 17 via adapter in expansion module. The subchannel address must be associated with a type 1 channel adapter (not a type 2 or type 3 adapter, which can be used only for operations in network control mode). The subchannel address specified for this line must not be the same as a subchannel address associated with any other line or used by any other program facility (such as the dynamic dump utility).

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

The subchannel address must be within the range specified by the HICHAN and LOCHAN operands of the BUILD macro.

For SDLC links:

If the communications controller is to transmit and receive over the *same* path (regardless of whether the communication facility constituting the path is half-duplex or duplex, specify a single line interface address. *Example:* ADDRESS=020.

If the communications controller is to transmit and receive over *separate* paths (regardless of whether the communications facility is half-duplex or duplex, specify *two* line interface addresses. *Example:* ADDRESS=(020,021).

Figure 5-11 gives the range of valie line addresses for this operand.

This operand is required.

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

Figure 5-11. Valid Line and Auto Call Interface Addresses for LINE Macro Instruction

`SPEED=(rate[,rate])`

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

If the modem at the controller is a dual-rate modem (whether the rate used is program selected or selected by a manual switch on the modem), specify the higher of the two rates.

If one line interface address is specified in the ADDRESS operand of this macro, specify SPEED=rate. If this LINE macro represents an SDLC link and two line interface addresses (xmt addr, rcv addr) are specified in the ADDRESS operand, code SPEED=rate, if the speed for both sides of the link are the same. Code SPEED=(rate,rate) for the transmit and receive addresses, respectively, if the rates differ.

If internal (business machine) clocking is used (see the CLOCKNG operand), the rate must be one of the four oscillator rates specified for the communication scanner to which this line is attached (SPEED operand of the CSB macro).

Specify the line speed in bits per second, omitting a fractional part, if any. For example, specify a line speed of 1200 bps as SPEED=1200; specify a line speed of 134.5 bps as SPEED=134 (omitting the decimal point and fraction).

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

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

If this LINE macro represents a multiple-terminal-access line used to communicate with terminals of different speeds, specify the speed of any of the terminals. For example, if terminals operating at speeds of 110 bps and 134.5 bps use this line, you may specify either SPEED=110 or SPEED=134.

This operand is required.

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

(call-in switched lines in network control mode only)

Specifies whether the network control program, upon answering a call over a call-in switched line, is to transmit an "answer tone" to the calling station to signify completion of the line connection.

It is necessary for the program to send the tone only when the modem that attaches the line to the communications controller does not itself send an answer tone. Most modems in the United States and Canada do provide the tone, therefore ANSTONE=NO is normally appropriate. If in doubt, consult the supplier or installer of the modem.

This operand is valid only for switched lines used for incoming calls (DIAL=YES [or DIAL=NO and CONFIG=SW] is specified).

```
[AUTO=address]
```

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

A line that the network control program is to use to call a station (that is, you code CALL=OUT or CALL=INOUT in this macro) may or may not be equipped with the ACU facility.

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

If the line does not have the ACU facility, omit the operand. Calls to stations over this line must in this case be made manually.

Figure 5-11 gives the range of valid addresses.

```
[BUFSIZE=n]
```

(lines serviced by type 3 scanner only; emulation mode only)

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

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

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

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than do smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

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

Note: Do not specify 4-byte buffers (*n*=4) if the line represented by this LINE macro is polled (POLLED=YES) and the size of the poll entries in the service order table is six bytes or more. (A poll entry comprises the address characters, ENQ, and index byte used to poll one station on the line [for example, C1 C1 40 40 ENQ Index]).

This operand is valid only for a communication line serviced by a type 3 scanner and applies to line operation only in emulation mode.

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

(switched BSC and start-stop lines only)

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. (Also specify in the DIALSET operand the name of the dial set to which this line belongs [start-stop and BSC lines only].)

If the line is to be used for both incoming and outgoing calls, code CALL=INOUT. (Also specify in the DIALSET operand the name of the dial set to which this line belongs [start-stop and BSC lines only].)

This operand is valid only if you specify DIAL=YES in the GROUP macro, and applies only to line operation in network control mode.

VTAM Note: For SDLC links, this operand provides information only to the VTAM initialization procedure and has no effect on the generation of the network control program. See the *VTAM System Programmer's Guide* for information on use of this operand for SDLC links.

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

(switched, duplex, start-stop lines in emulation mode only)

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

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

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

(applicable to BSC lines in emulation mode only)

Specifies the priority of the emulation subchannel associated with this line with respect to the other emulation subchannels used by the network control program being defined when communicating with the host processor in emulation mode.

This operand is valid only for a BSC line (LNCTL=BSC is specified in the GROUP macro) to be operated (sometimes or always) in emulation mode.

CHNPRI=NORMAL is ordinarily appropriate unless the line represented by this LINE macro is to operate at a data rate of 19,200 bps or more, and the majority of the remaining lines that are to operate in emulation mode are slow speed lines (2400 bps or less).

See the *EP Storage Estimates* manual for further information about subchannel priorities.

```
[CLOCKNG=( {INT} [, {INT} ] )]
           {EXT}    {EXT}
```

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

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

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—CLOCKNG=INT or CLOCKNG=EXT.

If the ADDRESS operand specifies two line interface addresses (SDLC links only), specify only the first suboperand, if both sides of the link use internal clocking or both use external clocking. If one uses internal clocking and the other uses external, specify CLOCKNG=(INT,EXT) or CLOCKNG=(EXT,INT), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

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

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

```
        {EBCDIC  }
        {USASCII }
        {BCD     }
        {EBCD    }
        {COR     }
[CODE=  {KATAKANA} ]
        {ITA2    }
        {ZSC3    }
        {DIC1    }
        {BCD2    }
        {COR2    }
        {DIC3    }
```

(start-stop and BSC lines only)

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 code used. (The *Program Reference Handbook*, in the section entitled "Line Character Codes," shows the contents of the translation tables associated with each transmission code.)

The network control program does not perform code translation for a line operating in emulation mode. The program transmits data over a communication line in the same transmission code in which it received the data over the emulation subchannel, and vice versa. However, for BSC lines operating in emulation mode, the program must be aware of the transmission code employed (EBCDIC or USASCII). For start-stop lines operating in emulation mode, on the other hand, the program need not know the code being used. For this reason, the code you specify in this operand applies to operation in network control mode, if a start-stop code, but if it is a BSC code, applies to operation in both emulation and network control modes.

If you code the MTALIST operand of the LINE macro, omit the CODE operand.

Omit the CODE operand if this LINE macro represents an SDLC link (LNCTL=SDLC is specified in the GROUP macro).

Figure 5-12 lists the codes that may be specified for each type of station. ("WT TTY" [World Trade Teletypewriters] refers to various European teletypewriters [teleprinters] employing either the ITA2 or ZSC3 transmission codes.)

The CODE operand should not be specified for any type of station not shown in Figure 5-12. (If a code is specified, it is ignored.) Also omit the CODE operand if the line is a multiple terminal access line.

			Type of Station				
Transmission Code	CODE=	Any BSC Station	IBM 1050	IBM 2740	IBM 2741	WU TWX	WT TTY
Extended BCD Interchange Code	EBCDIC	•[1]					
USA Standard Code for Information Interchange	USASCII	•					
USASCII (transparent)	USASCII-T	•[2]					
Extended Binary Coded Decimal	EBCD		•[1]	•[1]	•		
Binary Coded Decimal 1	BCD		•	•	•		
Binary Coded Decimal 2	BCD2		•		•		
Correspondence 1	COR			•	•[1]		
Correspondence 2	COR 2				•		
Katakana	KATAKANA		•	•			
Data Interchange Code 1	DIC1					•[1]	
Data Interchange Code 3	DIC3					•	
International Telegraph Alphabet No. 2	ITA2						•[1]
Figure Protected Code	ZSC3						•

[1] Code assumed if CODE operand is omitted.
[2] Valid only for a line serviced by a type 3 scanner and operated in emulation mode.

Figure 5-12. Valid CODE Operand Values

```
[CONFIG={SW     }]
        {NONSW}
```

(remote NCP only)

Specifies, within a remote network control program (TYPGEN=NCP-R), whether the physical communication path making up the local-remote SDLC link (as specified by LNCTL=SDLC and POLLED=NO) is switched or nonswitched. (Each local-remote SDLC link is operated by the NCP as a nonswitched link, regardless of whether the physical path is nonswitched or switched.) If the path is switched (CONFIG=SW is specified), the remote NCP can monitor the link for ring indicator signals (if you specify RING=YES) and present an answer tone (if you specify ANSTONE=YES) when the remote controller is called by the local NCP.

This operand is valid only in a remote network control program (TYPGEN=NCP-R) and the link to the local controller is specified as LNCTL=SDLC, DIAL=NO, and POLLED=NO.

```
[CRRATE=integer]
```

(start-stop lines only)

Specifies the number of print positions the carriage of printers connected to the line will return for each idle character the network control program sends. The minimum is 1; the maximum is 255.

If you omit this operand, a rate of 40 print positions per idle character is assumed for 83B3, 115A, and TWX teletypewriter terminals; 13 is assumed for IBM 1050 terminals equipped with the Accelerated Carrier Return feature (if ACR is included among the options specified in the FEATURE operand of all TERMINAL macros for stations attached to the line). Ten positions per idle character is assumed for all other terminals.

This operand is valid only if LNCTL=SS is specified in (or LNCTL operand omitted from) the GROUP macro, and applies only to line operation in network control mode.

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

(applicable to emulation mode only)

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

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

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

(start-stop and BSC lines only)

Specifies a maximum number of sub-blocks the network control program will accept from a station during line operation in network control mode, 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.

```
        {SDLC1}
        {2972 }
[CUTYPE={3271 }]
        {3275 }
```

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

Notes: (1) The CUTYPE operand as described above applies to operation of the line in emulation mode, except for SDLC1 (see note 3 below). The same operand, as described under the CLUSTER macro, applies to operation of the line in network control mode. If the line to which the clustered station is attached is to be operable alternately in either mode, you may specify the CUTYPE operand in both the LINE and CLUSTER macros; however, you must specify the same values in both. Alternatively, you may specify the CUTYPE operand only in the LINE macro, or in the GROUP macro, since this operand (among others) may be coded in a macro of a higher level than that under which the operand is described. (2) The values 2845 and 2848 cannot be coded in this CUTYPE operand, because the network control program cannot communicate with these types of control units in network control mode. (3) The value SDLC1 applies only to line operation in network control mode; CUTYPE=SDLC1 coded here is equivalent to CUTYPE=SDLC1 coded in the CLUSTER macro, in accordance with the operand hierarchy provided for configuration macro instructions.

```
[DATRATE=( {HIGH} [,{HIGH}] )]
           {LOW }   {LOW}
```

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

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—DATRATE=HIGH or DATRATE=LOW. 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.

If the ADDRESS operand in this LINE macro specifies two line interface addresses (SDLC links only), specify only the first suboperand, if both sides of the link use the high rate or both use the low rate. If the rates differ, specify DATRATE=(HIGH,LOW) or DATRATE=(LOW,HIGH), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

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

If you omit this operand, DATRATE=LOW or DATRATE=(LOW,LOW) is assumed. If you omit the second subparameter, and the ADDRESS operand specifies two addresses, the same value is assumed for this subparameter as you specified for the first subparameter.

```
[DIALALT={dial set name}]
         {NONE          }
```

(nonswitched point-to-point or BSC multipoint lines only)

Specifies the name of a dial set (DIALSET macro) from which the network control program can select a switched line (if available) to use in communicating with a station when unable to reach that station via the regular nonswitched point-to-point or BSC multipoint line. (This capability of the network control program is called the switched network backup function, and applies only to line operation in network control mode.) Upon request from the host processor, the network control program calls the station over a line from the dial set you specify.

For the alternate line to be used successfully, the lines in the dial set must have the same characteristics as the nonswitched 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 or BSC multipoint line, (2) you wish the network control program to reach the station over an alternate line when communication with the station over the regular nonswitched line is not possible, and (3) you have specified switched network backup among the options included in the network control program (see the SYSCNTRL macro).

This operand is not valid for an SDLC link (LNCTL=SDLC is specified in the GROUP macro).

```
[DIALSET={dial set name}]
         {NONE          }
```

(call-out BSC or start-stop switched lines only)

Specifies the name of the DIALSET macro for the dial set to which this line belongs. 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).

This operand is invalid for an SDLC link.

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

(applicable to emulation mode only)

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

Most modems do not require the long time-out, and DISABLE=NO is therefore the appropriate value. If in doubt, consult the supplier or installer of the modem.

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

(applicable to emulation mode only)

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

`lineaddress`

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

`A`

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

`B`

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

`DUALCOM=NONE`

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

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

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

```
LINE1 LINE  ADDRESS=(020,17), LINE2  LINE ADDRESS=(021,17),
            DUALCOM=(021,A),              DUALCOM=(020,B),
                  .                             .
                  .                             .
                  .                             .
```

```
[DUPLEX=( {HALF} [ , {HALF} ] )]
          {FULL}     {FULL}
```

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

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

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—DUPLEX=HALF or DUPLEX=FULL.

If the ADDRESS operand specifies two line interface addresses (SDLC links only), specify only the first suboperand, if both sides of the link are half-duplex or both are duplex; if one is half-duplex and the other is duplex, specify DUPLEX=(HALF,FULL) or DUPLEX=(FULL,HALF), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

Note: For BSC and start-stop lines, coding DUPLEX=FULL causes the request-to-send signal (sent from the communications controller to the modem) to be active when either sending or receiving data; coding DUPLEX=HALF causes the request-to-send signal to be active only when sending data. For SDLC links, this operand has no effect on the activation of the request-to-send signal; this signal is always active if the ADDRESS operand specifies two addresses and is active only when sending data if the ADDRESS operand specifies a single address.

```
[FEATURE=( . . . )]
```

(applicable to emulation mode only)

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

Note: The features and parameters below are those which apply only to operation of the line in emulation mode. The same operand, FEATURE, as described under the TERMINAL macro, specifies the features of which the network control program must be aware when operating the line in network control mode. If the line represented by this LINE macro is to be operable alternately in either mode, you may specify the FEATURE operand in both this LINE macro and in the TERMINAL macro or macros that follow this LINE macro. Specify the features for emulation mode operation in this LINE macro and those for network control mode in the TERMINAL macro(s). Alternatively, you may specify the FEATURE operand only in this LINE macro (or in the GROUP macro), and include both the features for emulation mode operation and those for network control mode operation.

```
[ {DUALCODE} ]
  {NODUALCD}
```

(BSC line only; emulation of 2701 only)

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

```
[{IMEND  }]
 {NOIMEND}
```

(specially-equipped start-stop terminals only)

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

```
[{LRC  }]
 {NOLRC}
```

(start-stop terminals only)

Specifies whether or not the start-stop terminals with which the network control program communicates in emulation mode over this line are equipped with record checking capability (either as an inherent function or as a feature).

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

Station:	*TERM operand specifies:*
IBM 1050	1050
IBM 1060	1060
IBM 2740 Model 1	2740-1
IBM 2740 Model 2	2740-2
IBM System/7	SYS7

```
[{SPACE  }]
 {NOSPACE}
```

(teletypewriter terminals only)

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

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

```
[FGSLTRS={IN }]
         {OUT}
```

(teletypewriter lines in network control mode only)

Specifies whether the network control program is to delete FIGS and LTRS characters from message text received from 83B3, 115A, or World Trade teletypewriter terminals (FGSLTRS=OUT) or to leave such characters within message text (FGSLTRS=IN), when the line is operating in network control mode.

```
         {0}   {0}
[INTPRI=( {1} [,{1}] )]
         {2}   {2}
         {3}   {3}
```

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

If the ADDRESS operand of this LINE macro specifies one line interface address, specify only the first suboperand.

If the ADDRESS operand of this LINE macro specifies two line interface addresses (SDLC links only), specify only the first suboperand, if the priority is the same for both sides of the link. Specify both suboperands if the priority differs. (The first and second suboperands are the priorities for the first and second addresses, respectively.)

If this line is serviced by a type 2 or type 3 communication scanner, the valid range for INTPRI is 0 through 3. If this line is serviced by a type 1 communication scanner, the only valid values are 0 and 1, with 1 being the higher priority.

Appendix J gives a method for determining the interrupt priorities for each line in the network.

```
[LINESIZ=integer]
```

(start-stop lines in network control mode only)

Specifies the length of the print line, in number of print positions, for printer-type devices connected to the line represented by this LINE macro. The minimum value for *integer* is 1; the maximum is 255. This operand applies only to line operation in network control mode.

If you omit this operand, a line length of 72 print positions is assumed for TWX, 83B3, 115A, and World Trade teletypewriter terminals; 130 print positions is assumed for all other types of terminals.

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

(start-stop lines in network control mode only)

Specifies the number of times the network control program will test the start-stop line following receipt of message data to ensure that the line has electrically "quiesced." (The line must become electrically quiet before the next data transmission begins or loss of message data may occur.)

The default value of 1 is appropriate for most start-stop lines under 1200 bps. For lines operating at 1200 bps or more, several tests in succession may be necessary to ensure that the line is quiet. The recommended value for 1200 bps start-stop lines is 5.

The minimum value you may code is 1; the maximum is 255.

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

(applicable to emulation mode only)

Specifies whether the communication line (1) is enabled immediately after the communication controller has been loaded (IPL) or the System Reset key has been pressed (MODEM=OPTION1); or (2) is disabled after the controller has been loaded or the System Reset key has been pressed, and must subsequently be enabled by command from the access method (MODEM=OPTION2). (Until the line is enabled, most commands issued to the line by the access method will result in an intervention required indication to the access method.) (Specify MODEM=NTT if the modems on the line are those of Nippon Telegraph and Telephone Co.)

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

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

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

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

(IBM 1050, 2741; WU TWX; and multiple-terminal-access lines in network control mode only)

Specifies whether the network control program is to monitor the line, for an attention signal sent by the terminal or a "disconnect" condition at the terminal. (The program monitors the line whether or not it is currently executing a command for that line.)

The program notifies the access method when it detects either an attention signal or a disconnect condition.

Note: Use of the keyboard lock option for TWX terminals (GROUP macro: KBDLOCK operand) requires that MONITOR=YES be specified.

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

(switched call-out BSC line only)

Specifies whether the switched BSC line represented by this LINE macro can be used as an alternate line for communicating with BSC stations normally reached over a nonswitched multipoint line. When failure of the multipoint line prevents the network control program from communicating with a station over that line, the switched network backup facility can be used to reach the station over any switched call-out BSC line for which you specify MPTALT=YES.

(Use of the switched network backup option requires that the BSC station [including modems] be appropriately configured for switched backup operation.)

This operand is valid only if the line represented by this LINE macro is a switched call-out BSC line (LNCTL=BSC, DIAL=YES, and CALL=OUT or CALL=INOUT are specified). If you so specify these operands and omit the MPTALT operand, MPTALT=NO is assumed.

```
[MTALIST=entry]
```

(Call-in multiple terminal access lines only)

Specifies that the line represented by this LINE macro is to be used for call-in 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.

This operand is valid only if you specify DIAL=YES and LNCTL=SS in the GROUP macro, and CALL=IN or CALL=INOUT in this LINE macro (or in the GROUP macro).

```
[NEGPOLP={n    }]
         {NONE }
```

(applicable to BSC lines in network control mode only)

Specifies the duration of a pause following receipt of a negative response to polling by a multipoint BSC station, before polling is resumed. The pause may be specified as seconds or as seconds and tenths. The minimum is .1 second; the maximum is 23.5 seconds. (The average pause imposed is the value of *n* plus 50 milliseconds.)

If you omit this operand or specify NEGPOLP=NONE, no pause is imposed between receipt of the negative response and continuation of polling.

Use of a negative polling pause limits the amount of non-productive polling on a BSC multipoint line and therefore reduces the amount of processing overhead associated with such polling. On the other hand, too large a negative polling pause can increase the response time experienced by the operators of terminals on the line. For further information on the use of the negative polling pause, see the NCP Storage and Performance publication.

This operand is valid only for a BSC line (LNCTL=BSC) on which multipoint line control is used (POLLED=YES), and is applicable only when the line is in network control mode.

```
[NEWSYNC=( {YES}  [,{YES}] ) ]
           {NO }    {NO }
```

(BSC and SDLC lines only)

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

NEWSYNC=YES is valid only if (1) the modem (at the controller) serving the line has the new sync feature, *and* (2) if the communications controller is the multipoint master (not tributary) station for a duplex (*not* half-duplex) line on which multipoint line control is used, *and* (3) the modem at the remote station does not continuously send carrier signal to the modem at the controller. (The modem at the remote station sends carrier continuously if it is wired internally to do so [consult the installer or supplier of the modem if in doubt] or the remote station continuously sends the "request-to-send" signal to the modem [consult your IBM representative to determine if this is the case].)

NEWSYNC=NO is required if there is only one remote station on the communication line.

If you omit the NEWSYNC operand, NEWSYNC=YES is assumed if you specify LNCTL=BSC or LNCTL=SDLC, POLLED=YES, DUPLEX=FULL, CLOCKING=EXT, and DIAL=NO. (All five operands must be so specified.) If you omit the operand and specify LNCTL=BSC or LNCTL=SDLC, but any of the four remaining operands is not as shown, NEWSYNC=NO is assumed. For LNCTL=SS, the NEWSYNC operand has no meaning.

Determine from your IBM representative or the installer or supplier of the modem (if other than an IBM modem) whether the appropriate conditions above prevail. If they do not, the new sync function cannot be used.

If the ADDRESS operand in this LINE macro specifies one line interface address, specify only the first suboperand—NEWSYNC=YES or NEWSYNC=NO.

If the address operand specifies two line interface addresses, specify only the first suboperand, if the new sync option is to be used for both sides of the link (NEWSYNC=YES) or for neither (NEWSYNC=NO). If the option is required for only one of the two, specify NEWSYNC=(YES,NO) or NEWSYNC=(NO,YES), as appropriate. (The first and second suboperands apply respectively to the first and second addresses specified in the ADDRESS operand.)

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

(SDLC lines only)

Specifies whether the data terminal equipment (controller and remote station) at the ends of the SDLC link must operate in "non-return-to-zero-inverted" (NRZI) mode (NRZI=YES) or in "non-return-to-zero" (NRZ) mode (NRZI=NO).

If internal (business machine) clocking is used on the link, specify NRZI=YES or omit the operand.

If external (modem) clocking is used on the link, specify NRZI=YES or omit the operand *unless* the modem is sensitive to repeated binary 10 bit patterns (that is, 10101010...). (Sensitivity to this pattern can cause the modem to lose synchromism when it encounters sufficiently long sequences of this pattern in message data.) If the modem is sensitive to this bit pattern, specify NRZI=NO. To determine whether your modems are sensitive to this pattern, consult your IBM representative (for IBM modems) or the supplier or installer of the modem (for non-IBM modems).

Note: NRZI=NO is the proper choice for most non-IBM modems; NRZI=YES is appropriate for IBM modems.

Caution: All business machine equipment (terminal equipment) on the same SDLC link must use the same encoding scheme—that is, all use NRZI mode (NRZI=YES) or all use NRZ mode (NRZI=NO). Mixing of modes on the same SDLC link will result in total lack of communication between stations on the link. In the case of an SDLC link between a local and a remote communications controller, the equivalent choice (NRZI=YES or NRZI=NO) is required for the LINE macros in each program (local and remote) that represent the SDLC link, and must correspond to the setting of the NRZI bit in the IPL configuration data set of the remote program loader. (The IBM customer engineer sets the NRZI bit.)

This operand is valid only if LNCTL=SDLC is specified in the GROUP macro.

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

(BSC lines in emulation mode only)

Specifies whether or not the communications controller, when emulating an IBM 2703 Transmission Control, is to verify that the first four bits of trailing pad characters received from lines are all 1s, that is, hexadecimal 'F'.

If you specify PAD=YES or omit the operand, the controller checks each pad character received and indicates a data check error if the fist four bits are not all 1s.

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

```
          {ODD      }
[PARCHK={ EVEN     }]
          {NOCHECK}
```

(TWX lines in network control mode only)

Specifies whether the network control program is to perform parity checking for TWX terminals on the line represented by this LINE macro, and whether the checking is for odd parity or even parity.

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

(multipoint start-stop or BSC lines in network control mode)

For BSC and start-stop lines: specifies the number of seconds of delay between successive service cycles when no sessions currently exist. *t* may be from 0 to 255; if you omit the operand, 0 is assumed, i.e., there is no delay between successive cycles. This operand applies only to line operation in network control mode.

```
[PAUSE={t  }]
       {0.1}
```

(SDLC primary station line only)

For SDLC links: specifies the average duration of the polling cycle in seconds or seconds and tenths of seconds. The polling cycle extends from the moment polling begins with the first active entry in the service order table to the moment polling next begins at the same entry. If the time expended in a complete polling cycle—that is, in servicing all active entries in the service order table—equals or exceeds the time you specify as *t*, the next polling cycle begins immediately. If, on the other hand, the time expended in a complete polling cycle is less than *t*, the beginning of the next polling cycle is deferred until *t* seconds have elapsed since the beginning of the cycle just completed. During the pause, the SDLC link is in "poll-wait" state; any outgoing data ready for transmission to the SDLC stations on the link is sent during this pause.

Allowing a pause to elapse when activity on the link is relatively low can reduce the amount of processing time consumed by unproductive polling.

Note: The larger the number of active entries in the service order table, the greater the likelihood will be that polling cycles will proceed continuously, without intervening pauses.

t may be from 0 to 25.5; if you omit this operand, a pause of 0.1 seconds is assumed.

This operand is valid only if you code POLLED=YES in this LINE macro (or the GROUP macro) and you code DIAL=NO in (or omit the DIAL operand from) the GROUP macro.

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

(nonswitched multipoint start-stop or BSC lines in network control mode only)

Specifies (1) the number of consecutive negative responses to polling the network control program will accept from a station before breaking the logical connection; (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 line operation in network control mode and 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.

This operand is valid only if the line is a nonswitched multipoint line (POLLED=YES is specified in this LINE macro or in the GROUP macro); DIAL=NO is specified in [or DIAL operand omitted from] the GROUP macro.

Note: For a switched line used to communicate with IBM 1050 terminals, the network control program accepts up to 16 consecutive negative responses to polling and then executes the WAIT option as described below.

```
  {WAIT  }
[ {NOWAIT} ]
  {QUEUE }
```

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 the line.

`NOWAIT`

Specifies that the network control program is to break the logical connection with the station, notify the host processor, and terminate the current read request.

`QUEUE`

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

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

(lines in network control mode only)

Specifies whether or not the stations attached to the line must be polled and addressed.

Note: All lines in an SDLC line group must be specified as POLLED=YES, or all must be specified as POLLED=NO. Both options cannot be included in the same line group.

Code POLLED=YES if:

- The line is a start-stop or BSC multipoint line (all types of stations attachable to a multipoint line require polling and addressing); or
- The line is a nonswitched or switched point-to-point line (including a multiple-terminal-access line)used to communicate with an IBM 1050 terminal; or
- The line is an SDLC link, if the controller in which the program being defined will be executed is the *primary* station on the link.

Exception: Do not code POLLED=YES if the controller in which the network control program will be executed is a tributary station on a BSC multipoint line (that is, if you have coded a tributary address character in the TADDR operand of this LINE macro).

Code POLLED=NO if:

- The line is a nonswitched or switched point-to-point, start-stop or BSC point-to-point line (unless the terminal is an IBM 1050); or
- The line is a multiple-terminal-access line (MTALIST operand of this LINE macro is coded) over which no IBM 1050 terminals will communicate with the controller; or

- The line is an SDLC link, if the controller in which the program is executed is the *secondary* station on the link. (POLLED=NO is valid only if the program is a remote NCP [TYPGEN=NCP-R], the link represented by this LINE macro is a nonswitched line [DIAL=NO], and the physical unit on the link is a type 4 unit [PU macro, PUTYPE=4 or INNODE macro represents the unit].)

If you omit this operand and the line is a start-stop or BSC line (LNCTL=SS, LNCTL=BSC), POLLED=NO is assumed.

If you omit this operand, the line is a nonswitched SDLC link (LNCTL=SDLC, DIAL=NO), and:

- this is the *first* LINE macro following the GROUP macro, then POLLED=NO is assumed if the program is to be executed in a remote controller (TYPGEN=NCP-R). Otherwise, POLLED=YES is assumed.
- this is *not* the first LINE macro following the GROUP macro, then the value assumed is the same as that specified in the first LINE macro.

```
[POLLTO={ERROR  } ]
        {NEGRESP}
```

(start-stop lines in network control mode only)

Specifies whether a timeout occurring during polling (when the line is in network control mode) is to be treated as an error condition or a negative response. If treated as an error condition, the value specified in the CRETRY operand of the GROUP macro determines the number of subsequent attempts at polling. If treated as a negative response, the value specified in the POLIMIT operand of this LINE macro (or in the GROUP macro) determines the number of times the station is re-polled.

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

(applicable to emulation mode only)

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

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

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

(switched BSC and start-stop lines in network control mode only)

Specifies how many times the network control program is to perform the dialing operation in attempting to reach a station over a switched line in network control mode. 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.)

This operand is valid only for a switched line (DIAL=YES in the GROUP macro) equipped with an automatic calling unit (AUTO operand of this macro) and only if the line is used for outgoing calls (CALL=OUT or CALL=INOUT is specified in this LINE macro [or in the GROUP macro]). The operand is not valid for a switched SDLC link.

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

(applicable to network control mode only)

Specifies the number of attempts, via retransmission, to recover from text errors in message data when the line is operating in network control mode.

Text-write errors: When an error occurs while sending to the station (text-write errors), the network control program retransmits the block on which the error occurred. If the error recurs, retransmission is repeated. The network control program persists in retransmitting until it successfully sends the block or until a maximum number of retries (retransmissions) has occurred. This maximum is specified as the *m* parameter of the RETRIES operand. *m* may be from 0 to 128, for SDLC stations, and 0 to 255, for start-stop and BSC stations. RETRIES=0 specifies no retry attempts, and is equivalent to specifying RETRIES=NONE. For start-stop and BSC stations, RETRIES=255 specifies unlimited attempts; that is, repeated retransmissions without limit. (Unlimited retries are not permitted for SDLC stations.)

Optionally, you may also specify that the network control program pause after completing the retry sequence, then begin a new retry sequence. This second sequence continues until the block is successfully transmitted or the maximum, *m*, is again reached. Alternation of retry sequence and pause continues until the error is cleared or the maximum number of retry sequences is reached. The pause, specified by the *t* parameter, may be from 0 to 255 seconds. The maximum number of retry sequences, specified by *n*, may be from 1 to 255, for start-stop and BSC stations, and from 0 to 127, for SDLC stations.

The maximum number of retries—$(n+1)*m$—for SDLC stations is 128. The maximum number of retries per sequence, *m*, is always specified in the RETRIES operand of the LINE macro.

For SDLC stations, you may specify the *t* and *n* parameters individually for each such station connected to the SDLC link represented by this LINE macro. (Specify these values in the RETRIES operand of the PU (or CLUSTER) macro, for SDLC cluster control units, or in the RETRIES operand of the PU (or INNODE) macro, for a remote communications controller.

The *t* and *n* parameters can be specified in a PU (or INNODE) macro associated with a LINE macro for a backup SDLC local-remote link (that is, a PU or INNODE macro from which the SUBAREA operand is omitted).

For start-stop and BSC stations, the *t* and *n* values must always be specified in the LINE macro; the same values apply to all stations connected to the line.

Text-read errors: For an error occurring when receiving from a station (text-read error), the network control program sends a negative response to the station, causing the station to retransmit the block in error. If the error recurs, the network control program again sends a negative response and receives the block again. The network control program persists in this way until (1) the block is successfully received, or (2) the station sends an EOT character or sequence instead of retransmitting the block, or (3) the network control program has sent a maximum number of negative responses. This maximum is specified as the *m* parameter of the RETRIES operand. *m* may be from 0 to 255. RETRIES=0 specifies no retry attempts, and is equivalent to RETRIES=NONE. RETRIES=255 specifies unlimited attempts; that is, no limit to the number of times the program sends a negative response to the station. Values from 1 through 254 indicate a specific number of retries.

The parameters t and n do not apply to text-read errors.

Maximum, Minimum, and Default Values:

The maximum number of retries (retransmissions) for text-write errors equals $(n+1)m$; the maximum number for text-read errors is m. (If m is 255, however, there is no maximum.)

If m is 0 or 255 (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 n is omitted and t is *not* zero: n=1 is assumed if $(n+1)\ n \leq 128$, and $n=0$ is assumed if $n+1\ n \geq 128$.

If n is omitted and t is 0, n=0 is assumed.

If you code RETRIES=NONE, no error recovery is attempted for read or write text errors.

If you omit this operand entirely, the assumed values will be $m=2$ (for start-stop stations) or 7 (for BSC and SDLC 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 two retries for start-stop stations, or seven retries, for BSC stations. 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 }
```

(switched lines only; not applicable to U.S. and Canada)

Specifies whether or not the ring indicator mode of automatic answer operation is to be used for the switched line represented by this LINE macro. This depends solely upon the type of modem (data set) that connects the line to the controller. Determine from the modem supplier or installer whether it has a "ring indicator interface" lead.

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

The RING operand is valid only for a switched line (DIAL=YES is specified, or (in a remote NCP only) DIAL=NO and CONFIG=SW are specified).

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

(telegraph lines in network control mode only)

Specifies whether or not the communication line represented by this LINE macro is attached to a type 2A line set (telegraph single current line set). SCLSET=YES is valid only for a start-stop line (LNCTL=SS is specified in [or LNCTL operand omitted from] the GROUP macro). If you omit this operand, SCLSET=YES is assumed for AT & T 83B3, WU 115A, and World Trade teletypewriter terminals; SCLSET=NO is assumed for all other terminals.

```
[SERVLIM=count]
```

(multipoint lines and SDLC links in network control mode only)

For start-stop or BSC lines: Specifies the number of entries in the service order table for this line that the network control program is to check each time the program performs service seeking. The maximum you may specify is 255.

If you omit this operand and this LINE macro represents a start-stop or BSC line (LNCTL=SS, LNCTL=BSC), the program checks one-half of the table entries each time it performs service seeking.

For SDLC links: Specifies the maximum number of regular scans of the service order table that the network control program will make for normal servicing of physical and logical units on the link before it makes a special scan of the table. The regular scan of the table accommodates normal transmission of path information units between the access method or host application programs and physical or logical units on the link. In the special scan of the service order table, the network control program determines whether there are any outstanding commands from the access method to interrogate or alter the on-line status of any physical units on the link. If so, the program fulfills the first such command, then resumes regular scans of the table to perform normal servicing. If no status commands are outstanding, the program immediately resumes regular scans unless, in the previous regular scan, the program found no stations to be in the "contacted" state—that is, no station is presently active. In this case, resumption of regular scans occurs after a delay of 2.2 seconds. If more than one status command is outstanding, only one is honored each time the special scan is made; the remaining status commands are fulfilled one at a time, in turn, during subsequent special scans of the table.

Upon completing a regular scan, the program begins the special scan when (1) in the regular scan just completed the program found no active stations, or (2) the maximum number of regular scans specified by SERVLIM has been reached—whichever occurs first.

Specifying a low value in SERVLIM gives the network control program more frequent opportunities to fulfill accumulated status commands than does specifying a higher value. Such status commands can be fulfilled more promptly, but at the cost of frequent interruptions to normal servicing. Conversely, specifying a higher value in SERVLIM causes fewer interruptions to normal servicing of stations than does a lower value, but delayed fulfillment of the status commands is more likely to result. The relative number of status commands the access method will issue for the link served by the service order table, the relative importance of the alternatives described above, and experience should influence your selection of a value for the SERVLIM operand.

Caution: The network control program will perform a timeout for any status command issued for the physical unit of an SDLC station whose power is off. The duration of this timeout interval is as specified in the REPLYTO operand of the GROUP macro.

Normal servicing of physical and logical units is interrupted during the timeout interval because regular scanning of the service order table is suspended. Only when the specified interval expires does regular scanning (and therefore normal servicing) resume. A low value for SERVLIM (which causes relatively frequent special scans) and/or a high value for REPLYTO (resulting in long timeout delays) will result in serious degradation of message throughput on the SDLC link if status commands are received for SDLC stations whose power is off. (The timeout interval recurs for each successive special scan.)

You may minimize the possibility of throughput degradation by specifying a high value in the SERVLIM operand and/or a low value in the REPLYTO operand.. Alternatively, the risk may be averted by arranging network operating procedures to avoid sending commands to physical units whose power is off.

If you omit this operand and this LINE macro represents an SDLC link (LNCTL=SDLC), a value of 4 (four regular scans of the table) is assumed.

This operand is valid only for a nonswitched start-stop, BSC or SDLC link (operating in network control mode) for which DIAL=NO and POLLED=YES are specified in the GROUP and LINE macros, respectively.

```
[SERVPRI={OLD} ]
         {NEW}
```

(multipoint start-stop and BSC lines in network control mode only)

Specifies whether the network control program is to give priority to servicing current sessions (SERVPRI=OLD) or to establishing new sessions (SERVPRI=NEW) on the line represented by this LINE macro.

This operand is valid only for a nonswitched line (operating in network control mode) on which multipoint line control is used (GROUP macro specifies DIAL=NO [or DIAL operand omitted] and this LINE macro [or the GROUP macro] specifies POLLED=YES).

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

(multipoint start-stop and BSC lines in network control mode only)

Specifies the number of sessions the network control program is to attempt to maintain concurrently on a nonswitched line (operating in network control mode) for which multipoint line control is required. 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 valid only for a nonswitched line on which multipoint line control is used (DIAL=NO [or operand omitted] is specified in the GROUP macro and POLLED=YES in this LINE macro [or the GROUP macro]). (This operand does not limit the number of sessions if clustered stations are attached to the line and general polling is used for soliciting input [GPOLL operand is specified in the CLUSTER macro]. The network control program cannot control the number of terminals that may respond to a general poll, and a separate session is established with each responding terminal.)

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

(lines operating in network control mode only)

Specifies, for a line connected to a modem capable of transmitting at either of two data rates, whether the data rate may be changed by request from the access method.

Specify SPDSEL=YES if you wish to allow the program to change the data rate of the modem upon request from the access method. SPDSEL=YES is valid only if external clocking is used on the line represented by this LINE macro (CLOCKNG=EXT is specified). The data rate specified in the SPEED operand must be the higher of the two data rates.

Do not specify SPDSEL=YES if the modem used on this line has only a single data rate.

Specify SPDSEL=NO (or omit the operand) if the modem has one data rate or if it has two data rates but the network control program is not to change the rate on request from the access method.

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

(teletypewriter lines operating in network control mode only)

Specifies whether or not the network control program is to react to space characters received from 83B3, 115A, or World Trade teletypewriter terminals as downshift characters, when the line is in network control mode. If you code SPSHIFT=YES, each space character received from a terminal causes the program to send all subsequent text characters to the host processor in their downshifted form.

If you code SPSHIFT=NO (or omit the operand), the program does not convert the text characters to their downshifted form, but instead sends them as they were received from the terminal.

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

(tributary controller on BSC line or remote controller on SDLC link only)

Specifies, if this program is to be executed in a tributary controller on a BSC line or a remote controller on an SDLC link, the one-character address you wish to assign to this controller.

If this program is to be executed in a tributary controller on a BSC line: Code TADDR=char, where *char* is the two-digit hexadecimal representation of a single character (in transmission code). The character specified must be the same as the polling character that is specified in the network control program for the controller that controls this line. If, for example, *A* is the polling character to which this triburary controller is to respond, you would specify *A* as the tributary address in this LINE macro, and in the other network control program you would specify *A* in the POLL operand of the appropriate TERMINAL macro. (See also the ADDR operand of the TERMINAL macro.)

Although the same character—*A*—is required in the TADDR and POLL operands in the respective programs, the way in which you specify the character may differ. In the POLL operand you code the hexadecimal representation of the *EBCDIC* letter A—POLL=C1. In the TADDR operand of this LINE macro you code the hexadecimal representation of the *transmission code* bit pattern. If the transmission code used on the line is EBCDIC, the hexadecimal representation is the same: POLL=C1, TADDR=C1. If, however, the transmission code is USASCII, you would code the USASCII bit pattern for the letter A in the TADDR operand: TADDR=41.

Note: The polling character you assign to the tributary controller must have a bit pattern in which the third bit from the left (bit 2) is zero. Any uppercase alphabetic character is either EBCDIC or USASCII (except EBCDIC letters S-Z) meets this requirement. Any other bit pattern that meets this condition can also be assigned, whether or not that bit pattern represents a character. For example, all of the following are valid polling characters for a tributary controller:

EBCDIC	*Bit Pattern*	
Character	*(binary)*	*(hex)*
A	1100 0001	C1
!	0101 1010	5A
(none)	0101 0100	54

USASCII	*Bit Pattern*	
Character	*(binary)*	*(hex)*
A	0100 0001	41
[	0101 1011	5B

If this program is to be executed in a remote communications controller on an SDLC link (TYPGEN=NCP-R is specified in the BUILD macro):

Code TADDR=char, where *char* is the two-digit hexadecimal representation of a single, EBCDIC SDLC station address (the address of the remote controller). You may assign as the address any bit configuration except hexadecimal 00 or FF. However, both the address you specify in the TADDR operand and the address you specify in the ADDR operand of the PU (or INNODE) macro (within the local NCP) representing the remote controller must be identical to the physical address entered into the IPL configuration data set in the diskette contained within the remote controller. (This address is entered by the customer engineer.)

This operand is valid only if POLLED=NO is specified in this LINE macro.

`[TERM=type]`

(start-stop and BSC lines only)

Specifies the type of station with which the network control program communicates over the line represented by this LINE macro. It must be one of the types listed in Figure 5-13.

This operand is required if the line is to be operated in emulation mode. If the line is to be operated only in network control mode, the type of station may be specified in the TERM operand of the TERMINAL macro, or in this TERM operand (of the LINE macro), or in the GROUP macro.

Note that in Figure 5-13 some types of stations are restricted to operation in network control mode. Stations of these types (IBM 3780 and System/7; MTA terminals) must not be specified in this TERM operand if the line is to operate in both network control and emulation mode. Stations that cannot be specified for line operation in emulation mode do not appear in Figure 5-13 because they can be specified only in a LINE macro for a line that operates only in emulation mode.

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

[1] Specify this type only if the line is to operate only in network control mode.

[2] Specify a System/7 (BSC version) as follows: (1) Specify TERM=SYS3 in the TERMINAL (or LINE or GROUP) macro if the line is to operate only in network control mode. (2) Specify TERM=SYS3 in the LINE (or GROUP) macro if the line is to operate in both network control and emulation modes (alternately). (3) Specify TERM=SYS7 in the LINE (or GROUP) macro if the line is to operate only in emulation mode.

Figure 5-13. Values for TERM Operand of LINE Macro Instruction

[TRANSFR=count]

(stations on multipoint lines operating in network control mode only)

For start-stop and BSC stations:

Specifies a limit on the number of buffers that the network control program will obtain to receive message text from a station before transferring 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 or end-of-transmission character (or until the sub-block limit you have specified in the CUTOFF operand of this LINE macro [or the GROUP macro] is reached). This operand applies only to line operation in network control mode.

For SDLC stations:

Specifies the maximum number of buffers that the network control program will obtain and fill with message data from an SDLC station before transferring them to the host processor. If no ending characters have been received from the station by the time the specified number of buffers are filled with message data, the network control program discards all the data received and sends a negative acknowledgment to the station. (The discarded data is *not* sent to the host processor.)

The minimum value of *count* is 1. The maximum value is the smaller of (1) 255 or (2) the result of multiplying the values of the MAXBFRU and UNITSZ operands of the HOST macro, subtracting the value of the BFRPAD operand of the HOST macro, and dividing the result by the buffer size specified in the BFRS operand of the BUILD macro. Expressed as a formula:

$$\text{count} \leq \frac{\text{(MAXBFRU)(UNITSZ) - BFRPAD}}{\text{BFRS}}$$

If you omit this operand, the network control program uses the lesser of (1) 255 or (2) the result of the calculation expressed by the preceding formula.

```
      {EP }
[TYPE={NCP}]
      {PEP}
```

(start-stop and BSC lines only)

Specifies whether the line represented by this LINE macro is to operate in network control mode only (TYPE=NCP) or alternately in either network control mode or emulation mode (TYPE=PEP).

The valid choices for this operand depend upon (1) whether the program can perform only network control functions, or only emulation functions, or both (as specified in BUILD: TYPGEN), and (2) whether the line group to which the line belongs is defined in the GROUP macro as operable in network control mode, or in emulation mode, or alternately in either mode, as follows:

If...	*And...*	*Then valid choices are...*
BUILD: TYPGEN=	**GROUP: TYPE=**	**LINE: TYPE=**
NCP or NCP-LR or NCP-R	NCP	NCP
EP	EP[1]	EP[1,2]
PEP or PEP-LR	PEP[1]	PEP[1] NCP EP[1,2]
	NCP	NCP
	EP[1]	EP[1,2]

[1] Not valid if LINE macro represents an SDLC link.

[2] If the line is always to operate in emulation mode, refer to the next section, *Macro Instructions for Operation in Emulation Mode Only*, rather than the present section.

Default values: If you omit the TYPE operand from the LINE macro, the value assumed is the same as specified in the TYPE operand of the GROUP macro.

If you specify TYPE=PEP, all valid operands of this LINE macro may be specified. If you specify TYPE=NCP, only the valid operands for network control mode should be coded. Ignore any operands labelled as applicable to emulation mode only. (If you specify such operands, the generation procedure will ignore them.)

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

(applicable to emulation mode only)

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

It is normally appropriate to specify UNITXC=YES (or omit the operand), which causes the program to signal Unit Exception status upon receiving an EOT.

However, if read and write commands within the access method are command chained, UNITXC=NO may be appropriate. UNITXC=NO, by suppressing the Unit Exception indication, prevents the command chain from being broken. (Unit Exception status always breaks the command chain.)

```
[USE={NCP} ]
     {EP }
```

(applicable only to lines operable alternately in network control and emulation modes)

Specifies whether the line represented by this LINE macro is to operate initially (that is, when the line is first activated after the program is loaded into the controller) in network control mode (USE=NCP) or emulation mode (USE=EP).

If TYPE=PEP is specified in this macro, either USE=NCP or USE=EP is valid; if you omit the USE operand, USE=NCP is assumed—that is, the line will initially operate in network control mode.

Switching a line from one mode to another is accomplished by a request from the access method sent to the network control program over the network control subchannel.

This operand has no meaning unless TYPE=PEP is specified for this line.

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

(BSC lines in network control mode only)

Specifies whether or not the controller is to be the secondary station on a nonswitched point-to-point (contention) line. (The secondary station yields to the primary station when contention occurs.) This operand applies only to line operation in network control mode.

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.

This operand is valid only for a BSC point-to-point line (GROUP macro specifies LNCTL=BSC, LINE [or GROUP] macro specifies POLLED=NO [or POLLED operand is omitted]).

Specifying Lower-Level Operands in a Higher-Level Macro

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 5-9 shows which of the lower-level operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the LINE macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this LINE macro description means only that the NCP generation assembly procedure accepts them as valid operands of the LINE macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

CLUSTER Macro Instruction

The CLUSTER macro represents either a BSC station of the "clustered" type (IBM 3271, 3275, or 2972) or an SDLC station of the clustered type.

BSC Clustered Stations:

For a BSC clustered station, the CLUSTER macro specifies:

- The type of station.
- Whether the station has the Batched Message Input feature (2972 only).
- The general polling characters of the station, if required.
- The block handler set, if any, associated with the station, and the points of execution of block handlers within the set.
- Certain procedural options the network control program is to use when communicating with the station.

Use a CLUSTER macro to represent an IBM 3271, 3275, or 2972 only if the line is a nonswitched multipoint line (DIAL=NO is specified in [or omitted from] the GROUP macro and POLLED=YES is specified in the LINE [or GROUP] macro.

The following operands (exclusive of VTAM-only operands) are valid in a CLUSTER macro representing a BSC clustered station:

BHEXEC
BHSET
CDATA
CUTYPE
FEATURE
GPOLL
INHIBIT
ITBMODE
LGRAPHS
PT3EXEC
XMITLIM

Note: In version 3 of the network control program, SDLC type 2 physical units are represented by CLUSTER macros. For compatibility with version 3, versions 4 and 5 accept CLUSTER macros and for this reason, operands pertaining to SDLC stations are shown in this CLUSTER macro description. However, when defining a new source program to be generated under NCP version 4 or 5, or when adding new SDLC stations to an existing version 3 source program to be regenerated under version 4 or 5, use PU macros, not CLUSTER macros, to represent the SDLC stations.

SDLC Clustered Stations:

For an SDLC clustered station, the CLUSTER macro specifies:

- The type of station (SDLC1).
- The address of the SDLC station.
- Whether immediate polling retry is required.
- The maximum number of path information units the network control program will send to the SDLC station (physical unit) before awaiting a response from the station.
- The maximum number of consecutive path information units or PIU segments the network control program will send at one time to the station.
- The number of error recovery attempts, via retransmission, the network control program will make when transmission errors occur.

The following operands (exclusive of VTAM-only operands) are valid in a CLUSTER macro representing an SDLC clustered station:

ADDR
CUTYPE
IRETRY
MAXDATA
MAXOUT
PASSLIM
RETRIES

Do not code CLUSTER macros following a LINE macro if any of the following is true:

- The line is a start-stop line (LNCTL=SS is specified in the GROUP macro).
- The line is an SDLC link (LNCTL=SDLC) and the program being defined will be executed in a remote communications controller (TYPGEN=NCP-R is specified in the BUILD macro).
- The line operates only in emulation mode (TYPE=EP is specified in the LINE macro).
- The line is a switched line (DIAL=YES is specified in the GROUP macro).
- The line does not use multipoint line control (POLLED=NO is specified in the LINE macro).

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Each CLUSTER macro coded causes a resource name to be generated only if the GPOLL operand is coded. (Omit the GPOLL operand if the general polling function is not required for this station, as its omission causes a saving in controller storage.)

All operands of the CLUSTER macro apply only to line operation in network control mode.

```
Name      Operation      Operands

symbol    CLUSTER        For BSC stations:

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

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

                         [,CDATA={YES}]
                                 {NO }

                                  {2972 }
                         [,CUTYPE={3271 }]
                                  {3275 }

                         [,FEATURE={BATCH  }]
                                   {NOBATCH}

                         [,GPOLL=chars]

                         [,INHIBIT={( [WACKCNT] [,SUBBLOCK] [,ERPR] [,ERPW] )}]
                                   {NONE                                     }

                         [,ITBMODE=( [{YES}] [,{YES}] )]
                                      {NO }     {NO }

                         [,LGRAPHS=( [{REJECT}] [,{REJECT}] )]
                                      {ACCEPT}     {ACCEPT}
```

```
Name     Operation     Operands

                       [,PT3EXEC={YES}]
                                 {NO }

                       [,XMITLIM={count}]
                                 {NO   }

                       For SDLC stations:

                       CUTYPE=SDLC1

                       [,ADDR=chars]

                       [,IRETRY={YES}]
                                {NO }

                       [,MAXDATA=size]

                       [,MAXOUT={n}]
                               {1}

                       [,PASSLIM={n}]
                                 {1}

                       [,RETRIES=([t[,n]])]

                       VTAM-only operands:

                       FEATUR2=
                       ISTATUS=
                       LOGAPPL=
                       LOGTAB=
                       VPACING=
```

`symbol`

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

Operands for BSC Stations

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

(BSC stations only)

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 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.

The section, *Block Handling Definition (BH) Macro Instructions* explains how to define a block handler set.

Code this operand only if you specify BHSET=setname. (If you specify BHSET=setname, you must specify at least one execution point.) If you specify BHSET=NONE or BHSET=DYNAMIC, do not specify an execution point.

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

(BSC stations only)

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.

`DYNAMIC`

Specifies that no block handler set is to be initially assigned to this station but one may be assigned dynamically by command from the host processor.

`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.

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

(BSC stations only)

Specifies whether or not the data sent to and received from the station represented by this CLUSTER macro is "critical" (for security reasons). If you code CDATA=YES, the network control program automatically clears all buffers containing data associated with the terminal before returning them to the buffer pool. If you code CDATA=NO (or omit the operand), the program does not clear the buffers. CDATA=YES is valid only if you specify ERASE=YES in the BUILD macro.

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

(BSC stations only)

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

Note: If the line to which 2972, 3271, or 3275 stations are connected is to operate in eumulation mode as well as in network control mode, specify the CUTYPE operand in the LINE (or GROUP) macro instead of the CLUSTER macro.

Specifying Polling and Addressing Characters for IBM 2972, 3271, 3275:

IBM 2972: Code the general polling characters in the GPOLL operand of this CLUSTER macro. In addition, directly following this macro, code a TERMINAL macro for each terminal address available on the control unit, from the lowest address to the highest address to which a terminal is attached (regardless of whether some intermediate addresses are unused). For example, if terminals were attached to the third, fourth, eighth, and ninth addresses, you would code nine TERMINAL macros; the first representing the first (lowest) address, the last representing the ninth address.

In the ADDR operand of each TERMINAL macro representing a 2980, code 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 3271: If general polling is required, code the general polling character in the GPOLL operand of this CLUSTER macro. In addition, directly following this macro, code a TERMINAL macro for each terminal address available on the control unit, from the lowest address to the highest address to which a terminal is attached (regardless of whether some intermediate addresses are unused). For example, if terminals were attached to the third, fourth, eighth, and ninth addresses, you would code nine TERMINAL macros; the first representing the first (lowest) address, the last representing the ninth address.

In the POLL and ADDR operands of each TERMINAL macro, code the polling and addressing characters assigned to the device. (Determine the characters from the system designer.)

If a 3277 is used as an input-only device and you specify general polling characters in the GPOLL operand of this CLUSTER macro, you may omit the ADDR and POLL operands from the TERMINAL macro representing that 3277.

IBM 3275: If general polling is required, code the general polling characters in the GPOLL operand of this macro. In addition, directly following this CLUSTER macro, code a single TERMINAL macro in which TERM=3275 is specified. If the 3275 is to be individually polled and addressed, specify the polling and addressing characters in the POLL and ADDR operands of the TERMINAL macro. If you specify addressing characters in the ADDR operand, you must also code polling characters in the POLL operand (the reverse is not true).

```
[FEATURE={BATCH  }]
         {NOBATCH}
```

(IBM 2972 only)

Specifies whether or not the station has the batched message input feature (2972 only). If you specify FEATURE=BATCH, the network control program does not deblock messages from the 2972 and therefore does not identify the individual 2980 terminals from which each message was received.

```
[GPOLL=chars]
```

(BSC stations only)

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. (The GPOLL operand must be coded if the control unit represented by this CLUSTER macro is a 2972 (CUTYPE=2972), since individual polling of terminals attached to a 2972 is not possible.)

This operand is required if this CLUSTER macro represents an IBM 3271.

```
[INHIBIT={([WACKCNT][,SUBBLOCK][,ERPR][,ERPW])}]
         {NONE                               }
```

(BSC stations only)

Specifies which, if any, of the network control program facilities are to be inhibited from functioning initially—that is, when the program begins execution after being loaded into the controller.

```
[WACKCNT]
```

Specifies that the WACK limit specified by the WACKCNT operand of the GROUP macro is to be inhibited—that is, the first WACK received from a the station causes the network control program to return the request to the host processor rather than to respond to that and subsequent WACKs with an ENQ character. This parameter is valid only if LNCTL=BSC is specified in the GROUP macro.

```
[SUBBLOCK]
```

Specifies that the sub-blocking indicated by the TRANSFR operand of the LINE (or GROUP) macro is to be inhibited—that is, if the number of buffers specified by the TRANSFR operand are filled by received text, the network control program terminates the receiving operation just as if the cutoff limit (specified by the CUTOFF operand) were reached.

```
[ERPR]
```

Specifies that recovery procedures for text-read errors are to be inhibited.

```
[ERPW]
```

Specifies that recovery procedures for text-write errors are to be inhibited.

NONE

Specifies that none of the network control program facilities are to be inhibited; that is, all will become effective when execution of the network control program begins.

```
[ITBMODE=( [{YES}] [,{YES}] )]
           {NO }     {NO }
```

(BSC stations only)

Specifies how the network control program is to handle ITB characters in text received from the station represented by this CLUSTER macro.

The first suboperand specifies whether the program is to insert an EIB (error information block) character following each ITB character received from the station.

The second suboperand specifies whether or not each ITB character received from the access method is followed by an EIB character. If you specify YES, the network control program removes from the message data it sends to the station the first character following each ITB character received from the access method.

```
[LGRAPHS=( [{REJECT}] [{REJECT}] )]
            {ACCEPT}    {ACCEPT}
```

(BSC stations only)

Specifies whether leading graphics received from the station are to be accepted by the network control program or rejected (that is, treated as an error condition).

The first parameter specifies acceptance or rejection of leading graphics for read operations; the second parameter specifies acceptance or rejection for write operations.

```
[PT3EXEC={YES}]
         {NO }
```

(BSC stations only)

Specifies whether or not a block handler set that is executed at point 3 (see the BHSET macro) is to be associated with the station represented by this CLUSTER macro.

This operand is ignored if you omit the BHSET macro, or if you code BHSET=NONE, BHEXEC=PT3, or BHEXEC=ALL in this CLUSTER macro.

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

(BSC stations only)

Specifies the maximum number, if any, of transmissions 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 suspend the session.

The maximum value you may specify is 255; the minimum is 1.

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

XMITLIM=count is valid only for stations associated with a line for which you have specified a service order table (that is, a SERVICE macro follows the LINE macro).

Operands for SDLC Stations

CUTYPE=SDLC1

Specifies that the control unit of the station is an SDLC cluster controller, (3600, 3650, 3660 only).

CUTYPE=SDLC1 is valid only if the GROUP macro assicated with this CLUSTER macro represents a line group comprising SDLC links (LNCTL=SDLC is specified in the GROUP macro). If you specify LNCTL=SDLC in the GROUP macro and you omit the CUTYPE operand from this cluster macro, CUTYPE=SDLC1 is assumed.

(ADDR=chars]

(SDLC clustered stations only)

Specifies the hexadecimal representation of the eight-bit address of the SDLC clustered station represented by this CLUSTER macro (CUTYPE=SDLC1 must be specified in this macro).

This operand is required if the station is on an SDLC link (LNCTL=SDLC is specified in the GROUP macro for the line group).

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

(SDLC stations only)

Specifies whether or not the network control program, when an "idle detect timeout" condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program re-polls the control unit; otherwise, the program bypasses the unsuccessfully polled control unit and services the station represented by the next entry in the service order table.

This operand is valid only for SDLC stations (LNCTL=SDLC is specified in the GROUP macro).

[MAXDATA=size]

(SDLC stations only)

Specifies the maximum amount of data, in bytes (including the transmission header and request/response header), that the SDLC control unit can receive in one data transfer–that is, one entire PIU or a PIU segment.

To determine the amount of data that the physical unit can receive, consult the publications for the specific type of SDLC station represented by this CLUSTER macro.

The maximum amount of user data the network control program sends to the physical unit in one data transfer is the value of MAXDATA minus nine bytes (three bytes for the request/response header, and six bytes for the transmission header).

This operand is valid only for an SDLC station (LNCTL=SDLC is specified in the GROUP macro).

If you omit this operand, and the station is an SDLC control unit, the program sends the station a maximum of one network control program buffer, plus six bytes.

```
[MAXOUT={n}]
        {1}
```

(SDLC stations only)

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the SDLC cluster control unit represented by this CLUSTER macro before requesting a response from the control unit. The minimum is one; the maximum is seven. The network control program will send one PIU or segment if you omit this operand.

This operand is valid only for an SDLC station (LNCTL=SDLC is specified in the GROUP macro).

```
[PASSLIM={n}]
        {1}
```

(SDLC stations only)

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the SDLC cluster control unit represented by this CLUSTER macro for any one appearance of the control unit in the service order table. The minimum is one PIU or segment; the maximum is 254.

This operand is valid only for an SDLC station (LNCTL=SDLC is specified in the GROUP macro).

If you omit this operand and the station is an SDLC cluster control unit, the program sends one PIU (or segment) at a time.

```
[RETRIES=([,t[,n]])]
```

(SDLC stations only)

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts—via retransmission—to recover from text errors in message data sent to the SDLC cluster control unit specified by this CLUSTER macro.

The meanings of the *t* and *n* parameters are the same as for those parameters in the RETRIES operand of the LINE macro.

Specifying these parameters in the CLUSTER macro allows you to designate different values for each cluster control unit attached to the line. If this CLUSTER macro represents a BSC station, you must specify *t* and *n* in the RETRIES operand of the LINE macro.

If you specify RETRIES=NONE in the LINE macro for the SDLC link to which the SDLC cluster control units are connected, no error recovery is attempted for read-text or write-text errors.

Specifying Lower-Level Operands in a Higher-Level Macro

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 5-9 shows which of the TERMINAL macro operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the CLUSTER macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this CLUSTER macro description means only that the NCP generation assembly procedure accepts them as valid operands of the CLUSTER macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

PU Macro Instruction

The PU macro represents a physical unit of any type (1, 2, or 4) with which the network control program communicates over a nonswitched or switched SDLC link.

The PU macro specifies:

- The station address of the physical unit.
- The physical unit type (1, 2, or 4).
- The maximum amount of data the physical unit can receive from the network control program in one data transfer.
- The maximum number of consecutive path information units (PIU) or PIU segments the network control program will send to the physical unit before servicing other physical units on the link.
- Whether immediate polling retry is required.
- The number of error recovery attempts the network control program will make when transmission errors occur.
- The maximum number of path information units or PIU segments the network control program will send to the physical unit before requesting a response from the unit.
- The subarea address of the physical unit (type 4 only).

The sequence of PU macros (with intervening LU macros, if required) must appear directly following the SERVICE macro that defines the service order table for the SDLC link:

```
LINE POLLED=YES
SERVICE
PU
LU
LU
PU
LU
LU
 .
 .
 .
```

If the local and remote controllers (physical unit type 4) can communicate over one or more backup (alternate) SDLC links in lieu of the principal link, a PU macro must appear after each LINE (or SERVICE) macro corresponding to an alternate link. Omit the SUBAREA operand from each of these alternate PU macros. Also, if the alternate PU macro is coded in a *local* network control program (and thus represents the remote controller} omit all other PU macro operands except IRETRY and RETRIES; or, if the alternate PU macro is coded in a *remote* network control program (and thus represents the local controller), only the MAXOUT and PASSLIM operands are valid; they should specify the same values as these operands specify in the *principal* PU macro.

Exception: If the PU macro appears in the network control program executed in the controller that is the secondary station on the SDLC link (and the PU macro thus represents the primary station on the link), no SERVICE macro is used. The sequence then is simply:

```
LINE
PU
```

Example:

Local communications controller: *Remote communications controller:*

Local NCP (subarea 4) Remote NCP (subarea 5)

```
L1    LINE  POLLED=YES,    LA    LINE  POLLED=NO
            RETRIES=5      LCI   PU    SUBAREA=4,
                                       PUTYPE=4,
                                       DATMODE=FULL,
      SERVICE ORDER=(RC1)              MAXOUT=7,
                                       PASSLIM=200
RC1   PU    SUBAREA=5,
            PUTYPE=4,
            DATMODE=FULL,  LAALT LINE  POLLED=NO
            MAXOUT=7,
            PASSLIM=200,   LC1ALTPU    PUTYPE=4,
            RETRIES=(10,5)             MAXOUT=7,
                                       PASSLIM=200
L1ALT LINE  POLLED=YES,
            RETRIES=5

      SERVICE ORDER=(RC1ALT)

RC1ALTPU    PUTYPE=4,
            RETRIES=(10,5)
```

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only for the VTAM initialization procedure and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each PU macro coded causes a resource name to be generated.

```
Name       Operation    Operands

symbol     PU           [ADDR=chars]

                        [,BNNSUP=3270]

                        [,DATMODE={HALF}]
                                  {FULL}

                        [,IRETRY={YES}]
                                 {NO}

                        [,MAXDATA=size]

                        [,MAXLU=count]

                        [,MAXOUT={n}]
                                 {1}

                        [,PASSLIM={n}]
                                  {1}

                                 {1    }
                        [,PUTYPE={2    }    ]
                                 {4    }
                                 {(1,2)}

                        [,RETRIES=(,[t][,n])]

                        [,SUBAREA=n]

                        VTAM-only   operands:
                        BUFLIM=
                        DISCNT=
                        ISTATUS=
                        LOGAPPL=
                        LOGTAB=
                        MODETAB=
                        SSCPFM=
                        USSTAB=
                        VPACING=
```

`symbol`

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

[ADDR=chars]

Specifies the hexadecimal representation of the eight-bit address of the physical unit represented by this PU macro. Any bit configuration except hexadecimal 00 or FF is valid. This operand is invalid if (1) POLLED=NO is specified in the LINE macro; (2) POLLED=YES and PUTYPE=4 are specified but the SUBAREA operand is omitted; or (3) DIAL=YES is specified in the GROUP macro.

This operand is required if the communication facility is nonswitched (DIAL=NO), the physical units on the link must be polled (POLLED=YES), and the physical unit type is 1, or 2, or 4 (and the SUBAREA operand is specified).

Note: For a type 4 physical unit, the address specified in the ADDR operand in the *local* network control program and the address specified in the TADDR operand of the LINE macro within the remote network control program must be identical to the physical address of the remote controller. (The customer engineer enters the physical address into the IPL configuration data set in the diskette contained within the remote controller.)

[BNNSUP=3270]

Specifies that the physical unit represented by this PU macro is a 3270 (operating in SDLC mode).

This operand is invalid if the line is switched (DIAL=YES) or if the physical unit type is not 1.

[DATMODE={HALF}]
{FULL}

(nonswitched link only)

Specifies, for a communications controller (type 4 physical unit) on a nonswitched SDLC link that allows simultaneous transmission in both directions, whether the network control program is to communicate with the distant controller in half-duplex or duplex data mode. This operand is valid only if the ADDRESS operand of the corresponding LINE macro specifies both a receive and a transmit address and the LINE macro represents the principal (not backup) SDLC link.

If you (1) omit this operand, (2) code the SUBAREA operand, and (3) specify, in the LINE macro, both a receive and a transmit address, then DATMODE=FULL is assumed.

This operand is invalid if (1) the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R), or (2) the program is to be executed in a local controller (TYPGEN=NCP-LR) and the SUBAREA operand is omitted from this macro, or (3) the physical unit type is not 4.

[IRETRY={YES}]
{NO}

Specifies whether or not the network control program, when an 'idle detect timeout' condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program re-polls the controller; otherwise, the program services the station represented by the next entry in the service order table.

This operand is valid only if POLLED=YES is specified in the LINE macro. If you omit this operand and the LINE macro specifies POLLED=YES, IRETRY=NO is assumed.

Note: You may specify IRETRY in a PU macro associated with either the principal or the backup SDLC link.

`[MAXDATA=size]`

(type 1 and 2 physical units only)

Specifies the maximum amount of data, in bytes (including the transmission header and request/response header), that the physical unit can receive in one data transfer—that is, one entire PIU or a PIU segment.

To determine the amount of data that the physical unit can receive, consult the publications for the specific type of SDLC station represented by this PU macro.

The maximum amount of user data the network control program sends to the physical unit in one data transfer is the value of MAXDATA minus five bytes (for a type 1 physical unit) or minus nine bytes (for a type 2 physical unit). (These values represent the lengths of the request/response header (three bytes) and the transmission header (two bytes, for a type 1 physical unit; six bytes for a type 2 physical unit.)

The value you specify for MAXDATA, divided by the NCP buffer size (specified in the BFRS operand of the BUILD macro), determines the number of NCP buffers sent in one data transfer to the physical unit represented by this PU macro.

This operand is valid only for a nonswitched SDLC station of physical unit type 1 or 2.

If you omit this operand, BFRS+2 is assumed as the maximum amount of data for PUTYPE=1 and BFRS+6 is assumed for PUTYPE=2.

`[MAXLU=count]`

(switched link only)

Specifies, for a physical unit on a switched link (DIAL=YES), the maximum number of logical units associated with any physical unit that can communicate with the network control program over the link. The minimum is 1; the maximum is 255.

This operand is required if the physical unit is on a switched link (DIAL=YES); otherwise it is invalid.

```
[MAXOUT={n}]
        {1}
```

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the physical unit represented by this PU macro before requesting a response from the physical unit. The minimum is one; the maximum is seven.

If you omit this operand and specify DIAL=NO, (1) MAXOUT=1 is assumed if you specify POLLED=YES or (2) MAXOUT=7 is assumed if you specify POLLED=NO.

```
[PASSLIM={n}]
         {1}
```

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the physical unit represented by this PU macro.

If this physical unit is associated with a LINE macro in which POLLED=YES is specified, the program services the station represented by the next entry in the service order table when the pass limit value is reached.

If this PU macro is coded in a remote network control program and represents a local communications controller, the remote program stops sending to the local controller when the pass limit is reached. Transmission to the local controller resumes when the local network control program again polls the remote controller.

The minimum is one PIU or segment. The maximum is 254. The default value assumed if you omit this operand is 1 (PASSLIM=1).

If this PU macro appears in a remote network control program and represents a local communications controller, the maximum value is 254. This is also the recommended value and the default value assumed if you omit this operand.

```
         {1     }
[PUTYPE={2     }     ]
         {4     }
         {(1,2)}
```

Specifies the physical unit type constituting the SDLC station represented by this PU macro. PUTYPE=(1,2) specifies, only for a group of switched SDLC links, that physical units of either type 1 or type 2 can communicate with the communications controller over the same link.

PUTYPE=1 and PUTYPE=2 are valid only for polled links (POLLED=YES).

PUTYPE=1 and PUTYPE=2 are valid for either nonswitched or switched links (DIAL=NO or DIAL=YES); PUTYPE=4 is valid only for nonswitched links (DIAL=NO).

`[RETRIES=( [,t] [,n] )]`

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts—via retransmission—to recover from text errors in message data sent to the physical unit represented by this PU macro.

The meanings of the *t* and *n* parameters are the same as for these parameters in the RETRIES operand of the LINE macro.

`[SUBAREA=n]`

(nonswitched SDLC link only; specified in local NCP only)

Specifies the subarea address assigned to the network control program in the remote communications controller (physical unit type 4) represented by this PU macro. (The network control program in each controller in the network must have a unique subarea address.) The value of *n* specified in this SUBAREA operand and in the SUBAREA operand of the BUILD macro for the remote NCP must be identical.

This operand is invalid for a switched SDLC link or a remote network control program.

The minimum valid subarea address is 1—that is, SUBAREA=1. The maximum address is the value you specify in the MAXSUBA operand of the BUILD macro.

If this PU macro and its associated LINE macro represent the principal SDLC link to a physical unit type 4 (remote communications controller), SUBAREA must be specified.

If this PU macro and its associated LINE macro represent a *backup* SDLC link to a remote controller, omit the SUBAREA operand and all other operands except PUTYPE, IRETRY, and RETRIES. (When communicating with the remote controller over the backup link, the network control program uses the same values as specified in the PU macro for the principal link except for the values of PUTYPE, IRETRY, and RETRIES.)

(At least one PU [or INNODE] macro in the source program must specify the SUBAREA operand if TYPGEN=NCP-LR is specified.)

Note: A backup SDLC link can be defined only if the line constituting the link is serviced by a type 2 (not type 1) communication scanner (TYPE=TYPE2 is specified in the CSB macro).

Specifying Lower-Level Operands in a Higher-Level Macro

In addition to the preceding operands, some operands of the LU macro can be specified in the PU (or LINE or GROUP) macro instead of the LU macro. Similarly, most operands of the PU macro can be specified instead in the LINE or GROUP macro. Figure 5-9 shows which of the lower-level operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the PU macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

LU Macro Instruction

The LU macro instruction represents a logical unit associated with an SDLC station (type 1 or 2 physical unit) attached to a nonswitched SDLC link, and specifies:

- The local address of the logical unit.
- Use of the pacing option.
- Whether data transfer from VTAM to the logical unit is in batch mode (VTAM users only).
- The BSC station start is to be associated with the logical unit, when the SDLC/BSC path function is to be used.

Each logical unit associated with a type 1 or 2 physical unit on a nonswitches SDLC link must be represented by a separate LU macro instruction. The sequence of LU macros must immediately follow the PU (or CLUSTER) macro representing the physical unit. The sequence must be in ascending order of local addresses assigned to logical units (as specified in the LOCADDR operand of this macro). That is, the LU macro specifying the lowest address must appear first, following the PU (or CLUSTER) macro, and the LU macro specifying the highest address must appear last. (LU macros are not required for any local addresses with which no logical units are associated; however, the generation procedure generates logical units for each local address not defined, up to the highest address for which there is an LU macro.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information for the VTAM initialization procedure and are not required (though are permissible) in the card deck used as input for the NCP generation procedure.

Each LU macro coded causes a resource name to be generated.

Name	*Operation*	*Operands*
symbol	LU	LOCADDR=n [,BATCH={YES}] {NO } [,DATASW=termname] [,PACING={(n[,m])}] {1,1 } *VTAM-only operands:* BUFLIM= ISTATUS= LOGAPPL= LOGTAB= MODETAB= SSCPFM= USSTAB= VPACING=

symbol

Provides a name for the logical unit and is required. *symbol* may be any valid assembler-language symbol; the first character may not be $. (This name must not be specified in the service order table.)

```
LOCADDR=n
```

Specifies the local address (in decimal, without leading zeros) of the logical unit. For type 1 physical units, the minimum is 1; the maximum is 255.

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

(VTAM users only)

Specifies whether the data transfer from VTAM to the logical unit is in batch mode. (Lower data transfer priority is provided for transfers in batch mode than for interactive mode.)

```
[DATASW=term name]
```

(SDLC/BSC path function only; VTAM users only)

Specifies the name of the TERMINAL macro representing the BSC station to which the network control program is to directly send data PIUs originated by the logical unit. Specifying this operand invokes the SDLC/BSC path function by which data PIUs (other than control and error messages) originating at the logical unit are sent directly to a BSC station rather than to VTAM. (Control and error message PIUs originating at the logical unit are sent to VTAM.)

Note: If you specify this operand, you must also specify the SPAFPT 3 block-handling macro instruction for the BSC station (or assign to the BSC station a user block-handling routine that performs the equivalent function. See the description of the SPAFPT3 macro. Also see the *SDLC/BSC Path Function System Programmer's Guide*, GC30-3029.

```
[,PACING={(n[,m] )}]
         {1,1     }
```

Specifies whether the network control program is to pace the logical unit; that is, whether while sending to the logical unit the program is to require the logical unit to acknowledge, at intervals, receipt of the message data and its ability to accept more data.

n

Specifies the number of requests the network control program is to send to the logical unit before stopping transmission to await a pacing response from that unit. The minimum is 1; the maximum is 255.

m

Specifies in which of the *n* requests the network control program is to turn on the pacing bit. The minimum is 1 (that is, the first request); the maximum is the value specified for *n*.

If you omit *m*, the network control program turns on the pacing bit in the last (*n* th) request sent.

0

Specifies that the network control program is not to pace the logical unit.

If the logical unit is associated with a type 1 physical unit (PUTYPE=1), PACING=(1,1) must be specified (or the PACING operand omitted).

Note: Pacing applies only to normal (synchronous) requests.

TERMINAL Macro Instruction

The TERMINAL macro represents a start-stop or BSC 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.

Some of the items in the preceding list apply only to certain types of stations. The descriptions of the individual operands give the conditions under which these items are specified.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Each start-stop and each BSC station attached to a nonswitched point-to-point or multipoint line must be represented in the network control program by a separate TERMINAL macro.

Stations that will call the communications controller over the switched telephone network are not individually represented by TERMINAL macros. Instead, a TERMINAL macro is required for each of the controller's connections ("ports") to the switched network that is to be used for receiving calls. The control blocks generated by each TERMINAL macro are used to represent whichever station is connected to the controller for the duration of any given call.

Stations that will be called by the controller, on the other hand, may be represented either by individual TERMINAL macros or by a common TERMINAL macro that represents whichever station the controller is connected to during any given call.

For each station whose telephone number you wish to be maintained within the controller, code an individual TERMINAL macro. In the DIALNO operand of the macro specify the telephone number of the station.

Code a single TERMINAL macro to represent all stations for which VTAM will supply the telephone number.

Each TERMINAL macro coded causes a resource name to be generated.

All operands of the TERMINAL macro apply only to line operation in network control mode.

If the station for which you code a TERMINAL macro is an IBM 3275 or a device attached to an IBM 2972 or 3271 control unit, the 2972, 3271, or 3275 must be represented by a CLUSTER macro. (See the explanation of the CUTYPE operand of the CLUSTER macro for requirements regarding use of TERMINAL macros.) Do not code a TERMINAL macro for a printer attached to an IBM 3275 (TERM=3275).

Note: SDLC stations are represented by PU macros rather than by TERMINAL macros.

```
Name     Operation    Operands

symbol   TERMINAL     TERM=type

                              {addr chars                               }
                      [,ADDR={selection chars                           }]
                              {([normal addr chars][,alt addr chars])}

                      [,ATTN={ENABLED }]
                              {DISABLED}

                      [,BFRDLAY=count]

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

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

                      [,CDATA={YES}]
                               {NO }

                      [,CONV={YES}]
                              {NO }

                      [,CRDLAY={YES}]
                                {NO }

                      [,CRITSIT={YES}]
                                 {NO }

                      [,CTERM={YES}]
                               {NO }

                                 {n  }
                      [,CUIDLEN={TWX}]
                                 {ALL}
                                 {0  }

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

                      [,DIALSET={dialset name}]
                                 {NONE        }

                                 {IN   }
                      [,DIRECTN={OUT  }]
                                 {INOUT}

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

Name	*Operation*	*Operands*

```
[,FANOUT=terminal name]

[,FEATURE=( [{ACR  }] [,{ATTN  }]
             {NOACR}     {NOATTN}

             [,{BREAK  }] [,{CHECK  }]
               {NOBREAK}     {NOCHECK}

             [,{SCTL  }] [,{TOSUPPR}]
               {NOSCTL      {NOTOSUP}

             [,{XCTL  }] )]
              {NOXCTL}

         {symbol}
[,IDSEQ={NONE  }]
         {PASS  }
         {IGNORE}

[,INHIBIT= {( [TEXTTO] [,TIMEFILL] [,WACKCNT]   ]
             [,SUBBLOCK] [,ERPR] [,ERPW] )}

             {NONE}

[,ITBMODE=( [{YES}] [,{YES}] )]
             {NO }     {NO }

[,LCST={mtalcst name}]
        {NONE        }

[,LGRAPHS=( [{REJECT}] [,{REJECT]} )]
             {ACCEPT}     {ACCEPT }

[,POLL=chars]

[,PT3EXEC={YES}]
           {NO }

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

VTAM-only operands:

```
BUFLIM=
DEVICE=
FEATUR2=
ISTATUS=
LOGAPPL=
LOGTAB=
UTERM=
```

`symbol`

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

`TERM=type`

Specifies the type of station represented by this TERMINAL macro. It must be one of the types listed in Figure 5-14.

Note 1: If the line to which the station represented by this TERMINAL macro is to operate in emulation mode as well as in network control mode, specify the TERM operand in the LINE (or GROUP) macro instead of the TERMINAL macro. (Operands specified in the LINE macro apply to line operation in network control mode and emulation mode; operands specified in the TERMINAL macro apply only to line operation in network control mode.)

Note 2: If this TERMINAL macro defines a call-in logical-connection station (CTERM=YES) representing more than one type of BSC station, specify any of the types in this operand. For example, if this logical-connection station represents IBM 1130, 2780, and System/370 Model 135 stations, you may specify TERM=1130, or TERM=2780, or TERM=3135.

This operand is required; you may code it in this TERMINAL macro or in a higher-level (LINE or GROUP) macro.

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

[1] Attached to an IBM 2972 Model 8 or 11. 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.

[3] Not supported on switched lines.

[4] Specify a System/7 (BSC version) as follows: (1) Specify TERM=SYS3 in the TERMINAL (or LINE or GROUP) macro if the line is to operate only in network control mode. (2) Specify TERM=SYS3 in the LINE (or GROUP) macro if the line is to operate in both network control and emulation modes (alternately). (3) Specify TERM=SYS7 in the LINE (or GROUP) macro if the line is to operate only in emulation mode.

Figure 5-14. Values for TERM Operand of TERMINAL Macro Instruction

```
      {addr chars                            }
[ADDR={selection chars                       }]
      {([normal addr chars][,alt addr chars])}
```

Specifies, in hexadecimal representation, the EBCDIC addressing or selection characters assigned to the station represented by this TERMINAL macro.

`ADDR=addr chars`

Specifies the addressing characters for a start-stop or BSC station on a multipoint line, or an IBM 1050 on a switched point-to-point line (POLLED=YES in the LINE [or GROUP] macro).

This operand is invalid (1) if this TERMINAL macro represents a call-in multiple-terminal-access, logical-connection station (TERM=MTA and CTERM=YES), or (2) if POLLED=NO is specified in (or POLLED operand omitted from) the LINE macro and LNCTL=SS is specified in (or LNCTL operand omitted from) the GROUP macro for the line over which this station communicates with the controller.

If this TERMINAL macro represents a *call-out* MTA terminal (TERM=MTA, CTERM=NO) and if any of the MTALCST macros named in the LCST operand of this TERMINAL macro represents IBM 1050 terminals, specify the 1050 polling and/or addressing characters in the POLL and ADDR operands of this TERMINAL macro.

Code only the alphameric addressing characters, omitting any control characters. For example, for an IBM 2740 with Station Control whose address is *A,* you would code the hexadecimal representation of the EBCDIC character *A,* thus: ADDR=C1.

Do not include in the ADDR operand the start-of-address character that is transmitted before the addressing character or the space character transmitted following it (S A SP).

Note: The character you specify in this ADDR operand for a tributary controller on a multipoint line must have the same bit pattern as the corresponding polling character specified in the POLL operand, except that the third bit from the left (bit 2) must be 1. (Bit 2 in the polling character is always 0.)

`ADDR=selection chars`

Specifies (if component selection is required) the component selection characters for one of the output components attached to the station, if on a BSC point-to-point line (LNCTL=BSC is specified in the GROUP macro). For an IBM 2780, specify the alphameric component selection character, but not the escape (ESC) character that precedes it. For an IBM 2770, specify the device control character—DC1, DC2, DC3, or)—that serves as the component selection character. For all BSC stations, specify all characters except ENQ, ESC, and ETB.

`ADDR=([normal addr chars][,alt addr chars])`

Specifies, for a 2980 terminal only, the normal and/or alternate addressing characters to be used.

The addressing characters (escape sequence) assigned to the station should be determined from the system designer.

```
[ATTN={ENABLED } ]
      {DISABLED}
```

(IBM 1050, 2741; AT&T 83B3; WU 115A, TWX; World Trade teletypewriters terminals

Specifies whether the network control program is to assume that the Attention feature of the terminal is 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 and to notify the host processor. If the feature is disabled, the program ignores it.

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

```
[BFRDLAY=count]
```

(buffered terminals only)

Specifies the delay, in seconds, between successive transmissions to the device represented by this TERMINAL macro. The maximum value for *count* is 255; the minimum is 0 (that is, no delay). This operand is valid only for buffered devices, as follows:

IBM 2740 Model 2 with the Buffered Receive feature
IBM 2770
IBM 2980
IBM 3275
IBM 3277
IBM 3284
IBM 3286
IBM 3780

If this TERMINAL macro represents an IBM 3275 (TERM=3275) to which a printer is attached, or if it represents an IBM 3284 or 3286 (TERM=3284 or TERM=3286), the buffer delay should be 13 seconds or more. BFRDLAY=0 is invalid for a 3284 or 3286 printer.

If you omit this operand, the network control program observes a delay of 13 seconds for 3284 or 3286 printers and observes no delay for any other device.

This operand is valid only for a nonswitched multipoint line (DIAL=NO is specified in [or DIAL operand omitted from] the GROUP macro; POLLED=YES is specified in the LINE [or GROUP] macro).

```
[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 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=setname is specified, ALL is assumed.

At least one execution point must be specified.

The section, *Block Handler Definition Macro Instructions*, explains how to define a block handler set.

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

```
       {NONE                 }
[BHSET={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.

`DYNAMIC`

Specifies that no block handler set is to be initially assigned to this station, but one may be assigned dynamically by command from the host processor.

`setname`

Specifies a block handler set to be assigned to this station; *setname* must be the name of a BHSET macro.

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

Specifies whether or not the block handler set is to be initially executable—that is, executable immediately when execution of the network control program begins.

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.

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

Specifies whether or not the data sent to and received from the station represented by this TERMINAL macro is "critical" (for security reasons). If you code CDATA=YES, the network control program automatically clears all buffers containing data associated with the terminal before returning them to the buffer pool. If you code CDATA=NO (or omit the operand), the program does not clear the buffers.

CDATA=YES is valid only if you specify ERASE=YES in the BUILD macro.

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

Specifies whether or not the network control program, upon receiving a message block from a station, sends the station a message block (instead of a positive acknowledgment) in response. Sending a message block in response (called a "conversational response"), is possible only if the program currently holds a request to send to the station. If it has no request, the program sends the usual positive acknowledgment.

Stations capable of accepting conversational responses are:

IBM 1050
IBM 2740 with Record Checking feature
IBM 2770 with Conversational Mode feature
All IBM BSC stations except IBM 2715 and 2780

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

Specifies whether or not the network control program is to delay executing a write operation that follows a read operation for the terminal. The delay allows the carriage of the terminal printer time enough to return to the left margin.

CRDLAY=YES is valid only if this TERMINAL macro represents an IBM 1050, 2740 Model 1, 2741, or WU TWX terminal or a MTA terminal.

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

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.

Note: A critical situation message cannot be sent from a tributary controller to its control station. Therefore, if the network control program being defined is to be executed in a tributary controller, and this TERMINAL macro represents the control station to which the tributary controller is connected, CRITSIT=YES is invalid in this TERMINAL macro. (That is, if the TADDR operand is coded in the LINE macro preceding this TERMINAL macro, do not code CRITSIT=YES in this macro.)

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.

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.

If this TERMINAL macro represents a station on a multipoint line or a 1050 on a switched point-to-point line (POLLED=YES is specified in the LINE macro), you must specify the addressing characters of the station in the ADDR operand in order for the critical situation message to be sent.

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

(switched lines only)

Specifies whether or not this TERMINAL macro 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 operating in network control mode. 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, if the LINE macro represents a switched line. It is not required, and should be omitted (or CTERM=NO coded), for a LINE macro in which CALL=OUT is coded.

TERMINAL macros that represent terminals to be called over switched call-out lines have no required positional relationship to the LINE macros for such lines. The association of terminals and call-out lines in this case is made via DIALSET operands that specify the names of logical groups (called dial sets) of call-out lines to be used for calling the terminals. The DIALSET macro defines dial sets. The TERMINAL macros may appear following any LINE macro (or macros) representing a switched line (or lines) whose line control (LNCTL=BSC or LNCTL=SS) matches that used by the terminals to be called. Example: the following statements specify two switched lines to be used for call-out only, one line to be used for call-in and call-out use, and four terminals. All three lines are included in one dial set. Five TERMINAL macros are required. One, in which CTERM=YES is coded, represents the logical connection terminal; the remaining macros represent each of the four terminals to be called over lines in the dial set.

```
DSET    DIALSET       LINES=(A,B,C)
G       GROUP         DIAL=YES,...
A       LINE          CALL=OUT,DIALSET=DSET,...
T1      TERMINAL      DIALSET=DSET,DIALNO=(5142),...
T2      TERMINAL      DIALSET=DSET,DIALNO=(7615),...
T3      TERMINAL      DIALSET=DSET,DIALNO=(8204),...
T4      TERMINAL      DIALSET=DSET,DIALNO=(7382),...
B       LINE          CALL=OUT,DIALSET=DSET,...
C       LINE          CALL=INOUT,DIALSET=DSET,...
LOGCON  TERMINAL      CTERM=YES,...
```

(In this example, one or more of the TERMINAL macros could instead appear following LINE macro *B* or LINE macro *C*; the resulting configuration would be identical to that achieved by coding the macros in the sequence shown.)

Note: If you specify CTERM=YES, no operand in any macro can refer to the name of this TERMINAL macro. For example, this TERMINAL macro cannot be named in the IDSEQ operand of the IDLIST macro.

```
           {n  }
[CUIDLEN={TWX}]
           {ALL}
           {0  }
```

(switched BSC or TWX station only)

Specifies the length of the ID sequence the network control program is to send to the BSC or TWX station represented by this TERMINAL macro, or specifies that no ID sequence be sent.

`n`

Specifies the length of the ID sequence the program will send.

If this TERMINAL macro represents a BSC station (LNCTL=BSC is specified in the GROUP macro), the first *n* characters of the sequence specified by the CUID operand of the BUILD macro will be sent.

If this TERMINAL macro represents a TWX terminal (LNCTL=SS is specified in [or LNCTL operand omitted from] the GROUP macro and TERM=TWX is specified in this TERMINAL macro), *n* must equal the length of the *call-out* sequence specified by the TWXID operand of the BUILD macro, if this TERMINAL macro specifies CTERM=NO (or CTERM operand is omitted). Or, *n* must equal the length of the *call-in* sequence specified by the TWXID operand, if this TERMINAL macro specifies CTERM=YES.

The maximum value for *n* is 20 (20 EBCDIC characters).

`TWX`

Specifies that the program is to send the TWX ID to the terminal represented by this TERMINAL macro (which must be a TWX terminal). If CTERM=NO is specified (or CTERM operand omitted), the entire sequence specified by the *call-out* parameter of the TWXID operand is sent; if CTERM=YES is specified, the entire sequence specified by the *call-in* parameter of the TWXID operand is sent.

`ALL`

Specifies that the program is to send all of the appropriate ID sequence (TWXID for TWX terminals, CUID for BSC stations) to the station represented by this TERMINAL macro.

`0`

Specifies that no ID sequence is to be sent to the station represented by this TERMINAL macro.

Note: This operand is valid only if the GROUP macro specifies DIAL=YES and the LINE macro specifies POLLED=NO (or POLLED operand omitted); and either (1) the GROUP macro specifies LNCTL=SS (or LNCTL operand omitted) and this TERMINAL (or higher-level) macro specifies TERM=TWX, or (2) the GROUP macro specifies LNCTL=BSC.

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

Specifies the numerals and, optionally, the length of a telephone number by which the network control program can call the station represented by this TERMINAL macro.

`[chars]`

Specifies the telephone number of the station. *chars* represents the dial digits (EBCDIC numeric characters) 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. The minimum value is 1; the maximum, 32.

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*. *Example:* DIALNO=19195678888.

(2) Code *chars* if the host processor will update the dial digits using the dynamic control facility. Also code *count* if the number of replacement dial digits may exceed the number of digits originally specified by the *chars* suboperand. *Example:* DIALNO=(19195678888,12). Otherwise, you may omit the *count* suboperand.

This operand is invalid if this TERMINAL macro specifies CTERM=YES or DIALSET=NONE.

Separator and End-of-Number Characters: The sequence of dial digits may include separator and/or end-of-number characters, if the modem at the communications controller is designed to accept these characters. A separator character causes the modem to delay sending the next dial digit on the communication line until the modem receives a secondary dial tone. An end-of-number character sent to the modem after the last dial digit signals the modem that it has received the last digit of the telephone number and causes it to start monitoring the line for an answer tone from the distant station. Use of the end-of-number character can reduce the time needed to establish a connection, thus increasing line utilization.

The separator character can be coded in the dial digit sequence wherever a pause for a secondary dial tone is needed. The end-of-number character must be coded at the end of the sequence.

The communications controller sends to the modem only the four low-order bits of the digits specified in the sequence. The end-of-number bit pattern is 1100 (hexadecimal C); the separator bit pattern is 1101 (hexadecimal D). Any EBCDIC characters whose four low-order bits equal these patterns may be used (for example, * [hex 5C] for end-of- number and ' [hex 7D] for separator). *Example:* DIALNO=(8'5799*)

Programmed dialing pause: If the modem is not equipped to use separator characters, you may introduce a programmed dialing pause into the dialing action to allow time to recieve a secondary dial tone. At the point in the sequence at which the pause is required, code one vertical bar character (hex FA) for each second of the pause. For instance, code three such characters if a three-second pause is required. *Example:* DIALNO=(8 | | | 5799)

```
[DIALSET={dialset name}]
        {NONE         }
```

(call-out switched line only)

Specifies the name of the primary dial set to be associated with the station represented by this TERMINAL macro. This operand is required if the network control program will call the station represented by this TERMINAL macro. It is not required if the program will not call the station.

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 (or omit the operand). This choice is appropriate when the station can call the communications controller but the network control program cannot call the station.

Note: The lines in the dial set specified by this operand must have the same operating characteristics as specified by the LINE macro that precedes this TERMINAL macro.

```
          {IN    }
[DIRECTN={OUT   }]
          {INOUT}
```

Specifies, for the station represented by this TERMINAL macro, whether the network control program will only receive text from the station (DIRECTN=IN), only send text (DIRECTN=OUT), or both send and receive (DIRECTN=INOUT [or DIRECTN operand omitted]).

If you omit this operand and you code POLLED=YES in the LINE macro, IN is assumed if you specify polling characters (POLL operand) in this macro; OUT is assumed if you specify addressing characters (ADDR operand) in this macro; and INOUT is assumed if you specify both polling and addressing characters.

If you omit this operand and you code POLLED=NO in (or omit the POLLED operand from) the LINE (or GROUP) macro, DIRECTN=INOUT is assumed.

Note: For IBM 3284 and 3286 terminals, DIRECTN=OUT is assumed because these devices transmit only status information (not message data) when polled.

```
[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; it is, however, invalid for an IBM 2740 Model 2.

```
[FANOUT=terminal name]
```

(BSC terminals only)

Specifies that this TERMINAL macro is the first of a sequence of TERMINAL macros representing stations attached to the same modem (called a "fanout" modem because several stations can be attached to it), and names the last such TERMINAL macro. (The modem referred to is the modem at the distant end of the communication line, not the modem attached to the communications controller.)

Example: Assume that four terminals are attached to a "fanout" modem, represented by four TERMINAL macros named T1, T2, T3, T4. In the FANOUT operand of the first TERMINAL macro you would specify the name of the last TERMINAL macro representing a terminal attached to the same modem, thus:

```
T1    TERMINAL   FANOUT=(T4),...
T2    TERMINAL   ...
T3    TERMINAL   ...
T4    TERMINAL   ...
```

The FANOUT operand appears only in the first TERMINAL macro, as shown. If only one terminal is attached to the "fanout" modem, omit the FANOUT operand.

All TERMINAL macros representing stations attached to the same modem must appear in a single sequence, with no other intervening TERMINAL macros for other stations.

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

[FEATURE=...]

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

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

Specifies whether or not the station is equipped with the Accelerated Carrier Return feature. (The network control program makes use of the Accelerated Carrier Return feature only if you specify FEATURE=ACR in *all* TERMINAL macros following the LINE macro. If you omit FEATURE=ACR in any TERMINAL macro, the network control program ignores the presence of the feature when sending to any terminal on the line.)

[{ATTN }]
{NOATTN}
(IBM 1050, 2741, 3767 (in 2741 mode); AT&T 83B3; WU 115A, TWX; World Trade teletypewriters; and MTA lines only)

Specifies whether or not the terminal is able to send attention signals to the controller. For IBM 1050 or 2741 terminals, specify FEATURE=ATTN only if the terminal is equipped with the Receive Interrupt (1050) or Interrupt (2741) feature.

[{BREAK   }]
{NOBREAK}
(IBM 1050, 2741 3767 (in 2741 mode); AT&T 83B3; WU 115A, TWX; World Trade teletypewriters; and MTA lines only)

Specifies whether or not the station is capable of interrupting its transmission to the controller upon receiving a break signal from the network control program. For IBM 1050 or 2741 terminals, specify FEATURE=BREAK only if the terminal is equipped with the Transmit Interrupt feature. FEATURE=BREAK is valid only if DUPLEX=FULL or SCLSET=YES is specified in the LINE macro for the line over which this terminal communicates with the controller. *Exception:* For a 3767 specified as a 2741 (in TERM operand of the TERMINAL macro), FEATURE=BREAK is also valid if DUPLEX=HALF is specified.

```
[{CHECK   }]
 {NOCHECK}
```

(IBM 2740 only)

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

```
[{SCTL   }]
 {NOSCTL}
```

(IBM 2740 Model 1 only)

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 1050 terminal represented by this TERMINAL macro is equipped with the timeout suppression feature.

```
[{XCTL   }]
 {NOXCTL}
```

(IBM 2740 Model 1 only)

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

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

(BSC and TWX stations only)

Specifies whether the station or stations represented by this TERMINAL macro will transmit identification (ID) sequences when calling or being called by the controller, and if so, how the network control program is to react to the sequence.

symbol

Specifies that the network control program is to expect and to verify ID sequences received from stations and names the identification list (IDLIST macro) with which the program is to check the sequences it receives. *symbol* is valid whether this TERMINAL macro represents a call-in logical-connection station (CTERM=YES) or a call-out station (CTERM=NO), but is valid only if the type of station is not “multiple terminal access” (TERM operand does not specify MTA).

NONE

Specifies that no ID sequences are expected from stations that call the controller or are called by the controller.

PASS

Specifies that the network control program is to expect ID sequences from stations and is to pass to VTAM all ID sequences it receives. (IDSEQ=PASS is valid only if this TERMINAL macro represents a call-in logical-connection station [CTERM=YES is specified].)

`IGNORE`

Specifies that the network control program is to ignore—that is, neither check nor pass to the host processor—any ID sequences it receives. (IDSEQ=IGNORE is valid only if the controller calls the station represented by this TERMINAL macro; if stations are to call the controller [CTERM=YES is specified], IDSEQ=IGNORE is invalid.)

This operand is valid only for BSC or TWX stations on a switched line (GROUP macro specifies LNCTL=BSC or LNCTL=SS, DIAL=YES; LINE macro specifies POLLED=NO [or POLLED operand is omitted]; this TERMINAL macro specifies a BSC or TWX station in TERM); and [if controller is to expect an ID sequence from a station it calls] this TERMINAL macro specifies a telephone number in DIALNO).

```
[INHIBIT={( [TEXTTO] [,TIMEFILL] [,WACKCNT] [,SUBBLOCK] [,ERPR] [,ERPW] )}]
         {NONE                                                          }
```

Specifies which, if any, of the network control program facilities are to be inhibited from functioning initially—that is, when the program begins execution after being loaded into the controller.

`[TEXTTO]`

Specifies that the network control program is to use the interval specified in the ITEXTTO operand of the BUILD macro, rather than the value specified by the TEXTTO operand of the GROUP macro, as the time limit between receipt of successive text characters.

`[TIMEFILL]` *(start-stop lines only)*

Specifies that automatic insertion of idle characters following carriage return and horizontal tab characters is to be inhibited— that is, no idle characters are to be sent. This parameter is valid only if LNCTL=SS is specified in (or LNCTL operand omitted from) the GROUP macro.

`[WACKCNT]` *(BSC lines only)*

Specifies that the WACK limit specified by the WACKCNT operand of the GROUP macro is to be inhibited—that is, the first WACK received from a the station causes the network control program to return the request to the host processor rather than to respond to that and subsequent WACKs with an ENQ character. This parameter is valid only if LNCTL=BSC is specified in the GROUP macro.

`[SUBBLOCK]`

Specifies that the sub-blocking indicated by the TRANSFR operand of the LINE (or GROUP) macro is to be inhibited—that is, if the number of buffers specified by the TRANSFR operand are filled by received text, the network control program terminates the receiving operation just as if the cutoff limit (specified by the CUTOFF operand) were reached.

`[ERPR]`

Specifies that recovery procedures for text-read errors (as specified by the RETRIES operand of the LINE macro) are to be inhibited. That is, the network control program will not attempt recovery for text-read errors.

```
[ERPW]
```

Specifies that recovery procedures for text-write errors (as specified by the RETRIES operand of the LINE macro) are to be inhibited. That is, the network control program will not attempt recovery for text-write errors.

```
NONE
```

Specifies that none of the network control program facilities are to be inhibited; that is, all will become effective when execution of the network control program begins.

```
[ITBMODE=( [{YES}] [,{YES}] )]
            {NO }    {NO }
```

(BSC lines only)

Specifies, for a BSC station, how the network control program is initially to handle ITB characters in text received from the station or the host processor.

The first suboperand specifies whether the program is to insert a EIB (error information block) character following each ITB character received from the station.

The second suboperand specifies whether or not each ITB character received from the access method is followed by an EIB character. If you specify YES, the network control program removes from the message data it sends to the station the first character following each ITB character received from the access method.

This operand is valid only if you code LNCTL=BSC in the GROUP macro.

```
[LCST={mtalcst name}]
      {NONE        }
```

Specifies the name of the MTALCST macro representing the set of device characteristics to be used by the network control program when calling the device represented by this TERMINAL macro.

Code LCST=mtalcst name in the TERMINAL macro for each station that the network control program is to call via a multiple terminal access line.

Code LCST=mtalcst name only if you specify DIAL=YES and LNCTL=SS (or you omit the LNCTL operand) in the GROUP macro; TERM=MTA in the TERMINAL macro (or a higher-level macro); CTERM=NO in (or omit CTERM operand from) the TERMINAL macro; and if you specify a telephone number in the DIALNO operand of the TERMINAL macro.

Code LCST=NONE (or omit the operand) if no MTALCST macros need to be specified.

```
[LGRAPHS=( [{REJECT}] [,{REJECT}] )]
            {ACCEPT}    {ACCEPT}
```

Specifies whether leading graphics received from the station are to be accepted by the network control program or rejected (that is, treated as an error condition).

The first parameter specifies acceptance or rejection of leading graphics for read operations; the second parameter specifies acceptance or rejection for write operations.

Note: For a 2740 model 2, the second parameter must be ACCEPT.

```
[POLL=chars]
```

Specifies, in hexadecimal representation, the EBCDIC polling characters assigned to this station.

Code only the alphameric polling character, omitting any control characters. For example, for an IBM 2740 with Station Control whose polling character is *A* you would code the hexadecimal representation of the EBCDIC character *A*, thus: POLL=C1. Do not code the space character that is transmitted after the polling character.

Note: If this TERMINAL macro represents an IBM 3275, 3277, 3284, or 3286, and the ADDR operand specifies the addressing character, the POLL operand must specify the polling character.

If this TERMINAL macro represents a call-in logical connection 1050 terminal (CTERM=YES, TERM=1050), you must specify polling characters in at least one of the macros representing this terminal (that is, in the TERMINAL macro or in one of the COMP macros [if any] representing terminal components).

This operand is invalid if the TERMINAL macro represents an IBM 2740 without Station Control, a 2741, or a 2980, as these types of terminals cannot be individually polled.

This operand is also invalid if this TERMINAL macro represents a call-in multiple-terminal-access, logical connection terminal (TERM=MTA, CTERM=YES).

If this TERMINAL macro represents an MTA terminal (TERM=MTA) that is not a call-in MTA terminal (CTERM=NO), the POLL and/or ADDR operands may be specified or omitted. Exception: If any of the MTALCST macros named in the LCST operand of this TERMINAL macro represents IBM 1050 terminals, either POLL, or ADDR, or both, must be specified in this TERMINAL macro.

Note: The character you specify in this POLL operand for a tributary controller must conform to the requirements described under the TADDR operand of the LINE macro.

```
[PT3EXEC={YES}]
         {NO }
```

Specifies whether or not a block handler set that is executed at point 3 (see the BHSET macro) is to be associated with the device represented by this TERMINAL macro.

This operand is ignored if you omit the BHSET operand, or if you code BHSET=NONE, BHEXEC=PT3, or BHEXEC=ALL in this TERMINAL macro.

```
[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 suspend the session.

The maximum value you may specify is 255; the minimum is 1.

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

Caution: In most applications, the default value, XMITLIM=NO, is *not* appropriate because it allows the station represented by this TERMINAL macro to monopolize the communication line indefinitely. You should therefore normally specify a count in this operand. The lower the count you specify for each station attached to a line, the greater will be the degree of line sharing.

XMITLIM=count is valid only for stations associated with a line for which you have specified a service order table (that is, a SERVICE macro follows the LINE macro).

Specifying Lower-Level Operands in a Higher-Level Macro

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 5-9 shows which of the COMP operands you may specify in the TERMINAL macro.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the TERMINAL macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this TERMINAL macro description means only that the NCP generation assembly procedure accepts them as valid operands of the TERMINAL macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

VTERM Macro Instruction (VTAM Only)

The VTERM macro instruction specifies, for VTAM, a name for each type of terminal that may call the controller on a multiple-terminal-access line, so that VTAM can associate the calling terminal with a specific application program. The macro also provides to VTAM information about the VTAM buffers required by the terminal, and specifies "log-on" information to be associated with the terminal operating parameters specified by an MTALCST macro.

The log-on information includes (1) the name of the application program to which the calling terminal is to be automatically logged on, and (2) the name of the VTAM interpret table that determines the application program to which the terminal is to be logged on.

This macro, if used, must appear directly following the TERMINAL macro representing the call-in MTA terminal (that is, CTERM=YES is specified in the TERMINAL macro). For example:

```
GROUP
LINE
TERMINAL    TERM=MTA,CTERM=YES,...
VTERM
VTERM
TERMINAL    TERM=MTA,CTERM=NO,...
TERMINAL    TERM=MTA,CTERM=NO,...
    .
    .
    .
```

See the *VTAM System Programmer's Guide* for a complete description of this macro and its operands and for the conditions of its use.

Name	*Operation*	*Operands*
symbol	VTERM	LCST=mtalcst name
		[,BUFLIM={n}] {2}
		[,LOGAPPL=name]
		[,LOGTAB=name]

COMP Macro Instruction

The COMP macro represents one input component, or one output component, or both, of a start-stop or BSC station in the teleprocessing network. 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.

The COMP macro is valid only for start-stop and BSC stations; do not use it for a station on an SDLC link.

COMP macros are required only for components the network control program must individually poll or address (select), using the specific polling or addressing (selection) characters assigned to the component. A COMP macro is not required, however, for a station having only one input component and one output component because the polling and addressing characters for these are specified in the TERMINAL macro. Thus, no COMP macro would follow a TERMINAL macro that represents a terminal having one of each type of component, for example, an IBM 1050 with no other components than a keyboard/printer. *Exception:* Each of two or more components with which separate sessions will be established concurrently must be represented by an individual (TERMINAL or COMP) macro. Thus, to establish concurrent sessions with a keyboard and a card punch, for example, would require that a separate TERMINAL (or COMP) macro be used to represent each device.

(VTAM Users Only) Appearing at the end of the list of operands below are the VTAM-only operands that may be coded in this macro instruction. These provide information only to the VTAM initialization process and are not required (though are permissible) in the card deck used as input to the NCP generation procedure.

Each COMP macro coded causes a resource name to be generated. All operands of the COMP macro apply only to line operation in network control mode.

Name	*Operation*	*Operands*
symbol	COMP	

```
[ADDR={addr chars      }]
       {selection chars}

[,ATTN={ENABLED }]
       {DISABLED}

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

```
Name    Operation    Operands

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

                     [,CDATA={YES}]
                             {NO }

                     [,CONV={YES}]
                            {NO }

                     [,ENDTRNS={EOT}]
                               {EOB}

                     [,INHIBIT= {([TEXTTO][,TIMEFILL][,WACKCNT] ]
                                 [,SUBBLOCK][,ERPR][,ERPW])}
                                {NONE}

                     [,ITBMODE=([{YES}][,{YES}])]
                                 {NO }   {NO }

                     [,LGRAPHS=([{REJECT}][,{REJECT}])]
                                 {ACCEPT}    {ACCEPT}

                     [,POLL=chars]

                     [,PT3EXEC={YES}]
                               {NO }

                     [,XMITLIM={count}]
                               {NO   }

                     VTAM-only  operands:
                     BUFLIM=
                     DEVICE=
                     ISTATUS=
                     LOGAPPL=
                     LOGTAB=
```

`symbol`

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

The operands shown above (exclusive of VTAM-only operands) 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.

Specifying COMP Operands in a Higher-Level Macro

Except for the POLL and ADDR operands, the operands of the COMP macro can be specified in the TERMINAL, LINE, or GROUP macro instead of the COMP macro. Figure 5-9 shows which of the COMP macro operands you may specify at a higher level.

VTAM-Only Operands

The VTAM-only operands listed at the beginning of the COMP macro description convey no information to the NCP generation assembly process. Certain of these operands must appear in the NCP generation input deck that serves as input to the

VTAM initialization process. See the *VTAM System Programmer's Guide* for descriptions of these operands and for information on the VTAM initialization process.

Appearance of these operands within this COMP macro description means only that the NCP generation assembly procedure accepts them as valid operands of the COMP macro and does not imply that they must be coded. If coded, they are not checked for proper syntax. Nor does the assembly process verify the presence or absence of related VTAM-only operands. See the *VTAM System Programmer's Guide* for the VTAM requirements regarding the coding of these operands.

INNODE Macro Instruction

The INNODE macro instruction represents an IBM 3704 or 3705 Communications Controller attached to an SDLC link.

In version 3 of the network control program, communications controllers with which the program communicates over an SDLC link are represented by INNODE macros. For compatibility with version 3, versions 4 and 5 accept INNODE macros to represent communications controllers, and for this reason, this INNODE macro is included in this publication. However, when defining a new source program to be generated under NCP version 4 or 5, or when adding new communications controllers to an existing version 3 program to be re-generated under version 4 or 5, use PU macros, not INNODE macros, to represent the controllers.

The INNODE macro specifies:

- The address of the communications controller.
- Whether immediate polling retry is required.
- The maximum number of path information units (PIU) or PIU segments the network control program will send to the communications controller before awaiting a response from the controller.
- The maximum number of consecutive path information units the network control program will send at one time to the communications controller.
- The number of error recovery attempts the network control program will make when transmission errors occur.
- The subarea address of the communications controller.

The INNODE macro is valid only for:

- A network control program, with or without the partitioned emulation programming (PEP) extension, that is to operate in a local communications controller and that is capable of communicating with a remote communications controller (TYPGEN=NCP-LR or PEP-LR is specified in the BUILD macro over a nonswitched link [DIAL=NO]); or
- A network control program that is to operate in a remote controller (TYPGEN=NCP-R is specified in the BUILD macro) that communicates with the local controller over a non-switched link (DIAL=NO).

The INNODE macro for the communications controller must appear directly following the SERVICE macro defining the service order table for the SDLC link to which the controller is attached, unless this network control program is to operate in a remote controller, in which case the INNODE macro must immediately follow the LINE macro.

If the local and the remote controller can communicate over one or more alternate (backup) SDLC links in lieu of the primary link, an INNODE macro must appear after each LINE (or SERVICE) macro corresponding to an alternate link. Omit the SUBAREA operand from each of these alternate INNODE macros. Also (1) omit all other INNODE macro operands except IRETRY and RETRIES, if the alternate INNODE macro is coded in a *local* network control program (and thus represents the *remote* controller); or (2) if the alternate INNODE macro is coded in a *remote* network control program (and thus represents the *local* controller), only the MAXOUT and PASSLIM operands are valid and must specify the same values as these operands specify in the *primary* INNODE macro.

Note: Communications controllers and cluster controllers cannot both be attached to the same SDLC link. Therefore, CLUSTER macros and INNODE macros cannot both appear following a SERVICE (or LINE) macro for an SDLC link.

```
Name          Operation     Operands

symbol        INNODE        [ADDR=chars]

                            [,DATMODE={HALF} ]
                                      {FULL}

                            [,IRETRY={YES} ]
                                     {NO }

                            [,MAXOUT=n]

                            [,PASSLIM={n} ]
                                      {1}

                            [,RETRIES=(,t[,n])]

                            [,SUBAREA=n]
```

symbol

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

[ADDR=chars]

(valid in local NCP only)

Specifies the hexadecimal representation of the eight-bit address of the controller represented by this INNODE macro. This address and the address specified in the TADDR operand of the LINE macro within the remote network control program must be identical to the physical address of the remote controller. (The customer engineer enters the physical address into the IPL configuration data set in the diskette contained within the remote controller.)

This operand is required if the program you are defining is to be executed in a local communications controller and can communicate with a remote controller (TYPGEN=NCP-LR or TYPGEN=PEP-LR is specified in the BUILD macro) and if you specify the SUBAREA operand in this macro; otherwise, this operand is invalid. 00 and FF are invalid address values.

```
[DATMODE={HALF} ]
         {FULL}
```

(valid in local NCP only)

Specifies for a communications controller on an SDLC link allowing simultaneous transmission in both directions, whether the network control program is to communicate with the distant controller in half-duplex or duplex data mode. This operand is valid only if the ADDRESS operand of the corresponding LINE macro specifies both a receive and a transmit address.

DATMODE=FULL is assumed if you omit this operand and you (1) code the SUBAREA operand; (2) specify, in the LINE macro, both a receive and a transmit address; and (3) specify, in the BUILD macro, either TYPGEN=NCP-LR or TYPGEN=PEP-LR.

This operand is invalid if (1) the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R is specified in the BUILD macro) or (2) the program is to be executed in a local controller (TYPGEN=NCP-LR) and you omit the SUBAREA operand from this macro.

Minimizing Line Turnaround Delay:

Each reversal in direction of message transmission over an SDLC link causes a line turnaround delay of several milliseconds if the carrier signal transmitted by the modem is interrupted. Efficiency of the link is thus lower than if the carrier signal is transmitted continuously, because no message data can be transmitted during these intervals.

Interruption of the carrier signal, and hence line turnaround delays, are inherent and unavoidable if the communications facility (including lines and modems) is half-duplex. They may also occur, however, in a duplex facility and in this case they may be prevented by ensuring that the carrier signal is transmitted continuously. Continuous carrier transmission results from either (1) continuous activation of the "request-to-send" signal sent from the communications controller to the modem, or (2) "strapping" (internally connecting) the modem to transmit carrier continuously independent of whether the "request-to-send" signal is activated.

The "request-to-send" signal (and hence the carrier sent by the modem) is activated continuously if: (1) the communication facility is duplex, (2) the line set to which the SDLC link is attached is duplex, (3) separate receive and transmit addresses are specified in the ADDRESS operand of the LINE macro, and (4) DATMODE=FULL is specified in the LINE macro.

The "request-to-send" signal is not activated continuously if the line set is half-duplex *or* if DATMODE=HALF is specified in the LINE macro. The line turnaround delay can still be minimized or eliminated, however, if the modem is strapped for continuous carrier operation. Consult the supplier or installer of the modem to determine if it is capable of continuous carrier operation and, if so, have the modem so connected. (The modem cannot be strapped for continuous carrier if the communication facility is half-duplex.)

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

(valid in local NCP only)

Specifies whether or not the network control program, when an "idle timeout detect" condition follows a polling operation, is to immediately retry the operation. If you specify IRETRY=YES, the program repolls the controller; otherwise, the program bypasses the unsuccessfully polled controller and services the station represented by the next entry in the service order table.

This operand is invalid if the program you are defining is to be executed in a remote communications controller (TYPGEN=NCP-R is specified in the BUILD macro).

If you omit this operand and the program is executed in a local communications controller (TYPGEN: NCP-LR or PEP-LR), IRETRY=NO is assumed.

Note: You may specify IRETRY in an INNODE macro associated with either the principal or the backup SDLC link.

```
[MAXOUT=n]
```

Specifies the maximum number of path information units (PIU) (or PIU segments, if the program divides PIUs into segments) the network control program will send to the controller represented by this INNODE macro before requesting a response from the controller. The minimum is one; the maximum is seven.

If you omit this operand, a *local* network control program will send one PIU or segment to the remote controller represented by this INNODE macro; a *remote* network control program will send up to seven PIUs or segments to the local controller represented by this INNODE macro.

This operand is invalid in an INNODE macro associated with a backup SDLC link (SUBAREA operand is omitted).

```
[PASSLIM={n}]
        {1}
```

Specifies the maximum number of consecutive path information units (PIU) or PIU segments the network control program will send at one time to the communications controller represented by this INNODE macro. If this operand appears in a *local* network control program (TYPGEN: NCP-LR or PEP-LR), the program services the station represented by the next entry in the service order table when the pass limit value is reached.

If this operand appears in a *remote* network control program (TYPGEN=NCP-R), the program terminates sending to the local controller when the pass limit value is reached, and waits until polled again by the local network control program before resuming transmission.

The minimum is one PIU or segment. The maximum is 254.

The default value assumed if you omit this operand is 1 (PASSLIM=1).

If this INNODE macro appears in a remote network control program (and thus represents a local communications controller), the maximum value is 254. This is also the recommended value and the default value assumed if you omit this operand.

```
[RETRIES=([,t[,n])]
```

(valid in local NCP only)

Specifies, in conjunction with the RETRIES operand of the LINE macro, the number of attempts–via retransmission–to recover from text errors in message data sent to the communications controller represented by this INNODE macro.

The meanings of the *t* and *n* parameters are the same as for these parameters in the RETRIES operand of the LINE macro.

Specifying these parameters in the INNODE macro allows you to designate different values for each communications controller attached to the line. (You may instead specify the *t* and *n* parameters in the RETRIES operand of the LINE macro.)

Note: In order to avoid loss of contact with the remote controller when that controller must access its disk (as, for example, when the remote controller storage is being dumped), specify the number of retries and the value of REPLYTO (see the GROUP macro) such that a minimum of 30 seconds total retry time is assured.

```
[SUBAREA=n]
```

(valid in local NCP only)

Specifies the subarea address assigned to the network control program in the remote communications controller represented by this INNODE macro. (The network control program in each controller in the network must have a unique subarea address.) The value of *n* specified in this SUBAREA operand and in the SUBAREA operand of the BUILD macro for the remote NCP must be identical.

The minimum valid subarea address is 1—that is, SUBAREA=1. The maximum address is the value you specify in the MAXSUBA operand of the BUILD macro.

Code this operand only in the INNODE macro associated with the LINE macro representing the *principal* SDLC link to the remote controller. Omit it from any INNODE macro associated with a LINE macro representing a *backup* (alternate) link.

This operand is invalid for a remote network control program (TYPGEN=NCP-R).

If program is to be executed in a local communications controller, the following rules apply.

If this INNODE macro and its associated LINE macro represent the *principal* SDLC link to a remote controller, SUBAREA must be specified.

If this INNODE macro and its associated LINE macro represent a *backup* SDLC link to a remote controller, omit the SUBAREA operand and all other operands except IRETRY and RETRIES. (When communicating with the remote controller over the backup link, the network control program uses the same values as specified in the INNODE macro for the principal link except for the values of IRETRY and RETRIES.)

(At least one INNODE (or PU) macro in the source program must specify the SUBAREA operand.)

Note: A backup SDLC link can be defined only if the line constituting the link is serviced by a type 2 (not type 1) communication scanner (TYPE=TYPE2 is specified in the CSB macro).

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 access method over the network control subchannel or from stations on lines operating in network control mode. 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 BSC and start-stop 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 handling routines cannot be specified for processing message data originating from or destined for SDLC stations.

Block Handler Delimiter Macro Instructions

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

                                        {PT1}
bhname       STARTBH        [BHEXEC={PT2}]
                                        {PT3}
```

`bhname`

Specifies the name of the block handler set and is required; *bhname* may be any symbol valid in the assembler language; the first character must not be $. (This name is referred to by the BHSET macro.)

```
         {PT1}
[BHEXEC={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 received from the access method to be processed *after* the network control program has contacted the station, code BHEXEC=PT2.

Note: If you specify BHEXEC=PT2, a logical connection between the network control program and the station must exist before the block handler is executed. The interval during which the line is unavailable for communication with other stations is extended by the execution time of the block handler.

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 to allow communication over the line to continue while the block handler is processing the received data, code BHEXEC=PT3.

If you wish the network control program to suspend further communication over the line until the block handler has processed the received message data, code BHEXEC=PT2.

Note: If you specify BHEXEC=PT2, the line is unavailable for further communication with stations during the time occupied by block handler execution.

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 a DATETIME, EDIT, or UBHR macro.

The IBM-supplied routine that converts path information units (PIV) from BSC to SDLC format can be included only in a block handler that is executed at point 3. The macro that calls this routine, SPAFPT3, is used, most appear after a STARTBH macro in which you specify BHEXEC=PT3.

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]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

This macro has no operands.

Block Handling Function Macro Instructions

Block-handling 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 }
```

```
[symbol]
```

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

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

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

```
         {YY.DDD   }
[DATEFMT={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, 75.294 (October 21, 1975).

Code DATEFMT=MM/DD/YY if you wish the date to appear in the month/day/year format; for example, 10/21/75.

Code DATEFMT=YY/MM/DD if you wish the date to appear in the year/month/day format; for example, 75/10/21.

Code DATEFMT=DD/MM/YY if you wish the date to appear in the day/month/year format; for example, 21/10/75.

```
[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 point two, 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.

Do not code both DATE=NO and TIME=NO.

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.

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).

If you specify both EDIT and DATETIME in the same block handler, the EDIT macro must precede DATETIME.

Name	*Operation*	*Operands*
[symbol]	EDIT	[BKSP={char}] {16 }

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the first character may not be $. (The generation procedure does not check the symbol for validity.)

```
[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.

SPAFPT3 Macro Instruction

The SPAFPT3 macro causes the IBM-supplied PIU format conversion routine for the SDLC/BSC path function to be included in a block handler. A block handler that includes this macro (or includes a UBHR macro that specifies a user-written conversion routine) must be specified for any BSC station that is to communicate with an SDLC logical unit via the SDLC/BSC path function. The block handler must be specified as executable at point 3 (that is, message data is processed after it is received from the BSC station).

Do not specify both the SPAFPT3 macro and a UBHR macro (calling a user-written conversion routine) in the same block handler.

See the publication *Network Control Program/VS SDLC/BSC Path Function System Programmer's Guide* (GC30-3029) for information on the path function.

Name	*Operation*	*Operands*
`[symbol]`	`SPAFPT3`	

`[symbol]`

Provides a name for the macro. *symbol* must be any valid assembly language symbol (the generation procedure does not check the symbol for validity).

This macro has no operands.

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.

A maximum of 65 uniquely named user modules may be specified by UBHR macros. However, the generation procedure does not limit the number of entry points (ENTRY operand) you may specify in a user-block-handling routine.

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.

See *Guidelines for Writing User Block Handling Routines* in Chapter 4 for more information on these routines.

```
Name         Operation    Operands

[symbol]     UBHR         NAME=member name

                          [,ACCESS={GOOD }]
                                   {ERROR}

                                    {READ  }
                          [,COMMAND={INVITE}]
                                    {BOTH  }

                          [,ENTRY=entry point name]

                          [,PT2EXEC={BEFORE}]
                                    {AFTER }
```

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the first character may not be $. (The generation procedure does not check the symbol for validity.)

`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.

The ACCESS operand may be coded 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.

```
          {READ   }
[COMMAND={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.

This operand is valid only if ACCESS=GOOD is specified in (or omitted from) this macro.

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. You may generate a maximum of 255 block handler sets.

Name	Operation	Operands
setname	BHSET	[EXEC={YES}] {NO } [,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.)

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

Specifies whether or not the block handler set is to be executable when it is associated with a device by command from the host processor.

Do not confuse this operand with the EXEC operand of the TERMINAL macro, which specifies whether or not the block handler set is to be *initially* executable (that is, as soon as the network control program begins execution after being loaded into the controller.)

`[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 ends, 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.

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

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

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

`[symbol]`

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

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

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

`mask1...mask4`

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

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

LIB Portion and HSS Mask Bit Position	*Bit Value*	*Scanner Position*	*Address Scanned*	*Addresses Not Scanned*
0	1	First	020	021,030,031,040,041
		Second	0A0	0A1,0B0,0B1,0C0,0C1,0D0,0D1
		Third	120	121,130,131,140,141,150,151
		Fourth	1A0	1A1,1B0,1B1,1C0,1C1,1D0,1D1

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

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

The bit settings you specify should correspond to the high speed lines requiring increased scanning. For each such line interface installed in the controller, a high speed select feature is present that blocks the attachment of lines to all but the lowest address in the corresponding LIB portion.

Example: Assume that a 3705 having three modules is equipped with type 3 scanners in the first and second expansion modules, but not in the base module. If high speed select features are present in the second scanner for LIB portions 3 and 7 (thus allowing high speed lines to be attached to addresses 0A6 and 0AE), you would specify HSPDSEL=(,00010001,00000000). The first comma signifies that no type 3 scanner is installed in the base module; the first eight-bit mask indicates that increased scanning frequency is required for addresses 0A6 and 0AE in LIB portions 3 and 7, respectively; and the second mask indicates that no addresses in the second expansion module (scanner position 3) require increased scanning frequency.

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

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

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

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

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

`limit1...limit4`

Specifies the scan limits for each installed type 2 or type 3 scanner. Each limit can be from 0 to 3; these values have the meanings shown below. *Limit1* specifies the scan limit for the first scanner position (base module), *limit2* for the second position (first expansion module) etc. All addresses associated with a scanner are scanned if the scan limit for that scanner is 0. Scan limits of 1, 2, and 3 reduce the number of addresses scanned to 8, 48, and 16, respectively. If a scanner position does not contain a type 2 or type 3 scanner, code a comma for the corresponding limit [for example, SCANCTL=(limit1,,limit3,, asmask)]. If a type 2 or type 3 scanner is installed but you specify no limit, the generation procedure assigns the appropriate limit based on the range of actual installed addresses and line speeds as specified in the LINE macros.

The scan limits have the following meanings:

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

*If 3704 is equipped with two LIBs and the speed of any line(s) is 19,200 bps, specify a scan limit of 2 and do not use address substitution.

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 substitution; all addresses enabled.
3	1	Address substitution is to be performed for address 6 in LIB position 1. Addresses 8 and 9 in all LIB positions are disabled.
3	0	No address substitution; all addresses enabled.

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

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.

Part IV.
Defining Emulation Functions: BSC and/or Start-Stop Networks

This Part (Chapters 6 and 7) is of interest only to users needing to define emulation programs (or network control programs with the partitioned emulation feature) for networks that include one or more lines or line groups that will operate exclusively in emulation mode. You may wish to remove and file one or both of these chapters if unneeded—see *How to Use this Book* and the associated chart *To Define a Program* at the front of this book.

Chapter 6: Emulation Functions for BSC and/or Start-Stop Networks

This chapter describes the aspects of a teleprocessing subsystem that you must identify to an emulation program. This chapter gives the same information for emulation mode operation as is given in Chapter 5, and is included for the convenience of those who presently have no need for information on network control operation. This entire chapter pertains only to start-stop and binary synchronous communications. SDLC lines cannot be operated in emulation mode.

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

- The stations and lines of the teleprocessing network
- The communications controller hardware configuration
- The type of transmission control unit to be emulated
- The procedural options governing message traffic between the controller and the network

The remaining two sections explain the optional service aid facilities that may be included in the program and the program generation options and data sets that the generation procedure will use in creating the network control program load module.

The description of each characteristic and option given in this chapter is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the program generation macro instructions given in Chapter 7.

Once you are familiar with those characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions. At that point you should go on to Chapter 7.

Teleprocessing Network Characteristics

The descriptions of the network characteristics give the names of the applicable macro instructions and operands. For your convenience, many operands of the LINE macro can be specified instead in the GROUP macro, as explained in Chapter 7. The description of the operand always appears under the macro named, however.

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 communications controller. For operations in emulation mode, this definition includes (1) computers, (2) transmission control units such as the IBM 2701, 2702, and 2703, and (3) the input/output units (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as *terminals.*

Type of Station

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

Terminal Features

The presence of the features below is specified in the FEATURE operand of the LINE macro representing the line over which the controller communicates with the terminal.

Record Checking: Some start-stop terminals have the record-checking capability (also called longitudinal redundancy checking), and others do not. For each line in the network you must specify to the network control program whether the terminals with which the program communicates over that line have the record-checking capability. If the terminal is an IBM 1050, 1060, 2260, 2265, 2845, or 2848, or System/7, which do have this capability, specify LRC in the FEATURE operand of the LINE macro. Also specify LRC for an IBM 2740 (Model 1 or 2), if it is equipped with the Record Checking feature.

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

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

Dual Code: Either of two transmission codes (EBCDIC and USASCII) can be transmitted on a binary synchronous communication line attached to an IBM 2701 Data Adapter Unit equipped with the Dual Code feature for that line. The code used is changed from one to the other by command from the access method. The same function can be performed when the IBM 3704 or 3705 is installed in place of the 2701. Specify DUALCODE in the FEATURE operand of the LINE macro representing the line, if the Dual Code feature was used for that line when the line was attached to the 2701. Otherwise, specify NODUALCD or omit the parameter. (In addition to EBCDIC and USASCII, transparent USASCII is supported as a dual code option for a line serviced by a type 3 scanner.)

Communication Line Characteristics

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

Line characteristics refer to (1) the functional attributes of the transmission path, for example, whether the line is half-duplex or duplex; (2) logical characteristics, such as the transmission code and line control scheme employed; and (3) related aspects of the line such as the address by which it is known by the network control program.

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

Whether a line is switched or nonswitched must be specified in the DIAL operand of the GROUP macro representing the line group. However, you need not specify whether a nonswitched line is multipoint or point-to-point. It is the responsibility of the access method to be aware of this, and to issue the appropriate command sequences for each type of line.

Half-Duplex vs. Duplex Lines

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

Line Speeds and Clocking

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

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

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

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

Line and Subchannel Addresses

Each communication line attached to the communications controller (in emulation mode) requires a nonshared subchannel address on a byte multiplexer channel. Each line is identified to the network control or emulation program by a line interface address representing the physical location in the controller at which the line is attached (via line set and line interface base [LIB]). Associated with each line address is the CPU subchannel address(es) with which the control program communicates when transmitting message data to or from that line. Specify the line interface address and associated subchannel address(es) in the ADDRESS operand of the corresponding LINE macro. (The multi-subchannel line access facility of the network control program permits two or more emulation subchannels to communicate, alternately, with the same communication line. The address of each subchannel to be associated with a line must be specified in the ADDRESS operand.)

Modems and Automatic Calling Units

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

New Sync Feature

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

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

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

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

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

Automatic Calling Units

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

Communications Controller Hardware Configuration

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

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

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

For each scanner, you must specify to the emulation program (1) the type of scanner, (2) the machine module in which it is installed, and (3) the bit rates of the oscillators with which it is equipped. This information should be obtained from the system designer before you code the program generation macro instructions. Specify the details of the scanners in the TYPE, MOD, and SPEED operands of a CSB macro—one macro for each scanner in the controller.

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

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

Appendix J gives a method for determining the interrupt priority for each line in the network.

Communication Between Controller and Host Processor

Associated with each communication line specified as operable in emulation mode, and serviced by a type 3 scanner, is a pair of buffers contained within the control blocks related to the line. The size of each buffer in the pair is user specified as 4, 8, 16, 32, 64, 96, 128, 160, 192, or 224 bytes.

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than do smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

The size of the emulation mode buffers for a line serviced by a type 3 scanner is specified in the BUFSIZE operand of the LINE macro for the line. If you do not specify a size, 32-byte buffers are provided for lines operating at speeds of 9600 bps or less, and 64-byte buffers are provided for lines operating at higher speeds (as specified in the SPEED operand of the LINE macros).

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

Procedural Options

When defining a program that performs only emulation functions, there are only three procedural options: the type of line control discipline to be used for each line, the terminal timeouts required, and—for World Trade teletypewriters (teleprinters) only—the end-of-block and end-of-transmission sequences to be recognized by the program.

Type of Line Control

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

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

Terminal Timeouts

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

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

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the program uses the timeout intervals indicated in the descriptions of these two operands for all lines in the group represented by that macro. Some applications may justify unlimited intervals, that is, no timeout at all. This also may be specified in the REPLYTO or TEXTTO operands.

EOB and EOT Sequences for World Trade Teletypewriter Terminals

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

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

Multi-Subchannel Line Access Facility

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

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

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

The MSLA facility can be used in the following ways:

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

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

Service Aids

The emulation program (or network control program, when specified to perform only emulation functions), provides the emulation mode line trace, panel test, and dynamic dump facilities to aid in diagnosing difficulties in network operation. Although optional, inclusion of these facilities in the program is recommended.

Line Trace Facility for Emulation Mode

The line trace facility of the program is a service aid that permits detailed analysis of the operation of any communication line controlled by the program. This facility records operating parameters of a line each time a level two interrupt (except bit service interrupt) or level three interrupt occurs for that line. (Level two is the program level at which bit service or character service for the communication line is performed. Level three is the program at which servicing of channel interrupts is performed.) The program accumulates this information in a trace table within controller storage. (Unlike a program that includes network control functions, a program that performs only emulation functions does not accumulate the trace information in buffers and does not transfer the buffer contents to the host processor. The contents of the controller storage must be dumped in order to make the line trace records available. Inclusion of the line trace facility, the number of lines to be traced, and the size of the trace table are specified in the LINETRC operand of the BUILD macro.

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

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

Panel Tests

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

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

Dynamic Dump Facility

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

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

Program Generation Options and Data Sets (Files)

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

Program Generation Options

Program generation options pertain to the type of functions (emulation) that the program is to perform, the type of communications controller in which the program will be executed (3704 or 3705), and several assembly and link-editing options. All program generation options are specified in the BUILD macro.

Type of Program to be Generated

It is assumed that the reader of this chapter intends to generate a program that performs only emulation functions. A network control program that performs only emulation functions is generated if you specify TYPGEN=EP in the BUILD macro (or if you omit the TYPGEN operand). If you intend, however, to generate a program that performs both network control and emulation functions, or network control functions only, consult Chapter 4 for the characteristics and procedural options that you must consider when defining a program.

Model of Controller

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

Other Options

The remaining program generation options, and the operands of the BUILD macro by which you specify them, are:

- Whether stage two of the generation procedure is to consist of a single, multistep job of a separate job for each step, and whether a job card is required (JOBCARD).
- The region size for the stage two linkage editor job steps (LESIZE).
- The type of device or class of devices to be used for utility data sets during stage two (UNIT) [applicable only for OS/VS].

Data Sets (Files) Used in the Generation Procedure

The names of the various program data sets to be used in the generation procedure when generating under OS/VS are specified by the LOADLIB, OBJLIB, QUALIFY, UT1, UT2, and UT3 operands of the BUILD macro.

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

The only file name required when generating under DOS/VS is NEWNAME.

Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks

System Definition Macro Instruction

This section contains the single system definition macro instruction, BUILD, to be used in defining a program that performs only emulation functions. If you wish to define a program that performs network control functions, or both network control and emulation functions, see Chapter 5.

BUILD Macro Instruction

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

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

```
Name        Operation    Operands

[symbol]    BUILD        HICHAN=(addr1[,addr2]),

                         LOADLIB=dsname,        (OS/VS only)

                         LOCHAN=(addr1[,addr2])],

                         OBJLIB=dsname          (OS/VS only)

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

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

                         [,JOBCARD={YES}]       (OS/VS only)
                                   {NO }

                         [,LESIZE=size]         (OS/VS only)

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

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

                         [,NEWNAME={EP001 }]
                                   {symbol}

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

```
Name        Operation       Operands

                                     {symbol}
                            [,QUALIFY={NONE   }]   (OS/VS only)
                                     {SYS1   }

                            [,TEST={YES}]
                                   {NO }

                            [,TYPGEN=EP]

                            [,TYPSYS={OS }]
                                     {DOS}

                            [,UNIT=unit type]      (OS/VS only)

                            [,UT1=dsname]          (OS/VS only)

                            [,UT2=dsname]          (OS/VS only)

                            [,UT3=dsname]          (OS/VS only)
```

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`HICHAN=(addrl[,addr2])`

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

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

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

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

For a program to be executed in a 3705, the address must equal 4*n*-1, where *n* is an integer equal to or exceeding 1. Choose the appropriate subchannel address from the list below:

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

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

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

This operand is required.

Note: Also see description of LOCHAN operand.

`LOADLIB=dsname`

(generation under OS/VS only)

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

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

`LOCHAN=(addr1[,addr2])`

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

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

This operand defines for each channel adapter the lowest emulation subchannel address associated with that adapter, regardless of whether this address is used for an emulation-mode line or for dynamic dump data transfer.

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

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

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

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

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

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

The address range specified by the LOCHAN and HICHAN operands applies only to emulation subchannels. The native subchannel address may, but need not, fall within this range. *Exception:* The native subchannel address must not be within the LOCHAN-HICHAN range if the communications controller is equipped with two type 4 channel adapters.

This operand is required.

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

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

This operand is required for generation under OS/VS and is not applicable for generation under DOS/VS.

`[CA=(adapter1[,adapter2])]` *(local controller only)*

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

`adapter1`

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

`adapter2`

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

See Figure 7-1 for valid specifications of the CA operand.

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

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

If CA operand is omitted:

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

Figure 7-1. CA Operand Specifications Valid for Each 3704 and 3705 Channel Adapter Configuration (TYPGEN=EP Only)

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

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

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

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

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

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

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

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

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

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

Figure 7-2 shows, for each CA operand specification, the valid ways the DYNADMP operand can be used to specify subchannels to be used for transfer of dynamic dump data to the host processor.

[JOBCARD={YES}]
{NO }

(generation under OS/VS only)

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

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

[LESIZE=size]

(generation under OS/VS only)

Specifies the OS/VS region size, in K (1024) bytes, to be used by all linkage editor job steps during stage two of program generation. *size* must exceed 200 and be less than 16384 (16,384 K bytes).

If you omit the LESIZE operand, no REGION parameter appears on the EXEC cards for the linkage editor job steps.

If CA=	*Then to allow dynamic dump data transfer over these subchannels:* *	*Code DYNADMP=*
TYPE1	NSC$_1$	(YES,NSC)
TYPE4	NSC$_1$	(YES,NSC)
	ESC$_1$	(YES,*addr1*)
(TYPE4,TYPE4)	NSC$_1$	(YES,NSC)
	ESC$_1$	(YES,*addr1*)
	NSC$_2$	(YES,NONE,NSC)
	ESC$_2$	(YES,NONE,*addr2*)
	NSC$_1$ or NSC$_2$ †	(YES,NSC,NSC)
	ESC$_1$ or ESC$_2$	(YES,*addr1*,*addr2*)
	NSC$_1$ or ESC$_2$ ††	(YES,NSC,*addr2*)
	ESC$_1$ or NSC$_2$ ††	(YES,*addr1*,NSC)

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

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

*ESC – emulation subchannel
NSC – native subchannel

Subscripts indicate channel adapter
^{1}first (base) adapter
^{2}second (expansion) adapter

Figure 7-2. Valid Subchannel Address Specifications for Dynamic Dump Data Transfer

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

Specifies whether or not the channel and line interrupt line trace option is to be included in the program. The trace functions may be initiated from the control panel of the communications controller only if you specify LINETRC=YES. (The *Control Panel Guide* (see Preface) explains the use of the trace option.)

`lines`

Specifies the maximum number of lines that are to be traced concurrently. If you specify inclusion of the trace option and omit this parameter, all lines currently operating in emulation mode can be traced at once. The minimum value of *lines* is 1; the maximum is 352.

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

Specifies whether the generated program is to be loaded into and executed by a 3705-II (MODEL=3705-2), a 3705-I (MODEL=3705), or 3704 (MODEL=3704). (A 3705-I may be specified as either MODEL=3705 or MODEL=3705-1.)

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

```
[NEWNAME={EP001  }]
         {symbol}
```

Specifies the name to be given to the generated program.

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

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

Specifies that a 20-byte data buffer is to be provided for communication lines (1) that are serviced by a type 2 communication scanner, (2) for which you have specified CHNPRI=HIGH in the LINE macro. The buffers, which are permanently assigned to the line, provide extra protection against overruns that can result from temporary slowdowns in channel operation or temporary peaks in data traffic in the network. Lines serviced by a type 2 scanner for which you do not specify OPCSB2=YES have two four-byte buffers. If you omit this operand, OPCSB2=NO is assumed for lines associated with subchannels on a type 1 channel adapter, and OPCSB2=YES is assumed for lines associated with subchannels on a type 4 channel adapter for which CHNPRI=HIGH is specified.

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

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

(generation under OS/VS only)

Specifies the first-level qualifier for OS/VS data sets specified by the LOADLIB, OBJLIB, 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 alphameric characters; the first character must be alphabetic or $, #, or @. (Omit the period [.] that separates the qualifier and the data set name; the generation procedure appends the period to the qualifier you specify.)

NONE

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

SYS1

Specifies that SYS1 is to be used as the qualifier.

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

Specifies whether or not the panel-initiated line test function is to be included in the program. The test routines may be initiated from the control panel of the controller only if you specify TEST=YES. (The *Control Panel Guide* (see Preface) explains the line test function.)

```
[TYPGEN=EP]
```

Specifies that the program to be generated is to perform only emulation functions. You may omit this operand. (If you wish to generate a program that performs network control functions, instead of or in addition to emulation functions, see Chapter 5; do not use this chapter (7), which applies to a program that performs *only* emulation functions.)

```
[TYPSYS={OS }]
        {DOS}
```

Specifies whether stage two of the network control program generation procedure is to be run under OS/VS or DOS/VS.

```
[UNIT=unit type]
```

(generation under OS/VS only)

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

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and SYSDA is assumed for the linkage editing steps.

Note: The utility data sets for the linkage editor must reside on a direct-access device.

```
[UT1=dsname]
```

(generation under OS/VS only)

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

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

`[UT2=dsname]`

(generation under OS/VS only)

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

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

`[UT3=dsname]`

(generation under OS/VS only)

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

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

Configuration Definition Macro Instruction

The CSB macro instruction is the only configuration definition macro required when defining an emulation program.

Chapter 5 contains the configuration definition macros to be used in defining a program that performs network control functions only, or both network control and emulation functions. Use that chapter instead of the present chapter (7) if you wish your program to include network control functions.

CSB Macro Instruction

The CSB macro specifies:

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

Each scanner in the controller must be represented by a CSB macro.

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

`[symbol]`

Provides a name for the macro. *symbol* may be any valid assembler-language symbol; the generation procedure does not check the symbol for validity.

`SPEED=( rate,... )`

Specifies the oscillator (business machine clock) bit rates for up to four oscillators installed in the communication scanner. (Do not confuse this SPEED operand, which specifies the oscillator bit rates, with the SPEED operand of the LINE macro, which specifies the data rate for the communication line.) The speeds must be specified in the same order that the oscillators are installed on the scanner, in ascending order according to speed. Standard oscillator bit rates are shown in Figure 7-3.

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

Rate	*Represents:*	*Rate*	*Represents:*
45	45.5 bps	150	150.0
50	50.0	200	200.0
56	56.89	300	300.0
74	74.2	600	600.0
75	75.0	950	950.0
100	100.0	1200	1200.0
110	110.0	2000	2000.0
134	134.5	2400	2400.0

Figure 7-3. Standard Communications Scanner Oscillator Bit Rates

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

This operand is required.

```
WRAPLN=line addr
```

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

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

Specify the hexadecimal line address without framing quote characters, for example, WRAPLN=02F. The address specified must be within the range shown in Figure 7-4, and must appear in the ADDRESS operand of one of the LINE macros.

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

This operand is required.

```
[MOD={n} ]
     {0}
```

Specifies the location of the communication scanner, as shown in Figure 7-4. The line interface addresses valid for each scanner type and module location are given. (The 3704 has only one module.)

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

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

Valid designations for scanner type are:

If controller is a 3704 (MODEL=3704): TYPE1
TYPE2

If controller is a 3705-I (MODEL=3705): TYPE1 (valid only for MOD=0)
TYPE2
TYPE3 (valid only for MOD=1, 2, or 3)

If controller is a 3705-II (MODEL=3705-2): TYPE2
TYPE3

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

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

Figure 7-4. Location of Communication Scanners and Valid Line Interface Addresses

Network Configuration Macro Instructions

This section contains the teleprocessing network configuration macro instructions (GROUP, LINE) to be used in specifying communication lines that are to operate only in emulation mode. (Chapter 5 contains the network configuration macro instruction to be used in defining a program that performs network control functions only, or both network control and emulation functions. Use that chapter instead of the present chapter (7) if you wish your program to include network control functions.)

GROUP Macro Instruction

A communication line group to be operated in emulation mode consists of lines that have the following characteristics in common:

- All lines in the group are either nonswitched or switched (lines in a nonswitched group may be point-to-point or multipoint or a combination of the two).
- All stations connected to lines in the group are start-stop, or all are binary synchronous. (Lines to which SDLC stations are connected cannot operate in emulation mode.)

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

For each line group, one GROUP macro is required.

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

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

```
Name     Operation   Operands

                                { [XONOFF] [,chars] }
symbol   GROUP       [CHAREC=(  { XON,chars         }  ) ]
                                { XOFF,chars        }
                                { NO,chars          }

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

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

Name	*Operation*	*Operands*
		[,EOB=(char[,F])]
		[,EOT=(char[,F])]
		[,LNCTL={SS }] {BSC}
		[,QUIETCT= { count }] { 0 }
		[,REPLYTO={count}] {3.0 }
		[,TEXTTO={count}] {25.6 }

symbol

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

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

(83B3,115A,TWX lines only)

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

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

Specify this operand as follows:

For TWX terminals:

If you wish the program to recognize, as ending characters:

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

Note: The program will not recognize ENQ (enquiry) characters, or any other characters other than WRU, XON, and XOFF, unless you specify the required characters in this operand.

For 83B3 and 115A terminals:

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

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

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

(World Trade teletypewriters only)

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

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

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

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

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

```
[EOB=(char[,F])]
```

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

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

```
[EOT=(char[,F])]
```

(World Trade teletypewriters only)

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

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

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

Note: Appendix E lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

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

```
[LNCTL={SS }]
       {BSC}
```

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

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

(start-stop lines only)

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

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

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

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

```
[REPLYTO={count}]
        {3.0  }
```

(start-stop lines only)

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

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

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

The maximum value is 51.1 seconds.

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

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

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

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

(start-stop lines only)

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

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

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

Note: The value you specify for *count* is a nominal value. The actual elapsed value may vary anywhere between the nominal value and twice the nominal value. A nominal value of 30 seconds, for example, will result in an actual elapsed interval of from 30 to 60 seconds.

The maximum value is 51.1 seconds.

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

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

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

LINE Macro Instruction

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

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

Name	*Operation*	*Operands*
symbol	LINE	

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

                    SPEED=rate

                    [,AUTO=address]

                    [,BUFSIZE=n]

                    [,CHECK={DCD   }]
                            {NODCD }

                    [,CHNPRI={NORMAL }]
                             {HIGH   }

                    [,CLOCKNG={INT}]
                              {EXT}

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

Name *Operation* *Operands*

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


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

[,DATRATE={HIGH}]
           {LOW }

[,DISABLE={YES}]
           {NO }

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

[,DUPLEX={HALF}]
          {FULL}

[,FEATURE=( [{DUALCODE}] [,{IMEND  }]
             {NODUALCD}     {NOIMEND}

            [,{LRC  }]    [,{SPACE  }])]
              {NOLRC}       {NOSPACE}

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

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

[,MULTI={YES}]
         {NO }

[,NEWSYNC={YES}]
           {NO }

[,PAD={YES}]
       {NO }

[,QUIET={YES}]
         {NO }

[,RING={YES}]
        {NO }

[,TADDR={character}]
         {NONE     }

[,TERM=type]

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

`symbol`

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

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

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

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

The subchannel address must be associated with a type 1 or type 4 channel adapter (not a type 2 or type 3 adapter, which can be used only for network control mode). The subchannel address specified for this line must not be the same as a subchannel address associated with any other line or used by any other program facility (such as the dynamic dump utility).

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

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

Figure 7-5 gives the range of valid line addresses for this operand.

This operand is required.

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

Figure 7-5. Valid Line and Auto Call Interface Addresses for LINE Macro Instruction

SPEED=rate

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

If the modem at the controller is a dual-rate modem (whether the rate used is program-selectable or selected by manual switch), specify the higher of the two rates.

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

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

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

This operand is required.

[AUTO=address]

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

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

If the line does not have the ACU facility, omit the operand.

Figure 7-5 gives the range of valid addresses.

[BUFSIZE=n]

(lines serviced by type 3 scanner only)

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

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

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

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than do smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

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

Note: Do not specify 4-byte buffers (*n*=4) if the line represented by this LINE macro is polled (POLLED=YES) and the size of the poll entries in the service order table is six bytes or more. (A poll entry comprises the address characters, ENQ, and index byte used to poll one station on the line [for example, C1 C1 40 40 ENQ Index]).

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

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

(switched, duplex, start-stop lines in emulation mode only)

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

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

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

(BSC lines only)

Specifies the priority of the emulation subchannel associated with this line with respect to the other emulation subchannels used by the program.

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

CHNPRI=NORMAL is ordinarily appropriate unless the line represented by this LINE macro is to operate at a data rate of 19,200 bps or more, and the majority of the remaining lines are slow speed lines (2400 bps or less).

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

See the *EP Storage Estimates* manual for further information about subchannel priorities.

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

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

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

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

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

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

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

(binary synchronous lines only)

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

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

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

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

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

Note: The control program does not perform code translation for lines in emulation mode. The program transmits data over a communication line in the same transmission code in which it received the data over the emulation subchannel, and vice versa. However, the program must be aware of the code used on BSC lines (but not start-stop lines).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Most modems do not require the long timeout, and DISABLE=NO is therefore the appropriate value. If in doubt, consult the supplier or installer of the modem.

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

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

`lineaddress`

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

A

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

B

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

DUALCOM=NONE

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

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

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

```
LINE1 LINE  ADDRESS=(020,17),  LINE2  LINE  ADDRESS=(021,17),
            DUALCOM=(021,A),               DUALCOM=(020,B),
                 .                                .
                 .                                .
                 .                                .
```

[DUPLEX= {HALF}]
        {FULL}

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

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

[FEATURE=(. . .)]

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

[{DUALCODE}]
{NODUALCD}

(BSC lines only)

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

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

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

[{IMEND }]
{NOIMEND}

(specially-equipped start-stop terminals only)

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

[{LRC }]
{NOLRC}

(start-stop terminals only)

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

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

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

```
[{SPACE  }]
 {NOSPACE}
```

(teletypewriter terminals only)

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

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

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

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

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

Specifies whether the communication line (1) is enabled immediately after the communication controller has been loaded (IPL) or the System Reset key has been pressed (MODEM=OPTION1); or (2) is disabled after the controller has been loaded or the System Reset key has been pressed, and must subsequently be enabled by command from the access method (MODEM=OPTION2). (Until the line is enabled, most commands issued to the line by the access method will result in an intervention required indication to the access method.) (Specify MODEM=NTT if the modems on the line are those of Nippon Telegraph and Telephone Co.)

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

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

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

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

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

(IBM 2845, 2848 only)

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

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

(BSC lines only)

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

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

If you omit the NEWSYNC operand, NEWSYNC=YES is assumed if you specify LNCTL=BSC, DUPLEX=FULL, CLOCKNG=EXT, and DIAL=NO. (All operands must be so specified.) If you omit the operand and specify LNCTL=BSC, but any of the three remaining operands is not as shown, NEWSYNC=No is assumed. For LNCTL=SS, the NEWSYNC operand has no meaning.

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

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

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

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

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

Specifies whether or not the program is to observe a "long line quiet" timeout of 25.6 seconds when receiving from the line represented by this LINE macro. If you specify QUIET=YES, the program observes the long timeout. If you specify QUIET=NO (or omit the operand), the normal timeout of 3.0 seconds is observed. QUIET=YES should not be specified if FEATURE=IMEND is specified for the line represented by this LINE macro.

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

(not applicable to U.S. and Canada)

Specifies whether or not the ring indicator mode of 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 modem supplier or installer whether it has a "ring indicator interface" lead.

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

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

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

(tributary controller on BSC line only)

Specifies, if this program is to be executed in a tributary controller on a BSC line, the one-character address you wish to assign to this controller.

Code TADDR=character, where *character* is one or two EBCDIC characters specified in hexadecimal form (for example, C1 or C1C2). A one-character address (C1) represents the station address. The second character in a two-character address (C1C2) represents the group address. Each address must be equivalent to the address the access method sends to solicit input from the tributary controller.

```
[TERM=type]
```

Specifies the type of station with which the program will communicate over the line represented by this LINE macro. It must be one of the types listed in Figure 7-6.

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

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

It is normally appropriate to specify UNITXC=YES (or omit the operand), which causes the program to signal Unit Exception status upon receiving an EOT.

However, if read and write commands within the access method are command chained, UNITXC=NO may be appropriate. UNITXC=NO, by suppressing the Unit Exception indication, prevents the command chain from being broken. (Unit Exception status always breaks the command chain.)

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

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

Figure 7-6. Values for TERM Operand of LINE Macro (Lines in Emulation Mode Only)

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.

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

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

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

`[symbol]`

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

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

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

> `mask1...mask4`
>
> Specifies eight-bit binary sequences (for example, 00101000) constituting the masks. For scanning purposes, the line interface bases (LIB) serviced by a type 3 scanner is divided into eight portions. The eight bit positions of a mask correspond to the eight portions (0-7) within all LIBs serviced by the scanner. See Appendix K for an illustration.

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

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

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

The bit settings you specify should correspond to the high speed lines requiring increased scanning. For each such line interface installed in the controller, a high speed select feature is present that blocks the attachment of lines to all but the lowest address in the corresponding LIB portion.

Example: Assume that a 3705 having three modules is equipped with type 3 scanners in the first and second expansion modules, but not in the base module. If high speed select features are present in the second scanner for LIB portions 3 and 7 (thus allowing high speed lines to be attached to addresses 0A6 and 0AE), you would specify HSPDSEL= (,00010001,00000000). The first comma signifies that no type 3 scanner is installed in the base module; the first eight-bit mask indicates that increased scanning frequency is required for addresses 0A6 and 0AE in LIB portions 3 and 7, respectively; and the second mask indicates that no addresses in the second expansion module (scanner position 3) require increased scanning frequency.

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

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

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

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

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

`limit1...limit4`

Specifies the scan limits for each installed type 2 or type 3 scanner. Each limit can be from 0 to 3; these values have the meanings shown below. *Limit1* specifies the scan limit for the first scanner position (base module), *limit2* for the second position (first expansion module), etc. All addresses associated with a scanner are scanned if the scan limit for that scanner is 0. Scan limits of 1, 2, and 3 reduce the number of addresses scanned to 8, 48, and 16, respectively. If a scanner position does not contain a type 2 or type 3 scanner, code a comma for the corresponding limit [for example, SCANCTL=(limit1,,limit3,,asmask)]. If a type 2 or type 3 scanner is installed but you specify no limit, the generation procedure assigns the appropriate limit based on the range of actual installed addresses and line speeds as specified in the LINE macros.

The scan limits have the following meanings:

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

*If 3704 is equipped with two LIBs and the speed of any line(s) is 19,200 bps, specify a scan limit of 2 and do not use address substitution.

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 substitution; all addresses enabled.
3	1	Address substitution is to be performed for address 6 in LIB position 1. Addresses 8 and 9 in all LIB positions are disabled.
3	0	No address substitution; all addresses enabled.

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

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.

Part V.
OS/VS Generation and Utilities

This Part (Chapters 8, 9 and 10) is of interest only to users needing to generate, load, and dump a network control program or emulation program under OS/VS. You may wish to remove and file one or more of these chapters if unneeded for your installation.

Chapter 8: Program Generation Under OS/VS

The control program generation procedure under OS/VS is a two-stage process consisting of a series of jobs executed under control of the operating system. The procedure need not be executed in the host processor attached to the communications controller into which the program will be loaded.

Stage one of the generation procedure is an assembly job using either the communications controller assembler (CWAX00) or an OS/VS assembler to prepare, from the program generation macros defining the program, a job stream (sequential data set) for input to stage two. The data set may be placed on cards, tape, or a direct-access device. The stage one output (stage two input) contains (1) data constants, (2) macros that will cause stage two to generate the control tables and conditionally assemble the required program modules, (3) job control statements for stage two, and (4) linkage editor control statements.

Note: Only one stage one generation may be assembled in a job.

Stage two of the generation procedure first uses the communications controller assembler (CWAX00) to assemble the control tables and those program modules that require conditional assembly, and places the resultant object modules on the library you have specified in the OBJLIB operand of the BUILD macro. Stage two then link edits these modules and other, preassembled, modules (located in SYS1.OBJ3705) into a network control or emulation program load module and places this module on the library you have specified in the LOADLIB operand of the BUILD macro. From this library either the access method loader or the independent loader provided in the system support programs may load the control program into the communications controller.

Note: If unresolved external references appear in the linkage editor output, disregard any such references that (1) are listed in the Memo to Users that accompanies the NCP distribution medium as received from the IBM Program Information Department (for unresolved references in a network control program [with or without PEP]); or (2) are listed below in this chapter (for unresolved references in an emulation program).

For an NCP (with or without PEP), stage two also produces a *resource resolution table* load module and, if you have coded any block handling routines, a *block handler set resolution table* load module. These load modules contain information required by the access method, which must obtain them from the library you have specified in the LOADLIB operand.

The generation procedure is the same for complete generation and for partial generation. (Partial generation is possible only if the program includes network control functions; that is, NCP, NCP-LR, NCP-R, PEP, or PEP-LR is specified in the TYPGEN operand of the BUILD macro.) The only difference is that fewer modules are conditionally assembled in a partial generation; in some cases only the control tables are reassembled.

Partial generation is possible only if the object library previously produced during the complete generation procedure is available. (The object library contains the conditionally assembled modules.) You should therefore always save the object library. In addition, you should retain the stage one and stage two assembly listings produced by the complete generation procedure that preceded the partial generation.

The same source deck used to generate a given control program may be modified and used to generate a different program, through either a partial or a complete generation procedure. Care must be exercised in specifying program and library names associated with the subsequent program, as follows. (1) If the new program is to replace the original one, and the original form of the object modules placed in the object library will not be subsequently needed, the NEWNAME, LOADLIB, and OBJLIB operands of the BUILD macro in the source deck need not be changed. The subsequent control program load module will be cataloged in place of the original one and the original load module will not be retrievable thereafter. (2) If both the original and the subsequent load modules will be needed, either change the NEWNAME operand to specify a different name for the subsequent load module or change the LOADLIB operand to specify a different library.

In addition, be sure to save the object modules associated with the original load module before submitting the modified source deck to the generation procedure that will produce the new load module. Failure to save the old object modules will cause them to be replaced in the object library with the modified object modules; this would prevent your later re-generating the original program should you wish to do so. To save the object modules, simply specify a different data set name in the OBJLIB operand of the BUILD macro before submitting the revised source deck to the subsequent generation procedure (either partial or complete). The original object modules will thus remain unaltered in their data set. Another step is necessary if the subsequent generation is partial: you must copy the original object modules onto the new data set you specify in the OBJLIB operand of the modified source deck. This step allows the partial generation procedure to obtain the original modules for updating and link editing into the new load module.

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

Caution (VTAM Users): Because the VTAM initialization does no validity checking of parameters validity checked by the NCP generation procedure, it is imperative that the NCP source statements be entirely free of errors before being given to the VTAM initialization procedure. Therefore, the network control program must be assembled, via stage one of the generation procedure, and reassembled if necessary, until the stage one output listing shows no MNOTE statements having severity codes of 4 or 8.

Figures 8-1 and 8-2 show the content of the stage one input job stream and stage one output (stage two input) job stream using the communications controller assembler.

Providing User Job Cards

The format of the stage two job cards produced when you specify JOBCARD=YES or JOBCARD=MULTI in the BUILD macro (valid only if NCP, NCP-LR, NCP-R, PEP, or PEP-LR is specified in the TYPGEN operand) is:

```
//NCPGENnn    JOB   1,'NCPSYSGEN',MSGLEVEL=1
```

where *nn* is a sequential identification number provided by the program generation procedure. You may provide a job card with different parameters, before generating the program, by using the OS/VS IEBUPDTE utility program to modify the job statement information in member JOBCARD of the stage one macro library. (Refer to the *OS/VS Utilities* manual, GC35-0005, for information on the IEBUPDTE program.)

When you modify the job statement, the *name* parameter must be *jobname* & SNOA. *jobname* consists of one through six alphameric characters (including &, @, #), the first being alphabetic or &, @, or #. &SNOA is a counter that will be incremented by the program generation procedure to provide unique job names. (See the *OS/VS Job Control Language* manual, GC28-0618, for information on the job statement.)

```
//STAGE1     JOB      MSGLEVEL=1
//STEP1      EXEC     PGM=CWAX00
//SYSLIB     DD       DSN=SYS1.GEN3705
        .
        .
    (JCL statements for assembler)
        .
        .
//SYSIN      DD       *
        .
        .
             BUILD    ...
                      ...
        .                    Control
        .                    Program
        (etc.)               Generation
        .                    Macro
        .                    Statements
             GENEND   ...
             END
/*
```

Figure 8-1. OS/VS Generation Stage One Input Job Stream

The following example illustrates how the job statement may be changed:

```
//CHNGJOB   JOB     (G40,060,SG,-,2),NAME,MSGLEVEL=1
//UPDATE    EXEC    PGM=IEBUPDTE
//SYSPRINT  DD      SYSOUT=A
//SYSUT1    DD      DSN=SYS1.GEN3705,DISP=OLD
//SYSUT2    DD      DSN=SYS1.GEN3705,DISP=OLD
//SYSIN     DD      *
./          REPL    NAME=JOBCARD,LIST=ALL
./          NUMBER  NEW1=100,INCR=100
            PUNCH   '//MYGEN&SNOA   Job (81,3,B62),MYNAME,'
            PUNCH   '//MSGLEVEL=(1,1),CLASS=A'
./          ENDUP
/*
```

```
//NCPGEN1   JOB     MSGLEVEL=1,...
//S1        EXEC    PGM=CWAX00,...   (communications controller assembler)
//SYSLIB    DD      DSN=SYS1.MAC3705
                                    Two assembly job steps
                                    for control tables plus
    (JCL statements for assembler)  one assembly job step
                                    for each program module
                                    requiring conditional
//SYSIN     DD      *               assembly (JOB card is
                                    provided for each
                                    assembly job step only if
    (Data for conditional assembly) JOBCARD=MULTI is coded
                                    in BUILD macro)

/*
//S2        EXEC    PGM=CWAX00,...
//SYSLIB    DD      DSN=SYS1.MAC3705


    (JCL statements for assembler)


//SYSIN     DD      *


    (Data for conditional assembly
/*


//Sn        EXEC    PGM=IEWL,...   (First linkage editor job step)


    (JCL statements for OS/VS linkage editor)


//SYSIN     DD      *


    (INCLUDE statements for linkage editor)


/*
//Sn+1      EXEC    PGM=IEWL,...   (Second linkage editor job step)


    (JCL statements for OS/VS linkage editor)


//SYSIN     DD      *


    (INCLUDE statements for linkage editor)


                    END
/*
//


    (INCLUDE statements specify object modules obtained from
    SYS1.OBJLIB and object modules obtained from the library
    specified in the OBJLIB operand of the BUILD macro.  The
    load module is placed on the library specified in the
    LOADLIB operand.)
```

Figure 8-2. OS/VS Generation Stage One Output (Stage Two Input) Job Stream

Chapter 9: Loader Utility under OS/VS

This chapter explains the use of the independent loader utility provided by OS/VS. (Chapter 12 explains the equivalent DOS/VS utility.) The independent loader utility program is for use when you wish to load a local communications controller before assigning the controller to the access method. VTAM and TCAM also have a facility for loading a communications controller. In many instances, using the access method loader to load a local communications controller may be preferable to using the independent utility. Moreover, *only* the access method loader may be used to load a *remote* communications controller. (This is true because the local controller to which the remote controller is connected must be communicating with the access method before the remote unit can be loaded.)

This chapter describes only the independent loader program provided by the NCP system service programs under OS/VS.

The independent loader must be run as a job or job step under OS/VS. If you wish to load several controllers at the same time, you may do so by a sequence of job steps under the same loader job.

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

Before the loader utility loads the network control program into the controller, it loads a diagnostic routine, called the *initial test* routine. If the initial test routine detects no malfunctions, the loader then loads the network control or emulation program into the controller. If the initial test routine does detect trouble, that routine stops and the loader issues error message IFL004I indicating the fact. The loader will then load the remaining controllers, if any, specified in the loader job.

Loading and execution of the initial test routine is optional (it is run unless you specify its omission in the LOAD control statement), but is recommended because it can detect conditions that can cause later failure of the network control program.

Successful completion of the network control or emulation 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, when the loader job specifies multiple controllers.

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

Messages issued by the loader are given in Appendix C.

Host Processor and Controller Requirements

The virtual storage requirements (OS/VS) are:

- OS/VS1 loader utility operates in a minimum virtual partition.
- OS/VS2 loader utility operates in a minimum virtual region.

Direct-access residence requirements are:

Device	*Tracks*	*Directory Blocks*
2311	14	2
2314	8	2
3330	5	2
3340	8	2

No work data sets are required for loader execution.

The controller module of the loader can be executed in any local communications controller. In order to be loaded, the controller: (1) must have its power on, (2) must be identified to the operating system under which the loader utility is running, (3) must be free to be allocated to the loader job step, and (4) must not be in a program-stop condition.

Note: Once the loader has been started, the load job should not be cancelled before normal completion.

The loader consists of the load modules IFLOADRN, IFLLD1P2, and IFLLD2P2. These modules must be in the SYS1.LINKLIB data set or on a partitioned data set pointed to by a STEPLIB or JOBLIB statement.

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 or emulation program load module to be loaded.
- A data set containing LOAD statements specifying the names of the network control or emulation program load modules and the controller into which each is to be loaded.
- A partitioned data set containing the initial test routine (consisting of load modules IFL3705A, IFL3705B, IFL3705D, and IFL3705E) to be loaded before control program loading. This data set is optional; it may be omitted if the initial test is not desired (as indicated by DIAG=NO 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 independent loader program are as follows:

```
//jobname  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
                   device, 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 controller
                   initial test load modules; not required
                   if DIAG=NO is specified in LOAD statement)

//ccname   DD     (Specifies the unit address of the
                   communications controller to be loaded)

                  One DD statement is required for each
                  communications controller to be loaded.

//SYSIN    DD     (Specifies the data set [input stream]
                   containing the LOAD control statement)

/*
```

Utility Control Statement (LOAD)

There is one utility control statement: the LOAD statement. It specifies (1) which member of the input data set contains the 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 the control program is loaded.

```
  LOAD  LOADMOD=member name,

        3705=ddname

               {Y6}
        [,DIAG={Y8}]
               {NO}
```

LOADMOD=member name

Specifies which member of the input data set indicated by SYSUT1 contains the desired control program load module. The member must be in standard OS/VS 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. (Code 3705 = *ddname* whether the controller to be loaded is a 3705 or a 3704.)

```
      {Y6}
[DIAG={Y8}]
      {NO}
```

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

Example of Job and Utility Control Statements

Assume that 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=3330,       X
//                VOL=SER=111111,DISP=OLD
//SYSUT3   DD     DSN=SYS1.LINKLIB,DISP=SHR
//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 include DIAG=NO and the SYSUT3 DD statement would be omitted.

Chapter 10: Dump Utilities under OS/VS

This chapter explains the use of the independent and dynamic dump utilities under OS/VS. (Chapter 13 explains the equivalent DOS/VS utilities.)

The Independent Dump Utility

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

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

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

To dump a local communications controller you may invoke either a standalone job using the independent dump program or the access method dump facility. To dump a remote communications controller, you must use the access method facility.

Note: Once the dump program has been started, the dump job should not be canceled before normal completion.

If you use the access method facility to dump controller storage, only the first step is executed. To produce a readable dump listing, you must then run a standalone job to execute step two. If, on the other hand, you invoke the independent dump program, step two will be executed immediately after step one. In the job control language, these two steps will appear to be one job step.

The two topics following *Host Processor and Controller Requirements* explain how to use the independent dump utility to (1) dump the controller storage and (2) print the dumped data. For information on dumping controller storage via the access method dump facility, see the *VTAM System Programmer's Guide* or the *OS/VS TCAM System Programmer's Guide.*

Host Processor and Controller Requirements

The dump program consists of eight load modules. Seven of these are executed in the host processor and the other module is executed in the communications controller.

The host processor modules of the dump program are executable in any System/370 that will accommodate OS/VS1 or OS/VS2.

The virtual storage requirements (OS/VS) are:

- OS/VS1 dump utility operates in a minimum virtual partition.
- OS/VS2 dump utility operates in a minimum virtual region.

Direct-access residence requirements are:

Device	*Tracks*	*Directory Blocks*
2311	8	3
2314	4	3
3330	3	2
3340	4	2

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 or type 4 channel adapter; the other, for a controller equipped with a type 2 or type 3 channel adapter.)

Dumping the Controller Storage

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

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

For 3705 equipped with a single channel adapter:

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

For 3704 or 3705 equipped with two channel adapters:

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

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

When step one is complete, the program informs the CPU operator. At this point the controller is idle and can be reloaded with a network control program via the loader utility (or the access method loader facility).

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

Printing the Dump Data

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

- Formatted or unformatted network control program control blocks. Specifying the formatted option causes certain control blocks associated with operations in network control mode to be labelled and printed at the beginning of the dump listing for convenient reference. (Control blocks associated with emulation operations are not formatted.) The control blocks are not formatted if you specify the unformatted option.
- Formatted or unformatted buffer pool. The area of controller storage occupied by the buffer pool is formatted into the individual buffers of which the pool consists, if you specify the formatted option; this is not done if you specify the unformatted option.
- Mnemonic operation codes may be shown or omitted.
- The complete contents of storage may be listed, or any specified portion or portions of storage.

The DUMP Control Statement

The dump program requires one control statement, DUMP. It specifies your choice of the four options mentioned immediately above. The control statement format is:

```
DUMP    [FROMADDR=address]

        [,TOADDR=address]

        [,BUF={Y}]
              {N}

        [,FORMAT={Y}]
                 {N}

        [,MNEMONIC={Y}]
                   {N}
```

`[FROMADDR=address]`

Specifies the lower limit of the controller storage to appear on the listing. If you omit FROMADDR, the listing will start at address X‘200’. (If you specify a value less than X‘200’, error message IFW201I is issued and a dump of the entire storage contents is produced.)

`[TOADDR=address]`

Specifies the upper limit of the controller storage to appear on the listing. If you omit TOADDR, the listing will end at the upper limit of storage. (If you specify a value higher than the upper limit of storage, message IFW201I is issued and a dump of the entire storage is produced.)

```
[BUF={Y}]
     {N}
```

Specifies whether or not the IFLDUMP program is to format the NCP buffer pool. The buffer pool will be formatted only if you specify BUF=Y.

```
[FORMAT={Y}]
        {N}
```

Specifies whether or not the IFLDUMP program is to format the NCP control blocks. These control blocks will be formatted only if you specify FORMAT=Y.

```
[MNEMONIC={Y}]
          {N}
```

Specifies whether or not the IFLDUMP program is to print the mnemonic operation codes in the dump listing. These codes are printed only if you specify MNEMONIC=Y.

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

How to Dump and Print Storage Contents

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

```
//jobname  JOB    (Initiates the job)

//stepname EXEC   (Specifies the program
                   IFLREAD, or the name of a procedure
                   containing the job control statements)

//SYSUT1   DD     (Specifies the communications
                   controller the contents of which
                   are to be dumped)

//SYSUT2   DD     (Specifies the DASD work data set
                   onto which the contents of the
                   controller are to be dumped)

//SYSPRINT DD     (Specifies a sequential data set
                   [system output device, magnetic
                   tape, or DASD volume] onto which the dump
                   program is to place the dump listing)

//SYSIN    DD     (Specifies the data set [input stream]
                   containing the utility control
                   statement, DUMP)

/*
```

Example: Assume that (1) a controller whose unit address is 030 is to be dumped; (2) the dump listing is to show the contents of controller storage from address X'17F0' to the end; (3) the NCP control blocks and buffer pool are to be formatted; and (4) the mnemonic operation codes are to be printed. The control and utility statements would be similar to:

```
//CCDUMP    JOB     123456,SMITH,MSGLEVEL=1
//JOBLIB    DD      DSN=SYS1.DUMPCC,DISP=SHR,UNIT=3330,               X
//                  VOL=SER=333333
//EXEC      EXEC    PGM=IFLREAD
//SYSUT1    DD      UNIT=030
//SYSUT2    DD      UNIT=SYSDA,DISP=NEW,                              X
//                  SPACE=(512,(513),,CONTIG),DCB=(DSORG=DA)
//SYSPRINT DD       SYSOUT=A
//SYSIN     DD      *
            DUMP    FROMADDR=17F0,FORMAT=Y,MNEMONIC=Y,BUF=Y
/*
```

How to Print Storage Contents Only

If the access method facility has been used to dump the contents of controller storage to a direct-access data set, you must run an independent job to produce a readable dump listing. The job control statements are:

```
//jobname  JOB      (Initiates the job)

//stepname EXEC     (Specifies the program IFLDUMP or
                     the name of a procedure containing
                     the job control statements)


//SYSUT2    DD      (Specifies the DASD work data set
                     onto which the storage contents have
                     been 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 DUMP control statement)
```

Example: Assume that a controller has been dumped by VTAM onto a data set called VTAM.DUMPDSET. The control and utility statements required to obtain the dump listing would be similar to:

```
//CCDUMP    JOB     123456,SMITH,MSGLEVEL=1
//EXEC      EXEC    PGM=IFLDUMP
//SYSUT2    DD      DSN=VTAM.DUMPDSET,DISP=OLD
//SYSPRINT DD       SYSOUT=A
//SYSIN     DD      *
            DUMP    FROMADDR=17F0,FORMAT=Y,MNEMONIC=Y,BUF=Y
/*
//
```

The PARM Field Option

| The only PARM field option recognized by the independent dump utility is LINECNT=nn.

```
PARM='LINECNT=nn' or PARM='LC=nn'
```

```
nn
```

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

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

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

```
//STEP1 EXEC PGM=IFLREAD(or PGM=IFLDUMP),PARM='LINECOUNT=40'
```

The Dynamic Dump Utility Under OS/VS (Emulation Mode Only)

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

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

Host Processor and Controller Requirements

The host processor module of the dynamic dump program is executable in any System/370 that will accommodate OS/VS1 or OS/VS2.

The OS/VS1 dynamic dump utility operates in a minimum virtual partition. The OS/VS2 dynamic dump utility operates in a minimum virtual region.

Residence requirements are:

2311 tracks—6, +2 directory blocks
2314 tracks—3, +2 directory blocks
3330 tracks—2, +1 directory block

The amount of work data set space required 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).

If a dynamic dump of trace table entries is requested, the work data set must be large enough to hold all of the trace data being dumped. A tape unit is preferable for this activity.

The controller module of the dynamic dump program uses the network control (native) subchannel of the type 1 or type 4 channel adapter to communicate with the host processor if a type 2 or type 3 channel adapter is installed. If the controller has only a type 1 channel adapter or one or two type 4 adapters, the dynamic dump program uses an emulation subchannel within the range recognized by the program and not used by a line. (The address[es] of the subchannel[s] are specified in the DYNADMP operand of the BUILD macro.)

Input to the Dynamic Dump Utility

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

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

Output from the Dynamic Dump Utility

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

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

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

Dynamic Dump Operational Characteristics

The dynamic dump utility is used when trouble or error conditions indicate that a dynamic dump of controller storage is desirable to help in isolating and fixing a problem.

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

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

Obtaining a Dynamic Dump of Trace Table Entries

The most important function of the dynamic dump utility is its ability to dynamically dump emulation mode line trace entries. Refer to *Utility Control Statements (DYNADMP)* for additional information on the control statements discussed below and to *Example of Dynamically Dumping Trace Table Entries* which follows this section for an illustration of the input stream.

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

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

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

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

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

To stop the trace activity the operator must first respond with 'S' to the message (IFL508I) issued when the trace was initiated. The 'S' response stops the transfer of trace blocks to the host module as soon as the next trace block is received. To stop the trace activity in the controller (which was initiated by the OPTION control statement) a second OPTION statement must be entered with X=3 specified. Alternatively, this statement may be entered at the controller's panel.

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

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

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

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

Utility Control Statements

The dynamic dump utility control statements:

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

In the explanation of each utility control statement, small letters represent parameters for which you supply a value. A combination of capital and small letters in control statements (for example, PRint) indicates that you may specify either the entire statement (PRINT) or only the capitalized part (PR).

The DYNADMP Statement

The DYNADMP statement requests a dump of the entire controller storage or of a specified portion. The controller does not become idle and does not require reloading.

Name	*Operation*	*Operands*

`[symbol]`	`DYnadmp`	`{Dynamic}`
		`{Storage}`
		`{Table  }`

`[symbol]`

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

`Dynamic`

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

`Storage`

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

`Table`

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

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

The DISPLAY Statement

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

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

`[symbol]`

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

`hhhhh`

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

`[n]`

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

The PRINT Statement

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

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

`[symbol]`

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

`START=hh:mm:ss`

Specifies that only those trace blocks which were written to the work data set after time hh:mm:ss (hours:minutes:seconds) are to be printed.

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

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

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

Example 2: Assume that a trace is started just before midnight. If the first trace block was written to the work data set at 23:25:23 (11:25 p.m.), and the last was written at 00:40:57 (12:40 a.m. the following day), then either of the following PRINT statement would produce the intended results:

PRINT	START=23:40:00	(trace entries written to work data set between 11:40 p.m. and 12:40 a.m. are printed)
PRINT	START=00:20:00	(trace entries written to work data set between 12:20 a.m. and 12:40 a.m. are printed)

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

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

The OPTION Statement

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

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

`[symbol]`

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

`ABCDE`

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

F *Function*	*BC* *Data*	*DE* *Bytes*	*Meaning (L=level)*
4	10	xx	Start L2 trace on subchannel *xx*
4	11	xx	Stop L2 trace on subchannel *xx*
4	20	xx	Start L3 trace on subchannel *xx*
4	21	xx	Stop L3 trace on subchannel *xx*
4	30	xx	Start L2 and L3 trace on subchannel *xx*
4	31	xx	Stop L2 and L3 trace on subchannel *xx*
4	70	00	Start L3 trace on trace defined subchannels
4	71	00	Stop L3 trace on trace defined subchannels
4	70	FF	Start L3 trace on all subchannels
4	71	FF	Stop L3 trace on all subchannels

A is used only when two type 4 channel adapters are installed. If A=1 is specified, the subchannel address specified as DE refers to subchannels on the *first* type 4 channel adapter; if A=2 is specified, the subchannel address specified as DE refers to subchannels on the *second* type 4 adapter.

The PAUSE Statement

The PAUSE statement allows control statements to be entered at the console of the host processor after the PAUSE statement is read from the input job stream or entered from the console.

Name	*Operation*	*Operands*
`[symbol]`	`PAUSE`	

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The END Statement

The END statement specifies the end of job and causes termination of the program after the trace output has been printed.

Name	*Operation*	*Operands*
`[symbol]`	`ENd`	

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The SYSIN Statement

The SYSIN statement is used by the operator to cause control statements to be read from the input stream.

Name	*Operation*	*Operands*
	`SYsin`	

The PARM Field Option

The only PARM field option recognized by the dynamic dump utility is LINECOUNT=nn.

`PARM='LINECOUNT=nn'` or `PARM='LC=nn'`

`nn`

Specifies a decimal number from 10 to 99 which represents the number of lines per page to be printed by the dynamic dump utility in producing its printed output.

If the LINECOUNT parameter is omitted, or if it is given but is syntactically incorrect, the default of 55 lines per page is assumed.

Example: To specify that the dynamic dump utility print 40 lines on each page of printed output, code the PARM field option as follows:

```
//STEP1 EXEC  PGM=IFLSVEP,PARM='LINECOUNT=40'
```

Job Control Statements

The OS/VS job control statements:

- Execute or invoke the program
- Define the output data set, the control statement data set, the work data set, and the communications controller.

If a trace entry dump is requested, the work data set must be large enough to hold all of the trace data being dumped. If, however, the work data set is exhausted, the job will abnormally end. It is preferable to use a tape unit for this activity.

The job control statements needed to invoke the dynamic dump utility are as follows:

```
//JOBNAME   JOB   (Initiates the job).
//[name]    EXEC  (Specifies PGM=IFLSVEP or the procedure
                   name if the job control statements
                   reside in a procedure library).
//SYSPRINT  DD    (Defines a sequential output data set.
                   This data set may be written onto a
                   system output device, a magnetic tape
                   volume, or a direct access volume.
                   The data control block's (DCB)
                   blocksize may be specified [optional]).
//SYSUT1    DD    (Defines the communications controller sub-
                   channel over which the EP dynamic dump utility
                   communicates with the host processor.  See Note
                   below.)
//SYSUT2    DD    (Defines a temporary work data set.  The
                   contents of the communications controller
                   are written to this data set [optional].
                   DISP=OLD must be specified).
//SYSIN     DD    (Defines the control data set [optional].
                   The data control block's (DCB) blocksize
                   may be specified.)
Control statements
/*
//
```

Note: See the description of the DYNADMP operand for requirements governing the selection of a subchannel for the dynamic dump function.

Example of Job Control and Dynamic Dump Utility Statements

The following example shows the statements required to dynamically dump, to the work data set, the entire contents of the controller whose subchannel address is 007. After the dump is complete, the contents of the work data set are transferred to the output data set and printed. The job ends without operator intervention.

```
//SVEP      JOB    MSGLEVEL=(1,1),other parameters
//          EXEC   PGM=IFLSVEP
//SYSPRINT  DD     SYSOUT=A
//SYSUT1    DD     UNIT=007
//SYSUT2    DD     UNIT=2400,VOL=SER=SVTAPE,LABEL=(,NL),DISP=OLD,DSN=WORK
//SYSIN     DD     *
DUMP        DY     STORAGE
            PRINT
            END
/*
//
```

Example of Dynamically Dumping Trace Table Entries

The following example shows the statements required to dynamically dump the trace table entries as they are entered in the trace table.

```
//DYNADMP   JOB     MSGLEVEL=(1,1),other parameters
//STEP1     EXEC    PGM=IFLSVEP
//SYSPRINT  DD      SYSOUT=A
//SYSUT1    DD      UNIT=007
//SYSUT2    DD      UNIT=2400,VOL=SER=SVTAPE,LABEL=(,NL),DISP=OLD,DSN=WORK
//SYSIN     DD      *
            PAUSE   (Returns control to console)
/*
//
```

The following will appear at the operator's console:

```
 IEF403I DYNADUMP STARTED TIME=08.12.16 P00
*IEF233A M 282,SVTAPE,,DYNADUMP,STEP1  P00
@08 IFI501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 08, 'option 43023'     (this activates level 2 and level 3
                          trace activity in the emulation pro-
                          gram on subchannel 23.)
 +IFI503I  FUNCTION COMPLETED - 00  P00
@09 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 09,'dy dynamic'        (this starts the transmission of trace
                          entries [64 at a time] to the host
                          module and places them on the work
                          data set)
@10 IFL505E - REPLY 'S' TO STOP TRACE P00
 +IFI508I TRACE BLOCK     1 WRITTEN AT 08:15:24 P00
 +IFL508I TRACE BLOCK   200 WRITTEN AT 08:22:43 P00
 +IFI508I TRACE BLOCK   400 WRITTEN AT 08:31:09 P00
 +IFL508I TRACE BLOCK   600 WRITTEN AT 08:37:58 P00
!
r 10,'s'                 (this stops the transfer of trace blocks
                          to the host module)
 +IFI506I  STOP COMMAND ACKNOWLEDGED P00
 +IFI508I TRACE BLOCK   712 WRITTEN AT 08:40:12 P00
 +IFI503I  FUNCTION COMPLETED - 00 P00
@11  IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 11,'print start=08:35:00' (this causes a printout of only
                             those trace blocks written to the
                             work data set after 8:35 A.M.
                             This should be approximately 150
                             blocks)
 +IFI503I FUNCTION COMPLETED - 00 P00
@12 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 12,'option 43123'      (this halts trace activity on subchannel 23)
 +IFI503I FUNCTION COMPLETED - 00 P00
@13 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END'
!
r 13,'end'               (this terminates the DYNADUMP job)
 IEF404I DYNADUMP ENDED  TIME=08.46.07 P00
//
```

Part VI.
DOS/VS Generation and Utilities

This Part (Chapters 11, 12, and 13) is of interest only to users needing to generate, load, and dump a network control program or emulation program under DOS/VS. You may wish to remove and file one or more of these chapters if unneeded for your installation.

Chapter 11: Program Generation Under DOS/VS

The control program generation procedure under DOS/VS is a three-stage (for NCP or PEP) or two-stage (for EP) process consisting of a series of jobs executed under control of the operating system. The procedure need not be executed in the host processor attached to the communications controller into which the program will be loaded.

Procedure for NCP, PEP

Stage one of the generation procedure is a series of assembly jobs using the communications controller assembler (IFZASM) or a DOS/VS assembler to prepare, from the job stream program generation macros, a job stream program (sequential file) for input to stage two. The file may be placed on cards, tape, or a direct-access device. The stage one output (stage two input) contains (1) data constants, (2) macros that will cause stage two to generate the control tables and conditionally assemble the required program modules, (3) job control statements for stage two, and (4) an assembly step that punches stage two statements.

Stage two of the generation procedure assembles the control tables and those program modules that require conditional assembly, then punches job control and linkage editor control statements.

Stage three catalogs the tables and modules assembled in stage two and then link edits them into a load module. This module is placed on the core image library. From here the CSERV utility must be used to move it to a user-defined file.

Note: If unresolved external references appear in the linkage editor output, disregard any such references that are listed in the Memo to Users that accompanies the program distribution medium as received from the IBM Program Information Department.

The access method loader facility or the independent loader utility may then obtain the load module from the file and load it into the communications controller. (Chapter 12 tells how to use the independent loader utility under DOS/VS.)

Stage three also produces a *resource resolution table* load module and, if you have coded any block handling routines, a *block handler set resolution table* load module. These load modules contain information required by the access method, which must obtain them from the library on which you have placed them in stage three.

The generation procedure is the same for complete generation and for partial generation. (Partial generation is possible only if the program includes network control functions; that is, NCP, NCP-LR, NCP-R, PEP, or PEP-LR is specified in the TYPGEN operand of the BUILD macro.) The only difference is that fewer modules are conditionally assembled in a partial generation; in some cases only the control tables are reassembled.

Partial generation is possible only if the relocatable library previously used during the complete generation procedure is available. (The relocatable library contains the conditionally assembled modules.) You should therefore always save this library. In addition, you should retain the stage one, stage two, and stage three assembly listings produced by the complete generation procedure that preceded the partial generation.

Operator intervention is required between the stages of program generation. Diagnostic messages produced at the end of each stage indicate any errors that may have occurred. If these are serious errors, no job stream or partial job stream is produced. The source statements must be corrected and the stage must be re-executed. If no serious errors occur, the operator initiates the next stage, specifying as input the output of the previous stage. Refer to Appendix B for diagnostic messages that may appear in the output listings.

Caution: Because the VTAM initialization does no validity checking of parameters validity checked by the NCP generation procedure, it is imperative that the NCP source statements be entirely free of errors before being given to the VTAM initialization procedure. Therefore, the network control program must be assembled via stage one of the generation procedure, and reassembled if necessary, until the stage one output listing shows no MNOTE statements having severity codes of 4 or 8.

Figures 11-1, 11-2, and 11-3 show the content of each input job stream using the communications controller assembler.

Procedure for EP

Stage one of the generation procedure is an assembly job using the communications controller assembler (IFZASM) or a DOS/VS assembler to prepare, from the program generation macros, an object module (comprising the EP control tables) and a printed assembly listing. You must direct the object module to a sequential file (cards, tape, or direct-access device).

Stage one does not automatically produce the stage two job stream. You must prepare an input job stream that contains the job control and linkage editor statements required as input to stage two. The linkage editor statements comprise the INCLUDE and ENTRY statements appearing at the end of the stage one assembly listing; the sequence in the job stream must match the sequence in the listing.

Stage two of the generation procedure has three steps (see Figure 11-4). In step one, the MAINT utility places the object module produced by stage one on a private relocatable library that you specify. Step two link edits the object module with the preassembled EP object modules specified by the INCLUDE statements mentioned above, and places the resultant load module (phase) on the private core image library that you specify. Step three uses the CSERV utility to move the load module from the core image library to a private sequential load file from which the loader may obtain it.

Providing User Job Cards

The format of the stage two job cards produced when you specify JOBCARD=YES or JOBCARD=MULTI in the BUILD macro (valid only if NCP, NCP-LR, NCP-R, PEP, or PEP-LR is specified in the TYPGEN operand of the BUILD macro) is:

```
// JOB  module name
```

By submitting a job that modifies the job card format, as follows, you may provide different job cards before initiating the program generation procedure.

```
// JOB      CHGJOBCD
// EXEC     ESERV
   GENEND
   DSPCH    F.ASMJCL
)  COL 73,4
)  REP 1228
   PUNCH    '// JOB name'
)  END
/&
```

The preceding job creates a new ASMJCL macro. You must then assemble the macro with IFZASM and catalog it.

```
// JOB     jobname
// PAUSE   (Before executing stage one, assign appropriate
            source statement and private relocatable libraries)
// EXEC    IFZASM

   BUILD
        .  NCP generation
        .  macro
        .  statements
   GENEND
   END
/&
```

Figure 11-1. DOS/VS Generation Stage One Input Job Stream—NCP,PEP,EP

This input job stream is produced automatically by stage one of the generation procedure.

```
// JOB      jobname
// PAUSE    (Before executing stage two,
            assign appropriate libraries)
// OPTION   DECK
// EXEC     IFZASM
   PUNCH    '// JOB     jobname'
   PUNCH    '// EXEC    MAINT'
   PUNCH    ' CATALR    module name'

      (Source code for conditionally assembled modules)

   END
/*
// EXEC     IFZASM
   PUNCH    ' CATALR module name'

      (Source code for conditionally assembled modules)
   END

/*
      .
      .     (Other conditional assemblies as in above step)
      .

// EXEC     IFZASM
   PUNCH    ' CATALR    INITINCS'
   PUNCH    ' INCLUDE   ...' (Include statements for UBHR
                              modules, if specified)
   PUNCH    ' INCLUDE   ...' (Include statements for initial-
                              ization routines and tables)
   PUNCH    ' CATALR    LOADINCS'
   PUNCH    ' ACTION    MAP,NOAUTO'
   PUNCH    ' PHASE     NCP001,+0'
   PUNCH    ' INCLUDE   ...' (Include statements for remainder
                              of program modules)
      .         .
      .         .
      .         .
   PUNCH    '/*'
   PUNCH    '// OPTION  CATAL' (If program includes UBHRs,
                                the option is LINK)
   PUNCH    ' INCLUDE   LOADINCS'
   PUNCH    ' INCLUDE   INITINCS' (Omit this statement if
                                   program contains UBHRs)
   PUNCH    '// EXEC    LNKEDT'
   PUNCH    '/&&'
   END
/&
```

Figure 11-2. DOS/VS Generation Stage Two Input Job Stream—NCP,PEP

This input job stream is produced automatically by stage two of the generation procedure.

```
// JOB      jobname
// PAUSE    (Before executing stage three,
            assign appropriate libraries)
// EXEC     MAINT
   CATALR   module name
      .
      . (Object code)
      .
   CATALR   module name
      .
      . (Object code)
      .
      (etc.)
      .
      .
      .
   CATALR   INITINCS
   INCLUDE  ...(Include statements for UBHR modules, if
                present)
   INCLUDE  ...(Include statements for Initialization routines
                and tables)
   CATALR   LOADINCS
   ACTION   MAP,NOAUTO
   PHASE     phasename,+0  (Include statements for
   INCLUDE   module-1      remainder of
   INCLUDE   module-2      object modules)
      .
      .
      .
   INCLUDE   module-n
/*
// OPTION   CATAL  (If program includes UBHRs, the option
                    is LINK)
   INCLUDE LOADINCS
   INCLUDE  INITINCS (This statement omitted if program
                      contains UBHRs)
// EXEC     LNKEDT
/&
```

Figure 11-3. DOS/VS Generation Stage Three Input Job Stream–NCP,PEP

This input job stream must be prepared by the user and submitted to stage two of the generation procedure.

```
(step one)
//  JOB
//  PAUSE    (Assign appropriate private relocatable and core
             image libraries)


//  EXEC     MAINT
      (object deck)
/*
(step two)
//  OPTION   CATAL
    ACTION   MAP,NOAUTO,...
    PHASE    phasename,+0
    INCLUDE  module-1         (these Include and Entry statements
    INCLUDE  module-2         must appear in the same sequence as
       .                      listed in stage one assembly
       .                      listing)
       .
    INCLUDE  module-n
    ENTRY    CYASTART
//  EXEC     LNKEDT           (places EP load module [phase] on
                              private core image library)
(step three)
    DLBL     IJSYSPH,'phasename'   (defines disk area for load module)
    EXTENT   SYSPCH,...
    ASSGN    SYSPCH,X'xxx'
//  EXEC     CSERV            (moves load module from private core
    PUNCH    phasename        image library to load file specified
/*                            in preceding DLBL and EXTENT statements)
/&
    CLOSE    SYSPCH,X'xxx'
/&
```

Figure 11-4. DOS/VS Generation Stage Two Input Job Stream—EP

Chapter 12: Loader Utility under DOS/VS

This chapter explains the use of the independent loader utility provided by DOS/VS. (Chapter 9 explains the equivalent OS/VS utility.)

The independent loader utility program is for use when you wish to load a local communications controller before assigning the controller to VTAM. VTAM also has a facility for loading a communications controller. In many instances, using the VTAM loader to load a local communications controller may be preferable to using the independent utility. Moreover, *only* the VTAM loader may be used to load a *remote* communications controller. (This is true because the local controller to which the remote controller is connected must be communicating with VTAM before the remote unit can be loaded.)

This chapter describes only the independent loader program provided by the NCP system service programs under DOS/VS.

The independent loader must be run as a job or job step under DOS/VS. If you wish to load several controllers at the same time, you may do so by a sequence of job steps under the same loader job.

The loader has two modules. One is an operating system utility that may be invoked as any other DOS/VS utility. The other module runs in the local communications controller. When the loader is invoked, the controller module is contained within a data area in the host processor loader module. The host processor module loads the controller module into the controller via an initial program load (IPL) command.

Before the loader utility loads the network control program into the controller, it loads a diagnostic routine, called the *initial test* routine. If the initial test routine detects no malfunctions, the loader then loads the network control or emulation program into the controller. If the initial test routine does detect trouble, that routine stops and the loader issues error message IFU004I indicating the fact. The loader will then load the remaining controllers, if any, specified in the loader job.

Loading and execution of the initial test routine is optional (it is run unless you specify its omission in the LOAD control statement), but is recommended because it can detect conditions that can couse later failure of the network control program.

Successful completion of the network control or emulation 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, when the loader job specifies multiple controllers.

Syntax errors in the LOAD statement or permanent I/O errors occurring during loading are indicated by messages sent to the message file (SYSLST).

Messages issued by the loader are given in Appendix C.

Host Processor and Controller Requirements

The DOS/VS loader utility operates in a minimum virtual partition. Direct-access residence requirements are:

Core image library:

Device	*Tracks*
2314	7
3330	4
3340	7

Relocatable library:

Device	*Tracks*
2314	5
3330	3
3340	5

No work files are required for loader execution.

The controller module of the loader can be executed in any local communications controller. In order to be loaded, the controller: (1) must have its power on, (2) must be identified to the operating system under which the loader utility is running, (3) must be free to be allocated to the loader job step, and (4) must not be in a program-stop condition.

The loader consists of the load module IFULOAD.

Inputs to the Loader Program

Either two or three files are used as input to the loader:

- A DASD file (input file) containing the network control or emulation program load module to be loaded.
- A file containing LOAD statements specifying the names of the network control or emulation program load modules and the controller into which each is to be loaded.
- A file containing the initial test routine (consisting of load modules IFU3705D and IFU3705E) to be loaded before control program loading. This file is optional; it may be omitted if the initial test is not desired (as indicated by DIAG=NO in the LOAD statement).

Outputs from the Loader Program

The loader produces one output file, the message file (SYSLST). This contains completion or error messages produced by the loader.

Job Control Statements

The job control statements needed to invoke the independent loader program are as follows:

```
// JOB    (Initiates the job)

// ASSGN  (Specifies the unit address of the communications
           controller to be loaded. This statement may be
           omitted if a permanent assignment exists for
           the communications controller.)

// DLBL   (Defines a sequential file that contains a
           suitable formatted load module.)

// EXTENT

// ASSGN  (Assigns the file defined in the previous
           DLBL and EXTENT statements.)

// DLBL   DIAGFLE,'file-id'  Defines the sequential file
                             that contains the initial
// EXTENT SYS008,vol.id,1    test routine. Required only
                             if DIAG=Y6 or Y8 is specified
// ASSGN  SYS008,X'ccu'      or implied on any LOAD
                             statement.

// EXEC   (Specifies the program name, IFULOAD)
```

Utility Control Statement (LOAD)

There is one utility control statement: the LOAD statement. It specifies (1) which member of the input data set contains the 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 the control program is loaded.

```
LOAD   LOADMOD=file name

       3705=SYSxxx

               {Y6}
       [,DIAG={Y8}]
               {NO}

                 {2311}
       [,DEVICE={2314}]
                 {3330}
                 {3340}
```

`LOADMOD=file name`

Specifies the name of the file that contains the control program load module. This name must be the same as the file name specified in the DLBL statement.

```
3705=SYSxxx
```

Specifies the symbolic name of the communications controller to be loaded.

```
       {Y6}
[DIAG={Y8}]
       {NO}
```

Specifies whether the loader is to load the initial test routine into a 3704 or a 3705 without extended addressing (DIAG=Y6), a 3705 with extended addressing (DIAG=Y8), or is not to load the routine at all (DIAG=NO).

```
         {2311}
[DEVICE={2314}]
         {3330}
         {3340}
```

Specifies the type of direct access device on which the control program load module resides.

Example of Job and Utility Control Statements

Assume that the controller whose unit address is 001 is to be loaded, first with the initial test routine named INITTEST (residing on a file named DIAGFLE) and then with the NCP load module named NCP3MOD (residing on a file named NCPFILE).

The control and utility statements would be similar to:

```
//  JOB      LOAD3705
//  ASSGN    SYS007,X'001'
//  DLBL     DIAGFLE,'INITTEST'
//  EXTENT   SYS008,111111
//  ASSGN    SYS008,X'131'
//  DLBL     NCPFILE,'NCP3MOD'
//  EXTENT   SYS005,111111
//  ASSGN    SYS005,X'131'
//  EXEC     IFULOAD
    LOAD     LOADMOD=NCPFILE,3705=SYS007,DEVICE=3330,DIAG=Y8
/*
/&
```

If the initial test is not wanted, the LOAD statement would include DIAG=NO instead of DIAG=Y8 and you would omit the third, fourth, and fifth statements (DLBL, EXTENT, and ASSGN).

Link-Editing Modules from the Relocatable Library

If the host processor modules of the loader utility are cataloged in the relocatable library, the following control statements can be used for link-editing them into the core image library:

```
//  JOB      LINKLOAD
//  OPTION   CATAL
    INCLUDE  IFULINK
//  EXEC     LNKEDT
/&
```

Chapter 13: Dump Utilities under DOS/VS

This chapter explains the use of the independent and dynamic dump utilities under DOS/VS. (Chapter 10 explains the equivalent OS/VS utilities.)

The Independent Dump Utility

The independent dump utility program is used to dump the storage contents of a 3704 or 3705 communications controller. It accomplishes this by a two-step process:

Step 1: The storage contents of the controller are copied to a direct access file (SYS008).

Step 2: A printable copy of the controller's storage contents is produced and placed on a sequential output file (SYLST). The SYS008 file from step one serves as input to this step.

To dump a local communications controller you may invoke either a standalone job using the independent dump program or the VTAM dump facility. To dump a remote communications controller, you must use the VTAM facility.

If you use the VTAM facility to dump the controller storage, only the first step is executed. To produce a readable dump listing, you must then run a standalone job to execute step two. If, on the other hand, you invoke the independent dump program, step two will be executed immediately after step one. In the job control language, these two steps will appear to be one job step.

The two topics following *Host Processor and Controller Requirements* explain how to use the independent dump utility to (1) dump the controller storage and (2) print the dumped data. (See the *VTAM System Programmer's Guide* for information on dumping controller storage via the VTAM dump facility.)

Host Processor and Controller Requirements

The host processor module of the dump program is executable in any System/370 that will accommodate DOS/VS.

The DOS/VS dump utility operates in a minimum virtual partition.

Residence requirements are:

Core image library:

Device	*Tracks*
2314	9
3330	6
3340	9

Relocatable library:

Device	*Tracks*
2314	14
3330	9
3340	14

The amount of work file 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 or type 4 channel adapter; the other, for a controller equipped with a type 2 or type 3 channel adapter.)

Dumping the Controller Storage

Dumping from the controller to the direct access work file is performed by the first step of the dump utility. This step first transfers into the communications controller a module containing the utility code needed for the controller to participate in the dumping process. (This module is contained within the dump program in the host processor until transferred to the controller via an initial program load [IPL] command.)

Step one always transfers the entire contents of controller storage and local store registers to the host processor, which places them on a direct access file. However, a small portion of the storage data is overlaid by the dumping process. (The storage area does appear in the listing, but consists of read-only storage [ROS] or dump utility code.) The areas not available are as follows:

For 3705 equipped with a single channel adapter:

Hexadecimal addresses:	Overlaid by:
0 through 1FF	ROS
400 through 4E7	dump utility code
700 through 707	ROS
780 through 79F	ROS

For 3704 or 3705 equipped with two channel adapters:

Hexadecimal addresses:	Overlaid by:
0 through 3FF	ROS
400 through 4E7	dump utility code
700 through 70F	ROS
780 through 79F	ROS

Note: The contents of the controller's external registers are not transferred to the host processor. If the contents of these registers must be examined, they must be displayed on the controller's operator panel and the contents noted before the dump utility is invoked.

When the dumping process is complete, the program informs the CPU operator. At this point the controller is idle and can be reloaded with a network control program via the loader utility (or the VTAM loader facility).

For the job control statements needed to both dump and print the contents of controller storage, see the topic below, *How to Dump and Print Storage Contents.*

Printing the Dump Data

The second step of the dump utility converts all or a selected part of the dumped data to printable form, then places the data on a sequential output file. The output listing shows the hexadecimal representation of controller storage and register contents, and gives the character equivalents of all EBCDIC bit patterns that represent characters. Beyond this, four options are available, as specified by the DUMP control statement:

- Formatted or unformatted network control program control blocks. Specifying the formatted option causes certain control blocks associated with operations in network control mode to be labelled and printed at the beginning of the dump listing for convenient reference. (Control blocks associated with emulation operations are not formatted.) The control blocks are not formatted if you specify the unformatted option.

- Formatted or unformatted buffer pool. The area of controller storage occupied by the buffer pool is formatted into the individual buffers of which the pool consists, if you specify the formatted option; this is not done if you specify the unformatted option.
- Mnemonic operation codes may be shown or omitted.
- The complete contents of storage may be listed, or any specified portion or portions of storage.

The DUMP Control Statement

The dump program requires one control statement, DUMP. It specifies your choice of the four options mentioned immediately above. The control statement format is:

```
DUMP    [FROMADDR=address]

        [,TOADDR=address]

        [,BUF={Y}]
              {N}

        [,FORMAT={Y}]
                 {N}

        [,MNEMONIC={Y}]
                   {N}
```

```
[FROMADDR=address]
```

Specifies the lower limit of the controller storage to appear on the listing. If you omit FROMADDR, the listing will start at address X'200'. (If you specify a value less than X'200', error message IFW201I is issued and a dump of the entire storage contents is produced.)

```
[TOADDR=address]
```

Specifies the upper limit of the controller storage to appear on the listing. If you omit TOADDR, the listing will end at the upper limit of storage. (If you specify a value higher than the upper limit of storage, message IFW201I is issued and a dump of the entire storage is produced.)

```
[BUF={Y}]
     {N}
```

Specifies whether or not the IFUDUMP program is to format the NCP buffer pool. The buffer pool will be formatted only if you specify BUF=Y.

```
[FORMAT={Y}]
        {N}
```

Specifies whether or not the IFUDUMP program is to format the NCP control blocks. These control blocks will be formatted only if you specify FORMAT=Y.

```
[MNEMONIC={Y}]
          {N}
```

Specifies whether or not the IFUDUMP program is to print the mnemonic operation codes in the dump listing. These codes are printed only if you specify MNEMONIC=Y.

Note: If the controller storage to be dumped contains line trace information, omission of mnemonic operation codes is recommended.

How to Dump and Print Storage Contents

To use the dump utility to both dump and print the controller storage contents, you provide job control statements only for the first step and the DUMP statement(s) for the second step. The first step generates the required control statements for the second step. The statements are:

```
// JOB                     (Initiates the job)

// EXEC                    (Specifies the first step)

// ASSGN  SYS007           (Specifies the unit address  of the
                           communications controller to be
                           dumped.  You may omit this statement
                           if a permanent assignment was made
                           for the controller during the NCP
                           generation process.)

// ASSGN  SYS008,X'nnn'    (Specifies the unit address  of the
                           direct-access device that contains
                           the dump file.  You must define the
                           file with DLBL and EXTENT statements
                           if the file was not permanently as-
                           signed.  The DLBL statement must have
                           a file-id of NCPDUMP.)
```

Note: The symbolic unit address of the controller and the dump file must be SYS007 and SYS008, respectively, as shown above.

Example: Assume that (1) a controller whose unit address is 019 is to be dumped; (2) the dump listing is to show the contents of controller storage from address '17F0' to the end; (3) the NCP control blocks and buffer pool are to be formatted; and (4) the mnemonic operation codes are to be printed. The control and utility statements would be similar to:

```
// JOB     DUMP
// ASSGN   SYS007,X'019'
// DLBL    NCPDUMP,'NCP3DUMP',,DA
// EXTENT  SYS008,111111,,,2000,80
// ASSGN   SYS008,X'131'
// EXEC    IFUREAD
   DUMP    FROMADDR=17F0,FORMAT=Y,MNEMONIC=Y,BUF=Y
/*
/&
```

How to Print Storage Contents Only

If the VTAM facility has been used to dump the contents of controller storage to a direct-access file, you must run an independent job to produce a readable dump listing. The job control statements are:

```
// JOB                      (Initiates the job)
// ASSGN  SYS008,X'nnn'     (Specifies the unit address of the
                            direct-access device that contains
                            the dump file.)  Unless permanently
                            defined, the file must be defined
                            with DLBL and EXTENT statements.
                            Note that VTAM applies a file-id of
                            'NCPDUMP' to the dump file it cre-
                            ates.
// EXEC   IFUDUMP
```

Example: Assume that VTAM has dumped the storage contents of a communications controller onto a file whose unit address is 131. The control and utility statements required to obtain the dump listing would be similar to:

```
// JOB      DUMPRT
// ASSGN    SYS008,X'131'
// DLBL     NCPDUMP,'NCP3DUMP',,DA
// EXTENT   SYS008,111111
// EXEC     IFUDUMP
   DUMP     FROMADDR=17F0,FORMAT=Y,MNEMONIC=Y,BUF=Y
/*
/&
```

Link-Editing Modules from the Relocatable Library

If the host processor modules of the independent dump utility are cataloged in the relocatable library, the following control statements can be used for link-editing them into the core image library:

```
//  JOB      LINKDUMP
//  OPTION   CATAL
    INCLUDE  IFUWLINK
//  LBLTYP   NSD(1)
//  EXEC     LNKEDT
/&
```

The Dynamic Dump Utility Under DOS/VS (Emulation Mode Only)

The dynamic dump utility is an optional utility that provides the following services that are useful in debugging. This utility can be used to:

- Obtain, without terminating the execution of the program, (1) a storage dump (from location 0 through the end of storage) of the communications controller, or (2) a display, on the operator's console at the host processor, of portions of controller storage (up to 144 bytes) starting at any location, or (3) a dump of the emulation mode trace table only.
- Activate or deactivate the emulation mode line trace function.
- Obtain a dynamic dump of emulation mode trace table entries as they are entered into the trace table.

Host Processor and Controller Requirements

The host processor module of the dynamic dump program is executable in any System/370 that will accommodate DOS/VS.

The DOS/VS dynamic dump utility operates in a minimum virtual partition.

Residence requirements are:

Core image library:

Device	*Tracks*
2314	10
3330	7
3340	10

Relocatable library:

Device	*Tracks*
2314	15
3330	10
3340	15

The controller module of the dynamic dump program uses the network control subchannel of the type 1 channel adapter to communicate with the host processor if a type 2 channel adapter is installed. If the controller has only a type 1 channel adapter, or one or two type 4 adapters the dynamic dump program uses an emulation subchannel within the range recognized by the program and not used by a line. (The address(es) of the subchannels(s) are specified in the DYNADMP operand of the BUILD macro.)

Input to the Dynamic Dump Utility

Control statements are used to request the various functions of the dynamic dump utility. These control statements may reside in the input stream or they may be entered via the operator's console.

Initially, the dynamic dump utility reads control statements from the input job stream until either an END statement or a PAUSE statement is read. The PAUSE statement instructs the dynamic dump utility to read control statements only from the operator's console until either an END statement or a SYSIN statement is read. The SYSIN statement is the opposite of the PAUSE statement: it instructs the dynamic dump utility to return to the input job stream for control statements (beginning with the next statement after the last PAUSE statement). An END statement either encountered in the input job stream or entered from the operator's console causes the dynamic dump utility to be terminated.

Output from the Dynamic Dump Utility

Work File—This is a temporary file on which the contents of storage are written. (This file usually resides on a tape unit.)

Output File—This is the file on which the trace or storage dump is printed from the work file. It also contains the dynamic dump control statements and applicable error messages.

Operator's Console—The operator's console at the host processor may receive output as a result of a DISPLAY statement, control statement responses, and error conditions.

Dynamic Dump Operational Characteristics

The dynamic dump utility is used when trouble or error conditions indicate that a dynamic dump of controller storage is desirable to help in isolating and fixing a problem.

The dynamic dump utility physically consists of two modules. One module resides in the host processor (as load module IFUSVEP), and the other resides in the controller as part of the network control or emulation program. (This module is included in the program only if DYNADMP=YES is specified in the BUILD macro during program generation.) These two modules communicate with each other to transfer specified controller storage to the host module. If the DISPLAY command is used to enter a request, the transferred storage is displayed at the operator's console; otherwise, the host module writes the received storage to the work file in 516-byte blocks. You may then invoke the PRINT facility of the dynamic dump utility to print the contents of this work file.

When a particular user request has been satisfied, the host module of the dynamic dump utility issues message IFU503I to inform the operator that the transfer of data to the work file is complete.

Obtaining a Dynamic Dump of Trace Table Entries

The most important function of the dynamic dump utility is its ability to dynamically dump emulation mode line trace entries. Refer to *Utility Control Statements (DYNADMP)* for additional information on the control statements discussed below and to *Example of Dynamically Dumping Trace Table Entries* which follows this section for an illustration of the input stream.

To dynamically dump the emulation mode line trace table entries, first start the trace on the desired range of emulation mode subchannels by using the OPTION control statement. To begin the transfer of 516-byte blocks of trace entries (hereafter referred to as *trace blocks*) to the dynamic dump utility host module enter the DY DYNAMIC control statement next. Message IFU505E is sent to the operator's console where an eventual response is required. This message informs the operator as to how to stop the trace.

Each of the trace blocks received by the host module is time-stamped before being written to the work file. The time stamp is of the form hh:mm:ss (hours:minutes:seconds) and indicates the time that the trace block was received by the host module. Periodically (for the first and last trace block and every 200th block between) the operator is informed of these time stamps via message IFU508I. A typical IFU508I message might be:

```
IFU508I TRACE BLOCK     15,000 WRITTEN AT 13:40:42
```

This message indicates that the 15,000th trace block was written to the work file at 1:40:42 p.m. This information may be used when you prepare to print the work file.

Note: A total of 200 trace blocks is equivalent to approximately 72 pages of printed output (assuming 55 lines per page).

To stop the trace if running in the background partition, press the External Interrupt button on the CPU; to stop the trace if running in a foreground partition, press the console Interrupt Request button and enter 'MSG Fx' (*x* represents the number of the partition desired).

This action will halt the transfer of trace blocks to the work file as soon as the next trace block is received from the controller and placed on the work file. To stop the trace activity in the controller (which was initiated by the OPTION control statement) a second OPTION statement must be entered with A=3 specified. Alternatively, this statement may be entered at the controller's panel.

With the trace activity completed, a readable output listing of the trace blocks can be obtained by entering the PRINT command. This command causes the entire work file to be formatted and printed. Suppose, however, that you are interested only in printing the last portion of the trace blocks. For example, the trace is run to trap a sporadic line error. The trace is stopped when the line error occurs and a printout of the last portion of the trace blocks is required. To obtain this printout, a PRINT command like the following can be entered:

```
PRINT START=13:40:00
```

This command results in a printout of only those trace blocks written to the work file after 1:40 p.m.

Utility Control Statements

The dynamic dump utility control statements:

- Obtain a full storage dump.
- Dump the trace table area.
- Dump trace entries dynamically.
- Specify trace options.
- Request printing of the information dumped.

In the explanation of each utility control statement, small letters represent parameters for which you supply a value. A combination of capital and small letters in control statements (for example, PRint) indicates that you may specify either the entire statement (PRINT) or only the capitalized part (PR).

The DYNADMP Statement

The DYNADMP statement requests a dump of the entire controller storage or of a specified portion. The controller does not become idle and does not require reloading.

Name	*Operation*	*Operands*
[symbol]	DYnadmp	{Dynamic} {Storage} {Table}

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`Dynamic`

Specifies that the trace table is to be dumped dynamically as entries are made. This type of dump requires operator intervention to stop the trace. A trace must be started on a communication line via the control panel of the controller or via the dynamic dump facility (the OPTION control statement) before a dynamic trace can be started.

`Storage`

Specifies that the entire contents of controller storage are to be dumped. The execution of the network control program continues both during the operation and after the storage contents have been dumped.

`Table`

Specifies that only the trace table portion of controller storage is to be dumped. If no operand is specified, a full storage dump will be produced.

The DISPLAY Statement

The DISPLAY statement is used to request a display of a portion of the controller storage on the operator's console at the host processor.

Name	*Operation*	*Operands*
`[symbol]`	`DIsplay`	`hhhhh[,n]`

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`hhhhh`

Specifies the beginning address, in hexadecimal, of the storage to be displayed.

`[n]`

Specifies the number of lines (16 bytes of storage per line) to be displayed. The maximum number of lines you may specify is nine. If *n* is omitted, 1 is assumed.

The PRINT Statement

The PRINT statement requests that a printout (32 bytes of storage per line) of the entire work file be sent to the SYSLST device (the output file).

Name	*Operation*	*Operands*
`[symbol]`	`PRint`	`START=hh:mm:ss`

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`START=hh:mm:ss`

Specifies that only those trace blocks which were written to the work file after time hh:mm:ss (hours, minutes:seconds) are to be printed.

hh:mm:ss should specify a time that is both later than (or equal to) the time stamp associated with the first trace block and earlier than (or equal to) the time stamp associated with the last trace block that was written to the work file; otherwise, message IFU510I will be issued to indicate that no trace blocks were found which satisfied the PRINT command.

Example 1: Assume that the first trace block is recorded at 09:05:00 (9:05 a.m.) and the last is recorded at 09:20:00 a.m. The statement PRINT START=09:17:30 would cause printing of trace blocks recorded between 9:17:30 a.m. and 9:20 a.m.

Note: The print facility correctly interprets a post-midnight time stamp (for a last-written trace block) as later than a pre-midnight time stamp (for a first-written trace block), even though the numeric value of *hh:mm:ss* is lower for the post-midnight time (as, for example, values of 23:55:00 and 00:02:00, representing the seven-minute interval from 11:55 p.m. to 12:02 a.m.)

Example 2: Assume that a trace is started just before midnight. If the first trace block was written to the work file at 23:25:23 (11:25 p.m.), and the last was written at 00:40:57 (12:40 a.m. the following day), then either of the following PRINT statement would produce the intended results:

`PRINT`	`START=23:40:00`	(trace entries written to work file between 11:40 p.m. and 12:40 a.m. are printed)
`PRINT`	`START=00:20:00`	(trace entries written to work file between 12:20 a.m. and 12:40 a.m. are printed)

If you omit the START operand, the entire work file is printed.

If you specify the START operand and there are storage dumps in the work file with the trace blocks, then these storage dumps are also printed regardless of whether or not they satisfy the START constraint. Storage dumps are not time stamped.

The OPTION Statement

The OPTION statement starts, stops, or alters the program interrupt levels being traced. Level 2 interrupts (line data), or level 3 interrupts (timeout complete or channel data, such as initial selection, data, and status), or both can be traced. Level 1 error log entries are traced continuously after a level 3 trace is started.

Name	*Operation*	*Operands*
`[symbol]`	`OPtion`	`F[A]BCDE`

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

F[A]BCDE

Specifies the trace functions desired: start trace or stop trace, program level to be traced, and subchannels to be traced. The values of F, A, B, C, D, and E and their meanings are as follows:

F *Function*	*BC* *Data*	*DE* *Bytes*	*Meaning (L=level)*
4	10	xx	Start L2 trace on subchannel *xx*
4	11	xx	Stop L2 trace on subchannel *xx*
4	20	xx	Start L3 trace on subchannel *xx*
4	21	xx	Stop L3 trace on subchannel *xx*
4	30	xx	Start L2 and L3 trace on subchannel *xx*
4	31	xx	Stop L2 and L3 trace on subchannel *xx*
4	70	00	Start L3 trace on trace-defined subchannels
4	71	00	Stop L3 trace on trace-defined subchannels
4	70	FF	Start L3 trace on all subchannels
4	71	FF	Stop L3 trace on all subchannels

A is used only when two type 4 channel adapters are installed. If A=1 is specified, the subchannel address specified as DE refers to subchannels on the *first* type 4 channel adapter; if A=2 is specified, the subchannel address specified as DE refers to subchannels on the *second* type 4 adapter.

The PAUSE Statement

The PAUSE statement allows control statements to be entered at the console of the host processor after the PAUSE statement is read from the input job stream or entered from the console.

Name	*Operation*	*Operands*
[symbol]	PAUSE	

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The END Statement

The END statement specifies the end of job and causes termination of the program after the trace output has been printed.

This statement does not end the trace, however, if trace table entries are being dynamically dumped; in this case the trace must be terminated at the console. Establish operator communication with the host processor for a background partition by pressing the console interrupt button. Reply 'MSG Fx' in reply to the attention routine for foregound partition Fx.

Name	*Operation*	*Operands*
[symbol]	ENd	

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The SYSIN Statement

The SYSIN statement is used by the operator to cause control statements to be read from the input stream.

Name	*Operation*	*Operands*
	SYsin	

Job Control Statements

The DOS/VS job control statements:

- Execute or invoke the program
- Define the output file, the control statement file, the work file, and the communications controller.

Work file requirements depend on the storage size of the controller whose storage is being dumped. The work file must be a tape unit under DOS/VS.

The job control statements needed to invoke the dynamic dump utility are as follows:

```
//  JOB        [name]      (Initiates the job)
//  ASSGN      SYSLST      (Defines the output file)
//  ASSGN      SYS011      (Defines the communications controller
                           subchannel over which the EP dynamic
                           dump utility communicates with the host
                           processor)
//  ASSGN      SYS010      (Defines a temporary work file;  the
                           contents of controller storage are
                           written to this file)
//  ASSGN      SYSIPT      (Defines the control statement
                           file)
//  EXEC       IFUSVEP     (Specifies the job step IFUSVEP)

(Control statements)

/*
/&
```

Example of Job Control and Dynamic Dump Utility Statements

The following example shows the statements required to dynamically dump, to the work file, the entire contents of the controller whose subchannel address is 001. After the dump is complete, the contents of the work file are transferred to the output file and printed. The job ends without operator intervention.

```
//  JOB        SEVP
//  ASSGN      SYSLST [...Parameters defining output file]
//  ASSGN      SYS010,X'280'
//  ASSGN      SYS011,X'001'
//  EXEC       IFUSVEP
    DYNADMP    STORAGE
    PRINT
    END
/*
/&
```

Example of Dynamically Dumping Trace Table Entries

The example below shows the statements required to dynamically dump the trace table entries while they are being made in the trace table. The entries are placed in the work file until operator communication is established. The work file contents are then transferred to the output file, from which they are sent to the printer. The job ends upon completion of the print operation.

```
//  JOB      SEVP
//  ASSGN    SYSLST          (...Parameters defining the output
                             file)
//  ASSGN    SYS010,X'280'   (X'280' represents a device ad-
                             dress)
//  ASSGN    SYS011,X'001'   (X'001' represents the controller
                             address)
//  EXEC     IFUSVEP
    PAUSE                    (Allows operator to enter control
                             statements from console)
```

```
Entered from console:

    OPTION   41023           (Start trace of level 2 activity on
                             subchannel 23)
    DYNADMP  DYNAMIC         (Dump trace table dynamically as
                             entries are made)
. . .                        (Establish operator communication
                             to stop the dynamic dump)
    SYSIN                    (Returns control to the control
                             statement file)
```

```
    PRINT                    (Print and end producing  n list-
                             ing on device specified in SYSLST
                             statement)
END
/*
/&
```

Requirements for Installing the Dynamic Dump Utility

The following Logical Input/Output Control System (LIOCS) modules must be cataloged in the relocatable library:

- IJCFZIW0
- IJDFAZZW
- IJFUZZWZ

The following macros can be assembled to provide the above modules if they are needed:

- CDMOD   TYPEFLE=INPUT,WORKA=YES,SEPASMB=YES
- PRMOD   CTLCHR=ASA,WORKA=YES,SEPASMB=YES
- MTMOD   RECFORM=UNDEF,WORKA=YES,SEPASMB=YES

The controller physical unit block must indicate TP device for SVC 27 (HALT I/O) to work. This can be accomplished by specifying 2701 on the ADD command.

The supervisor must be generated with AP=YES for the POST instruction to work, and TP=BTAM or TP=QTAM must be specified for the HALT I/O (SVC 27) instruction to be supported.

If the dynamic dump modules of the host processor portion of the utility are cataloged in the relocatable library, the following control statements can be used for link-editing them into the core image library:

```
//  JOB        LINKEDIT
//  OPTION     CATAL
    INCLUDE    IFUDYN
//  EXEC       LNKEDT
/&
```

Part VII.
Appendixes

Appendix A: Types of Stations Supported by the IBM 3704 and 3705

The IBM 3704 and 3705 Communications Controllers, when executing a network control program/VS (version 3), 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.

Stations supported in network control mode can be connected to either a local or a remote communications controller, except as otherwise indicated.

Terminals:

IBM 1030 Data Collection System (emulation mode only)
IBM 1050 Data Communication System[1,2]
IBM 1060 Data Communication System (emulation mode only)
IBM 2260 Display Station (via IBM 2848 Display Control) (emulation mode only)
IBM 2265 Display Station (via IBM 2845 Display Control) (emulation mode only)
IBM 2740 Communications Terminal (Models 1 and 2)[1,2,3]
IBM 2741 Communications Terminal[1,2,3]
IBM 2760 Optical Image Unit (via IBM 2740) (emulation mode only)
IBM 2770 Data Communications System[4,5]
IBM 2780 Data Transmission Terminal[4,5]
IBM 2972 General Banking Terminal System[4,5]
    IBM 2980 Models 1 and 4 Teller Station
    IBM 2980 Model 2 Administrative Station
IBM 3270 Information Display System[4,5,6]
    IBM 3275 Display Station
    IBM 3277 Display Station
    IBM 3284 Printer
    IBM 3286 Printer
IBM 3600 Finance Communication System (network control mode only)
    (via IBM 3601 Finance Communications Controller)
IBM 3614 Consumer Transaction Facility
IBM 3650 Retail Store System (via IBM 3651 Store Controller)
IBM 3660 Supermarket System (via IBM 3651 Model 60 Store Controller)
IBM 3671 Shared Terminal Control Unit
IBM 3735 Programmable Buffered Terminal[4,5]
IBM 3740 Data Entry System (emulation mode only)
    IBM 3741 Data Station
    IBM 3747 Data Converter
IBM 3767 Communications Terminal
IBM 3770 Data Communications System
IBM 3780 Data Communication Terminal[4]
IBM 3790 Communications System
IBM 3940 Banking Terminal (Emulation mode only)
IBM 3980 Banking Terminal (Emulation mode only)
IBM Communicating Magnetic Card Selectric® Typewriter

World Trade teleprinters that use CCITT (Consultative Committee on International Telegraphy and Telephony) No. 2 or No. 5 code on leased point-to-point, leased multipoint, or switched network lines.

Terminals using the following line control disciplines: AT & T 83B3 or WU 115A start-stop code, over point-to-point or multipoint leased telegraph lines; WU CPT-TWX (33/35) start-stop code over switched lines.

Attachment of non-IBM terminals is under the provisions of the IBM Multiple Supplier Systems Policy.

Communications Control Units:

IBM 2701 Data Adapter Unit[4,5]
IBM 2703 Transmission Control[4,5]
IBM 2715 Transmission Control Model 2[4]
IBM 3704 Communications Controller[4,5]
IBM 3705 Communications Controller[4,5]

Computers:

IBM System/3[4,5]
IBM System/7[7] (with asynchronous communications adapter) (supported as 2740 model 1)
IBM System/7[4,5] (with binary synchronous communications adapter) (supported as System/3 [BSC])
IBM System/32 (with BSC Adapter) (supported as System/3 [BSC])
IBM System/360 Model 20 (with BSC Adapter)[4,5]
IBM System/360 Model 25 (with Integrated Communications Attachment with Synchronous Data Adapter II) [4,5]
IBM System/370 Model 125 (with Integrated Communications Attachment with Synchronous Data Adapter II)[4,5]
IBM System/370 Model 135 (with Integrated Communications Attachment with Synchronous Data Adapter II) [4,5]
IBM 1130 Computing System (with Synchronous Communications Adapter)[4,5]
IBM 1800 Data Acquisition and Control System (via IBM 1826 Data Adapter Unit with Communication Adapter)[4,5]

---

[1] Binary Coded Decimal (BCD) code
[2] Extended Binary Coded Decimal (Extended BCD) code
[3] Correspondence code
[4] Extended Binary Coded Decimal Interchange Code (EBCDIC)
[5] USA Standard Code for Information Interchange (USASCII)
[6] not supported on switched lines
[7] PTTC/EBCD code

Appendix B: Network Control Program Generation Messages

The communications controller assembler program produces diagnostic error messages during expansion of the network control program generation macro instructions (stage one). The message identifier for each message begins with IFQ.

The format of the messages is as follows:

```
s,IFQnnnI -----text----
```

`s`
  Is the severity code.

  A code of 4 is a warning that the condition indicated by the message may cause errors in the 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 two 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; programmer action may be required.

`-----text-----`
  Is the text of the message explaining the error condition.

```
                               {REQUIRED.      }
IFQ001I aaa=bbb INVALID,       {ccc IS ASSUMED.}
                               {IGNORED.       }
```

Explanation:
bbb is not a valid specification for operand *aaa*.

System Action:
One of the following occurs:

1. The operand *aaa* is required. Generation is terminated.
2. The generation procedure assumes the default value *ccc*.
3. Operand *aaa* is not required; value *bbb* is ignored.

Programmer Response:
If operand *aaa* is required or the default value *ccc* is not acceptable, correct operand *aaa* and resubmit stage one.

```
IFQ002I yyy INVALID NAME, EXCEEDS 8 CHARACTERS, REQUIRED.
```

Explanation:
The symbol *yyy* specified in the name field of the macro instruction exceeds 8 characters.

System Action:
A name is required. Generation is terminated.

Programmer Response:
Correct the name and resubmit stage one.

```
IFQ003I yyy INVALID NAME, FIRST CHARACTER NOT ALPHABETIC, REQUIRED.
```

Explanation:
The symbol *yyy* specified in the name field has a non-alphabetic first character.

System Action:
A name is required. Generation is terminated.

Programmer Response:
Correct the name and resubmit stage one.

```
IFQ004I NAME OMITTED, REQUIRED.
```

Explanation:
No symbol was specified in the name field of the macro; a name is required.

System Action:
A name is required. Generation is terminated.

Programmer Response:
Correct the name and resubmit stage one.

```
IFQ005I bbb-INVALID SUBOPERAND, {IGNORED.       }
                               {ccc IS ASSUMED.}
```

Explanation:

The value *bbb* is invalid for this suboperand.

System Action:
The generation procedure takes one of these actions:

1. The invalid suboperand value is ignored. Generation continues.
2. The value *ccc* is assumed as the suboperand. 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 and resubmit stage one.

`IFQ006I SEQUENCE ERROR-mmm,... NOT DEFINED [,explanation].`

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). The explanation, if any, describes the conditions that require the macro or macros.

System Action:
Generation is terminated.

Programmer Response:
Insert the macro or macros *mmm* in the correct sequence in the input statements and resubmit stage one.

`IFQ007I SEQUENCE ERROR-mmm1 PRECEDES mmm2.`

Explanation:
Macro *mmm2* does not precede macro *mmm1* in the network control program generation input statements.

System Action:
Generation is terminated.

Programmer Response:
Correct the sequence of macros *mmm1* and *mmm2* and resubmit stage one.

`IFQ008I TERM=bbb, NON-SUPPORTED TERMINAL TYPE.`

Explanation:
The type of terminal indicated by *bbb* is not a type supported by the network control program.

System Action:
Generation is terminated.

Programmer Response:
Correct the value *bbb* and resubmit stage one.

```
                                              {IGNORED.         }
IFQ009I aaa=bbb INVALID, NOT WITHIN RANGE,   {ccc IS ASSUMED.}
                                              {REQUIRED.        }
```

Explanation:
The value *bbb* specified for operand *aaa* is not within the valid range of values.

System Action:
The generation procedure takes one of these actions:

- IGNORED.

  Operand *aaa* is not required and is ignored. Generation continues.
- *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 one.

```
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 one.

```
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, specify the correct value or values in the operand and resubmit stage one.

`IFQ012I yyy INVALID NAME, $ CANNOT BE FIRST CHARACTER IN A SYMBOL, REQUIRED.`

Explanation:
The symbol *yyy* specified in the name field begins with a $ character. Symbols beginning with $ are reserved for use by the program generation procedure.

System Action:
A name is required. Generation is terminated.

Programmer Response:
Correct the symbol and resubmit stage one.

`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 one.

`IFQ014I bbb BPS-NON STANDARD.`

Explanation:
The value *bbb* specified as one of the data rates (bits per second) for the CSB macro being processed is not one of the standard data rates for a 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 one. 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 one.

```
IFQ016I  bbb INVALID ENTRY, IDENTIFIED AND UNIDENTIFIED ENTRIES
         MAY NOT BE MIXED.
```

Explanation:

The IDSEQ operand of this IDLIST macro (or a continuation) contains entries of both forms *(chars)* and *(chars,termname)*. The two different forms are not permissible within the same IDLIST macro (or continuation).

System Action:
Generation is terminated.

Programmer Response:
Correct the IDLIST macro or macros and resubmit stage one.

```
IFQ017I 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 is terminated.

Programmer Response:
Correct the block handling routine function macros and resubmit stage one.

```
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 one. If no statements are missing, no action is required.

```
IFQ019I bbb INVALID ENTRY, ID CHARACTERS TRUNCATED TO 40 HEXADECIMAL CHARACTERS
```

Explanation:

Entry *bbb* in the IDLIST macro has more than the allowable maximum of 40 hexadecimal characters.

System Action:

Entry *bbb* is truncated to the first 40 hexadecimal characters. The truncated entry is used as the ID sequence.

Programmer Response:

If the truncated entry is not acceptable, correct the entry and resubmit stage one.

```
IFQ020I SEQUENCE ERROR—mmm MUST BE CONSECUTIVE, IGNORED.
```

Explanation:

The macro *mmm* is not in the proper sequence. All *mmm* macros must be coded in sequence.

System Action:

This statement of the *mmm* macro is ignored. Generation continues.

Programmer Response:

Place the *mmm* macro in proper sequence, then resubmit stage one.

```
                                           {IGNORED.        }
IFQ022I aaa=bbb INVALID, EXCEEDS n CHARACTERS, {ccc IS ASSUMED.}
                                           {REQUIRED.       }
```

Explanation:

The symbol *bbb* is specified as the name of a macro in operand *aaa. bbb* contains more than *n* characters, which is the maximum number allowed.

System Action:

The generation procedure takes one of the following actions.

- IGNORED

  Operand *aaa* is not required and is ignored. Generation continues.
- *ccc* is ASSUMED

  Value *ccc* is assumed for operand *aaa.* Generation continues.
- REQUIRED

  A value is required. Generation is terminated.

Programmer Response:
If *ccc* is not an acceptable value, correct it and resubmit stage one.

```
IFQ023I IDLIST HAS NO VALID ENTRIES, REQUIRED.
```

Explanation
The IDSEQ operand of this IDLIST macro specifies no valid entries; at least one valid entry is required.

System Action:
Generation is terminated.

Programmer Response:
Correct the IDLIST macro and resubmit stage one.

```
IFQ024I mmm EXCEEDS MAXIMUM NUMBER ALLOWED [,IGNORED].
```

Explanation:
Macro or operand *mmm* specifies a value exceeding the maximum allowable value.

The maximum allowable values are:

Macro or operand	*Maximum value*
MTALCST macro	63 LCST entries
UBHR macro	65 uniquely named user routines
BHSET macro	255 BH sets
LCST operand	10 LCST entries
LCTYPE operand	64 entries

System Action:
If *IGNORED* appears in this message, the value specified for *mmm* is ignored; generation continues. Otherwise, generation is terminated.

Programmer Response:
If the ignored macro or operand is required, or if generation is terminated, correct the macro or operand and resubmit stage one.

```
IFQ025I aaa=bbb INVALID, $ CANNOT BE FIRST CHARACTER IN A SYMBOL, {IGNORED.       }
                                                                  {REQUIRED.      }
                                                                  {ccc IS ASSUMED.}
```

Explanation:
The symbol *bbb* is specified as the name of a macro in operand *aaa.* The symbol *bbb* begins with the $ character; symbols beginning with this character are reserved for use by the program generation procedure.

System Action:

The generation procedure takes one of the following actions:

- IGNORED.

  Operand *aaa* is not required and value *bbb* is ignored. Generation continues.

- REQUIRED.

  Operand *aaa* is required. Generation is terminated.

- *ccc* IS ASSUMED.

  The default value *ccc* is assumed for operand *aaa.* Generation continues.

Programmer Response:

If the default value *ccc* is not acceptable or if the operand is required, specify a valid symbol for *aaa* and resubmit stage one.

```
IFQ026I MORE THAN 35 ENTRIES IN SERVICE ORDER TABLE, VALIDITY CHECKING HALTED.
```

Explanation:

The SERVICE macro (or continuations) specify more than 35 entries in the ORDER operand.

System Action:

The program generation procedure checks only the first 35 entires. No other entries will be checked to determine if they represent devices that are represented by PU, INNODE, CLUSTER, TERMINAL, or COMP macros. Generation continues.

Programmer Response:

Visually inspect the source statements to verify that each device represented by PU, INNODE, CLUSTER, TERMINAL, or COMP macros appears in the ORDER operand of the SERVICE macro.

```
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 one.

```
IFQ028I aaa=bbb INVALID, EXCEEDS n CHARACTERS [, ccc IS ASSUMED].
```

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 and this message does not specify a default value, *ccc, bbb* is ignored and generation continues. If the severity code is 4 and a default value is specified, generation continues. If the severity code is 8, generation is terminated.

Programmer Response:
If a value is required for operand *aaa*, or if the assumed value *ccc* is not acceptable, or if the severity code is 8, specify a correct value in operand *aaa* and resubmit stage one.

```
                                                          {IGNORED.        }
IFQ029I aaa=bbb INVALID, FIRST CHARACTER NOT ALPHABETIC, {ccc IS ASSUMED.}
                                                          {REQUIRED.       }
```

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:
The generation procedure takes one of the following actions:

- IGNORED

  Operand *aaa* is not required and the value bbb is ignored. Generation continues.

- *ccc* IS ASSUMED

  The default value *ccc* is assumed. Generation continues.

- REQUIRED

  Operand *aaa* is required. Generation is terminated.

Programmer Response:
If operand *aaa* is required, or the default value *ccc* is not acceptable, correct operand *aaa* and resubmit stage one.

```
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 one.

```
IFQ031I aaa=bbb INVALID, TIMEOUT TABLE IS FULL, ccc IS ASSUMED.
```

Explanation:
The timeout value *bbb* is specified for operand *aaa.* Sixteen timeout values have previously been specified and no new ones may be specified.

System Action:
The default value *ccc* is assumed. Generation continues.

Programmer Response:
If the default value *ccc* is not acceptable, respecify timeouts in program source statements so that no more than 16 different time values are specified; then resubmit stage one.

```
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 one.

```
IFQ033I  NO  { GROUPS }  DEFINED IN THIS GENERATION.
             { LINES  }
```

Explanation:
The program source statements do not contain any GROUP or LINE macros.

System Action:
Generation is terminated.

Programmer Response:
Add to the program source statements the GROUP and LINE macros required to define the teleprocessing network.

```
IFQ034I CUTYPE=bbb, NON-SUPPORTED CONTROL UNIT TYPE, 3271 IS ASSUMED.
```

Explanation:
The CUTYPE operand of the CLUSTER macro specifies a type of control unit that is not supported by the network control program.

System Action:
The generation procedure assumes the default value 3271; generation continues.

Programmer Response:
If the default value is not appropriate, correct the CUTYPE operand and resubmit stage one.

```
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 one.

```
IFQ036I SPEED=bbb INVALID, EXCEEDS MAXIMUM SPEED FOR TYPE 1 CSB, REQUIRED
```

Explanation:
bbb is specified as the line speed; this speed exceeds the maximum speed (7200 bps) at which a line attached to a type 1 communication scanner can operate.

System Action:
Generation is terminated.

Programmer Response:
Correct the specified speed and resubmit stage one.

```
            {BASE MODULE        }
IFQ037I CSB {EXPANSION MODULE 1}  NOT DEFINED.
            {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 one.

```
IFQ038I SPEED=bbb INVALID, CSB OSCILLATOR SPEED LESS THAN ONE HALF THE LINE SPEED NOT
        FOUND, REQUIRED FOR EXTERNAL CLOCKING.
```

Explanation:
SPEED=bbb is specified for this line which has external clocking. No speed less than one-half *bbb* was defined for the communication scanner to which this line is attached.

System Action:
Generation is terminated.

Programmer Response:
Correct one of the following and resubmit stage one:

1. If the line is attached to the wrong scanner, respecify ADDRESS.
2. If the CSB macro specifies the wrong oscillator bit rates, respecify the CSB macro.
3. If *bbb* is incorrect, respecify SPEED.

```
IFQ039I SPEED=bbb NOT CHECKED FOR OSCILLATOR ASSOCIATION, explanation.
```

Explanation:
bbb is specified as the speed for this line but it cannot be checked for validity against the communication scanner oscillator rates due to the reason shown in the explanation.

System Action:
Generation is terminated.

Programmer Response:
Correct the errors shown in the explanation and resubmit stage one.

```
IFQ040I SPEED=bbb INVALID, CSB SPEED EQUAL LINE SPEED NOT FOUND, REQUIRED FOR INTERNAL
        CLOCKING.
```

Explanation:
SPEED=bbb is specified for this line which has internal clocking. No speed equal to *bbb* was defined for the scanner to which this line is attached.

System Action:
Generation is terminated.

Programmer Response:
Correct one of the following and resubmit stage one:

1. If the line is attached to the wrong scanner, respecify ADDRESS.
2. If the CSB macro specifies the wrong oscillator bit rates, respecify the CSB macro.
3. If bbb is incorrect, respecify SPEED.

`IFQ041I aaa=bbb INVALID, REQUIRED FOR explanation.`

Explanation:
The value *bbb* is not a valid specification for operand *aaa* and is required for the reason shown in the explanation.

System Action:
Generation is terminated.

Programmer Response:
Specify an acceptable value for operand *aaa* and resubmit stage one.

`IFQ042I yyy INVALID NAME, PREVIOUSLY DEFINED ON THE LINE.`

Explanation:
The symbol *yyy* specified in the name field has been previously defined for another device on the same multipoint line.

System Action:
Generation is terminated.

Programmer Response:
Correct the conflict in names and resubmit stage one.

```
IFQ043I yyy INVALID NAME, NOT INCLUDED IN SERVICE ORDER TABLE, REQUIRED IF DIAL=NO and
        POLLED=YES.
```

Explanation:
The symbol *yyy* specified in the name field is not specified in the ORDER operand of the SERVICE macro for this multipoint line.

System Action:
Generation is terminated.

Programmer Response:
Respecify the SERVICE macro such that it contains the symbol *yyy* as an entry or specify the correct symbol in the name field of the PU, INNODE, CLUSTER, TERMINAL, or COMP macro for this device and resubmit stage one.

```
IFQ044I PREVIOUS {CLUSTER} HAS NO DEVICES.
                 {LINE   }
```

Explanation:
The previous CLUSTER macro or LINE macro (for a nonswitched line) is not followed by any CLUSTER, TERMINAL, or COMP macro representing a device. At least one must be specified.

System Action:
Generation is terminated.

Programmer Response:
Specify one or more devices following the CLUSTER or LINE macro and resubmit stage one.

```
IFQ045I TERM=3275 INVALID, ONLY ONE TERMINAL CAN BE ON A 3275 CLUSTER.
```

Explanation:
A second TERMINAL macro with TYPE=3275 was specified following a CLUSTER with CUTYPE=3275. Only one TERMINAL macro may follow the CLUSTER macro.

System Action:
Generation is terminated.

Programmer Response:
Remove the TERMINAL macro in error and resubmit stage one.

```
IFQ047I CRITICAL SITUATION MESSAGE [HEADER] NOT AN EVEN NUMBER OF HEXADECIMAL DIGITS,
        IGNORED.
```

Explanation:
The total number of digits specified by either CSMSG and CSMSGC or CSMHDR and CSMHDRC is not an even number.

System Action:
The operand is ignored. Generation continues.

Programmer Response:
If the message does not specify HEADER and critical situation notification is desired, respecify CSMSG and CSMSGC and resubmit stage one. If the message specifies HEADER and a critical situation message header is required, respecify CSMHDR and CSMHDRC and resubmit stage one.

```
IFQ048I CRITICAL SITUATION MESSAGE HEADER PLUS TEXT EXCEEDS LIMIT OF 476 HEXADECIMAL
        DIGITS, IGNORED.
```

Explanation:
The total number of hexadecimal digits specified in the CSMSG, CSMSGC, CSMHDR, and CSMHDRC operands exceeds the maximum of 476.

System Action:
The CSMSG, CSMSGC, CSMHDR, and CSMHDRC operands are ignored. Generation continues.

Programmer Response:
If critical situation notification is desired, respecify the CSMSG, CSMSGC, CSMHDR, and CSMHDRC operands such that the total number of hexadecimal digits does not exceed 476 and resubmit stage one.

`IFQ049I NO VALID BLOCK HANDLER SPECIFIED FOR BHSET, REQUIRED.`

Explanation:
No valid block handlers were specified by the PT1, PT2, and PT3 operands of the BHSET macro.

System Action:
Generation is terminated.

Programmer Response:
If this block handler set is required, specify the name of a valid block handler and resubmit stage one; otherwise, delete the invalid BHSET macro and resubmit stage one.

`IFQ050I SEQUENCE ERROR-PREVIOUS GROUP HAS NO LINES, REQUIRED FOR explanation.`

Explanation:
Two GROUP macros were encountered with no intervening macros. At least one intervening LINE macro must be specified unless DIAL=YES and TERM=1050, 2740-1, 2741, or TWX are coded in the previous GROUP macro. *explanation* defines the specific reason for requiring a LINE macro following the previous GROUP macro.

System Action:
Generation is terminated.

Programmer Response:
Either:

- Correct the errors indicated by the explanation, or
- Specify the LINE macro or macros belonging in the group represented by the previous GROUP macro and resubmit stage one.

```
IFQ051I LINE CONFLICT-SPEED GT speed AT address1-address2 AND LINE AT address3-
        address4.
```

Explanation:
During the automatic resolution of the scan limits, a line with speed greater than *speed* was found in the range of addresses *address1-address2* and another

was found in the range of addresses *address3-address4*. This is an invalid configuration and no valid scan limit can be set.

System Action:
Generation is terminated.

Programmer Response:
Either:

- If the configuration is incorrect, respecify the SPEED and/or ADDRESS and/or AUTO operands, and resubmit stage one; or
- Override the automatic scan limit processing by specifying the SCANCTL operand on the GENEND macro and resubmit stage one.

```
IFQ052I LINE address1, CONFLICTS WITH:  address2,...
```

Explanation:
Line address *address1* requires the use of address substitution. If address substitution is performed, any of the lines *address2,...* that are defined will be disabled.

System Action:
Generation is terminated.

Programmer Response:
Either:

- Respecify the configuration of lines to remove the conflict, and resubmit stage one; or
- Override the automatic substitution processing by specifying the SCANCTL operand on the GENEND macro and resubmit stage one.

```
IFQ053I THE NUMBER OF RESOURCES DEFINED EXCEEDS n, THE MAXIMUM AVAILABLE
        FOR THIS NETWORK
```

Explanation:
The number of resources defined in the generation input deck (via LINE, PU, CLUSTER, TERMINAL, COMP, LU, and INNODE macros) exceeds *n*, the maximum that can be specified.

System Action:
Generation is terminated.

Programmer Response:
Either reduce the number specified in the MAXSUBA operand to a value that will permit the program to accommodate the specified number of resources, (the smaller the subarea maximum, the larger the maximum number of resources); or remove some of the resource definitions from the deck. Then resubmit stage one.

```
IFQ054I SUBAREA=n PREVIOUSLY DEFINED, REQUIRED.
```

Explanation:
The subarea address specified by the SUBAREA operand duplicates a subarea address specified in an earlier SUBAREA operand.

System Action:
Generation is terminated.

Programmer Response:
Change one of the duplicate subarea addresses and resubmit stage one.

```
IFQ105I TERM=bbb INVALID,SHOULD BE CODED AS CUTYPE=bbb,IGNORED
```

Explanation:
Terminal type *bbb* is not a supported terminal type but is a valid control unit type (CUTYPE).

System Action:
Generation is terminated.

Programmer Response:
Specify *bbb* as a CUTYPE and resubmit stage one.

```
IFQ106I WRAPLN=bbb NOT CHECKED FOR MOD ASSOCIATION,explanation
```

Explanation:
Address *bbb* is specified but it cannot be checked for validity because of the reason shown in self-defining explanation.

System Action:
Generation is terminated.

Programmer Response:
Correct the error shown in the explanation and resubmit stage one.

```
IFQ107I SPEED=bbb INVALID,EXCEEDS MAXIMUM OF 2400 FOR INTERNAL CLOCKING,REQUIRED.
```

Explanation:
bbb is specified as the speed for this line. Internal clocking is specified so the maximum allowed is 2400.

System Action:
Generation is terminated.

Programmer Response:
Specify the correct speed and resubmit stage one.

IFQ108I chanaddr=bb NOT CHECKED FOR LOCHAN-HICHAN ASSOCIATION,ERROR IN LOCHAN OR HICHAN IN THE BUILD MACRO.

Explanation:
bb is not checked against the LOCHAN-HICHAN range of subchannels because of an error in the LOCHAN or HICHAN operand of the BUILD macro.

System Action:
The subchannel address *bb* is accepted as a value.

Programmer Response:
Specify the correct value for LOCHAN or HICHAN and resubmit stage one.

IFQ110I FEATURE=ffff IS STANDARD FOR TERMINAL TYPE bbb.

Explanation:
Feature *ffff* is standard when terminal type *bbb* is specified in the LINE macro. The FEATURE=ffff suboperand need not be specified.

System Action:
None.

Programmer Response:
None.

IFQ111I SEQUENCE ERROR,PREVIOUS GROUP HAS NO LINES.

Explanation:
There are no LINE macros following the preceding GROUP macro.

System Action:
Generation is terminated.

Programmer Response:
Correct the sequence of macros so at least one LINE macro follows the preceding GROUP macro, or remove the GROUP macro.

IFQ112I {HICHAN}=aa IS NOT COMPATIBLE WITH THE PLUGGABLE HARDWARE SUBCHANNELS
{LOCHAN}

Explanation:
(For 3705) LOCHAN address *aa* is not a subchannel equal to 16n where $n \geq 1$. When subchannels are installed in the communications controller, the lowest subchannel installed will have a subchannel number equal to 16n. (For 3704) LOCHAN address *aa* is not an *even* subchannel address.

(For 3705) HICHAN address *aa* is not a subchannel equal to 4n-1 where $n \geq 1$. When subchannels are installed in the communications controller, the highest subchannel installed will have a subchannel number equal to 4n-1. (For 3704) LOCHAN address *aa* is not an *odd* subchannel address.

System Action:
None.

Programmer Response:
Correct the HICHAN or LOCHAN value so that the highest or lowest subchannel specified is consistent with the subchannels installed. A subchannel installed in the controller and used by the host processor, but not specified to the program in the program source statements, produces unpredictable results.

IFQ113I OVER 20 DUALCOM PAIRS HAVE BEEN SPECIFIED

Explanation:
Over 20 DUALCOM pairs are specified in the program source statements. The generation procedure verifies proper pairing of a maximum of 20 DUALCOM pairs. This is a limit of the verification mechanism only. More DUALCOM pairs are allowed but not completely checked.

System Action:
None.

Programmer Response:
Ensure that this line has a matching DUALCOM line as explained in the DUALCOM operand description under the LINE macro.

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, independent dump, and dynamic dump programs.

Messages Issued by the Loader (OS/VS)

```
IFL000I ERROR-LOADING PROCESS TERMINATED ** (ddname) COULD NOT BE OPENED ***
```

Explanation:
The data set indicated by ddname could not be opened (message IFL010I is also sent). If the DD statement is missing, another system message (IEC130I) will also be issued.

Utility Action:
The loading process is terminated.

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 severity codes possible are:

00	The loading process was completed successfully; all controllers that were to be loaded are now loaded.
04	The loading process for at least one of the controllers to be loaded generated a warning or error message.
08	Because of a severe error, none of the controllers to be loaded was successfully loaded.

Utility Action:
The loader job is completed.

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 controllers, 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 IFL010I is also sent).

Utility Action:
The controller to be loaded with the indicated load module is not loaded; the loader utility processes the next utility control card, if any.

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. (Message IFL010I is also sent.)

Utility Action:
The controller to be loaded with the indicated load module is not loaded; the loader utility processes the next utility control card, if any.

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 controller during loading (message IFL010I is also sent.)

Utility Action:
Loading of the controller is terminated; the loader processes the next utility control card, if any.

Programmer Response:
Resubmit the loader job.

```
IFL005I ERROR–LOADING PROCESS TERMINATED ** INITIAL TEST DETECTED 3704/3705 ERROR ***
```

Explanation:
The initial test routine did not return control to the loader utility. This indicates that a hardware error occurred in the controller that would prevent the network control program from executing properly (message IFL010I is also sent).

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Operator Response:
The operator may follow the problem determination procedure.

```
IFL006I ERROR–LOADING PROCESS TERMINATED ** CONTROL STATEMENT ERROR ***
```

Explanation:
The LOAD utility control card contained a syntax error. (This message is sent to both SYSPRINT and the operator.)

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Programmer or Operator Response:
Correct the erroneous LOAD card and resubmit the loader job.

```
IFL007I ERROR–LOADING PROCESS TERMINATED ** PROGRAM FAILURE IN 3704/3705 ***
```

Explanation:
The controller module of the loader utility encountered a software or hardware error in the controller (message IFL010I is also sent).

Utility Action:
Loading of the controller is terminated; the loader utility processes the next utility control card, if any.

Operator Response:
The operator may follow the problem determination procedure.

```
IFL008I LOAD COMPLETE 3704/3705=xxx LOADMOD=(member)
```

Explanation:
The controller whose unit address is *xxx* was successfully loaded with the 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, if any.

Programmer or Operator Response:
None.

`IFL009I WARNING—LOADING PROCESS COMPLETED ** LOAD MODULE LARGER THAN 3704/3705 ***`

Explanation:
The load module is too large for the controller. The loader utility loaded as much of the load module as possible in the controller and attempted to give control to that load module. Either the LOADMOD parameter of the LOAD utility control card specified the wrong load module member name for the controller specified by the 3705= parameter, or the network control program specified is too large for the controller and must be reduced in size.

Utility Action:
The loader utility processes the next utility control card.

Programmer Response:
Correct the LOAD utility control card or regenerate a network control program of a size that the controller can accommodate.

`IFL010I LOAD FAILED 3704/3705-xxx LOADMOD=(member)`

Explanation:
The loading of the controller 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, if any.

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. (Message IFL006I is also sent to the operator.)

Utility Action:
The loader utility processes the next LOAD utility control card, if any.

Programmer Response:
Correct the erroneous LOAD card and resubmit the loader job.

```
IFL012D 3704/3705-xxx ACTIVE *** REPLY TO CONTINUE ***
```

Explanation:
An attempt has been made to load the *xxx* controller, which contains an active control program.

Utility Action:
The loader utility waits for the operator's reply.

Operator Response:
If the controller should be loaded, enter REPLY xx, 'U'. This causes the loader utility to continue the load for this controller. If the controller should not be loaded, enter REPLY xx, 'M' to terminate the load request; processing then continues with the next request.

```
IFL013I 3704/3705 ACTIVE—LOAD CANCELLED BY THE OPERATOR
```

Explanation
The controller 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, if any.

Programmer or Operator Response:
None.

```
IFL014I  ERROR—LOADING PROCESS TERMINATED
        ** UNEXPECTED END-OF-FILE ON MEMBER xxxxxxxx **
```

Explanation:
An end of file condition occurred on the indicated member before the load module and record produced by the linkage editor were found. If the member name is IFL3705x, the problem exists with the initial test routine on SYSUT3; otherwise, the problem exists with the LOADMOD member on SYSUT1.

Utility Action:
The loader continues with the next request.

Programmer or Operator Response:
Check the link-edit of the indicated member to ensure successful completion. If link-edit was not successful, repeat the link-edit and request the load operation again.

```
IFL015I ERROR—LOADING PROCESS TERMINATED
        ** 3704/3705 DEVICE TYPE CONFLICT **
```

Explanation:
The specified device is not identified as a 3704 or 3705 to the operating system.

Utility Action:
The loading process continues with the next request.

Programmer or Operator Response:
Ensure that the DD card describing the 3704 or 3705 has the correct unit and that DIAG=NO is specified correctly for this communications controller.

Messages Issued by the Loader (DOS/VS)

`IFU000I 3704/3705 LOAD COMPLETE 3704/3705 xxx LOADMOD=(file name)`

Explanation:
The controller whose address is *xxx* was successfully loaded with the network control program load module indicated by *file name.*

Utility Action:
The loaded utility processes the next control card, if any.

Programmer or Operator Response:
None.

`IFU001I ERROR--LOADING PROCESS TERMINATED ** CONTROL STATEMENT ERROR ***`

Explanation:
The LOAD utility control card contained a syntax error.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Programmer or Operator Response:
Correct the erroneous LOAD card and resubmit the job.

`IFU002I ERROR--LOADING PROCESS TERMINATED ** MISSING KEYWORD **`

Explanation:
A required keyword parameter is missing from the LOAD utility control card.

Utility Action:
The loader utility processes the next utility control card, if any.

Programmer or Operator Response:
Correct the erroneous LOAD card and resubmit the job.

`IFU003I ERROR--LOADING PROCESS TERMINATED ** xxx PERMANENT I/O ERROR ***`

Explanation:
A permanent I/O error occurred in the communications controller during the loading process.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Operator Response:
The operator may follow the problem determination procedure.

IFU004I ERROR--LOADING PROCESS TERMINATED ** INITIAL TEST DETECTED A HARDWARE FAILURE 3704/3705-xxx**

Explanation:
The initial test routine did not return control to the loader utility. This indicates that a hardware error occurred in the controller that would prevent the control program from executing properly.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Operator Response:
The operator may follow the problem determination procedure.

IFU005I ERROR--LOADING PROCESS TERMINATED ** INVALID 3704/3705 SYMBOLIC ADDRESS **

Explanation:
The symbolic address specified in the 3705= parameter of the LOAD card is invalid.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

Programmer or Operator Response:
Change the LOAD utility control card and the associated ASSGN statement to a valid symbolic address and resubmit the job.

IFU006I UTILITY END *xx* WAS THE HIGHEST SEVERITY CODE

Explanation:
The loader utility has processed all control cards in the input file. The severity codes are:

00	The loading process was completed successfully; all controllers that were to be loaded are now loaded.
04	The loading process for at least one of the controllers to be loaded generated a warning or error message; the controller was not loaded.
08	Because of a severe error, none of the controllers to be loaded was successfully loaded.

Utility Action:
The loader job is completed.

Programmer Response:
If the severity code is greater than 00, examine the message file for the appropriate messages. Correct the job control statements in error and resubmit the job. (The resubmitted job need not specify for loading those controllers, if any, that were successfully loaded).

IFU007I ERROR--LOADING PROCESS TERMINATED ** (file name) UNRECOVERABLE I/O ERROR **

Explanation:
A permanent I/O error occurred when the loader utility attempted to read from the file specified by *file name.*

Utility Action:
Loading of the controller is terminated the loader utility processes the next utility control card, if any.

Program Response:
Ensure that LOADMOD specifies the correct file. If the correct file is already specified, resubmit the job. If the file name is DIAGFILE, use DIAG=NO to bypass the problem. If the error persists, keep the program listings and call IBM.

IFU008I ERROR--LOADING PROCESS TERMINATED ** 3704/3705 DEVICE TYPE CONFLICT **

Explanation:
The specified device is not identified as a 3704 or 3705 to the operating system.

Utility Action:
The loader utility processes the next utility control card, if any.

Programmer or Operator Response:
Ensure that the symbolic address specified in the 3705= parameter of the LOAD control card corresponds to the proper communications controller and that the DIAG= parameter is specified correctly.

IFU009I ERROR--LOADING PROCESS TERMINATED ** DIAGFLE IN ERROR **

Explanation:
The diagnostic file (DIAGFLE) does not contain the required initial test routines in proper sequence.

Utility Action:
The loader utility processes the next utility control card, if any.

Programmer Response:
Ensure that the proper initial test file is specified in the DIAGFLE DLBL statement and that IFU3705D nad IFU3705E are specified in that sequence.

IFU010I 3704/3705 LOAD FAILED 3704/3705-xxx LOADMOD=(file name)

Explanation:
The loading of the controller indicated by *xxx* failed. This message is sent only to the operator.

Utility Action:
The loader utility processes the next utility control card, if any.

Operator Response:

Examine the SYSOUT output for messages defining the problem and respond accordingly.

Messages Issued by the Dump Program (OS/VS and DOS/VS)

`IFW100D 3704/3705-xxx ACTIVE ** REPLY TO CONTINUE (Y or N)`

Explanation:
An attempt was made to dump the controller indicated by *xxx* but this controller is currently executing a program.

Utility Action:
The operator's reply is awaited.

Operator Response:
If dumping of the controller is desired (thereby terminating execution of its program), the operator should respond Y to this message. Otherwise, he should respond N or cancel the job.

`IFW101I 3704/3705-xxx HAS BEEN DUMPED SUCCESSFULLY`

Explanation:
The contents of the controller whose unit address is *xxx* have been successfully copied to the work data set or file (SYSUT2 in OS/VS or SYS008 in DOS/VS).

Utility Action:
The dump utility program will next interpret the dump control cards and produce the requested outputs.

Programmer Response:
None.

`IFW102I 3704/3705-xxx HAS BEEN PARTIALLY DUMPED`

Explanation:
An I/O error has occurred and only part of the storage contents of the controller could be copied to the work data set or file (SYSUT2 in OS/VS or SYS008 in DOS/VS). *xxx* represents the unit address of the controller.

Utility Action:
The dump utility will next interpret the dump control cards and will produce as much of the requested output as it can.

Programmer Response:
None.

IFW103I THE 3704/3705 COULD NOT BE DUMPED

Explanation:
An error has occurred, and none of the contents of the controller could be copied to the work data set or file.

Utility Action:
The Dump job is terminated.

Programmer Response:
This message is preceded by another dump message which describes the particular problem which has arisen. Consult it for problem determination.

IEW104I 3704/3705-xxx ACTIVE ** OPERATOR CANCELED DUMP

Explanation:
Message IFW100D was issued, and the operator responded N. *xxx* represents the unit address of the controller.

Utility Action:
The Dump job is terminated.

Programmer Response:
None.

IFW110I PERMANENT I/O ERROR WITH xxxxxxxx

Explanation:
A permanent I/O error has occurred in communication with the device referred to by *xxxxxxxx.*

Utility Action:
Message IFW102I or IFW103I will be issued.

Programmer Response:
Resubmit the job. If the problem recurs, this is an indication that a hardware problem exists.

IFW111I xxxxxxxx COULD NOT BE OPENED

Explanation:
The data set referred to by *xxxxxxxx* could not be opened.

Utility Action:
Message IFW103I is issued, and the job is terminated.

Programmer Response:
Correct the DD statement indicated by *xxxxxxxx* and resubmit the job.

IFW112I xxxxxxxx REFERS TO AN INVALID DEVICE

Explanation:
(1) If *xxxxxxxx* is intended to denote the dump program's work data set: *xxxxxxxx* refers to a device that is not a 2311, 2314, 3330, or 3340 direct access storage device; or (2) if *xxxxxxxx* is intended to denote the 3704 or 3705: *xxxxxxxx* refers to a device that is not a 3704 or 3705. In DOS/VS, SYS007 should be assigned to the 3704 or 3705, and SYS008 should be assigned to the Dump work file. Reversing these assignments would cause this message to be issued.

Utility Action:
Message IFW103I is issued and the Dump job is terminated.

Programmer Response:
Correct the error and resubmit the job.

IFW113I xxxxxx WAS NOT ASSIGNED

Explanation:
This is a DOS/VS-only message. The LUB referred to by *xxxxxx* was not assigned to a device.

Utility Action:
Message IFW103I is issued, and the Dump job is terminated.

Programmer Response:
Ensure that SYS007 is assigned to your 3704 or 3705 and that SYS008 is assigned to your Dump work file. Then resubmit the job.

IFW200I xxxxxxxx COULD NOT BE OPENED

Explanation:
The data set referred to by *xxxxxxxx* could not be opened.

Utility Action:
The job is terminated.

Programmer Response:
Correct the DD statement indicated by *xxxxxxxx* and resubmit the job.

IFW201I 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 the 3704 or 3705.

Programmer Response:
Probable user error. If the unformatted dump provided does not provide sufficient information, correct the dump statement and resubmit the job.

`IFW202I THE 3704/3705 DOES NOT CONTAIN AN NCP--NCP CONTROL BLOCKS WILL NOT BE FORMATTED`

Explanation:
FORMAT=Y was specified on the DUMP control statement, but the 3704 or 3705 does not contain a network control program. Consequently, the dump listing does not provide NCP control blocks.

Utility Action:
NCP control blocks are not formatted.

Programmer Response:
None.

`IFW203I xxxxxxxx IS EMPTY`

Explanation:
The DASD data set or file referred to by *xxxxxxxx* does not contain a legitimate copy of the contents of a 3704 or 3705. This situation will arise when either one of the following has happened:

(1) The user has failed to run step 1 of the Dump program.

(2) The user has run step 1 correctly, but in coding his JCL for step 2 he has specified a DASD data set or file other than the one used in step 1.

Utility Action:
The Dump job is terminated.

Programmer Response:
Correct the JCL so that *xxxxxxxx* refers to the DASD data set or file that contains a copy of the storage contents of the 3704 or 3705 to be dumped. Or, if such a DASD data set or file does not exist, run step 1 of the Dump program to produce one.

`IFW205I CONTROL STATEMENT MISSING; DEFAULT TAKEN`

Explanation:
No DUMP control statement was provided.

Utility Action:
An unformatted dump of the entire storage is produced.

Programmer Response:
None.

`IFW206I xxxxxxxx REFERS TO AN INVALID DEVICE`

Explanation:
xxxxxxxx refers to a device that is not a 2311, 2314, 3330, or 3340 direct access storage device.

Utility Action:
The Dump job is terminated.

Programmer Response:
Correct the error and resubmit the job.

`IFW207I SYS008 WAS NOT ASSIGNED`

Explanation:
This is a DOS/VS-only message. The LUB referred to by SYS008 was not assigned to a device.

Utility Action:
The Dump job is terminated.

Programmer Response:
Ensure that SYS008 is assigned to the Dump work file (the *same* one used by step 1 of the Dump program), and then resubmit the job.

`IFW208I NCP RVT IS BAD -- FORMATTING OF THE NCP CONTROL BLOCKS COULD NOT BE CONTINUED`

Explanation:
FORMAT=Y was specified on the DUMP control card, indicating that NCP control blocks were to be formatted. However, a conflict was detected in attempting to format the NCP resource vector table, preventing completion of the control block formatting.

Utility Action:
The NCP control blocks are not formatted.

Programmer Response:
None.

`IFW209I THE PARM FIELD ON THE EXEC CARD IS INVALID--IT IS IGNORED`

Explanation:
The LINECOUNT parameter is syntactically incorrect.

Utility Action:

The incorrect parameter is ignored; a 55 lines-per-page dump is produced.

Programmer or Operator Response:

Correct the LINECOUNT parameter if the default value of 55 is not acceptable.

`IFW210I THE 3704/3705 DOES NOT CONTAIN AN NCP--NCP BUFFER POOL WILL NOT BE FORMATTED`

Explanation:

BUF=Y is specified on the DUMP control statement but the program being dumped is not a network control program.

Utility Action:

A dump is produced that does not contain a formatted buffer pool.

Programmer or Operator Response:

None.

Messages Issued by the Dynamic Dump Program (OS/VS)

IFL500I SYSUT1 NOT OPENED

Explanation:
SYSUT1 could not be opened. The DD statement defining the data set was not included in the input stream.

Utility Action:
The dynamic trace function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that a DD statement for SYSUT1 is included in the input stream and that the parameters on the DD statement are correct.Resubmit the job.

IFL501A REPLY WITH DESIRED FUNCTION OR 'END'

Explanation:
The dynamic dump utility is requesting control statement input from the operator because there is no input stream or because a PAUSE statement was encountered in the input stream.

Utility Action:
The utility waits for the operator's reply.

Operator Response:
Enter a control statement. If all desired functions are complete, enter REPLY xx, 'END'. If more control statements are in the input stream, enter REPLY xx,'SYSIN'.

IFL502I THE FUNCTION COULD NOT BE PERFORMED

Explanation:
A permanent I/O error was encountered while processing SYSUT1.

Utility Action:
The utility terminates the job step with a 'USER 0001' abend code.

Programmer Response:
Ensure that DYNADMP=YES is specified in the BUILD macro, and that the unit allocated to SYSUT1 is the type 1 channel adapter of the communications controller.

IFL503I FUNCTION COMPLETED - nn

Explanation:
The function has been completed with the value indicated by *nn.*

Utility Action:
The function is terminated.

Programmer or Operator Response:
If *nn* is 00, completion is normal and no action is required. If *nn* is 08, ensure that the control statements appear in the proper sequence. The function was not acceptable to the network control program.

`IFL504I INVALID CONTROL STATEMENT`

Explanation:
A control statement was incorrectly specified.

Utility Action:
The function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that the control statement is valid and retry the function.

`IFL505E REPLY 'S' TO STOP TRACE`

Explanation:
To stop the dynamic dumping of trace table entries from the controller to the work data set, reply 'S' to this message.

Utility Action:
The dynamic dump utility will reissue this message if a response other than 'S' is entered. 'S' is the only response that will stop the transfer of trace blocks to the host module; it will not stop the entire trace activity (the transfer of trace entries to the trace table) in the communications controller. To stop the entire trace, enter an OPTION command with X=3 specified.

Programmer or Operator Response:
Upon determining that enough trace data has been collected, reply 'S' to this message. When this has been acknowledged, you may wish to enter an OPTION command with X=3 specified to terminate the trace activity in the controller.

`IFL506I STOP COMMAND ACKNOWLEDGED`

Explanation:
The dynamic dump utility has acknowledged the 'S' response to message IFL505E.

Utility Action:
The transfer of trace blocks to the host processor module of the dynamic dump utility has been stopped.

Programmer or Operator Response:
None.

`IFL507I FUNCTION NOT AVAILABLE OR INVALID`

Explanation:
The desired function was not available due to unavailable devices or was found invalid by the controller portion of the dynamic dump utility.

Utility Action:
The function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that the control statement is valid and that required devices are available, then retry the function.

`IFL508I TRACE BLOCK nnn,nnn WRITTEN AT hh:mm:ss`

Explanation:
Trace block number *nnn,nnn* was written to the work data set at time *hh:mm:ss* (hours:minutes:seconds).

Utility Action:
All trace blocks are time-stamped before being written to the work data set. For the first trace block, for the last trace block, and for every 200th trace block in between, message IFL508I is sent to the host processor console.

Programmer or Operator Response:
Refer to the IFL508I messages when printing the work data set. The PRINT facility permits selective printing according to time.

`IFL509I INVALID PARM FIELD ON EXEC CARD - DEFAULTS ASSUMED`

Explanation:
An option specified in the PARM field of the EXEC card was misspelled or was syntactically incorrect.

Utility Action:
Default values for the options are assumed and processing continues.

Programmer or Operator Response:
Refer to *The PARM Field Option* in Chapter 5 or 6 of this manual as an aid to determining the error.

`IFL510I NO TRACE BLOCKS SATISFY THE PRINT COMMAND`

Explanation:
No trace blocks were found that were written to the work data set after the time specified (by START=) in the PRINT control statement.

Utility Action:
The dynamic dump utility does not print any trace blocks. If, however, there is data other than trace blocks on the work data set (for example, storage dumps), this nontrace data will be printed.

Programmer or Operator Response:

Refer to the IFL508I messages sent to the operator's console during the time the trace was taken. These will indicate the range of timestamps associated with the trace blocks written to the work data set. From this information, derive a meaningful START= time to specify in the PRINT command.

`IFL511I EP-DYNADMP IS UNDER THE CONTROL OF ANOTHER HOST`

Explanation:

The EP portion of the dynamic dump utility is busy handling the dynamic dump requests of another host processor. (This message can also occur if an incorrect subchannel address is specified in the SYSUT1 statement.)

Utility Action:

Message IFL512A is issued.

Operator Response:

None.

`IFL512A REPLY 'YES' TO CONTINUE UNCONDITIONALLY OR 'NO' TO END`

Explanation:

This message follows message IFL511I and requires an operator response to indicate the utility action required. The EP dynamic dump utility can transfer dynamic dump data to only one host processor at a time. This means that if host processor 'B' is currently conducting a dynamic dump session with the EP dynamic dump utility, then host processor 'A' cannot begin a session with the EP utility until (1) host processor 'B' voluntarily relinquishes control of the utility by ending its dynamic dump session or (2) host processor 'A' forces the utility to end its session with host processor 'B'.

Utility Action:

If the operator at host processor 'A' replies 'YES', the dynamic dump utility in host processor 'A' directs the EP dynamic dump utility to end its session with host processor 'B' and establish a dynamic dump session with host processor 'A'. If the operator at host processor 'A' replies 'NO', the dynamic dump request is cancelled and the dynamic dump session in progress (with host processor 'B') is allowed to continue.

Operator Response:

Determine from the individuals making the dynamic dump requests the priority of those requests, then respond 'YES' or 'NO' as appropriate.

`IFL513I UNCONDITIONAL CONTINUE REJECTED BY EP-DYNADMP`

Explanation:

After a 'YES' response to an IFL512A message, the host portion of the dynamic dump utility was unsuccessful in attempting to seize control of the EP portion of the utility. (This message can also occur if an incorrect subchannel address is specified in the SYSUT1 statement.)

Utility Action:

The host portion of the utility is abnormally terminated with a dump; this situation should not occur.

Programmer Response:

Probable user error. Ensure that the parameter on the SYSUT1 DD statement are correct and resubmit the job.

`IFL514I DYNADMP FUNCTIONS TERMINATED BY OPERATOR`

Explanation:

The operator replied 'NO' to an IFL512A message. Consequently, the only dynamic dump commands that the operator may enter during the remainder of the current dynamic dump operation are formatting commands such as PRINT.

Utility Action:

None.

Programmer Response:

None.

`IFL515I EP-DYNADMP HAS NOT RESPONDED WITHIN THE LAST 3 MINUTES`

Explanation:

The 3704/3705 portion of the dynamic dump program has not responded within the past three minutes to the last I/O operation initiated by the host portion of the dynamic dump program.

Utility Action:

The host portion of the program continues to wait for a response to the I/O operation. Message IFL515I will be issued every three minutes until a response is received.

Programmer or Operator Response:

The programmer or operator should examine the nature of his last dynamic dump request. If the line he is tracing has little activity, three minutes without a response may be normal. If, on the other hand, the line presumably has high activity, or if the last dynamic dump request was 'DY TABLE' or 'DY STORAGE', then three minutes without a response is unreasonable, and the programmer or operator should consider canceling the host portion of the dynamic dump program. He should examine the assembly listing of the emulation program for possible errors or omissions.

`IFL600I xxxxxxxx NOT OPENED`

Explanation:
The named data set (SYSUT2, SYSPRINT, or SYSIN) could not be opened. Either the DD statement defining the data set was not included in the input stream or a DCB parameter was fould invalid.

Utility Action:
The function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that the DD statements for SYSUT2, SYSPRINT, and SYSIN are included in the input stream and that their parameters are correct. Resubmit the job.

Messages Issued by the Dynamic Dump Program (DOS/VS)

`IFU500I SYS011 NOT OPENED`

Explanation:
SYS011 could not be opened. The ASSGN statement defining the controller was not included in the input stream.

Utility Action:
The dynamic trace function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that an ASSGN STATEMENT for SYS011 is included in the input stream and that the parameters on the statement are correct. Resubmit the job.

`IFU501A REPLY WITH DESIRED FUNCTION OR 'END'`

Explanation:
The dynamic dump utility is requesting control statement input from the operator because there is no input stream or because a PAUSE statement was encountered in the input stream.

Utility Action:
The utility waits for the operator's reply.

Operator Response:
Enter a control statement. If all desired functions are complete, enter 'END'. If more control statements are in the input stream, enter 'SYSIN'.

`IFU502I THE FUNCTION COULD NOT BE PERFORMED.`

Explanation:
A permanent I/O error was encountered while processing SYS011.

Utility Action:
The utility terminates the job step with a dump.

Programmer Response:
Ensure that DYNADMP=YES is specified in the BUILD macro, and that the unit allocated to SYS011 is the type 1 channel adapter of the communications controller.

`IFU503I FUNCTION COMPLETED - nn`

Explanation:
The function has been completed with the value indicated by *nn*.

Utility Action:
The function is terminated.

Programmer or Operator Response:

If *nn* is 00, completion is normal and no action is required. If *nn* is 08, ensure that the control statements appear in the proper sequence. The function was not acceptable to the network control program.

`IFU504I INVALID CONTROL STATEMENT`

Explanation:

A control statement was incorrectly specified.

Utility Action:

The function is terminated.

Programmer or Operator Response:

Probable user error. Ensure that the control statement is valid and retry the function.

`IFU505A OC EXIT TO FORCE STOP`

Explanation:

A dynamic trace dump has been requested; the dump may be stopped by establishing operator communications.

Utility Action:

The dynamic dump is terminated when operator communication is established.

Programmer or Operator Response:

Upon determining that enough trace information has been dumped, establish operator communication for a backgroupd partition by pressing the console interrupt button, or for foreground partition F *x* by entering 'MSG' F *x* in reply to the attention routine. This action ends the dynamic dump when the current trace table entry is complete. The dump can be stopped only when there is activity on the line being traced. If there is no current activity on any line being traced, start a trace on some line on which there is activity; the trace may be started from the control panel of the controller. If no line being traced has any activity, the trace can be stopped only by canceling the dynamic dump utility and reloading the communications controller.

`IFU506I FORCE STOP ACKNOWLEDGED`

Explanation:

The dynamic dump utility has received the STOP indication.

Utility Action:

The utility stops the dump.

Programmer or Operator Response:

None.

IFU507I FUNCTION NOT AVAILABLE OR INVALID

Explanation:
The desired function was not available due to unavailable device or was found invalid by the controller portion of the dynamic dump utility.

Utility Action:
The function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that the control statement is valid and that required devices are available, then retry the function.

IFU508I TRACE BLOCK nnn,nnn WRITTEN AT hh:mm:ss

Explanation:
Trace block number *nnn,nnn* was written to the work file at time *hh:mm:ss* (hours:minutes:seconds).

Utility Action:
All trace blocks are timestamped before being written to the work file. For the first trace block, for the last trace block, and for every 200th trace block in between, message IFU508I is sent to the host processor console.

Programmer or Operator Response:
Refer to the IFU508I messages when printing the work file. The PRINT facility permits selective printing according to time.

IFU510I NO TRACE BLOCKS SATISFY THE PRINT COMMAND

Explanation:
No trace blocks were found written to the work file after the time specified (START=) in the PRINT control statement.

Utility Action:
The dynamic dump utility does not print any trace blocks. If, however, there is data other than trace blocks on the work file (for example, storage dumps) this nontrace data will be printed.

Programmer or Operator Response:
Refer to the IFU508I messages sent to the operator's console during the time the trace was taken. These will indicate the range of time stamps associated with the trace blocks written to the work file. From this information, derive a meaningful START time to specify in the PRINT command.

IFL511I EP-DYNADMP IS UNDER THE CONTROL OF ANOTHER HOST

Explanation:
The EP portion of the dynamic dump utility is busy handling the dynamic dump requests of another host processor. (This message can also occur if an incorrect subchannel address is specified in the SYSUT1 statement.)

Utility Action:
Message IFL512A is issued.

Operator Response:
None.

IFL512A REPLY 'YES' TO CONTINUE UNCONDITIONALLY OR 'NO' TO END

Explanation:
This message follows message IFL511I and requires an operator response to indicate the utility action required. The EP dynamic dump utility can transfer dynamic dump data to only one host processor at a time. This means that if host processor 'B' is currently conducting a dynamic dump session with the EP dynamic dump utility, then host processor 'A' cannot begin a session with the EP utility until (1) host processor 'B' voluntarily relinquishes control of the utility by ending its dynamic dump session or (2) host processor 'A' forces the utility to end its session with host processor 'B'.

Utility Action:
If the operator at host processor 'A' replies 'YES', the dynamic dump utility in host processor 'A' directs the EP dynamic dump utility to end its session with host processor 'B' and establish a dynamic dump session with host processor 'A'. If the operator at host processor 'A' replies 'NO', the dynamic dump request is cancelled and the dynamic dump session in progress (with host processor 'B') is allowed to continue.

Operator Response:
Determine from the individuals making the dynamic dump requests the priority of those requests, then respond 'YES' or 'NO' as appropriate.

IFU513I UNCONDITIONAL CONTINUE REJECTED BY EP-DYNADMP

Explanation:
After a 'YES' response to an IFU512A message, the host portion of the dynamic dump utility was unsuccessful in attempting to seize control of the EP portion of the utility.

Utility Action:
The host portion of the utility is abnormally terminated with a dump; this situation should not occur.

Programmer Response:
Probable user error. Ensure that the parameters on the SYSUT1 DD statement are correct and resubmit the job.

IFU514I DYNADMP FUNCTIONS TERMINATED BY OPERATOR

Explanation:
The operator replied 'NO' to an IFU512A message. Consequently, the only dynamic dump commands that the operator may enter during the remainder of the current dynamic dump operation are formatting commands such as PRINT.

Utility Action:
None.

Programmer Response:
None.

IFU515I EP-DYNADMP HAS NOT RESPONDED WITHIN THE LAST 3 MINUTES

Explanation:
The 3704/3705 portion of the dynamic dump program has not responded within the past three minutes to the last I/O operation initiated by the host portion of the dynamic dump program.

Utility Action:
The host portion of the program continues to wait for a response to the I/O operation. Message IFU515I will be issued every three minutes until a response is received.

Programmer or Operator Response:
The programmer or operator should examine the nature of his last dynamic dump request. If the line he is tracing has little activity, three minutes without a response may be normal. If, on the other hand, the line presumably has high activity, or if the last dynamic dump request was 'DY TABLE' or 'DY STORAGE', then three minutes without a response is unreasonable, and the programmer or operator should consider canceling the host portion of the dynamic dump program. He should examine the assembly listing of the emulation program for possible errors or omissions.

IFU600I SYS010 NOT OPENED

Explanation:
The named file, SYS010, could not be opened. The ASSIGN statement specified 'ignore'.

Utility Action:
The function is terminated.

Programmer or Operator Response:
Probable user error. Ensure that the ASSGN statement is for included in the input stream and that its parameters are correct. Resubmit the job.

Appendix D: Coding Examples for Switched Lines and Multiple Terminal Access Operation

This appendix provides examples illustrating how operations over switched lines, with and without multiple terminal access operation, can be accommodated within the network control program. Multiple terminal access operation is possible only if the program includes network control functions (TYPGEN=NCP or TYPGEN=PEP is specified in the BUILD macro).

How to Establish Operation over the Switched Telephone Network (Network Control Mode)

Assume that your teleprocessing network includes the following types of terminals that can call the communications controller:

- IBM 1050
- IBM 2740 (basic)
- IBM 2740 with Record Checking

Assume also that two of these types—1050 and 2740 with Record Checking—can be called by the controller.

For each type of terminal that will call the controller a separate line is required. There are accordingly three lines that must be specified as capable of receiving incoming calls. The two lines associated with the 1050 and the 2740 with Record Checking must also be specified as capable of calling these terminals, and must therefore be placed in dial sets. In the LINE macro for the basic 2740 you would specify CALL=IN; in the LINE macros for the other two terminal types you would specify CALL=INOUT, and also specify the name of the DIALSET macro that represents the respective dial sets.

Because the lines are to be used for call-in operation, each line must have a logical connection definition. The call-in definition is a TERMINAL macro in which CTERM=YES is specified; the call-out definition is a TERMINAL macro in which CTERM=NO is specified. The basic 2740, because the controller does not call it, can be represented by a single TERMINAL macro in which CTERM=NO is specified.

The macro instructions for this combined call-in and call-out arrangement would appear as:

```
SW1050      DIALSET     LINES=(G1L1)

SW2740F     DIALSET     LINES=(G2L1)

G2740A      GROUP       LNCTL=SS,
                        DIAL=YES

BASICLN     LINE        ADDRESS=022,
                        SPEED=134,
                        POLLED=NO,
                        CALL=IN

BASIC       TERMINAL    CTERM=YES,
                        TERM=2740-1,
                        FEATURE=NOCHECK

GRSW1050    GROUP       LNCTL=SS,
                        DIAL=YES

G1L1        LINE        ADDRESS=020,
                        SPEED=134,
                        CALL=INOUT,
                        AUTO=021,
                        DIALSET=SW1050,
                        POLLED=YES

T1050LCD    TERMINAL    TERM=1050,
                        CTERM=YES,
                        ADDR=81F9,
                        POLL=81F0,
                        FEATURE=(ATTN,BREAK)

T1050       TERMINAL    TERM=1050,
                        CTERM=NO,
                        ADDR=81F9,
                        POLL=81F0,
                        FEATURE=(ATTN,BREAK),
                        DIALSET=SW1050,
                        DIALNO=1234

G2740F      GROUP       LNCTL=SS,
                        DIAL=YES

G2L1        LINE        ADDRESS=024,
                        SPEED=134,
                        CALL=INOUT,
                        AUTO=025,
                        DIALSET=SW2740F,
                        POLLED=NO

LCD2740F    TERMINAL    TERM=2740-1,
                        CTERM=YES,
                        FEATURE=CHECK

T2740F      TERMINAL    TERM=2740-1,
                        CTERM=NO,
                        FEATURE=CHECK,
                        DIALSET=SW2740F,
                        DIALNO=(5678)
```

Example of Alternate-Port Switched Network Backup Operation

The following source statements illustrate how switched network backup using the alternate-port technique can be established for BSC communication lines (not applicable for 3270s, however).

In this example, two switched lines—one with and one without an automatic calling unit (ACU) attached—are provided as backup to three principal non-switched BSC lines, one of which is multipoint. One of the backup lines is specified for call-in and call-out operation; the other is specified for call-out operation only. Only call-out operation is involved in switched backup use.

Operands illustrating how switched network backup operation is specified are shown. Other macros and operands that may be required, including configuration macros for other lines that may be in the network, are not shown.

```
          BUILD      TYPGEN=PEP,ENABLTO=90                1
          SYSCNTRL   OPTIONS=(BACKUP,...)
BKUPDSET  DIALSET    LINES=(BKUPLN1,BKUPLN2)              2
BSCLNGP   GROUP      LNCTL=BSC,DIAL=NO,TYPE=PEP,...       2
BSCLN1    LINE       TYPE=NCP,POLLED=YES,DIALALT=BKUPDSET

T1        TERMINAL   DIALNO=(5789),...                    3
T2        TERMINAL   DIALNO=(5841),...                    3
T3        TERMINAL   DIALNO=(6007),...                    3

BSCLN2    LINE       TYPE=NCP,DIALALT=BKUPDSET,...

T4        TERMINAL   DIALNO=(6217),...                    3

BSCLN3    LINE       TYPE=PEP,DIALALT=BKUPDSET,...        4

T5        TERMINAL   DIALNO=(5472),...                    3

BKUPGP    GROUP      LNCTL=BSC,DIAL=YES,TYPE=NCP,
                     DIALSET=BKUPDSET

BKUPLN1   LINE       AUTO=122,MPTALT=YES,CALL=OUT,...     5,6

BKUPLN2   LINE       MPTALT=YES,CALL=INOUT,...            6

CT1       TERMINAL   CTERM=YES,...                        7

          GENEND
```

---

[1] Timeout duration must be sufficient to permit operator to complete the manual dialing procedure for switched backup lines.

[2] All lines in the backup dialset and all lines for which the backup dial set provides backup lines must have the same characteristics (as explained in the description of the GROUP macro). All lines in the backup dial set must be specified as MPTALT=YES because one of the principal lines for which backup is provided is a multipoint line (POLLED=YES).

[3] DIALNO operand is required for use of automatic calling unit on line BKUPLN1. (Dial digits cannot be dynamically changed via TCAM.)

[4] Backup operation applies only when line is operating in network control mode.

[5] AUTO operand (and the ACU whose address it specifies) is required for automatic calling operation.

[6] MPTALT=YES is required because one of principal lines for which backup is provided is a multipoint line.

[7] CTERM=YES is required because line is specified for call-in as well as call-out operation.

How to Specify Multiple Terminal Access Operation (Network Control Mode)

The manner in which the multiple terminal access facility of the network control program can be put to use is illustrated by the following example.
Assume the following:

- The teleprocessing network includes three types of start-stop terminals:

  IBM 1050
  IBM 2740 Model 1 (basic)
  IBM 2740 Model 1 with Record Checking feature

- All three types of terminals will communicate with the controller over a single switched line, referred to as a multiple terminal access line (MTA line), as follows:

  —All three types are to be able to call the controller. This is referred to as *call-in MTA operation.*

  —In addition, the controller must be able to call 1050s and 2740s with record checking, but not basic 2740s, over the MTA line. This is known as *call-out MTA operation.*

- The teleprocessing network will include, in addition to these MTA terminals, other 1050 terminals, identical to the MTA 1050s.

To arrange for MTA operation based on the foregoing, you would proceed as follows:

Step 1: Define Line Groups to Represent Individual Terminal Types

In MTA operation, a *type* of terminal refers not to a particular numeric designation, such as IBM 1050 or 2740, but to a unique combination of the line control scheme and transmission code used. For example, a 1050 using BCD code and a 1050 using EBCD code represent two different types of MTA terminal, though they are both 1050s. Similarly, a 2740 without the Record Checking feature and a 2740 equipped with this feature represent two MTA terminal types.

In this example, there are three MTA terminal types. Each type must be represented in a line group definition. This may be a regular non-MTA group definition (GROUP, LINE and TERMINAL macros), or a "stand-alone" group definition consisting only of a GROUP macro. A stand-alone group need be defined only when there is no non-MTA group for the same type of terminal.

As stated above, one of the types of MTA terminals—the 1050—is also used in non-MTA operation. Because this type of terminal will be represented in a non-MTA line group, a stand-alone MTA line group for this type is not needed.

For the two types of 2740, however, stand-alone groups are required:

```
G2740A  GROUP   LNCTL=SS,
                POLLED=NO,  (for basic 2740)
                DIAL=YES,
                TERM=2740-1

G2740F  GROUP   LNCTL=SS,
                POLLED=NO,  (for 2740 with record
                DIAL=YES,   checking)
                TERM=2740-1,
                FEATURE=(CHECK)
```

Each stand-alone GROUP macro must be named; the name is referred to by the MTALCST macro.

(In this example, all operands relating to MTA operation are shown. For purposes of this example, it is assumed that the default values for omitted operands are appropriate. Some operands that could be omitted, such as LNCTL=SS, are nonetheless shown.)

Step 2: Define MTA Line Group

The next step is to define an *MTA line group.* This is a regular line group, defined in the same manner as any regular line group (not a stand-alone group), except that you specify MTA, rather than a specific type of terminal, in the TERM operand. (The TERM operand may appear in the TERMINAL or LINE macros, or in the GROUP macro.)

GROUP macro:

The GROUP macro need specify only DIAL=YES; for purposes of this example, assume that for all other operands the default values are appropriate.

```
G1       GROUP   LNCTL=SS,
                 DIAL=YES
```

LINE macro:

Assume that the MTA line (1) is a duplex line that will operate at 134.5 bits per second (the data rate of the 1050s and 2740s), (2) is attached to address 020 in the controller, and (3) has its automatic calling unit attached to address 021. You would therefore specify, in the LINE macro, ADDRESS=020, SPEED=134, AUTO=021, DUPLEX=FULL.

Because the line will communicate with 1050s, which must be polled, you would also code POLLED=YES.

You would code CALL=INOUT (or omit the CALL operand), because calls over the line will be initiated both by terminals (1050 and both 2740 types) and by the controller (for the 1050s and 2740s with record checking only), as stated at the beginning of this example.

Call-in operation requires that you associate an *MTA list* with this line. *Call-out* operation requires that you associate a *dial set* with the line. (These are defined as described under Steps 4 and 5, below.) You would therefore code the MTALIST and DIALSET operands in the LINE macro.

Assuming that no other operands are required because their default values are appropriate, the complete LINE macro would be:

```
G1L1     LINE    ADDRESS=020,
                 SPEED=134,
                 AUTO=021,
                 DUPLEX=FULL,
                 POLLED=YES,
                 CALL=INOUT,
                 MTALIST=MTALST,
                 DIALSET=SWITMTA
```

TERMINAL macro for call-in operation:

Following each LINE macro for an MTA line over which MTA terminals will call the controller there must be a single TERMINAL macro in which CTERM=YES is specified. CTERM=YES designates this TERMINAL macro as the "call-in logical connection" terminal. This TERMINAL macro must appear between the LINE macro and the next LINE macro. (If this MTA line group contained several lines, instead of just one, a separate logical connection TERMINAL macro would be required for each line.)

```
G1L1TU   TERMINAL    TERM=MTA,
                     CTERM=YES,
                     FEATURE=(ATTN,BREAK),
                     ATTN=ENABLED
```

TERMINAL macros for call-out operation:

Each MTA terminal type that the controller will call over the MTA line must be represented by its own TERMINAL macro. As mentioned earlier, in this example the controller will call 1050s and 2740s with record checking, but not basic 2740s.

Assume (1) that only one 1050 will be called, and that its telephone number is to be defined in the network control program, and (2) that several 2740s with record checking will be called, and that the host processor will provide the appropriate telephone number with each call-out request. One TERMINAL macro is accordingly required for the 1050 and one for all 2740s with record checking. The DIALNO operand for the 1050 specifies the digits in the telephone number, followed by the length of the number; DIALNO for the 2740s specifies only the length, as the number will be provided within each call-out request from the host processor.

In the TERMINAL macros you would specify in the TERM operand *not* the actual terminal type, but *MTA*. You would, however, specify the features with which the terminal is equipped, and in the case of the 1050, specify enabling of the attention feature, if required by the application. Also, for the 1050, you would specify the polling and addressing characters needed.

These TERMINAL macros must specify the line control selection table entry and the dial set to be associated with the terminal, as explained in Steps 3 and 5.

The TERMINAL macros for call-out operation are therefore:

```
G1L1TA   TERMINAL    TERM=MTA,
                     FEATURE=(ATTN,BREAK),
                     ATTN=ENABLED,
                     ADDR=81F9,           (for 1050)
                     POLL=81F0,
                     LCST=MTA1050,
                     DIALSET=SWITMTA,
                     DIALNO=(3251,4)

G1L1TB   TERMINAL    TERM=MTA,
                     LCST=MTA2740F,       (for 2740 with
                     DIALSET=SWITMTA,     record checking)
                     DIALNO=(,4)
```

Step 3: Define Line Control Selection Table

One entry in the line control selection table is required for each combination of MTA terminal type and set of operating parameters to be identified to the network control program. (*Operating parameters* refers to the characteristics of a terminal apart from its line control scheme and transmission code; for example, the carriage return rate and the length (in characters) of the print line.)

Assume that all of the 1050s and all of the 2740s with record checking have the same operating parameters. Also assume that some of the basic 2740s have a printer line length of 130 characters and others have a line length of 95 characters, all other parameters being identical. The basic 2740s therefore have two sets of operating parameters, thus requiring two MTALCST macros; the 1050s and 2740s with record checking require only one each:

```
MTA1050     MTALCST     GROUP=GRSW1050,
                        SPEED=134,
                        CODE=EBCD,
                        LCTYPE=1050,
                        LINESIZ=130

MTA274A1    MTALCST     GROUP=G2740A,
                        SPEED=134,
                        CODE=COR,
                        LCTYPE=2740A,
                        LINESIZ=130

MTA274A2    MTALCST     GROUP=G2740A,
                        SPEED=134,
                        CODE=COR,
                        LCTYPE=2740A,
                        LINESIZ=95

MTA2740F    MTALCST     GROUP=G2740F,
                        SPEED=134,
                        CODE=EBCD,
                        LCTYPE=2740F,
                        LINESIZ=95
```

If the network control program is to call the type of terminal represented by one of the MTALCST macros, the LCST operand of the TERMINAL macro representing that type must name the MTALCST macro. In this example, the program is to call the 1050 represented by the MTALCST macro labeled MTA1050 and 2740s represented by the MTALCST macro labeled MTA2740F. You would therefore code LCST=MTA1050 and LCST=MTA2740F in the TERMINAL macros labeled G1L1TA and G1L1TB, respectively, as shown in Step 2.

Step 4: Specify MTA Lists, MTA Tables, and 1050 Polling Character List

MTA lists: Associated with each MTA line must be a list of MTA terminal types for which the NCP must check when answering a call over the line. The MTALIST macro (which must directly follow the MTALCST macros) defines this list. In this example only one MTALIST macro is needed because there is only one combination of terminals for which the program must check. The name of the MTALIST macro must be specified in the MTALIST operand of the LINE macro.

The order in which you specify the MTA terminal types in the MTALIST macro is not important; the program always attempts to identify the terminal types in the same predetermined sequence.

```
MTALST  MTALIST LCTYPE=(1050,2740A,2740F)
```

MTA tables: For call-in operation, each MTA terminal type must be identified in an MTA table defined by an MTATABL macro.

Because there are three MTA terminal types, three MTATABL macros are required. Each must specify the names of the MTALCST macros representing the combinations of operating parameters that the network control program must recognize. For the 1050s and the 2740s with record checking, there is only one set of parameters. The basic 2740s, on the other hand, have two sets. These differ in the length of the print line. Notice that the LCST operand of the MTATABL macro for the basic 2740s therefore names two MTALCST macros, whereas the others name only one each.

```
        MTATABL     LCST=(MTA1050),
                    CODE=EBCD,
                    LCTYPE=1050

        MTATABL     LCST=(MTA274A1,MTA274A2),
                    CODE=COR,
                    LCTYPE=2740A

        MTATABL     LCST=(MTA2740F),
                    CODE=EBCD,
                    LCTYPE=2740F
```

1050 polling list: For the 1050 terminals that will call the controller, an MTAPOLL macro is required, listing all of the 1050 polling characters. Assuming for this example that all such terminals have the same polling characters, 81F0 (hexadecimal), you would code:

```
        MTAPOLL     POLL=(81F0)
```

Step 5: Specify Dial Sets for Call-Out Operation

Because the line over which the controller communicates with MTA terminals is to be used for call-out operation, it must be included in a dial set. (This is the normal requirement for call-out lines, regardless of whether they are used for MTA operation.)

The dial set in this example consists of one line. The DIALSET macro therefore names a single LINE macro. Assuming that no alternate dial set is required and that the default values for the QLIMIT, QLOAD, and RESERVE operands are appropriate, the DIALSET macro appears as:

```
SWITMTA     DIALSET     LINES=(G1L1)
```

The name of this dial set is referred to by the DIALSET operand of the LINE macro appearing in Step 2.

The DIALSET operand of the TERMINAL macro for each terminal that the controller will call must name the dial set to be used.

*****

All the macros required for the example have now been coded. Figure D-1 shows all of the macros in a single sequence, with all macro sequencing requirements observed. Figure D-2 shows the logical relationships between the macros, as established by the pointers shown.

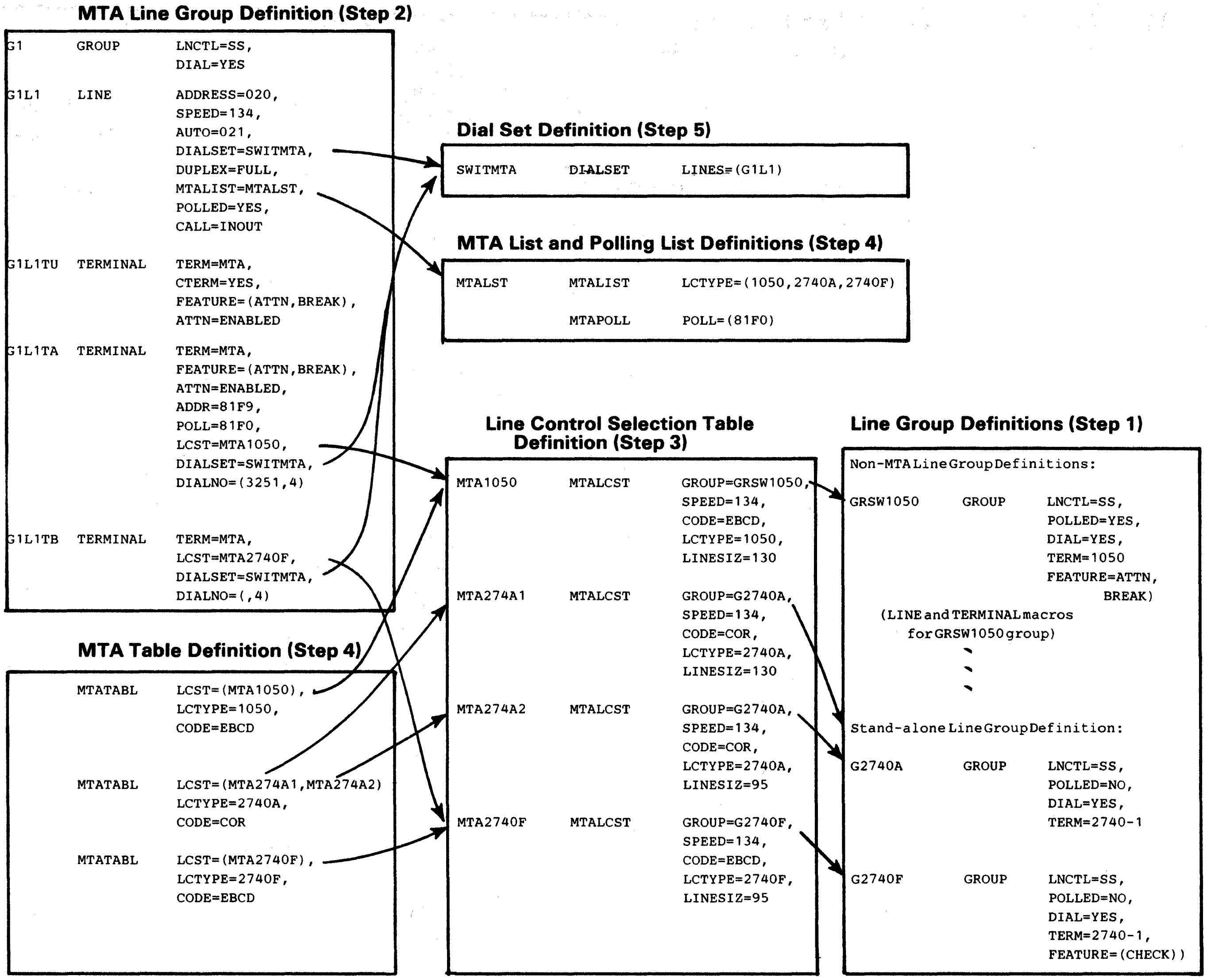
VTAM Note: Under certain conditions a VTERM macro may be needed directly following a TERMINAL macro in which CTERM=YES is specified. See the *VTAM System Programmer's Guide* for information on the purpose of the VTERM macro and for the conditions for its use.

```
SWITMTA     DIALSET     LINES=(G1L1)


MTA1050     MTALCST     GROUP=GRSW1050,
                        SPEED=134,
                        CODE=EBCD,
                        LCTYPE=1050,
                        LINESIZ=130



MTA274A1    MTALCST     GROUP=G2740A,
                        SPEED=134,
                        CODE=COR,
                        LCTYPE=2740A,
                        LINESIZ=130

MTA274A2    MTALCST     GROUP=G2740A,
                        SPEED=134,
                        CODE=COR,
                        LCTYPE=2740A,
                        LINESIZ=95


MTA2740F    MTALCST     GROUP=G2740F,
                        SPEED=134,
                        CODE=EBCD,
                        LCTYPE=2740F,
                        LINESIZ=95



MTALST      MTALIST     LCTYPE=(1050,2740A,2740F)



            MTAPOLL     POLL=(81F0)



            MTATABL     LCST=(MTA1050),
                        LCTYPE=1050,
                        CODE=EBCD



            MTATABL     LCST=(MTA274A1,MTA274A2),
                        LCTYPE=2740A,
                        CODE=COR


            MTATABL     LCST=(MTA2740F),
                        LCTYPE=2740F,
                        CODE=EBCD
```

Figure D-1. Macro Instructions Required for MTA Example (Part 1 of 2)

```
G2740A      GROUP       LNCTL=SS,
                        POLLED=NO,
                        DIAL=YES,
                        TERM=2740-1


G2740F      GROUP       LNCTL=SS,
                        POLLED=NO,
                        DIAL=YES,
                        TERM=2740-1,
                        FEATURE=(CHECK)


G1          GROUP       LNCTL=SS,
                        DIAL=YES

G1L1        LINE        ADDRESS=020,
                        SPEED=134,
                        AUTO=021,
                        DIALSET=SWITMTA,
                        DUPLEX=FULL,
                        MTALIST=MTALST,
                        POLLED=YES,
                        CALL=INOUT


G1L1TU      TERMINAL    TERM=MTA,
                        CTERM=YES,
                        FEATURE=(ATTN,BREAK),
                        ATTN=ENABLED


G1L1TA      TERMINAL    TERM=MTA,
                        FEATURE=(ATTN,BREAK),
                        ATTN=ENABLED,
                        ADDR=81F9,
                        POLL=81F0,
                        LCST=MTA1050,
                        DIALSET=SWITMTA,
                        DIALNO=(3251,4)

G1L1TB      TERMINAL    TERM=MTA,
                        LCST=MTA2740F,
                        DIALSET=SWITMTA,
                        DIALNO=(,4)


GRSW1050    GROUP       LNCTL=SS,
                        POLLED=YES,
                        DIAL=YES,
                        TERM=1050,
                        FEATURE=(ATTN,BREAK)

            (LINE and TERMINAL macros for
               GRSW1050 Group)

                  •
                  •
                  •
```

Figure D-1. Macro Instructions Required for MTA Example (Part 2 of 2)



MTA Line Group Definition (Step 2)

```
G1       GROUP       LNCTL=SS,
                     DIAL=YES

G1L1     LINE        ADDRESS=020,
                     SPEED=134,
                     AUTO=021,
                     DIALSET=SWITMTA,
                     DUPLEX=FULL,
                     MTALIST=MTALST,
                     POLLED=YES,
                     CALL=INOUT

G1L1TU   TERMINAL    TERM=MTA,
                     CTERM=YES,
                     FEATURE=(ATTN,BREAK),
                     ATTN=ENABLED

G1L1TA   TERMINAL    TERM=MTA,
                     FEATURE=(ATTN,BREAK),
                     ATTN=ENABLED,
                     ADDR=81F9,
                     POLL=81F0,
                     LCST=MTA1050,
                     DIALSET=SWITMTA,
                     DIALNO=(3251,4)


G1L1TB   TERMINAL    TERM=MTA,
                     LCST=MTA2740F,
                     DIALSET=SWITMTA,
                     DIALNO=(,4)
```

MTA Table Definition (Step 4)

```
         MTATABL     LCST=(MTA1050),
                     LCTYPE=1050,
                     CODE=EBCD


         MTATABL     LCST=(MTA274A1,MTA274A2)
                     LCTYPE=2740A,
                     CODE=COR

         MTATABL     LCST=(MTA2740F),
                     LCTYPE=2740F,
                     CODE=EBCD
```

Dial Set Definition (Step 5)

```
SWITMTA      DIALSET      LINES=(G1L1)
```

MTA List and Polling List Definitions (Step 4)

```
MTALST       MTALIST      LCTYPE=(1050,2740A,2740F)

             MTAPOLL      POLL=(81F0)
```

Line Control Selection Table Definition (Step 3)

```
MTA1050      MTALCST      GROUP=GRSW1050,
                          SPEED=134,
                          CODE=EBCD,
                          LCTYPE=1050,
                          LINESIZ=130

MTA274A1     MTALCST      GROUP=G2740A,
                          SPEED=134,
                          CODE=COR,
                          LCTYPE=2740A,
                          LINESIZ=130

MTA274A2     MTALCST      GROUP=G2740A,
                          SPEED=134,
                          CODE=COR,
                          LCTYPE=2740A,
                          LINESIZ=95

MTA2740F     MTALCST      GROUP=G2740F,
                          SPEED=134,
                          CODE=EBCD,
                          LCTYPE=2740F,
                          LINESIZ=95
```

Line Group Definitions (Step 1)

```
Non-MTA Line Group Definitions:

GRSW1050     GROUP        LNCTL=SS,
                          POLLED=YES,
                          DIAL=YES,
                          TERM=1050
                          FEATURE=ATTN,
                                  BREAK)
     (LINE and TERMINAL macros
        for GRSW1050 group)
               .
               .
               .

Stand-alone Line Group Definition:

G2740A       GROUP        LNCTL=SS,
                          POLLED=NO,
                          DIAL=YES,
                          TERM=2740-1


G2740F       GROUP        LNCTL=SS,
                          POLLED=NO,
                          DIAL=YES,
                          TERM=2740-1,
                          FEATURE=(CHECK))
```

Figure D-2. Logical Relationships between Macro Instructions in MTA Example

Appendix E: Transmission Codes for World Trade Teletypewriter Terminals

The chart below gives the eight-bit representation within communications controller storage of the transmission code bit patterns for each character in the ITA2 and ZSC3 character sets, as used by various European teleprinters (World Trade teletypewriters). In the WTTYEOB and WTTYEOT operands of the GROUP macro, if used, specify hexadecimal values appearing within the chart.

Note: The transmission code bit pattern, as it appears on the communication line, is the reverse of the value shown in this chart.

Character			Transmission Code	
Letters shift	Figures shift		Letters shift	Figures shift
ITA2 & ZSC3	ITA2	ZSC3	(hexadecimal)	(hexadecimal)
A	-	+	03	23
B	?	6	19	39
C	:	8	0E	2E
D	WRU	WRU	09	29
E	3	-	01	21
F		4	0D	2D
G		0	1A	3A
H		?	14	34
I	8	Bell	06	26
J	Bell	2	0B	2B
K	(	(	0F	2F
L	)	)	12	32
M	.	7	1C	3C
N	,	,	0C	2C
O	9	:	18	38
P	0	9	16	36
Q	1		17	37
R	4	/	0A	2A
S	’	’	05	25
T	5	.	10	30
U	7	1	07	27
V	=	=	1E	3E
W	2	3	13	33
X	/		1D	3D
Y	6	5	15	35
Z	+		11	31
CR	CR	CR	08	28
LF	LF	LF	02	22
LTRS	LTRS	LTRS	1F	-
FIGS	FIGS	FIGS	1B	-
Space	Space	Space	04	24

Appendix F: Required Coding Sequence for Program Generation Macro Instructions

This appendix shows the required sequence in which the network control program generation macro instruction statements must appear in the input job stream for stage one of the program generation procedure. (See Appendix I for specific macro sequences for each of several sample network programs.)

There are five distinct groups of program generation macro instructions. These groups are, in the required sequence:

1. System Macro Instructions
2. Configuration Definition Macro Instructions
3. Network Configuration Macro Instructions
4. Block Handling Macro Instructions
5. Generation Delimiter Macro Instruction

All of the macros within each group are applicable if the program being defined includes network control functions (TYPGEN operand of the BUILD macro specifies NCP, NCP-LR, NCP-R, PEP, or PEP-LR). If the program includes *only* emulation functions (TYPGEN=EP), only certain macros within each group are to be coded.

The block handling macro instruction group is valid only if start-stop and/or BSC lines are included in the network configuration macro instructions.

Shown in the chart below for each group are the names of all macros within the group, and for each macro:

- Whether the macro is required or optional, and the number that must or can be coded.
- The operands that are always required, regardless of the network configuration or program options to be specified.
- The position of the macro within the group, when a specific sequence is required.

Macros in heavily outlined boxes are always required; macros in lightly outlined boxes are not required if the program being defined is to perform only emulation functions (TYPGEN=EP is specified in the BUILD macro). Only those operands that must always be coded are shown; the configuration and characteristics of the teleprocessing network and the procedural options needed determine which other operands are required. Refer to Chapter 2 or 3 for explanations of the characteristics and options; refer to Chapter 4 for descriptions of each of the macro instructions and operands.

(VTAM Users Only) The three macros PCCU, VIDLIST, and VTERM are VTAM-only macros. That is, they provide information only to the VTAM initialization process. Their presence in the network control program generation deck is not required (but is permissible) for the generation procedure. They must, however, be present, as required by VTAM, when the same deck is used as input to the VTAM initialization process. (PCCU is always required; whether VIDLIST and VTERM are required depends upon VTAM application requirements.) See the *VTAM System Programmer's Guide* for details of the meaning and use of these macros.

1-SYSTEM MACRO INSTRUCTIONS

Usage	Macro	Required Operands	Macro Position
Required for VTAM initialization: 1	PCCU	(See VTAM System Programmer's Guide for required operands	VTAM-only macro: must precede BUILD macro for VTAM initialization.
Required: 1	BUILD (for TYPGEN=NCP or TYPGEN=PEP)	LOADLIB= MEMSIZE= OBJLIB= TYPGEN=	first macro in NCP source statements
	(for TYPGEN=EP)	HICHAN= LOADLIB= LOCHAN= OBJLIB=	
Required: 1	SYSCNTRL	OPTIONS=	directly following BUILD macro

2-CONFIGURATION DEFINITION MACRO INSTRUCTIONS

Usage	Macro	Required Operands	Macro Position
Required: 1	HOST	INBFRS= MAXBFRU= UNITSZ=	macros in this group (2) may appear in any sequence except as indicated for MTALCST and MTALIST macros
Required: 1 for each scanner in controller	CSB	SPEED= WRAPLN= (WRAPLN=required only for TYPGEN=EP)	
Optional: 1 for each list of ID sequences to be checked by NCP	IDLIST	IDSEQ=	
Optional: 1 for each list of ID sequences to be checked by VTAM (required only for VTAM initialization)	VIDLIST	(See VTAM System Programmer's Guide for required operands)	
Optional: 1 for each LCST entry to be defined	MTALCST	GROUP= SPEED=	must appear in one sequence
Optional: 1 for each list of line control types for MTA lines	MTALIST	LCTYPE=	must appear in one sequence
Optional: 1 for each unique combination of line control types and transmission codes for MTA lines	MTATABL	LCST=	
Optional: 1 only, for all 1050s that may call controller over any MTA line	MTAPOLL	POLL=	

Optional: 1 for each dial set to be defined	DIALSET	LINES=
Optional: 1 only, for all logical units to be associated with SDLC stations on switched links.	LUPOOL	NUMBER=

3-NETWORK CONFIGURATION MACRO INSTRUCTIONS

Usage	Macro	Required Operands	Macro Position
Required: 1 for each physical line group	GROUP		at beginning of line group definition
Required: 1 for each line within group	LINE	ADDRESS= SPEED=	directly following GROUP macro or another line definition
Required: at least 1 if line uses multipoint discipline; more as needed to accommodate all stations (omit if LINE macro is coded in remote NCP and represents SDLC link to remote controller)	SERVICE	ORDER=	directly following LINE macro

For all start-stop and all BSC stations except IBM 2972, 3271, 3275:

Required: 1 for each station to be identified to NCP; represents first or only input component and/or first or only output component	TERMINAL	TERM=	directly following SERVICE macro (if present) or LINE macro
Optional: Used only for call-in MTA terminals to be associated with specific VTAM application programs.	VTERM	(See VTAM System Programmer's Guide for required operands)	directly following TERMINAL macro in which TERM=MTA, CTERM=YES are coded.
Required: 1 for each additional input or output component (one COMP macro can specify both one input and one output component)	COMP		directly following the TERMINAL macro or another COMP macro

OR
For IBM 2972, 3271, 3275:

Required: 1 for each BSC cluster-type station (2972, 3271 [BSC], 3275 [BSC])	CLUSTER	directly following SERVICE macro (if present) or LINE macro; or following another station definition
Required: 1 for each terminal address on cluster control unit	TERMINAL TERM=	directly following CLUSTER macro or another station definition

OR
For SDLC stations:

Required: 1 for each SDLC station (e.g., 3270 [SDLC] 3600, 3650, 3660, 3767, 3770])	PU PUTYPE=	directly following SERVICE macro or another PU macro
Required: 1 for each logical unit associated with physical unit, if physcial unit is on nonswitched link.	LU LOCADDR=	directly following PU macro or another LU macro associated with same cluster.

OR
For remote or local 3704 or 3705:

Required: 1 for remote 3704 or 3705 (when macro appears in local NCP); 1 for local 3704 or 3705 (when macro appears in remote NCP)	PU PUTYPE=	directly following SERVICE macro, when coded in local NCP; directly following LINE macro, when coded in remote NCP.

4-BLOCK HANDLING MACRO INSTRUCTIONS (OPTIONAL)

Usage	Macro	Required Operands	Macro Position
Required: 1 for each block handler to be defined	STARTBH		at beginning of block handler
Optional: 1 allowed in each block handler	EDIT		preceding DATETIME, if DATETIME present
Optional: 1 allowed in each block handler	DATETIME		following EDIT, if EDIT present
Optional: no limit on number allowed	UBHR	NAME=	anywhere between STARTBH and ENDBH: may be intermixed with EDIT and DATETIME macros
Required: 1 for each block handler	ENDBH		at end of block handler
Optional: 1 for each block handler set needed (limit 255)	BHSET		following all block handler macros

5-GENERATION DELIMITER MACRO INSTRUCTION

Usage	Macro	Required Operands	Macro Position
Required: 1	GENEND		last macro in NCP source statements

Appendix G: Multiple Terminal Access Sign-On Procedure for Terminal Operators

Terminals under control of the multiple terminal access (MTA) facility of the network control program require a sign-on procedure to allow the program to determine the type of terminal that is calling the controller. The program analyzes the sign-on message and control characters received from the terminal to determine the type of terminal and the transmission code it employs, then selects the appropriate control procedures for use with that terminal. The program uses the selected procedure for as long as the connection to that terminal exists.

The sign-on procedure differs for the various types of terminals. The procedure to be used for each terminal should be posted at the terminal or otherwise given each user of the terminal.

The procedures are as follows.

All types of MTA terminals except TWX:

1. Dial the telephone number of the MTA line to be used for communicating with the controller.
2. After the keyboard unlocks (and the Proceed light comes on [for 1050] or the Bid message is printed [for basic 2740 and 2740 with Checking]), enter the two characters /" (slash double-quote). (Allow the Bid message to be completely printed before pressing the Bid key and entering the /" characters. Failure to do so may cause unpredictable results. The Bid message is transmitted in both Correspondence code and in EBCD/BCD code so that it will be intelligibly printed regardless of the code used.)
3. If more than one line control selection table (LCST) entry is defined in the MTATABL macro representing the terminal, enter a two-digit index number representing the LCST entry to be used by the program when communicating with the terminal. (See the explanation of the MTALCST and MTATABL macros in Chapter 4 of this publication.) The first entry named in the MTATABL macro associated with the terminal is represented by index number 00, the second by 11, and so on. The two digits must be identical. If the first entry is to be used, or only one entry is named in the MTATABL macro, the index number, 00, need not be entered. The network control program uses the first (or only) entry if no index number is entered.
4. End the sign-on procedure as follows.

   For 1050 (LCTYPE=1050), 2740 with Checking (LCTYPE=2740F), or 2740 with Transmit Control and Checking (LCTYPE=2740E):

      Press Return key, then enter EOB character. (If EOT rather than EOB is entered from a 1050, the network control program breaks the line connection; the controller must be re-dialed and the sign-on procedure repeated.)

   For basic 2740 (LCTYPE=2740A) or 2740 with Transmit Control (LCTYPE=2740D):

      Press Return key, then enter EOT character.

   For 2741 (LCTYPE=2741):

      Press Return key.

TWX Terminals

1. Dial the telephone number of the MTA line to be used for communicating with the controller.
2. Press WRU key within three seconds after the audible data tone begins. (Waiting longer than three seconds will cause a timeout to occur and the connection to be broken.)

(These two steps are the entire sign-on procedure; multiple LCST entries cannot be defined for TWX terminals, and there is consequently no index number to enter.)

Errors in Sign-On Procedure

If the typing mechanism of the terminal does not move after the sign-on message has been entered, the operator may assume that the sign-on was successful and may begin keying message data.

If, however, the operator makes an error in entering the sign-on message, or if the network control program cannot identify the type of terminal, the program sends a character sequence that "wiggles" the typing mechanism to signal the operator to re-enter the sign-on message. (If the terminal is a basic 2740 (LCTYPE=2740A), the operator must press the Bid key before re-entering the message.)

If the operator delays too long in completing the sign-on message, or if he enters it erroneously enough times to cause the retry limit to be reached, the network control program breaks the line connection. The operator must then redial the the controller and begin again. (The time allowed for completing the sign-on message is determined by the MTARTO operand of the BUILD macro, and the retry limit is specified by the MTARTRY operand of that macro.)

Specific Procedures for Each Type of Terminal

At each terminal should be posted the appropriate procedure from among those given below, modified as necessary to suit local conditions.

MTA Sign-On Procedure for IBM 2741

1. Dial the computer, using the following telephone number(s): *list numbers here.*
2. When the keyboard unlocks, enter /".

(2a. [optional] Enter MTA index number: [11,22, etc.])

3. Press Return key.

(If type element does not move within a few seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; repeat step 2, (2a), and 3.)

MTA Sign-On Procedure for IBM 2740 without Transmit Control or Checking

1. Dial the computer, using the following telephone number(s): *list numbers here*

   The Bid message is now printed.

2. After the Bid message is completed, press Bid key, then enter /".

(2a. [optional] Enter MTA index number: [11,22, etc.])

3. Press Return key.
4. Enter EOT.

(If type element does not move within several seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; press Bid key and then repeat steps 2, (2a), 3, and 4.)

MTA Sign-On Procedure for IBM 2740 with Checking

1. Dial the computer, using the following telephone number(s): *list numbers here*

   The Bid message is now printed.

2. After the Bid message is completed, press Bid key, then enter /".

(2a. [optional] Enter MTA index number: [11,22,etc.])

3. Press Return key.
4. Enter EOB. (If you enter EOT by mistake, the entire procedure, beginning with step one, must be repeated.)

(If type element does not move within a few seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; repeat steps 2, (2a), 3, and 4.)

MTA Sign-On Procedure for IBM 2740 with Transmit Control

1. Dial the computer, using the following telephone number(s): *list numbers here*
2. When the keyboard unlocks, enter /".

(2a. [optional] Enter MTA index number [11,22, etc.])

3. Press Return key.
4. Enter EOT.

(If type element does not move within a few seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; press Bid key and then repeat steps 2, (2a), 3, and 4.)

MTA Sign-On Procedure for IBM 2740 with Transmit Control and Checking

1. Dial the computer, using the following telephone number(s): *list numbers here*
2. When the keyboard unlocks, enter /".

(2a. [optional] Enter MTA index number [11,22, etc.])

3. Press Return key.
4. Enter EOB. (If you enter EOT by mistake, the entire procedure, beginning with step 1, must be repeated.)

(If type element does not move within a few seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; repeat steps 2, (2a), 3, and 4.)

MTA Sign-On Procedure for IBM 1050

1. Dial the computer, using the following telephone number(s): *list numbers here*
2. When Proceed light comes on, enter /".

(2a. [optional] Enter MTA index number [11,22, etc.])

3. Press Return key.
4. Enter EOB.

(If type element does not move within a few seconds, you have signed on successfully and may begin entering message data. If the type element "wiggles," sign-on is unsuccessful; repeat steps 2, (2a), 3, and 4.)

MTA Sign-On Procedure for Western Union TWX Terminals

1. Dial the computer, using the following telephone number(s): *list numbers here*
2. Press WRU key within three seconds after the audible data tone begins.

(If typing mechanism does not "jump" within a few seconds, you have signed on successfully and may begin entering message data. If mechanism does "jump," sign-on is unsuccessful; press WRU key again or repeat both steps.)

Appendix H: Partial Program Generation

As explained in Chapter 2, partial program generation permits you to modify an existing network control program by re-assembling selected modules and link editing them with object modules not requiring changes. Significant processing time can be saved by not performing a complete generation when only some modules require changes.

To perform a partial program generation, specify PARTIAL=YES in the BUILD macro, and in the CONDASM operand specify the modules requiring reassembly.

Note: Partial generation is not possible if (1) the program performs only emulation functions (TYPGEN=EP is specified in the BUILD macro); such a program consists of only one module; (2) the type of communication scanner in the controller is changed (CSB: TYPE); (3) the type of program generated is changed (BUILD: TYPGEN); or (4) the size of controller storage is changed from 64K bytes or less to more than 64K bytes (BUILD: MEMSIZE).

Partial generation under OS/VS is possible only if the library of object modules produced by the previous generation procedure is available to the partial generation procedure, and that library is named in the OBJLIB operand of the BUILD macro.

Modules Requiring Reassembly

For changes to the controller hardware, or for changes to the network configuration or program options, the modules requiring reassembly are as listed below. Specify the last two digits of the module names in the CONDASM operand of the BUILD macro, except for changes to program tables, for which you specify TABLE in the operand.

Any change not listed below requires a complete program generation.

References to "adding support" refer to adding the indicated type of line, station, etc., to a network control program that does not currently have *any* such lines or stations. For example, you would reassemble modules SYSCG00A, SYSCG00B, SYSCG00C, and SYSCG00D (and SYSCG003, if controller has a type 1 scanner) if you were adding the first start-stop line to the network, but not for any additional start-stop lines. Similarly, you would reassemble these modules if you were deleting *all* start-stop lines from the network, but not if you were simply reducing the number of such lines.

Add or delete Abend option (BUILD: ABEND)

SYSCG000
SYSCG001
SYSCG002
SYSCG010

Change size of network control program buffers (BUILD: BFRS)

SYSCG000

Add or delete automatic network shutdown (BUILD: ANS)

SYSCG000
SYSCG007

Add or delete address trace option (BUILD: TRACE)

SYSCG010
SYSCG007

Add or delete block handler support (any or all block handling macros)

SYSCG000
SYSCG001
SYSCG002

Add or delete date/time block handling routine (DATETIME)

SYSCG001
SYSCG002

Add or delete SESINIT dynamic control option (SYSCNTRL: OPTIONS)

SYSCG001
SYSCG002

Add or delete DVSINIT dynamic control option (SYSCNTRL: OPTIONS)

SYSCG001
SYSCG002

Add or delete Edit block handling routine (EDIT)

SYSCG001
SYSCG002

Add or delete critical situation notification message
(BUILD: CSMSG, CSMSGC, CSMHDR, CSMHDRC)

SYSCG000

Change buffer limit for slowdown mode (BUILD: SLODOWN)

SYSCG000

Add or delete ERASE option (BUILD: ERASE)

SYSCG002
SYSCG006 *(only if program is not a remote NCP [TYPGEN: NCP-R])*

Changes to number or size of access method buffers
(HOST: MAXBFRU, UNITSZ)

SYSCG000

Change to number of access method pad characters (HOST: BFRPAD)

SYSCG000
SYSCG006 *(only if program is not a remote NCP [TYPGEN: NCP-R])*

Change in type of primary channel adapter (BUILD: CHANTYP)

SYSCG000
SYSCG010
SYSCG002
SYSCG006 *(only if program is not a remote NCP [BUILD: TYPGEN=NCP-R])*
SYSCG007

Change of channel adapter from single type to mixed types, or vice versa (BUILD: CHANTYP)

Note: TYPE 2 and TYPE 3 are equivalent.

SYSCG000
SYSCG010
SYSCG007

Change from one to two channel adapters or vice versa (BUILD: CHANTYP)

SYSCG010
SYSCG007
SYSCG000 *(if channel adapter types are mixed)*

Change in the status modifier option (HOST: STATMOD)

SYSCG006 *(only if program is not a remote NCP [BUILD: TYPGEN=NCP-R])*

Change from no channel timeout value to a timeout value or vice versa (HOST: TIMEOUT)

SYSCG007
SYSCG006 *(only if program is not a remote NCP [BUILD: TYPGEN=NCP-R])*

Change from no channel delay to a channel delay or vice versa (HOST: DELAY)

SYSCG007
SYSCG006 *(only if program is not a remote NCP [BUILD: TYPGEN=NCP-R])*

Change in the number of lines/links in the network (LINE)

SYSCG000

Add or delete a communication scanner (CSB)

SYSCG007

Add or delete line trace option (SYSCNTRL: OPTIONS)

SYSCG010
SYSCG007
SYSCG00B

Add or delete online test option (BUILD: OLT)

SYSCG000
SYSCG009 *(only if program includes BSC support)*
SYSCG00B
SYSCG00C
SYSCG00D
SYSCG00A *(only if program includes start-stop support)*

Add, change, or delete any line speed (LINE: SPEED)

SYSCG000

Add, change, or delete any line address (LINE: ADDRESS)

SYSCG000

Change the system text timeout (BUILD: ITEXTTO)

SYSCG000

Change the system line disable timeout (BUILD: DSABLTO)

SYSCG000

Change the system line enable timeout (BUILD: ENABLTO)

SYSCG000

Change the system dial timeout (BUILD: DIALTO)

SYSCG000

Change the maximum subarea address (BUILD: MAXSUBA)

SYSCG000
SYSCG002

Change the network control programs subarea address (BUILD: SUBAREA)

SYSCG000
SYSCG010

Add or delete partitioned emulation line support (PEP) (LINE: TYPE)

SYSCG000
SYSCG010

Change controller storage size from a value equaling or exceeding 64K to another value equaling or exceeding 64K (BUILD: MEMSIZE)

SYSCG000

Change the number of break characters (BUILD: XBREAK)

SYSCG00B

Add or delete answer tone support (LINE: ANSTONE)

SYSCG00B

Add start-stop support (GROUP: LNCTL)

SYSCG00A
SYSCG00B
SYSCG00C
SYSCG00D
SYSCG003 *(only if scanner is type 1)*

Delete start-stop support (GROUP: LNCTL)

SYSCG00B
SYSCG00C
SYSCG00D
SYSCG003 *(only if scanner is type 1)*

Add BSC support (GROUP: LNCTL)

SYSCG009
SYSCG00B
SYSCG00C
SYSCG00D
SYSCG003 *(only if scanner is type 1)*
SYSCG00A *(only if program includes start-stop support)*

Delete BSC support (GROUP: LNCTL)

SYSCG00B
SYSCG00C
SYSCG00D
SYSCG003 *(only if scanner is type 1)*
SYSCG00A *(only if program includes start-stop support)*

Add SDLC support (GROUP: LNCTL)

SYSCG000
SYSCG007
SYSCG008
SYSCG00B
SYSCG00D
SYSCG00E
SYSCG003 *(only if scanner is type 1)*

Delete SDLC support (GROUP: LNCTL)

SYSCG000
SYSCG007
SYSCG00B
SYSCG00D
SYSCG003 *(only if scanner is type 1)*

If program already includes Start-Stop support:

Add or delete start-stop multipoint support
(LINE: POLLED)

SYSCG00A
SYSCG00B
SYSCG00C *(only if no BSC multipoint support included)*
SYSCG00D *(only if no BSC multipoint support included)*

Add or delete support for 83B3/115A terminals
(TERMINAL: TERM)

SYSCG00A
SYSCG00C
SYSCG00D

Add or delete support for TWX terminals
(TERMINAL: TERM)

SYSCG00A
SYSCG00C
SYSCG00D

Add or delete devices using LRC checking
(TERMINAL: TERM, FEATURE)

SYSCG00A

Add or delete support for IBM 2740 Model 2
(TERMINAL: TERM, FEATURE)

SYSCG00A
SYSCG00C
SYSCG00D

Add or delete start-stop point-to-point support
(LINE: POLLED)

SYSCG00C
SYSCG00D
SYSCG00B *(only if no BSC point-to-point support included)*

Add or delete support for World Trade teletypewriter terminals
(TERMINAL: TERM)

SYSCG00A
SYSCG00C
SYSCG00D

Add or delete multiple terminal access support
(MTALIST, MTALCST, MTATABL, MTAPOLL,
TERMINAL: TERM, LCST)

SYSCG00B
SYSCG00C
SYSCG00D

Add or delete carriage return delay support
(TERMINAL: CRDLAY)

SYSCG00B
SYSCG00C

Add or delete support for IBM 2740 Model 1
or change the type of 2740 Model 1 supported
(TERMINAL: TERM, FEATURE; MTALCST: LCTYPE)

SYSCG00C
SYSCG00D

Add or delete support for IBM 2741 (TERMINAL: TERM)

SYSCG00C
SYSCG00D

If BSC support is already included:

Add or delete either EBCDIC or USASCII code
(LINE: CODE)

SYSCG009

Add or delete BSC multipoint tributary support
(LINE: TADDR)

SYSCG009
SYSCG00B
SYSCG00C
SYSCG00D

Add or delete transmit ITB support (BUILD: XITB)

SYSCG009
SYSCG00C

Add or delete BSC point-to-point support

SYSCG00C
SYSCG00D
SYSCG00B (*only if no start-stop point-to-point support is included*)

Add or delete BSC multipoint support if no
start-stop multipoint support is included

SYSCG00B
SYSCG00C
SYSCG00D

Add or delete dial support (GROUP: DIAL)

SYSCG003 (*only if scanner is type 1*)
SYSCG009 (*only if program includes BSC support*)
SYSCG00A (*Only if program includes start-stop support*)
SYSCG00B
SYSCG00C
SYSCG00D

If manual dial or dial digits supplied by access method
(TERMINAL: DIALNO) see note below.

Also, if dial support is already included in program:

SYSCG001
SYSCG002

Add or delete manual dial support (TERMINAL: DIALNO)

SYSCG001
SYSCG002

Add or delete support for dial digits supplied
by access method (TERMINAL: DIALNO)

(**Note:** DIALNO specified with a count and no dial digits included)

SYSCG001
SYSCG002

Add or delete ring indicator mode support

SYSCG00B
SYSCG00C
SYSCG00D

Appendix I: Sample Network Control and Emulation Programs

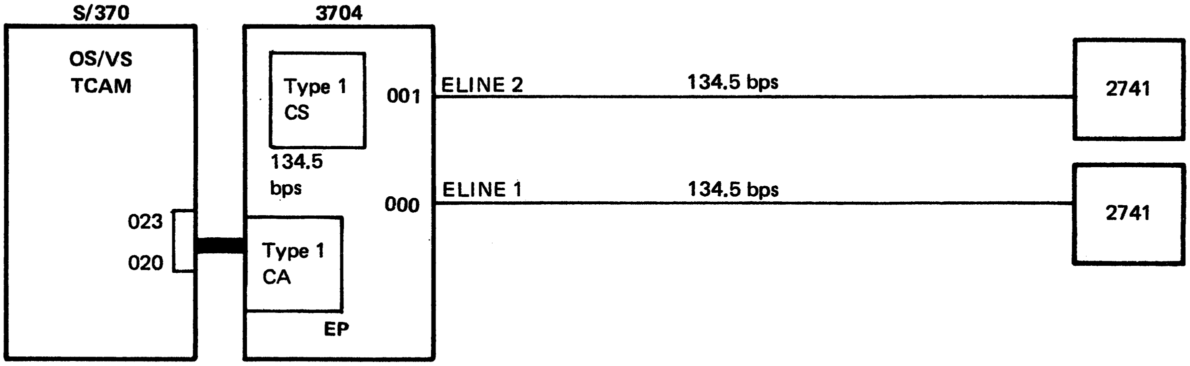
In this appendix are the source statements for nine sample communications controller control programs, representing a variety of programs: emulation functions only, network control functions only, combinations of the two, and one network control program for a remote communications controller. These samples progress from the simple, with one or two lines and few program options, to the complex, with many types of lines, stations, and program options. Preceding each sample program is the configuration of the network for which the program was coded, and accompanying remarks.

These programs are only representative samples showing in general how the source statements may be coded, and do not necessarily reflect the most appropriate choices of options for any particular application.

Sample 1

Emulation Program (TYPGEN=EP)

Lines and Stations:
Two start-stop, nonswitched point-to-point, IBM 2741.



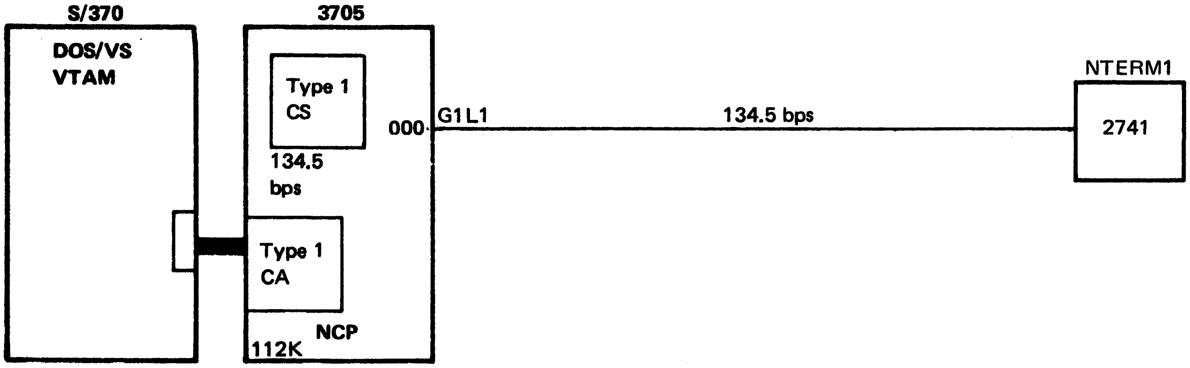
Remarks: This elementary example illustrates how to code a simple emulation program to control two point-to-point lines.

```
* SAMPLE PROGRAM #1
*
* EMULATION PROGRAM, GENERATED UNDER OS/VS, ACCESS METHOD: OS/VS TCAM
*
SAMPLE1  BUILD MODEL=3704,         CONTROLLER IS A 3704                X
               TYPGEN=EP,          EMULATION FUNCTIONS ONLY            X
               TYPSYS=OS,          GENERATION UNDER OS/VS              X
               LOADLIB=EPLIB,      LIBRARY NAME FOR EP LOAD MODULE     X
               HICHAN=023,         HIGHEST EMULATION SUBCHANNEL ADDRESSX
               LOCHAN=020,         LOWEST  EMULATION SUBCHANNEL ADDRESSX
               OBJLIB=EPOBJLIB     OBJECT LIBRARY FOR STAGE 2 OUTPUT
         CSB   TYPE=TYPE1,         SCANNER TYPE                        X
               MOD=0,              SCANNER LOCATION                    X
               SPEED=134,          OSCILLATOR SPEED                    X
               WRAPLN=001          LINE ADDRESS FOR WRAPLINE TEST
G1       GROUP LNCTL=SS            ALL DEFAULT VALUES ARE APPROPRIATE
ELINE1   LINE  ADDRESS=(000,20),   3704 AND EMUL  SUBCHNL ADDRESSES    X
               SPEED=134,          LINE SPEED 134.5 BPS                X
               CLOCKNG=INT,        INTERNAL (BUS MACHINE)CLOCKING USED X
               CU=2701,            EMULATED  TRANS CTL UNIT IS 2701    X
               MODEM=OPTION1,      LINE ENAPLED AFTER IPL OR SYS RESET X
               TERM=2741           TERMINAL ON LINE IS 2741
ELINE2   LINE  ADDRESS=(001,21),                                       X
               SPEED=134,                                              X
               CLOCKNG=INT,                                            X
               CU=2701,                                                X
               MODEM=OPTION1,                                          X
               TERM=2741
         GENEND
         END
/*
//
```

Sample 2

Network Control Program (TYPGEN=NCP)

Lines and Stations:
One start-stop, nonswitched point-to-point, IBM 2741 with Interrupt feature.



Remarks: Same configuration as for sample 1, except that there is only one line and operation is only in network control mode.

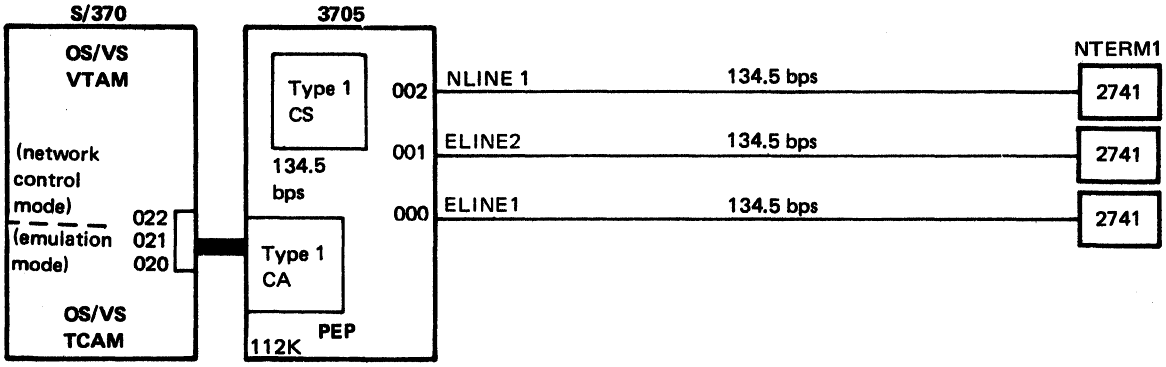
```
* SAMPLE PROGRAM #2
*
* NETWORK CONTROL PROGRAM (LOCAL), GENERATED UNDER DOS/VS, ACCESS
*   METHOD: DOS/VS VTAM
*
SAMPLE2  BUILD MODEL=3705,                                             X
               MEMSIZE=112,        STORAGE SIZE IS 112K BYTES          X
               TYPGEN=NCP,         LOCAL NCP-NETWORK CTL FUNCTIONS ONLYX
               SUBAREA=3,          SUBAREA ADDR OF NCP IS 3            X
               MAXSUBA=7,          HIGHEST SUBAREA ADDR POSSIBLE IS 7  X
               CA=TYPE1,                                               X
               CHANTYP=TYPE1,                                          X
               BFRS=80,            NCP BUFFER SIZE IS 80               X
               TYPSYS=DOS
         SYSCNTRL OPTIONS=(MODE,RCNTRL,RCOND,  VTAM-REQUIRED DYNAMIC   X
               RECMD,RIMM,ENDCALL,BHSASSC)    CONTROL OPTIONS
*                                  3 BFRS INITIALLY ALLOCATED FOR
         HOST  INBFRS=3,           DATA TRANSFERS FROM VTAM            X
               MAXBFRU=10,         MAX VTAM BFR UNITS FOR DATA FROM NCPX
               UNITSZ=88,          SIZE OF VTAM BFR UNITS              X
               BFRPAD=15           DOS/VS VTAM REQUIRES 15 BFR PADS
         CSB   TYPE=TYPE1,                                             X
               MOD=0,                                                  X
               SPEED=134
G1       GROUP TYPE=NCP            ALL DEFAULT VALUES ARE APPROPRIATE
NLINE1   LINE  ADDRESS=000,        3705 LINE INTERFACE ADDRESS         X
               SPEED=134,                                              X
               CLOCKNG=INT,                                            X
               CODE=EBCD,          TERMINAL USES EXTENDED BCD CODE     X
               DUPLEX=FULL         COMMUNICATIONS FACILITY IS DUPLEX
NTERM1   TERMINAL TERM=2741,                                           X
               ATTN=ENABLED,       NCP STOPS SENDING ON 2741 ATTN      X
               FEATURE=ATTN        2741 IS EQUIPPED WITH INTERRUPT FTR
         GENEND
         END
 /*
 /&
```

Sample 3

Network Control Program (local) with partitioned emulation programming extension (TYPGEN=PEP).

Lines and Stations:
One nonswitched point-to-point start-stop, IBM 2741 (network control mode).

Two nonswitched point-to-point start-stop, IBM 2741 (emulation mode).



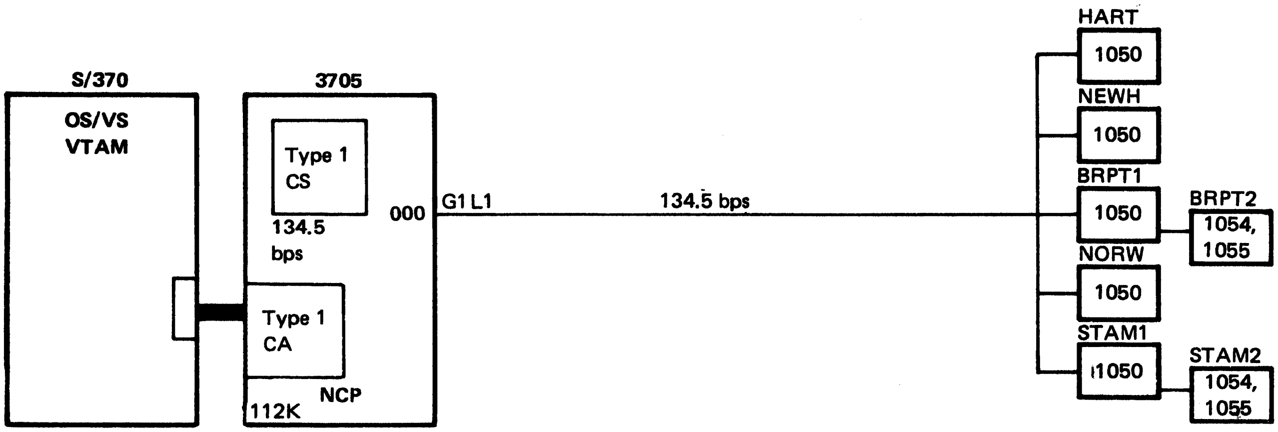
Remarks: This example combines the configurations of samples 1 and 2 into one configuration controlled by a network control program with the PEP extension. One line operates only in network control mode; two lines operate only in emulation mode.

```
* SAMPLE PROGRAM #3
*
* NETWORK CONTROL PROGRAM (LOCAL), GENERATED UNDER OS/VS, ACCESS
*   METHOD: OS/VS VTAM FOR NETWORK CONTROL MODE LINE,
*   OS/VS TCAM FOR EMULATION MODE LINES
*
* IN COMMENTS FIELD, E INDICATES OPERAND IS APPLICABLE TO EMUL MODE,
*                    N INDICATES OPERAND IS APPLICABLE TO NETW CTL MODE
SAMPLE3  BUILD MODEL=3705,         E N                                 X
               MEMSIZE=112,          N                                 X
               TYPGEN=PEP,         E N                                 X
               TYPSYS=OS,          E N                                 X
               LOADLIB=NCPLIB,     E N                                 X
               CA=TYPE1,           E N                                 X
               CHANTYP=TYPE1,      E N                                 X
               HICHAN=023,         E                                   X
               LOCHAN=020,         E                                   X
               OBJLIB=NCPOBJLB,    E N                                 X
               SUBAREA=3,            N                                 X
               MAXSUBA=7,            N                                 X
               BFRS=80               N
         SYSCNTRL  OPTIONS=(MODE,    N                                 X
               RCNTRL,RCCND,RECMD,   N                                 X
               RIMM,ENDCALL,BHSASSC) N
         HOST  INBFRS=3,             N                                 X
               NCPCHAN=023,          N                                 X
               MAXBFRU=10,           N                                 X
               UNITSZ=84,            N                                 X
               STATMOD=YES,          N                                 X
               BFRPAD=28             N
         CSB   TYPE=TYPE1,         E N                                 X
               MOD=0,              E N                                 X
               SPEED=134,          E N                                 X
               WRAPLN=001          E
G1       GROUP TYPE=PEP            E N
ELINE1   LINE  ADDRESS=(000,20),   E                                   X
               TYPE=EP,                                                X
               SPEED=134,          E                                   X
               CLOCKNG=INT,        E                                   X
               CU=2701,            E                                   X
               MODEM=OPTION1,      E                                   X
               TERM=2741           E      * * * * * * * * * * * * * *
*                                            THIS SAMPLE PROGRAM
ELINE2   LINE  ADDRESS=(001,21),   E      COMBINES THE CONFIGURATIONS  X
               TYPE=EP,                                                X
               SPEED=134,          E      OF SAMPLE PROGRAMS 1 AND 2   X
               CLOCKNG=INT,        E                                   X
               CU=2701,            E      * * * * * * * * * * * * * *  X
               MODEM=OPTION1,      E                                   X
               TERM=2741           E
NLINE1   LINE  ADDRESS=002,          N                                 X
               TYPE=NCP,                                               X
               SPEED=134,            N                                 X
               CLOCKNG=INT,          N                                 X
               CODE=EBCD,            N
               DUPLEX=FULL           N
NTERM1   TERMINAL TERM=2741,         N
               ATTN=ENABLED,         N
               FEATURE=ATTN          N
         GENEND                    E N
         END
/*
//
```

Sample 4

Network Control Program (local) (TYPGEN=NCP)

Lines and Stations:
One start-stop, nonswitched multipoint, IBM 1050.



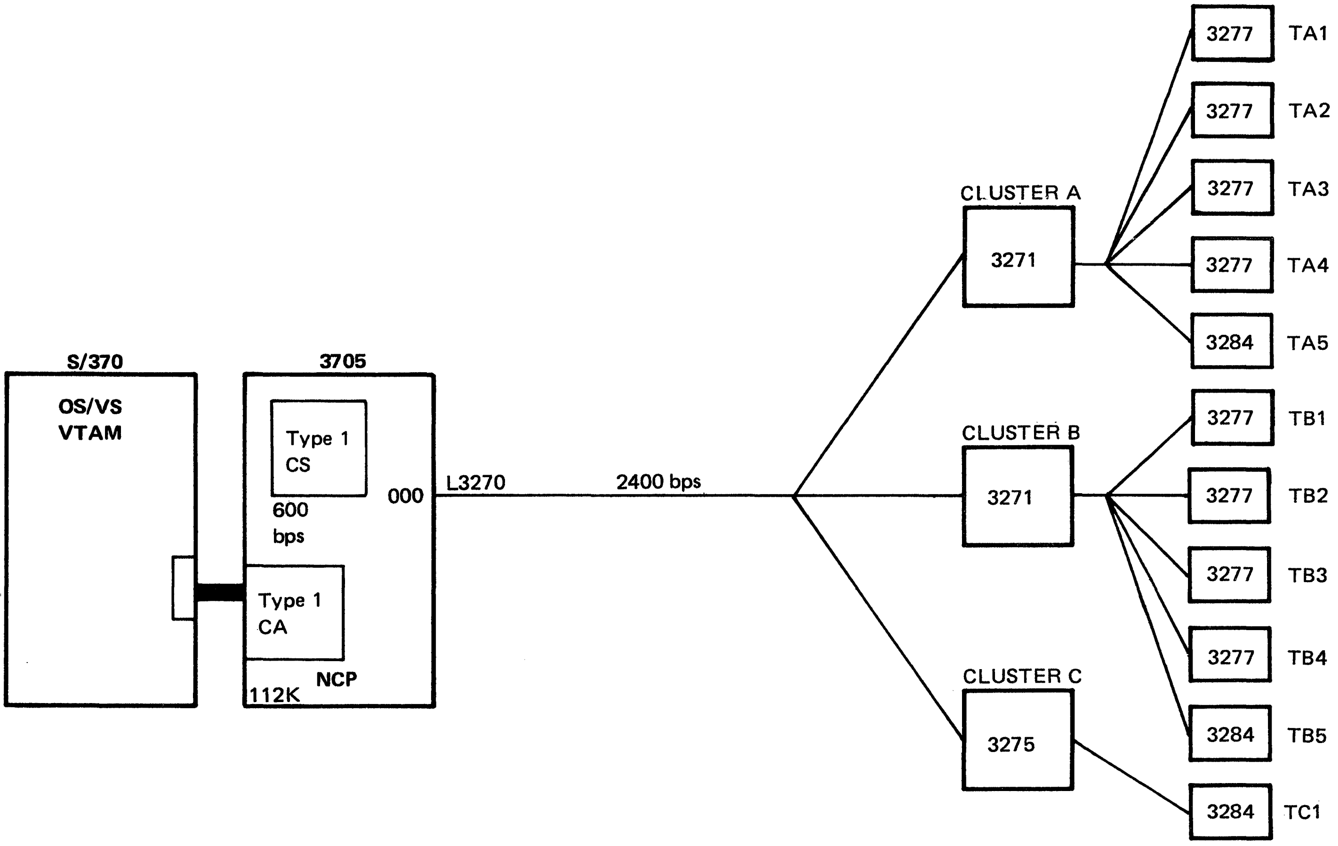
Remarks: This example illustrates the coding for a multipoint start-stop line on which multiple sessions are to be conducted. The use of COMP macros for subsidiary components of two of the terminals is shown. Notice that the GROUP macro is coded without operands; this reflects that all of the default values of this macro are appropriate.

```
* SAMPLE PROGRAM #4
*
* NETWORK CONTROL PROGRAM (LOCAL), GENERATED UNDER OS/VS, ACCESS
*   METHOD: OS/VS VTAM
*
SAMPLE4  BUILD MODEL=3705,                                             X
               MEMSIZE=112,                                            X
               TYPGEN=NCP,                                             X
               SUBAREA=3,                                              X
               MAXSUBA=7,                                              X
               CA=TYPE1,                                               X
               CHANTYP=TYPE1,                                          X
               BFRS=80,                                                X
               TYPSYS=OS,                                              X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM
        SYSCNTRL OPTIONS=(MODE,ENDCALL,RCOND,                          X
               RECMD,RIMM,BHSASSC,RCNTRL)
        HOST   MAXBFRU=10,                                             X
               UNITSZ=84,                                              X
               INBFRS=3,                                               X
               BFRPAD=28,                                              X
               STATMOD=YES,        STATUS MODIFIER OPTION AND CHANNEL  X
               DELAY=.2            DELAY AID CHANNEL PERFORMANCE
         CSB   TYPE=TYPE1,                                             X
               SPEED=134,                                              X
               MOD=0
G1       GROUP
G1L1     LINE  SPEED=134,                                              X
               CLOCKNG=INT,                                            X
               ADDRESS=000,                                            X
               TERM=1050,                                              X
               CODE=EBCD,                                              X
               POLLED=YES,         MULTIPOINT LINE CONTROL IS REQUIRED X
               SESSION=5,          MAX. CONCURRENT SESSIONS ON LINE    X
               SERVLIM=3,          SERV LMT: 3 SERV ORDER TABLE ENTRIESX
               PAUSE=20,           SERVICE-SEEKING PAUSE: 20 SECONDS   X
               TRANSFR=10,         TRANSFER LIMIT: 10 BUFFERS/SUB-BLOCKX
               CUTOFF=1            CUTOFF LIMIT: 1 SUB-BLOCK
         SERVICE ORDER=(HART,NEWH,HART,BRPT1,BRPT2,NORW,HART,          X
               STAM1,STAM2)
HART     TERMINAL ADDR=C1F9,       COMMON ADDR CHARACTERS: A9          X
               POLL=C1F0,          COMMON POLLING CHARACTERS: A0       X
               XMITLIM=3           TRANSMISSION LIMIT: 3
NEWH     TERMINAL ADDR=C2F1,       PRINTER ADDR CHARACTERS: B1         X
               POLL=C2F5,          KEYBOARD POLLING CHARACTERS: B5     X
               XMITLIM=3
BRPT1    TERMINAL ADDR=C3F1,       PRINTER ADDR CHARACTERS: C1         X
               POLL=C3F5,          KEYBOARD POLLING CHARACTERS: C5     X
               XMITLIM=3
BRPT2    COMP  ADDR=C3F3,          PAPER TAPE PUNCH ADDR CHARACTERS: C3X
               POLL=C3F6,          PAPER TAPE RDR POLLING CHARACTERS:C6X
               XMITLIM=6
NORW     TERMINAL ADDR=C4F9,       COMMON ADDR CHARACTERS: D9          X
               POLL=C4F0,          COMMON POLLING CHARACTERS: D0       X
               XMITLIM=3
STAM1    TERMINAL ADDR=C5F1,       PRINTER ADDR CHARACTERS: E1         X
               POLL=C5F5,          KEYBOARD POLLING CHARACTERS: E5     X
               XMITLIM=3
STAM2    COMP  ADDR=C5F3,          PAPER TAPE PUNCH ADDR CHARACTERS: E3X
               POLL=C5F6,          PAPER TAPE RDR POLLING CHARACTERS:E6X
               XMITLIM=6
         GENEND
         END
/*
//
```

Sample 5

Network Control Program (local) (TYPGEN=NCP)

Lines and Stations:
One BSC, nonswitched multipoint, IBM 3270 terminals (3271, 3275, 3277, 3284)



Remarks: As in sample 4, the line is nonswitched multipoint; in this case, however, the terminals are binary synchronous. Several more options are included in this program than in the previous sample; for example, NCP slowdown, buffer erase, and use of critical situation notification messages and headers. Notice how the general type of terminal (3277) is specified in the CLUSTER macros for clusters A and B, with the exception (3284 printers) specified in their respective TERMINAL macros (terminals TA5 and TB5). This illustrates the use of the operand hierarchy to save coding effort. (The TERM=3277 operand could just as well have been coded in the LINE or GROUP macro as in the CLUSTER macro.)

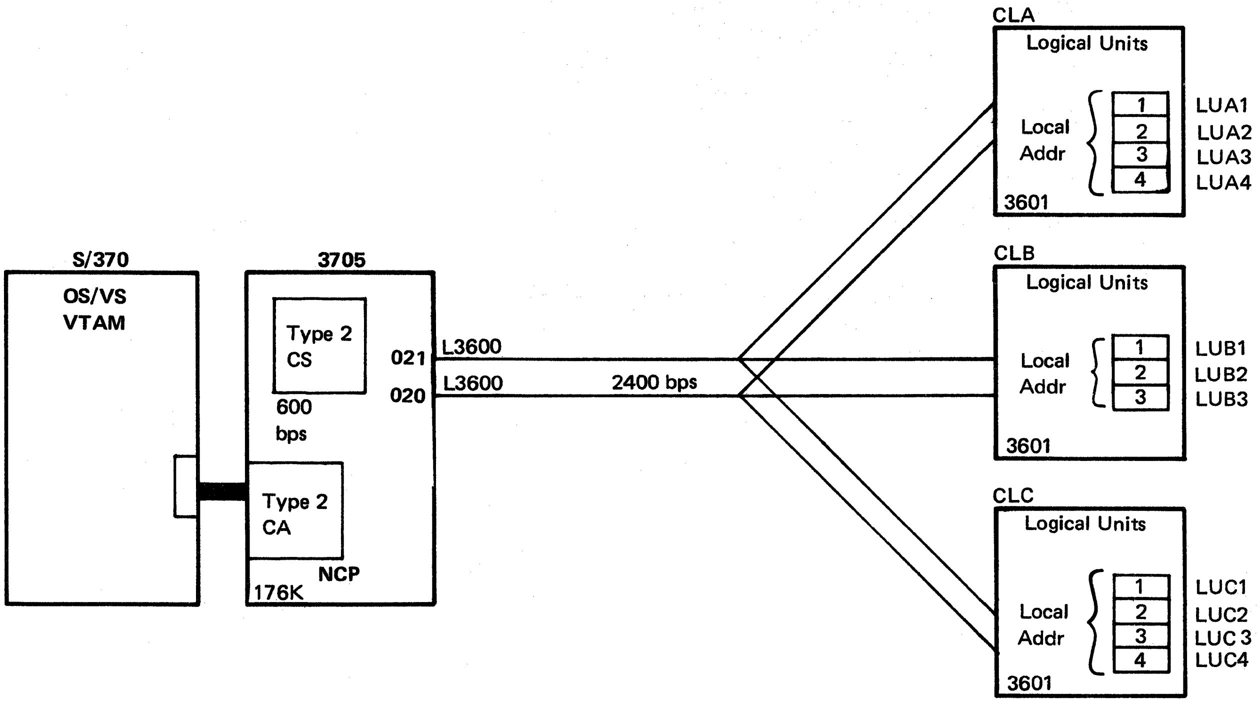
```
* SAMPLE PROGRAM #5
*
* NETWORK CONTROL PROGRAM (LOCAL) , GENERATED UNDER OS/VS, ACCESS
*   METHOD: OS/VS VTAM
*
SAMPLE5  BUILD MODEL=3705,                                             X
               MEMSIZE=112,                                            X
               TYPGEN=NCP,                                             X
               SUBAREA=3,                                              X
               MAXSUBA=7,                                              X
               BFRS=80,                                                X
               CA=TYPE1,                                               X
               CHANTYP=TYPE1,                                          X
               TYPSYS=OS,                                              X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM,                                         X
               NEWNAME=NCP3270,                                        X
               ERASE=YES,          BUFFER ERASE OPTION REQUIRED        X
               SLODOWN=25,         NCP SLOWDOWN REQD WHEN 1/4 BFRS LEFTX
               CSMHDR=27F5C8,      3270 CTL CHARS: ESC,ERASE/WRITE,WCC X
               CSMSG=5A5A5A40D5D640C6E4D9E3C8C5D940C9D5D7E4E340E4D5E3C9X
               D340D5D6E3C9C6C9C5C4405A5A5A  CS MSG TEXT: !!! NO FURTHER
                                   INPUT UNTIL NOTIFIED !!!
         SYSCNTRL OPTIONS=(RCNTRL,MODE,RCOND,RECMD,RIMM,               X
               BHSASSC,ENDCALL)
         HOST  INBFRS=3,                                               X
               MAXBFRU=10,                                             X
               UNITSZ=84,                                              X
               BFRPAD=28,                                              X
               STATMOD=YES,                                            X
               DELAY=.2
         CSB   TYPE=TYPE1,                                             X
               MOD=0,                                                  X
               SPEED=600           OSCILLATOR RATE LT 1/2 MODEM RATE
GP3270   GROUP LNCTL=BSC,                                              X
               TYPE=NCP
L3270    LINE  ADDRESS=000,                                            X
               SPEED=2400,                                             X
               CLOCKNG=EXT,        EXTERNAL (MODEM) CLOCKING USED      X
               CODE=EBCDIC,                                            X
               CRITSIT=YES,                                            X
               POLLED=YES,                                             X
               SESSION=14,         EQ OR EXCEED NO. OF DEV. IN S.O.T.  X
               SERVLIM=3,                                              X
               PAUSE=5,                                                X
               TRANSFR=3,                                              X
               CUTOFF=10,                                              X
               CDATA=YES           BUFFER ERASE REQUIRED               X
               POLIMIT=(20,QUEUE)  MAX. INPUT IS BFRS X TRANSFER X  CUTOFF
                                   = 80 X 3 x 10 = 2400 BYTES
         SERVICE ORDER=(CLUSTERA,TA1,TA2,TA3,TA4,TA5,CLUSTERB,TB1,     X
               TB2,TB3,TB4,TB5,CLUSTERC,TC1)
CLUSTERA CLUSTER CUTYPE=3271,      CLUSTER CONTROL UNIT TYPE           X
               TERM=3277,          TERMINAL TYPE (EXCEPT TA5)          X
               GPOLL=40407F7F,     GENERAL POLLING CHARACTERS          X
               XMITLIM=1           VTAM REQUIRES XMITLIM=1
TA1      TERMINAL ADDR=60604040,   SPECIFIC ADDR CHARACTERS            X
               POLL=40404040       SPECIFIC POLLING CHARACTERS
TA2      TERMINAL ADDR=6060C1C1,                                       X
               POLL=4040C1C1
TA3      TERMINAL ADDR=6060C2C2,                                       X
               POLL=4040C2C2
TA4      TERMINAL ADDR=6060C3C3,                                       X
               POLL=4040C3C3
TA5      TERMINAL ADDR=6060C4C4,                                       X
               POLL=4040C4C4,                                          X
               TERM=3284,          THIS TERMINAL IS 3284 PRINTER       X
               BFRDLAY=13          13 SEC. DELAY FOR BUFFERED PRINTER
```

```
CLUSTERB CLUSTER CUTYPE=3271,                                          X
               TERM=3277,          TERMINAL TYPE (EXCEPT  TB5)         X
               GPOLL=C1C17F7F,                                         X
               XMITLIM=3
TB1      TERMINAL ADDR=62614040,                                       X
               POLL=C1C14040
TB2      TERMINAL ADDR=6161C1C1,                                       X
               POLL=C1C1C1C1
TB3      TERMINAL ADDR=6161C2C2,                                       X
               POLL=C1C1C2C2
TB4      TERMINAL ADDR=6161C3C3,                                       X
               POLL=C1C1C3C3
TB5      TERMINAL ADDR=6161C4C4,                                       X
               POLL=C1C1C4C4,                                          X
               TERM=3284,                                              X
               BFRDLAY=13
CLUSTERC CLUSTER CUTYPE=3275,                                          X
               GPOLL=C2C27F7F,                                         X
               XMITLIM=5
TC1      TERMINAL TERM=3275
         GENEND
         END
/*
//
```

Sample 6

Network Control Program (local) (TYPGEN=NCP)

Lines and Stations:
One SDLC, nonswitched multipoint link, IBM 3600 terminals



Remarks: In this configuration, the 3705 communicates with three IBM 3601 controllers over a duplex, nonswitched multipoint SDLC link comprising separate paths (lines) for transmitting and receiving. This sample program shows the use of the text retries option, as specified in the LINE and PU macros.

```
* SAMPLE PROGRAM #6
*
* NETWORK CONTROL PROGRAM (LOCAL), GENERATED UNDER OS/VS, ACCESS
*   METHOD: OS/VS VTAM
*
SAMPLE6  BUILD MEMSIZE=176,                                            X
               TYPGEN=NCP,                                             X
               SUBAREA=3,                                              X
               MAXSUBA=7,                                              X
               CA=TYPE2,                                               X
               CHANTYP=TYPE2,                                          X
               BFRS=80,                                                X
               TYPSYS=OS,                                              X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM,                                         X
               NEWNAME=NCP3601,                                        X
               SLODOWN=25
         SYSCNTRL  OPTIONS=(MODE,RCNTRL,RCOND,RECMD,RIMM,BHSASSC,      X
               ENDCALL)
         HOST  INBFRS=3,                                               X
               MAXBFRU=10,                                             X
               UNITSZ=84,                                              X
               BFRPAD=28,                                              X
               STATMOD=YES,                                            X
               DELAY=.2
         CSB   TYPE=TYPE2,                                             X
               MOD=0,                                                  X
               SPEED=600
GP3601   GROUP LNCTL=SDLC
L3600    LINE  ADDRESS=(020,021),  TRANSMIT AND RECEIVE ADDRESSES      X
               DUPLEX=FULL,        SDLC LINK IS FULL-DUPLEX            X
               SPEED=2400,         3601'S OPERATE AT 2400 BPS          X
               POLLED=YES,                                             X
               RETRIES=5           5 RETRIES PER RECOVERY SEQUENCE     X
               MAXDATA=265,                                            X
               PASSLIM=3,                                              X
               PACING=(1,1)
         SERVICE ORDER=(CLA,CLB,CLC)
CLA      PU      ADDR=C1,          CLUSTER ADDRESS - A (EBCDIC)        X
                 PUTYPE=2,         TYPE 2 PHYSICAL UNIT                X
               MAXOUT=3,           MAX PATH INFO UNITS SENT BEFORE RESPX
               RETRIES=(,10,4)     4 RETRY SEQUENCES MAX, 10 SEC BETW'N
FALUA1XX LU    LOCADDR=1,PACING=(3,1)  (FA REQD 1ST 2 CHARS FOR 1ST
LUA2XXXX LU    LOCADDR=2                LOCAL ADDRESS (3601 LOAD ADDR))
LUA3XXXX LU    LOCADDR=3
LUA4XXXX LU    LOCADDR=4
CLB      PU    ADDR=C2,            ADDRESS IS B                        X
               PUTYPE=2,                                               X
               MAXOUT=3,                                               X
               RETRIES=(,10,4)
FALUB1XX LU    LOCADDR=1,PACING=(3,1)
LUB2XXXX LU    LOCADDR=2
LUB3XXXX LU    LOCADDR=3
CLC      PU    ADDR=C3,            ADDRESS IS C                        X
               PUTYPE=2,                                               X
               MAXOUT=3,                                               X
               RETRIES=(,10,4)
FALUC1XX LU    LOCADDR=1,PACING=(3,1)
LUC2XXXX LU    LOCADDR=2
LUC3XXXX LU    LOCADDR=3
LUC4XXXX LU    LOCADDR=4
         GENEND
         END
/*
//
```

Sample 7

Network Control Program (Local) with Partitioned Emulation Programming Extension (TYPGEN=PEP)

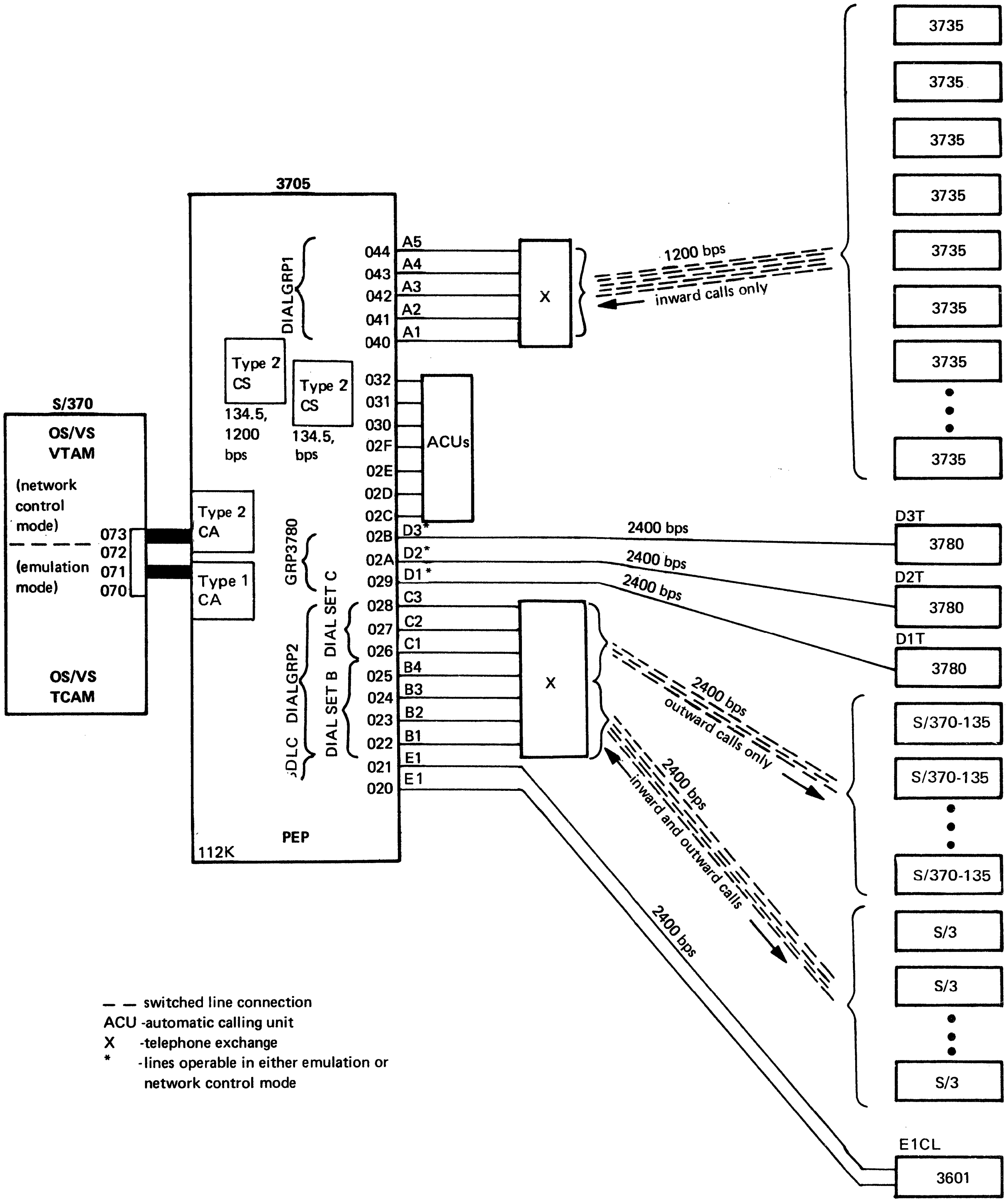
Lines and Stations:
Five BSC, switched call-in lines (network control mode only) for communicating with IBM 3735 terminals.

Four BSC switched call-in/call-out lines (network control mode only) for communicating with IBM System/3s (two lines reserved for call-in use).

Three BSC switched call-out lines (network control only) for communicating with IBM System/370 Model 135s.

Three BSC nonswitched point-to-point lines (network control or emulation mode) for communicating with IBM 3780s.

One SDLC nonswitched point-to-point link (network control mode only) for communicating with an IBM 3601 controller.



— — switched line connection
ACU -automatic calling unit
X -telephone exchange
* -lines operable in either emulation or network control mode

Remarks: This sample program is for a network configuration considerably more complex than those shown in previous samples. Illustrated is the coding required to establish switched lines and dial sets. In this example, five lines are used for receiving calls only (call-in lines), three lines for originating calls only (call-out lines), and four lines for either originating or receiving calls (call-in/call-out lines). The lines for outgoing calls are grouped into two dial sets, with one of these serving as an alternate dial set to the other.

This example also includes a single SDLC link (network control mode only) and three point-to-point BSC lines operable alternately in emulation mode and network control mode.

```
* SAMPLE PROGRAM #7
*
* NETWORK CONTROL PROGRAM (LOCAL) WITH PARTITIONED EMULATION PROGRAM-
*   MING EXTENSION, GENERATED UNDER OS/VS, ACCESS METHOD:  OS/VS VTAM
*   FOR NETWORK CONTROL MODE LINES, OS/VS TCAM FOR EMULATION MODE LINES
*
SAMPLE7  BUILD TYPGEN=PEP,                                             X
               MODEL=3705,                                             X
               MEMSIZE=176,                                            X
               SUBAREA=3,                                              X
               MAXSUBA=7,                                              X
               CA=(TYPE1,TYPE2),                                       X
               CHANTYP=(TYPE2,TYPE1), TYPE 2: NC MODE, TYPE 1: EM MODE X
               HICHAN=073,         SUBCHANNEL RANGE FOR                X
               LOCHAN=070,           EMULATION MODE DATA TRANSFER      X
               BFRS=80,                                                X
               ANS=NO,             NO AUTO NETWORK SHUTDOWN            X
               CUID=61C4C5D5E5D5D94BC3D7E4F261,   ID: /DENVER.CPU2/    X
               ERASE=YES,          BUFFER ERASE FOR LN GPS DIALGRP1,2  X
               TYPSYS=OS,                                              X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM,                                         X
               NEWNAME=PEP001
         SYSCNTRL OPTIONS=(MODE,RCNTRL,RCOND,RECMD,BHSASSC,            X
               RIMM,ENDCALL)
         HOST  NCPCHAN=019,        SUBCHAN FOR NETW CTL MODE DATA XFER X
               INBFRS=3,                                               X
               UNITSZ=84,                                              X
               MAXBFRU=10,                                             X
               BFRPAD=28,                                              X
               STATMOD=YES,                                            X
               DELAY=.2
         CSB   TYPE=TYPE2,                                             X
               MOD=0,                                                  X
               SPEED=134
         CSB   TYPE=TYPE2,                                             X
               MOD=1,                                                  X
               SPEED=(134,1200),                                       X
               WRAPLN=023          LINE ADDRESS FOR WRAPLINE TEST
*              ID LIST FOR 3735 TERMINALS
IDLISTA  IDLIST IDSEQ=(61F2F0F061,61F2F0F1C161,      /200/,/201A/      X
               61F2F0F1C261,61F2F0F561,61F2F0F761,   /201B/,/205/,/207/X
               61F5F0F961,61F5F1F261,61F5F1F7C161,   /509/,/512/,/517A/X
               61F5F1F7C261,61F5F1F7C361,61F5F3F861, /517B/,/517C/,/538/
               61F5F5F261,61F6F7F161,61F6F7F261,     /552/,/671/,/672/ X
               61F7F0F1C161,61F7F0F1C261),           /701A/,/701B/     X
               NOMATCH=STOP
         IDLIST IDSEQ=(61F7F0F661,61F7F0F761,        /706/,/707/       X
               61F7F1F261,61F7F4F461)                /712/,/744/
DIALSETB DIALSET LINES=(B1,B2,B3,B4),                                  X
               DIALALT=DIALSETC,                                       X
               RESERVE=2,          2 LINES ALWAYS RESERVED FOR         X
               QLIMIT=3,             INCOMING CALLS                    X
               QLOAD=2
DIALSETC DIALSET LINES=(C1,C2,C3),                                     X
               QLIMIT=3
DIALGRP1 GROUP LNCTL=BSC,                                              X
               DIAL=YES,                                               X
               CALL=IN,            *                                   X
               TYPE=NCP,                                               X
               CODE=EBCDIC,        *                                   X
               SPEED=1200,         *                                   X
```

```
               CLOCKNG=INT,        *                                   X
               TERM=3735,          *                                   X
               CDATA=YES,          *                                   X
               IDSEQ=IDLISTA       *
*                                  * - OPERANDS OF LINE OR TERMINAL
*                                       MACROS CODED HERE TO SAVE
*                                       CODING EFFORT
A1       LINE  ADDRESS=040
CTA1     TERMINAL CTERM=YES
A2       LINE  ADDRESS=041
CTA2     TERMINAL CTERM=YES
A3       LINE  ADDRESS=042
CTA3     TERMINAL CTERM=YES
A4       LINE  ADDRESS=043
CTA4     TERMINAL CTERM=YES
A5       LINE  ADDRESS=044
CTA5     TERMINAL CTERM=YES
DIALGRP2 GROUP LNCTL=BSC,                                              X
               DIAL=YES,                                               X
               REDIAL=5,           *    MAX. TIMES NCP WILL REDIAL     X
               TYPE=NCP,                                               X
               CODE=EBCDIC,        *                                   X
               SPEED=2400,         *                                   X
               CLOCKNG=EXT,        *                                   X
               TERM=SYS3,          *    (EITHER SYS3 OR 3135 ACCEPTABLE)
               CDATA=YES           *
*                                  * -  LINE OR TERMINAL MACRO OPERANDS
B1       LINE  ADDRESS=022,                                            X
               AUTO=02C,                                               X
               CALL=INOUT,         LINE USED FOR CALLING IN AND OUT    X
               DIALSET=DIALSETB
CTB1IN   TERMINAL CTERM=YES        LOGICAL TERMINAL FOR INCOMING CALLS
CTB1OUT  TERMINAL DIALNO=(,7),     LOG TERM FOR OUTGOING CALLS, VTAM   X
               CUIDLEN=ALL           SUPPLIES TEL NO., NCP SENDS ENTIREX
                                     CU ID SEQUENCE
B2       LINE  ADDRESS=023,                                            X
               AUTO=02D,                                               X
               CALL=INOUT,                                             X
               DIALSET=DIALSETB
CTB2IN   TERMINAL CTERM=YES
CTB2OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
B3       LINE  ADDRESS=024,                                            X
               AUTO=02E,                                               X
               CALL=INOUT,                                             X
               DIALSET=DIALSETB
CTB3IN   TERMINAL CTERM=YES
CTB3OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
B4       LINE  ADDRESS=025,                                            X
               AUTO=02F,                                               X
               CALL=INOUT,                                             X
               DIALSET=DIALSETB
CTB4IN   TERMINAL CTERM=YES
CTB4OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
C1       LINE  ADDRESS=026,                                            X
               AUTO=030,                                               X
               CALL=OUT,           LINE USED FOR CALLING OUT ONLY      X
               DIALSET=DIALSETC    (NO ALTERNATE DIALSET FOR THIS LINE)
```

```
CTC1OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
C2       LINE  ADDRESS=027,                                            X
               AUTO=031,                                               X
               CALL=OUT,                                               X
               DIALSET=DIALSETC
CTC2OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
C3       LINE  ADDRESS=028,                                            X
               AUTO=032,                                               X
               CALL=OUT,                                               X
               DIALSET=DIALSETC
CTC3OUT  TERMINAL DIALNO=(,7),                                         X
               CUIDLEN=ALL
GRP3780  GROUP LNCTL=BSC,                                              X
               DIAL=NO,                                                X
               POLLED=NO,          *                                   X
               TYPE=PEP,                GROUP OPERATES IN BOTH NETWORK X
               CODE=EBCDIC,        *      CONTROL AND EMULATION MODES  X
               SPEED=2400,         *                                   X
               CLOCKNG=EXT,        *                                   X
               TERM=3780,          *                                   X
               CU=2701,            *    TCU EMULATED IN EMULATION OPERNX
               YIELD=NO            *    (3705 IS PRIMARY STN ON LINE)
*                                  * -  LINE OR TERMINAL MACRO OPERANDS
D1       LINE  ADDRESS=(029,70),                                       X
               TYPE=PEP,                                               X
               USE=EP              LINE OPERATES INITIALLY IN EMUL MODE
D1T      TERMINAL ,                (SEE GROUP MACRO FOR OPERANDS)
D2       LINE  ADDRESS=(02A,71),                                       X
               TYPE=PEP,                                               X
               USE=EP
D2T      TERMINAL
D3       LINE  ADDRESS=(02B,72),                                       X
               TYPE=PEP,                                               X
               USE=EP
D3T      TERMINAL
GRPSDLC  GROUP LNCTL=SDLC,                                             X
               TYPE=NCP
E1       LINE  ADDRESS=(020,021),                                      X
               SPEED=2400,                                             X
               POLLED=YES
         SERVICE ORDER=(E1CL)
E1CL     PU    ADDR=C1,            CLUSTER ADDRESS IS A (EBCDIC)       X
               PUTYPE=2            TYPE 2 PHYSICAL UNIT
FAE1LU1X LU    LOCADDR=1
E1LU2XXX LU    LOCADDR=2
E1LU3XXX LU    LOCADDR=3
         GENEND
         END
/*
//
```

Sample 8

The configuration below is the most complex of those in this appendix, as it includes both local and remote communications controllers. Two sample programs are given—8-L and 8-R—for one of the local controllers and the remote controller. (The program for the other local controller [subarea5] attached to the BSC line is not included.)

Lines and Stations—Local:
Three start-stop lines (emulation mode only):

One nonswitched multipoint line, IBM 2848/2260
One nonswitched point-to-point line, IBM 2848/2260
One nonswitched multipoint line, IBM 2845/2265

One BSC nonswitched multipoint line, IBM 2701, 3705, and 1130 (network control mode only)

One SDLC nonswitched point-to-point (principal) link to a remote 3705.

One SDLC switched point-to-point (backup) link to the same remote 3705 (nonswitched line control used).

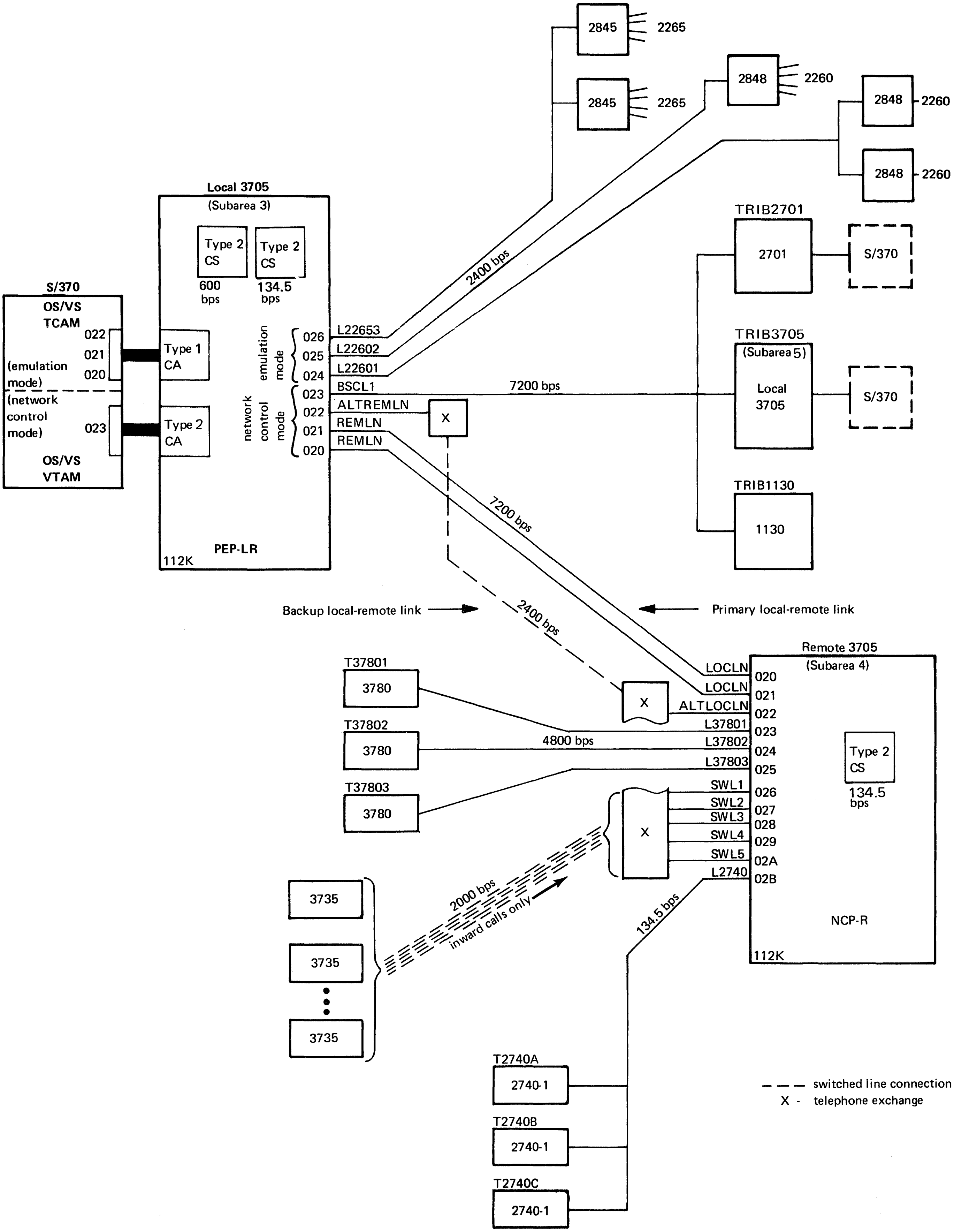
Lines and Stations—Remote:
Three BSC nonswitched point-to-point lines, IBM 3780.

Five BSC switched call-in lines, IBM 3735

One start-stop nonswitched multipoint line, IBM 2740 Model 1.

One SDLC nonswitched point-to-point (principal) link to a local 3705.

One SDLC switched point-to-point (backup) link to the same local 3705 (nonswitched line control used).


2845
2265
2845
2265
2848
2260
2848
2260
2848
2260
Local 3705
(Subarea 3)
Type 2
CS
600
bps
Type 2
CS
134.5
bps
S/370
OS/VS
TCAM
022
021
020
(emulation
mode)
(network
control
mode)
023
OS/VS
VTAM
Type 1
CA
Type 2
CA
emulation
mode
network
control
mode
026
025
024
023
022
021
020
L22653
L22602
L22601
BSCL1
ALTREMLN
REMLN
REMLN
2400 bps
7200 bps
TRIB2701
2701
S/370
TRIB3705
(Subarea 5)
Local
3705
S/370
TRIB1130
1130
X
7200 bps
PEP-LR
112K
Backup local-remote link
2400 bps
Primary local-remote link
Remote 3705
(Subarea 4)
T37801
3780
T37802
3780
T37803
3780
4800 bps
X
LOCLN
LOCLN
ALTLOCLN
L37801
L37802
L37803
SWL1
SWL2
SWL3
SWL4
SWL5
L2740
020
021
022
023
024
025
026
027
028
029
02A
02B
Type 2
CS
134.5
bps
X
3735
3735
3735
2000 bps
inward calls only
134.5 bps
NCP-R
112K
T2740A
2740-1
T2740B
2740-1
T2740C
2740-1
– – – switched line connection
X - telephone exchange

Remarks—Sample 8-L: This sample program, executed in a local 3705, supports emulation mode lines, network control mode lines, and a remote communications controller; it is therefore specified as TYPGEN=PEP-LR. Notice that the 3705 in which the program will be executed is equipped with separate channel adapters for network control mode and emulation mode data transfer. Also notice the use of the address substitution option (GENEND macro), required because of the presence of lines operating at 7200 bps.

Remarks—Sample 8-R: Because the sample program is to be executed in a remote 3705, it is specified as TYPGEN=NCP-R. Notice the absence of channel information in the BUILD macro; the remote controller has no channel adapter. The operands and parameters of the HOST macro are specified identically to those in the HOST macro of the local NCP (sample 8-L). Notice further that because all BSC switched lines are used only for incoming calls, no dial sets are specified, as is the case in sample 7. Finally, observe that most operands of the LINE and TERMINAL macros specifiable at a higher level appear in the GROUP macro, to save coding effort.

```
* SAMPLE PROGRAM #8-L
*
* NETWORK CONTROL PROGRAM WITH PARTITIONED EMULATION PROGRAMMING EXTEN-
*   SION -- LOCAL WITH REMOTE SUPPORT, GENERATED UNDER OS/VS,
*   ACCESS METHOD: OS/VS VTAM FOR NETWORK CONTROL MODE LINES,
*   OS/VS TCAM FOR EMULATION MODE LINES
*
SAMPLE8L BUILD MODEL=3705,                                             X
               MEMSIZE=112,                                            X
               TYPGEN=PEP-LR,      LOCAL NCP W/ PEP AND REMOTE SUPPORT X
               SUBAREA=3,                                              X
               MAXSUBA=7,                                              X
               CA=(TYPE1,TYPE2),                                       X
               CHANTYP=(TYPE2,TYPE1), TYPE 2 CA FOR NETW CTL MODE DATA X
               HICHAN=023,            TRANSFER, TYPE 1 FOR EMUL MODE   X
               LOCHAN=020,                                             X
               ANS=YES,            AUTO NETWORK SHUTDOWN               X
               BFRS=80,                                                X
               TYPSYS=OS,                                              X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM,                                         X
               NEWNAME=LOCNCP
         SYSCNTRL OPTIONS=(MODE,RCNTRL,RCOND,RECMD,RIMM,               X
               ENDCALL,BHSASSC)
         HOST  INBFRS=3,                                               X
               MAXBFRU=10,                                             X
               BFRPAD=28,                                              X
               UNITSZ=84,                                              X
               DELAY=.2,                                               X
               STATMOD=YES
         CSB   TYPE=TYPE2,                                             X
               MOD=0,                                                  X
               SPEED=600,                                              X
               WRAPLN=023
         CSB   TYPE=TYPE2,                                             X
               MOD=1,                                                  X
               SPEED=134,                                              X
               WRAPLN=0A3
BSCGRP   GROUP LNCTL=BSC,                                              X
               TYPE=NCP
BSCL1    LINE  CODE=EBCDIC,                                            X
               SPEED=7200,                                             X
               CLOCKNG=EXT,                                            X
               ADDRESS=023,                                            X
               POLLED=YES
         SERVICE ORDER=(TRIB3705,TRIB2701,TRIB1130)
TRIB3705 TERMINAL TERM=3705,                                           X
               ADDR=E1,                                                X
               POLL=C1,                                                X
               XMITLIM=10
TRIB2701 TERMINAL TERM=2701,                                           X
               ADDR=E2,                                                X
               POLL=C2,                                                X
               XMITLIM=10
TRIB1130 TERMINAL TERM=1130,                                           X
               ADDR=E3,                                                X
```

```
               POLL=C3,                                                X
               XMITLIM=3
EM2260GP GROUP LNCTL=SS,                                               X
               TYPE=EP,                                                X
               SPEED=2400,                                             X
               CLOCKNG=EXT,                                            X
               CU=2701,                                                X
               DUPLEX=FULL
L22601   LINE  TERM=2260,                                              X
               MULTI=YES,          MULTIPLE DISPLAYS ON LINE           X
               ADDRESS=(024,20)
L22602   LINE  TERM=2260,                                              X
               MULTI=NO,                                               X
               ADDRESS=(025,21)
L22653   LINE  TERM=2265,                                              X
               MULTI=YES,                                              X
               ADDRESS=(026,22)
REMOTEGP GROUP LNCTL=SDLC,                                             X
               TYPE=NCP
REMLN    LINE  ADDRESS=(020,021),  PRINCIPAL SDLC LINK TO REMOTE CTLR  X
               DUPLEX=FULL,        COMMUNICATIONS FACILITY IS DUPLEX   X
               SPEED=7200,                                             X
               CLOCKNG=EXT,                                            X
               POLLED=YES,                                             X
               RETRIES=(3,10,10)
         SERVICE ORDER=REM3705A
REM3705A PU     ADDR=C1,           PHYSICAL ADDR (A) OF REMOTE 3705    X
               PUTYPE=4,           TYPE 4 PHYSICAL UNIT                X
               MAXOUT=4,           MAX OF 4 PIU'S SENT BEFORE RESPONSE X
               DATMODE=FULL,       DATA XFER SIMULTANEOUS IN BOTH DIREC-
               SUBAREA=4           SUBA ADDR OF REMOTE 3705       TIONS
ALTREMLN LINE  ADDRESS=022,        ALT (BACKUP) SDLC LINK TO REM CTLR  X
               DUPLEX=HALF,                                            X
               SPEED=2400,                                             X
               CLOCKNG=EXT,                                            X
               POLLED=YES,                                             X
               RETRIES=(3,10,10)
         SERVICE  ORDER=REM3705B
REM3705B PU    PUTYPE=4
         GENEND SCANCTL=(,,,,1100) ADDR SUBSTITUTION MASK -- ADDR SUBST.
                                   REQUIRED TO ACCOMMODATE 7200 BPS    X
                                   LINES
         END
 /*
 //
```

```
* SAMPLE PROGRAM #8-R
*
* NETWORK CONTROL PROGRAM (REMOTE), GENERATED UNDER OS/VS, ACCESS
*   METHOD: OS/VS VTAM
*
SAMPLE8R BUILD MODEL=3705,                                             X
               MEMSIZE=112,                                            X
               TYPGEN=NCP-R,                                           X
               SUBAREA=4,          (NOTE ABSENCE OF CHANNEL            X
               MAXSUBA=7,           INFORMATION, NOT APPLICABLE        X
               BFRS=80,             FOR A REMOTE NCP)                  X
               TYPSYS=OS,                                              X
               ANS=YES,            AUTO NETWORK SHUTDOWN               X
               LOADLIB=NCPLIB,                                         X
               OBJLIB=STG2ASM,                                         X
               NEWNAME=REMNCP
         SYSCNTRL OPTIONS=(MODE,RCNTRL,RCOND,RECMD,                    X
               RIMM,ENDCALL,BHSASSC)
         HOST  INBFRS=3,                                               X
               MAXBFRU=10,         CODE SAME OPERANDS AND VALUES       X
               BFRPAD=28,          AS IN HOST MACRO OF LOCAL NCP       X
               UNITSZ=84,                                              X
               STATMOD=YES
         CSB   TYPE=TYPE2,                                             X
               SPEED=134
GP3780   GROUP LNCTL=BSC,                                              X
               CODE=EBCDIC,                                            X
               SPEED=4800,                                             X
               CLOCKNG=EXT,                                            X
               TERM=3780
L37801   LINE  ADDRESS=023
T37801   TERMINAL ,
L37802   LINE  ADDRESS=024
T37802   TERMINAL ,
L37803   LINE  ADDRESS=025
T37803   TERMINAL ,
SW3735GP GROUP LNCTL=BSC,                                              X
               SPEED=2000,                                             X
```

```
               CLOCKNG=EXT,                                            X
               TERM=3735,                                              X
               CALL=IN,                                                X
               DIAL=YES,                                               X
               CODE=EBCDIC
SWL1     LINE  ADDRESS=026
SWT1     TERMINAL CTERM=YES
SWL2     LINE  ADDRESS=027
SWT2     TERMINAL CTERM=YES
SWL3     LINE  ADDRESS=028
SWT3     TERMINAL CTERM=YES
SWL4     LINE  ADDRESS=029
SWT4     TERMINAL CTERM=YES
SWL5     LINE  ADDRESS=02A
SWT5     TERMINAL CTERM=YES
GP2740   GROUP LNCTL=SS,                                               X
               SPEED=134,                                              X
               CODE=COR,                                               X
               POLLFD=YES,                                             X
               CLOCKNG=INT,                                            X
               TERM=2740-1,                                            X
               FEATURE=SCTL,       TERMINALS HAVE STN CONTROL FEATURE  X
               XMITLIM=3           TRANSMISSION LIMIT FOR TERMINALS
L2740    LINE  ADDRESS=02B
         SERVICE ORDER=(T2740A,T2740B,T2740C)
T2740A   TERMINAL POLL=C1,         POLLING AND ADDRESSING CHARACTER:   X
                  ADDR=C1          A (EBCDIC)
T2740B   TERMINAL POLL=C2,         B (EBCDIC)                          X
                  ADDR=C2
T2740C   TERMINAL POLL=C3,         C (EBCDIC)                          X
                  ADDR=C3
SDLCGP   GROUP LNCTL=SDLC,                                             X
               DIAL=NO
LOCLN    LINE  ADDRESS=(020,021),  PRINCIPAL SDLC LINK TO LOCAL CTLR   X
               SPEED=7200,                                             X
               CLOCKNG=(EXT,EXT),                                      X
               TADDR=C1,           PHYSICAL ADDR OF REMOTE CONTROLLER  X
               NEWSYNC=NO,         IN WHICH THIS PROGRAM WILL BE       X
               DUPLEX=FULL         EXECUTED--NOTE THAT ADDR IS SAME AS X
                                   SPECIFIED IN PU MACRO OF LOCAL NCP
LOC3705A PU    PUTYPE=4            TYPE 4 PHYS UNIT; MAXOUT DEFAULT=7
ALTLOCLN LINE  ADDRESS=022,        ALT (BACKUP) SDLC LINK TO LOCAL CTLRX
               SPEED=2400,                                             X
               CLOCKNG=EXT,                                            X
               TADDR=C1,                                               X
               NEWSYNC=NO,                                             X
               DUPLEX=FULL
LOC3705B PU    PUTYPE=4
         GENEND
         END
/*
//
```

Appendix J: Procedure for Determining Line Interrupt Priorities

This appendix gives a recommended procedure for determining the interrupt priorities for lines serviced by type 2 and type 3 communication scanners.

You can achieve optimum servicing of communication lines and maximum protection from overruns (1) by using the priority registers associated with all four interrupt priorities and (2) by evenly distributing the total throughput of the network, in bits per second, among the four different priorities. This is the case whether lines in the network operate in network control mode or emulation mode.

The following procedure ensures that the four interrupt priorities are evenly distributed among lines in the network.

1. Divide by 8 the speed (as specified by the SPEED operand of the LINE macro) of each line serviced by a *type 3* communication scanner. The result is the adjusted line speed for such lines and is to be used in the remaining steps of the procedure. For lines serviced by a *type 2* scanner, the adjusted line speed is the same as the speed specified in the SPEED operand.
2. For each speed category, multiply the adjusted speed by the number of lines to which that speed applies.
3. Calculate the total throughput rate for all lines (in bits per second) by adding up the values calculated in step 2. Then divide the result by 4 to determine one-fourth of the throughput in bits per second.
4. List all lines in the network in the sequence of their adjusted line speeds. Those lines having the highest adjusted speed should appear at the top of the list and those having the lowest adjusted speed should appear at the bottom. Within any speed category, the sequence of lines does not matter.
5. Divide the list of lines into four sections such that the throughput for each section is roughly the same as one-fourth of the total throughput.
6. Assign priority 3 to each line in the first section of the list, priority 2 to the lines in the second section, priority 1 to those in the third section, and priority 0 to those lines in the last section.
7. Specify the assigned priority in the INTPRI operand of the LINE macro representing each line.

For purposes of the foregoing procedure, approximately the same proportion of lines in each section of the list are assumed to be active at any given moment. If the planned use of the network or experience shows that the proportions are markedly different, you may wish to adjust the distribution of lines to the sections of the list to compensate, then respecify the values in the INTPRI operands for the affected lines.

For instance, if experience shows that several of the lines in the last section of the list are relatively inactive compared to lines in the other sections, you could adjust each of the section boundaries upward so that more lines appear in the last group and fewer in each of the other sections.

The use of the procedure is illustrated by the following examples.

Example One:

The network has six lines rated at 9600 bps and serviced by a type 3 scanner, and thirteen lines—seven rated at 2400 bps and six rated at 600 bps—serviced by a type 2 scanner.

Determine the total throughput and one-fourth of the total:

Line ID (Name of LINE Macro)	*Line Speed (LINE: SPEED)*	*Type of Scanner*	*Adjusted Line Speed*	*Number of Lines*	*Throughput*
			(Step 1)		(Step 2)
LH1-LH6	9,600	3	1,200	6	7,200
LM1-LM7	2,400	2	2,400	7	16,800
LL1-LL6	600	2	600	6	3,600

Total throughput (Step 3) 27,600
One-fourth of total 6,900

List lines in order of adjusted speed (step 4), divide list into four sections (step 5), and assign priorities to each section (step 6):

Line ID (Name of LINE Macro)	*Adjusted Line Speed*	*Total Bit Rate for Section*	*Interrupt Priority*
LM1	2,400	7,200	3
LM2	2,400		
LM3	2,400		
LM4	2,400		
LM5	2,400	7,200	2
LM6	2,400		
LM7	2,400		
LH1	1,200	6,000	1
LH2	1,200		
LH3	1,200		
LH4	1,200		
LH5	1,200		
LH6	1,200		
LL1	600		
LL2	600	7,200	0
LL3	600		
LL4	600		
LL5	600		
LL6	600		

As step 7, specify the priority values in the INTPRI operands of the LINE macros.

Example Two:

The network has eleven lines serviced by a type 3 scanner—three lines at 19,200 bps, three at 9,600 bps, and five at 2,400—and the following lines serviced by a type 2 scanner: ten lines at 1,200 bps, nine at 600 bps, seven at 150 bps, and eight at 134.5 bps.

As in example one, first determine the total throughput and one-fourth of that value, then list the lines in order of adjusted line speed, divide into four sections, and assign priorities.

Line ID (Name of LINE Macro)	*Line Speed (LINE: SPEED)*	*Type of Scanner*	*Adjusted Line Speed*	*Number of Lines*	*Throughput*
LA1-LA3	19,200	3	2,400	3	7,200
LB1-LB3	9,600	3	1,200	3	3,600
LC1-LC5	2,400	3	300	5	1,500
LD1-LD10	1,200	2	1,200	10	12,000
LE1-LE9	600	2	600	9	5,400
LF1-LF7	150	2	150	7	1,050
LG1-LG8	134.5	2	134*	8	1,072

*Decimal fraction dropped as insignificant

Total throughput 31,822
One-fourth of total 7,956

Line ID (Name of LINE Macro)	*Adjusted Line Speed*	*Total Bit Rate for Section*	*Interrupt Priority*
LA1	2,400		
LA2	2,400	7,200	3
LA3	2,400		
LB1	1,200		
LB2	1,200		
LB3	1,200		
LD1	1,200	8,400	2
LD2	1,200		
LD3	1,200		
LD4	1,200		
LD5	1,200		
LD6	1,200		
LD7	1,200		
LD8	1,200	8,400	1
LD9	1,200		
LD10	1,200		
LE1	600		
LE2	600		
LE3	600		
LE4	600		
LE5	600		
LE6	600		
LE7	600		
LE8	600		
LE9	600		
LC1	300		
LC2	300		
LC3	300		
LC4	300		
LC5	300		
LF1	150		
LF2	150	7,822	0
LF3	150		
LF4	150		
LF5	150		
LF6	150		
LF7	150		
LG1	134		
LG2	134		
LG3	134		
LG4	134		
LG5	134		
LG6	134		
LG7	134		
LG8	134		

Appendix K: Upper Scan Limits, Address Substitution, and High Speed Select Options

The maximum data rate, or speed, at which a communication line can operate is limited by the frequency at which that line's interface address is scanned by the communication scanner. In the absence of upper scan limits, address substitution, and high speed select options, each line interface address associated with a type 2 or type 3 communication scanner is scanned once per scanning cycle. The maximum line speed in this case is 4800 bits per second (bps). (The type of line set and oscillator or modem clocking rate determines the actual line speed. To accommodate higher maximum line speeds requires the imposition of upper scan limits or the application of the address substitution or high speed select technique.

Upper Scan Limits:

Imposing an upper scan limit is a means of increasing the frequency at which a selected range of line addresses is scanned, at the expense of not scanning the remaining addresses associated with the scanner. (No lines attached to the unscanned addresses can be active while the upper scan limit is in effect.) By not scanning some addresses, the communication scanner can scan the others more often within each scanning cycle, thus raising the maximum line speed. (The number of scans per cycle is constant regardless of how they are distributed to the line addresses.)

Upper scan limits can be specified individually for each of the type 2 and type 3 scanners in the communications controller. Figure K-2 shows for each scan limit value (from 0 to 3, as specified in the GENEND macro of the emulation program or network control program), the range of addresses that are scanned (light boxes) and those not scanned (dark boxes). Also given is the maximum line speed for the addresses scanned (disregarding any lower limit that may be imposed by choice of line set and oscillator speeds).

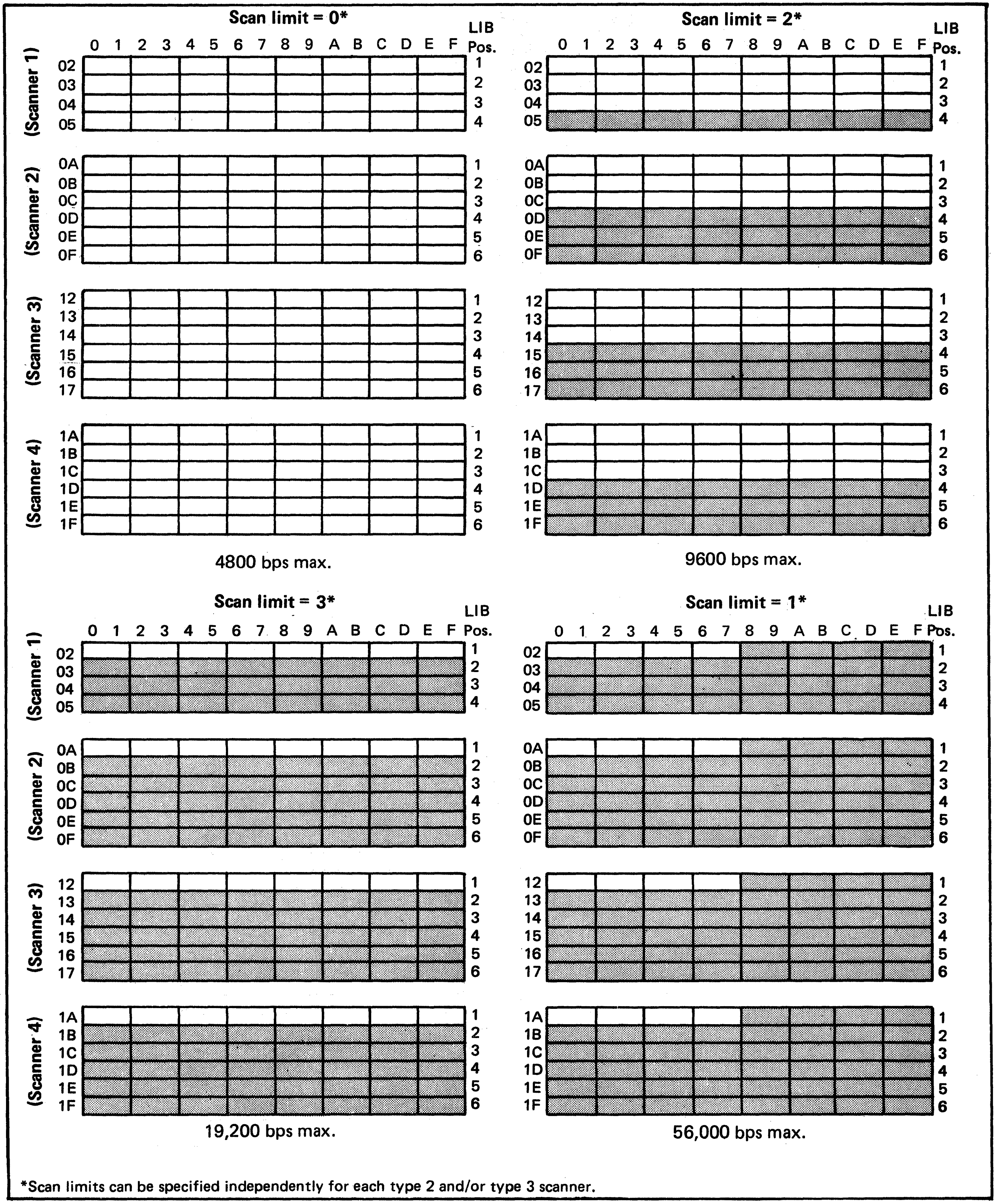
Address Substitution

A technique similar to the use of upper scan limits alters the scanning pattern so that a single, predetermined address is scanned several times per scanning cycle, again at the expense of not scanning other addresses. In this technique, however, only one address, of the range of addresses scanned, receives the benefit of increased scanning frequency. Further, address substitution applies uniformly to *all installed type 2 (and type 3) scanners,* rather than being specified separately for the individual scanners. Address substitution should not be specified if the controller contains one or more type 3 scanners.

Figure K-2 shows, for each bit position in the address substitution mask (specified in the GENEND macro of the emulation program or network control program), the selected addresses that will be scanned more often, and the group of addresses that will accordingly not be scanned. As is the case for upper scan limits, no lines attached to the unscanned addresses can be active while address substitution is in effect.

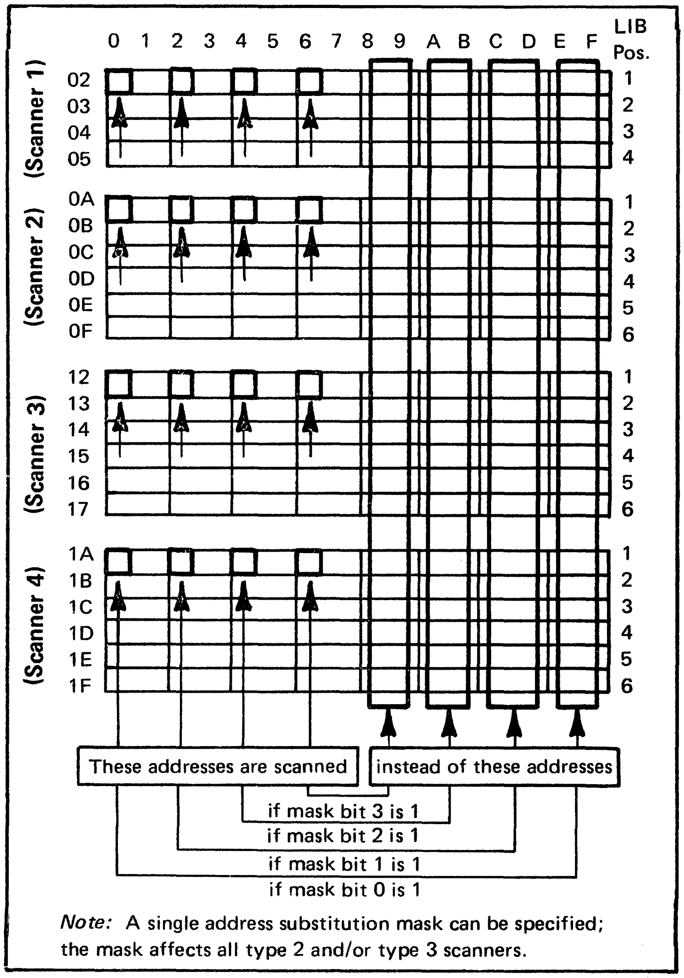
High Speed Select Option

The high speed select option is similar to address substitution in that bit settings within a mask alter the scanning pattern so that a predetermined address is scanned several times per scanning cycle, at the expense of not scanning other addresses. This option differs from address substitution, however, in that (1) up to eight addresses serviced by the scanner can receive the increased scanning frequency, and (2) masks are individually specified for each of the installed scanners, thus allowing more flexibility in selecting addresses to receive the increased scanning. Figure K-3 shows, for each bit position in the high speed select mask (specified in the GENEND macro) the selected addresses that will be scanned more often and the group of addresses that will not therefore be scanned. No lines attached to the unscanned addresses can be active while the high speed select mask is in effect.



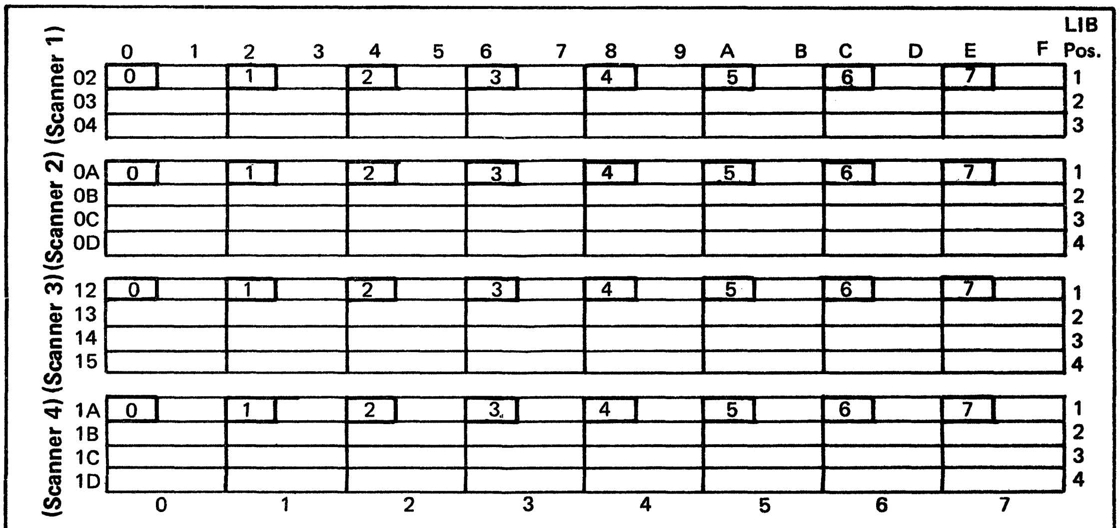
*Scan limits can be specified independently for each type 2 and/or type 3 scanner.

Figure K-1. Addresses Scanned and Not Scanned When Upper Scan Limits Are Used



Note: A single address substitution mask can be specified; the mask affects all type 2 and/or type 3 scanners.

Figure K-2. Addresses Scanned and Not Scanned When Address Substitution Mask Is Used



[n] = the address scanned at increased frequency when mask bit *n* equals 1; all other addresses within the LIB portion are not scanned.

Note: High speed select masks can be specified independently for each type 3 scanner.

Figure K-3. Addresses Scanned and Not Scanned When High Speed Select Mask Is Used

Appendix L: Supplemental Information for Airlines Line Control Users

This appendix contains information needed to specify communication lines to use the airlines line control (ALC) procedure. The information herein, which supplements the information in the remainder of this publication, is applicable only when you are defining a control program for an IBM 3705-II Communications Controller in which RPQ numbers 858911 and 858912 are installed.

To define a line as using airlines line control (ALC), rather than BSC or start-stop line control, specify LNCTL=ALC in the GROUP macro and observe the requirements indicated under each of the macro instructions and operands mentioned below.

Note: LNCTL=ALC is valid only for a line serviced by a type 3 communication scanner in a 3705-II controller.

Certain operands of the LINE macro may be specified instead in the GROUP macro. The operands for which you may do so are the same for an LLC line as for a non-ALC line.

BUILD Macro Instruction

Use of ALC imposes no special requirements on coding of the BUILD macro and its operands.

CSB Macro Instruction

No line for which LNCTL=ALC is specified should be specified in the WRAPLN operand of the CSB macro unless all lines serviced by the scanner represented by the CSB macro are ALC lines. Otherwise, use of ALC imposes no special requirements on coding of the CSB macro and its operands.

GROUP Macro Instruction

Each ALC line must be included in a line group, represented by a GROUP macro. No ALC line may be included in more than one line group, and all lines in an ALC line group must be ALC lines.

All ALC lines are nonswitched (DIAL=NO).

In general, the operands of the GROUP macro that apply to nonswitched BSC lines apply also to ALC lines. Specific requirements for coding these operands are as follows.

Invalid Operands

CHAREC
DELAY
EOB
EOT
QUIETCT
REPLYTO
TEXTTO

Valid Operands

The following operands are valid for an ALC line group. Specific requirements are indicated.

DIAL:	Specify DIAL=NO or omit this operand.
LNCTL:	Specify LNCTL=ALC.

In general, the operands of the LINE macro that apply to nonswitched BSC lines apply also to ALC lines. Specific requirements for coding these operands are as follows.

Invalid Operands

The following operands are invalid for ALC lines and if coded are ignored.

AUTO
CHECK
CODE
DUALCOM
MULTI
QUIET
RING
TADDR
TERM
UNITXC

Valid Operands

The following operands are valid for an ALC line. Specific requirements are indicated.

ADDRESS and RCVADDR: ALC lines are arranged in pairs and attached to consecutive line interface addresses, the lower of which must be an even address. The even address is used for transmit operations. The next higher (odd) address is used for receive operations. In the ADDRESS operand specify only the even (transmit) address and the associated emulation subchannel address. *Example:* ADDR ADDRESS=(0A0, 43). In the RCVADDR operand specify the odd (receive) address and the associated emulation subchannel address. *Example*: RCVADDR=(0A1, 42).

The ADDRESS and RCVADDR operands are required.

Note: Any line interface address you specify for an ALC line must be an address associated with a type 3 communication scanner.

Specify the subchannel address in the ADDRESS and RCVADDR operands in the same way as for a non-ALC line. Because an ALC line must be associated with a type 4 channel adapter, you may specify multiple subchannel address for the line interface address you specify. (See Figure J-2 for an example.)

BUFSIZE: Specify BUFSIZE=(n1 ,[n2]), in which n1 represents the emulation mode buffer size for the transmit (even) line address specified in the ADDRESS operand and n2 represents the emulation mode buffer size for the receive (odd) line address specified in the RCVADDR operand.

n1 and n2 may be any of the following values:

8	64	160
16	96	192
32	128	224

If you omit this operand, a buffer size of 32 [BUFSIZE=(32,32)] is assumed.

CHNPRI: No special requirements apply to the use of this operand.

CLOCKNG: No special requirements apply to the use of this operand.

CU: Specify CU=2703 or omit this operand.

CUTYPE: Specify CUTYPE=1006 or CUTYPE=2946 or CUTYPE=2948, as appropriate for the type of station attached to the ALC line. If a mixture of station types are attached to the line, specify CUTYPE=MIXD.

DATRATE: No special requirements apply to the use of this operand.

DISABLE: Specify DISABLE=NO or omit this operand.

DUPLEX: Specify DUPLEX=FULL.

INTPRI: No special requirements apply to the use of this operand.

MODEM: Specify MODEM=OPTION2 or omit this operand.

NEWSYNC: No special requirements apply to the use of this operand.

PAD: Specify PAD=YES or omit this operand.

RCVADDR: (See ADDRESS operand in this appendix.)

SPEED: Line speeds between 2400 and 9600 bps, inclusive, are valid.

GENEND Macro Instruction

Use of ALC imposes no special requirements on coding of the GENEND macro and its operands.

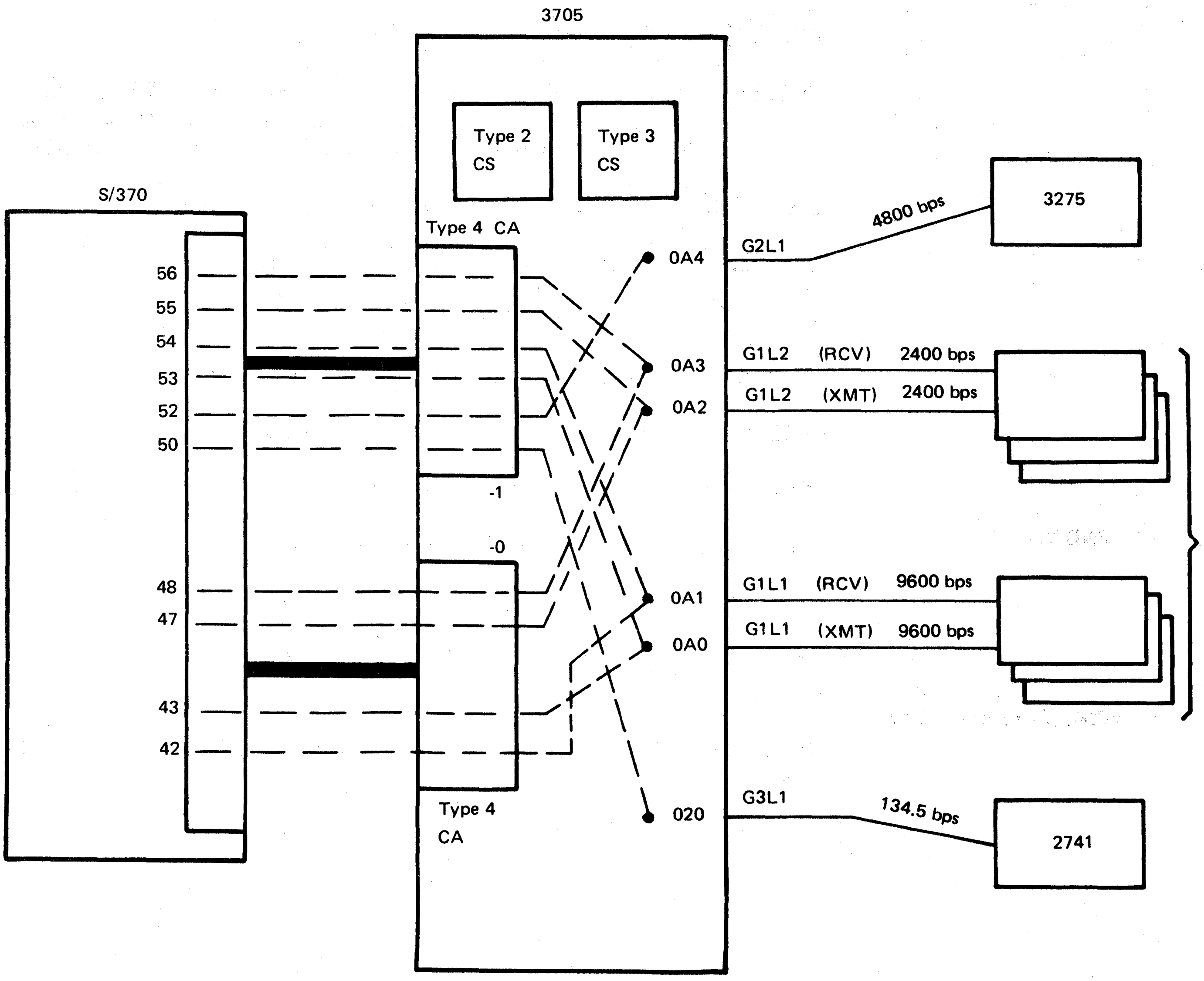


Figure L-1. Network Configuration for Sample Program for ALC Users

```
         BUILD   MODEL=3705-2,                                         X
                 CA=(TYPE4, TYPE4),                                    X
                 LOCHAN=(40, 50),                                      X
                 HICHAN=(4F, 5F),                                      X
                 TEST=YES,                                             X
                 LINETRC=YES,                                          X
                 DYNADMP=(YES,, 5F),                                   X
                 LOADLIB=LOD3705,                                      X
                 OBJLIB=OBJ3705,                                       X
                 NEWNAME=ALC01

         CSB     TYPE=TYPE2,                                           X
                 MOD=0,                                                X
                 SPEED=(134, 600),                                     X
                 WRAPLN=020

         CSB     TYPE=TYPE3,                                           X
                 MOD=1,                                                X
                 SPEED=(150, 600, 1200),                               X
                 WRAPLN=0A4

G1       GROUP   LNCTL=ALC,                                            X
                 CLOCKNG=EXT,                                          X
                 DUPLEX=FULL,                                          X
                 CUTYPE=MIXD

G1L1     LINE    ADDRESS=(0A0, 43, 53-1),                              X
                 RCVADDR=(0A1, 42, 54-1),                              X
                 SPEED=9600,                                           X
                 BUFSIZE=(64, 32),                                     X
                 INTPRI=3,                                             X
                 CHNPRI=HIGH

G1L2     LINE    ADDRESS=(0A2, 47, 55-1),                              X
                 RCVADDR=(0A3, 48, 56-1),                              X
                 SPEED=2400

G2       GROUP   LNCTL=BSC,                                            X
                 CODE=EBCDIC

G2L1     LINE    ADDRESS=(0A4, 52-1),                                  X
                 SPEED=4800,                                           X
                 TERM=3275,                                            X
                 CU=2701

G3       GROUP   LNCTL=SS,                                             X
                 CLOCKNG=INT

G3L1     LINE    ADDRESS=(020, 50-1),                                  X
                 SPEED=134,                                            X
                 DUPLEX=FULL,                                          X
                 UNITXC=NO,                                            X
                 TERM=2741

         GENEND
         END
```

Figure L-2. Sample Emulation Program for ALC Users

Glossary

access method: A data management technique for transferring data between main storage and an input/output device. In this publication, teleprocessing access method refers to the data management technique, executed in the host processor, that transfers data between the host processor and the 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.

block: The smallest data unit recognized by the network control program. 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 controller hardware unit that provides attachment of the controller to a System/360 or System/370 channel.

cluster: A station that consists of a control unit (cluster controller) and the terminals attached to it.

communication scanner: A controller hardware unit that provides the interface between line interface bases and the central control unit. The communication scanner monitors the communication lines for service requests.

component: An independently addressable part of a station that performs either an input or an output function, 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 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 controller and stations, and between the controller 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.

duplex line: A communication line having two independent data paths over which data can be transmitted in both directions simultaneously. (Also called full-duplex line.) Contrast with *half-duplex line.*

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 automatically attempts to correct a transmission error.

formatted dump: A dump in which certain network control program control blocks are isolated and identified.

full-duplex line: See *duplex line.*

generation delimiter macro: The network control program generation macro that marks the end of the network control program generation input stream.

half-duplex line: A communication line having a single data path over which data can be transmitted in either direction, but not simultaneously. Contrast with *duplex line.*

host processor: The central processing unit to which a local communications controller is attached by a channel and that executes the teleprocessing access method that supports the controller.

initial test routine: A diagnostic program executed in the controller before the network control program is loaded. The initial test routine tests the controller 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.

line control character: A special character that controls transmission of data over a start-stop or BSC 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 controller.

line interface base (LIB): A controller hardware unit that provides for the attachment of communication lines to the controller.

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 controller.

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.

local communications controller: A communications controller attached to a CPU (the host processor) by channel adapter.

logical unit: An application program within an SDLC cluster controller, represented within the NCP by a LU macro instruction.

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 user defines for the controller 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 control program generation language.

network control program: A control program for the controllers, generated by the user from a library of IBM-supplied modules.

on-line testing: Diagnostic aids by which (1) 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 (on-line terminal testing); or (2) a similar group of tests may be performed on a communications line (on-line line testing).

pacing: A means for limiting the number of path information units (PIU) sent to a logical unit on an SDLC link until the logical unit acknowledges its ability to receive more PIUs. Use of this option can prevent needless transmission of PIUs to a logical unit before it is ready to accept them.

parameter: A variable that is given a constant value for a specific purpose or process.

path information unit (PIU): The basic unit of transmission in a teleprocessing network. Path information units may request a particular teleprocessing operation (request PIU) or indicate the result of an operation (response PIU).

pause-retry: A network control program option that allows the user to specify how many times the program should try to retransmit data after a transmission error occurs, and how long the program should wait between successive attempts.

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.

remote communications controller: A communications controller that communicates over a communications line with a local communications controller, instead of being attached directly to the host processor by a channel adapter.

remote power off: An optional, program-supported feature of a remote communications controller by which the controller power can be turned off by command from the host processor.

request: A directive from the access method that causes the network control program to perform a data transfer operation or auxiliary operation.

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.)

SDLC link: A communications line over which communications are conducted using the synchronous data link control (SDLC) scheme.

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 start-stop or BSC multipoint line 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 start-stop or BSC device.

session limit: The maximum number of concurrent sessions that can be initiated on a multipoint line.

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, cluster control units, and terminals in the teleprocessing network connected to the controller.

switched network backup: 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.

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 controller.

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 controller via a communication line and identified as a cluster, terminal, or component at the time the network control program is generated.

teleprocessing network: The stations that are controlled by a single access method (or, in the controller, by a single network control program), and the communication lines by which they are connected to the communications 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, modems (data sets), communication lines, and the communications control unit.

terminal: A teleprocessing device capable of transmitting or receiving data (or both) over a communication line.

test request message: A message entered from a terminal or console requesting that a specified on-line 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 or line 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 start-stop or BSC device during one session on a multipoint line 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 network control program during program generation.

List of Abbreviations

ACU	automatic calling unit
bcc	block checking character
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 communication
BTAM	Basic Telecommunications Access Method
CPU	central processing unit
CR	carriage return
DLE	data link escape
DOS	Disk Operating System
DOS/VS	Disk Operating System/Virtual Storage
EBCD	Extended Binary Coded Decimal
EBCDIC	Extended Binary Coded Decimal Interchange Code
EIB	error information block
EOB	end of block
EOT	end of transmission
EP	emulation program
ERP	error recovery procedure
ETB	end of transmission block
ETX	end of text
HT	horizontal tab
ID	identification
I/O	input/output
IPL	initial program load
ITB	intermediate transmission block
K	thousand (1,024, when referring to bytes of storage)
MTA	multiple terminal access
NCP	network control program
OLTT	on-line terminal test
OS	Operating System
OS/VS	Operating System/Virtual Storage
PEP	partitioned emulation programming (extension)
PIU	path information unit
QTAM	Queued Telecommunications Access Method
SDLC	Synchronous Data Link Control
SOH	start of heading
STX	start of text
TCAM	Telecommunications Access Method
TCU	transmission control unit
TSO	time-sharing option
TWX	teletypewriter exchange service
UCW	unit control word
USASCII	United States of America Standard Code for Information Interchange
VS	virtual storage
VTAM	virtual Telecommunications Access Method
WRU	"who-are-you" control characters

Note: *Page references in chapters 2 and 3 apply to SDLC networks only. References in Chapters 6 and 7 apply to emulation-mode operations only. References in Chapters 8, 9, and 10 apply to program generation and utilities under OS/VS; references in Chapters 11, 12, and 13 apply to program generation and utilities under DOS/VS.*

IBM 3704 and 3705
Communications Controllers
Network Control Program/VS
Generation and Utilities
Guide and Reference Manual
(for OS/VS and DOS/VS VTAM Users)

READER'S COMMENT FORM

Your comments, accompanied by answers to the following questions, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. Comments and suggestions become property of IBM.

Please direct *any requests for copies of publications or for assistance in using your IBM System or equipment,* to your IBM representative or to the IBM sales office serving your locality.

- Did this publication meet your needs?
- What is your occupation?
- How did you use this publication?

COMMENTS

We would appreciate your comments: please give specific page and line references when appropriate. Additional comments may be made on the other side of this form.

If you would like a reply, complete the following (Please Print):

Your Name ______
Company Name ______
Department ______
Street Address ______
City ______ State ______ Zip Code ______

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along This Line

GC30-3008-5

ADDITIONAL COMMENTS:

Cut or Fold Along Line

.... Fold .... Fold ....

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Dept. E01
P.O. Box 12195
Research Triangle Park
North Carolina 27709

.... Fold .... Fold ....

IBM®

3704/3705 Control Program Generation and Utilities Guide and Reference Manual (File No. S36-/S370) Printed in U.S.A. GC30-3008-5

IBM 3704 and 3705
Communications Controllers
Network Control Program/VS
Generation and Utilities
Guide and Reference Manual
(for OS/VS and DOS/VS VTAM Users)

READER'S COMMENT FORM

Your comments, accompanied by answers to the following questions, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. Comments and suggestions become property of IBM.

Please direct *any requests for copies of publications or for assistance in using your IBM System or equipment,* to your IBM representative or to the IBM sales office serving your locality.

- Did this publication meet your needs?
- How did you use this publication?
- What is your occupation?

COMMENTS

We would appreciate your comments: please give specific page and line references when appropriate. Additional comments may be made on the other side of this form.

If you would like a reply, complete the following (Please Print):

Your Name ____________________
Company Name ____________________
Department ____________________
Street Address ____________________
City ____________ State ____________ Zip Code ____________

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along This Line

GC30-3008-5

ADDITIONAL COMMENTS:

Cut or Fold Along Line

Fold .... Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Dept. E01
P.O. Box 12195
Research Triangle Park
North Carolina 27709

Fold .... Fold

3704/3705 Control Program Generation and Utilities Guide and Reference Manual (File No. S36-/S370) Printed in U.S.A. GC30-3008-5

IBM 3704 and 3705
Communications Controllers
Network Control Program/VS
Generation and Utilities
Guide and Reference Manual
(for OS/VS and DOS/VS VTAM Users)

READER'S COMMENT FORM

Your comments, accompanied by answers to the following questions, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. Comments and suggestions become property of IBM.

Please direct *any requests for copies of publications or for assistance in using your IBM System or equipment,* to your IBM representative or to the IBM sales office serving your locality.

- Did this publication meet your needs?
- How did you use this publication?
- What is your occupation?

COMMENTS

We would appreciate your comments: please give specific page and line references when appropriate. Additional comments may be made on the other side of this form.

If you would like a reply, complete the following (Please Print):

Your Name ____________________
Company Name ____________________
Department ____________________
Street Address ____________________
City __________ State __________ Zip Code __________

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along This Line

GC30-3008-5

ADDITIONAL COMMENTS:

Cut or Fold Along Line

3704/3705 Control Program Generation and Utilities Guide and Reference Manual (File No. S36-/S370) Printed in U.S.A. GC30-3008-5

Fold

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Dept. E01
P.O. Box 12195
Research Triangle Park
North Carolina 27709

Fold

Fold

Macro Instruction Index

The table at right is an index to the applicability of program generation macro instructions for various combinations of types of stations (SDLC and non-SDLC) and operating modes (network control and emulation).

The seven numbered columns of this table correspond to the seven columns of the *To Define a Program* chart in *How to Use This Book* following the Preface.

The symbols used in the chart at right are as follows:

R	Macro is always required in the program
R	Macro is required if network includes type of station shown at left
O	Macro is optional (that is, needed only under certain conditions)
O	Macro is optional (needed only under certain conditions) if network includes type of station shown at left

Qualifying symbols and notes appear at the bottom of the table at right.

Page numbers on which the macro instruction descriptions begin appear at the right under columns headed *Page.*

				1		2	3	4	5	6		7		
TYPE OF STATION:														
SDLC only				▨										
SDLC & BSC or SDLC & SS or SDLC, BSC & SS														
BSC and/or SS only									▨	▨		▨		
OPERATING MODE:														Type of Generation (BUILD macro, TYPGEN=)
Network Control only				▨		▨			▨					NCP
				▨		▨								NCP-LR
				▨		▨								NCP-R
Network Control and Emulation							▨	▨		▨				PEP
							▨	▨						PEP-LR
Emulation only												▨		EP
Station types for which macro is valid														
Macro Instruction	SDLC	BSC	SS		*Page*		*Note 1*	*Note 2*			*Page*		*Page*	
System Definition Macro Instructions					3-						5-		7-	
BUILD	●	●	●	R	3	R	R	R	R	R	3	R	1	
SYSCNTRL**	●	●	●	R	17	R	R	R	R	R	27			
Configuration Definition Macro Instructions														
HOST	●	●	●	R	19	R	R	R	R	R	29			
CSB	●	●	●	R	23	R	R	R	R	R	33	R	11	
IDLIST		●	●			O	O		O	O	36			
SERVICE	●	●	●	O	25	O	O	O	O	O	40			
LUPOOL	●			O	27	O	O	O			42			
MTALCST			●			O	O		O	O	44			
MTALIST			●			O	O		O	O	51			
MTAPOLL			●			O	O		O	O	52			
MTATABL			●			O	O		O	O	53			
DIALSET		●	●			O	O		O	O	56			
Network Configuration Macro Instructions														
GROUP	●	●	●	R	37	R	R	R	R	R	72	R	14	
LINE	●	●	●	R	42	R	R	R	R	R	86	R	19	
PU	●			R†	58	R†	R†	R†			131			
CLUSTER (BSC)		●				R	R		R	R	122			
CLUSTER (SDLC)	●			O*	55	O*	O*	O*			122			
LU	●			O	65	O	O	O			138			
TERMINAL		●	●			O	O		R	R	140			
COMP			●			O	O		O	O	161			
INNODE	●			O*	67	O*	O*	O*			164			
Block Handler Definition Macro Instructions														
STARTBH		●	●			O	O		O	O	169			
ENDBH		●	●			O	O		O	O	171			
DATETIME		●	●			O	O		O	O	172			
EDIT		●	●			O	O		O	O	174			
UBHR		●	●			O	O		O	O	176			
BHSET		●	●			O	O		O	O	178			
Generation Delimiter Macro Instruction														
GENEND	●	●	●	R	72	R	R	R	R	R	179	R	32	
For descriptions of program functions for . . .														
SDLC stations—see Chapter:				2		2	2	2						
BSC, start-stop stations—see Chapter:						4	4	6	4	4		6		
For macro descriptions—see Chapter(s):				3		5	5 7††	5 7	5	5 7††		7		

R = required

O = optional

*For compatibility with existing network control programs; use PU macro when coding a new network control program or when adding more SDLC stations to an existing network control program. **Not required, and invalid, for emulation program.

†Required unless, in an existing network control program, SDLC stations are represented by CLUSTER and/or INNODE macros.

††If you intend to define one or more BSC and/or start-stop lines or line groups exclusively in emulation mode, you may use the GROUP and LINE macro descriptions in Chapter 7 for such lines and groups. The macro descriptions in Chapter 7 include only the operands applicable to emulation mode.

Note 1: SS and/or BSC stations in both network control and emulation modes.

Note 2: SS and/or BSC stations in emulation mode only.

GC30-3008-5

3704/3705 Control Program Generation and Utilities Guide and Reference Manual (File No. S360/S370) Printed in U.S.A. GC30-3008-5