LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii	Original
iii	Original
iv	Blank
v thru ix	Original
x	Blank
1-1 thru 1-3	Original
1-4	Blank
2-1 thru 2-6	Original
3-1 thru 3-88	Original
4-1 thru 4-5	Original
4-6	Blank
5-1 thru 5-3	Original
5-4	Blank
6-1 thru 6-12	Original
A-1 thru A-7	Original
A-8	Blank
B-1 thru B-6	Original
C-1 thru C-8	Original
D-1 thru D-3	Original
D-4	Blank
1 thru 4	Original

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

SECTION 1
INTRODUCTION

General

This document describes the B 1000 Network Controller installation and the operation of the SYSTEM/CONFIGURE program which creates line configurations for the B 1965/B 1995 systems.

Audience Level

This document is intended for the operations manager.

Manual Structure

A brief description of each section and appendix follows:

Section 1: Introduction
Describes the purpose and structure of the manual, and provides a list of related documentation.

Section 2: Network Controller Overview
Provides a general description of the network controller mechanisms and is intended for all audiences.

Section 3: Network Controller Symbolic Specifications
Describes the basic symbolic specifications required to generate a network controller. An example of a network controller symbolic file is included.

Section 4: Network Controller Generation
Describes the generation of the network controller using the symbolic specifications described in section 3.

Section 5: Network Controller Installation
Describes the network controller installation and the program switches.

Section 6: SYSTEM/CONFIGURE
Describes the use of the SYSTEM/CONFIGURE program.

Appendix A: Glossary of Terms
Provides a glossary of the terms and acronyms used in this document.

Appendix B: Data Communications Control Characters
Describes the various data communications control characters and provides a brief explanation of their usage.

Appendix C: Character Sets
Describes the EBCDIC and ASCII character sets.

Appendix D: Syntax and Notation Conventions
Describes the syntax and notation conventions used in in this document.

Related Documents

The following documents are referenced in this document:

B 1000 Software Manuals

B 1000 Systems System Software Operation Guide, Volume 1, form number 1151982.

B 1000 Systems System Software Operation Guide, Volume 2, form number 1152097.

B 1000 Systems Network Definition Language (NDL) Language Manual, form number 1152014.

B 1000 Systems NDL/LIBRARY Functional Description Manual, form number 1152246.

B 1000 Systems Data Communications Functional Description Manual, form number 1152030.

B 1000 Systems Message Control System (MCS) Functional Description Manual, form number 1152253.

B 1000 Systems Command AND Edit (CANDE) Installation and Operation Manual, form number 1152006.

B 1000 Systems EM3270/BSC Installation, Operation and Functional Description Manual, form number 1152337.

B 1000 Systems EM3270/SNA Installation, Operation and Functional Description Manual, form number 1152345.

B 1000 Systems HASP Installation, Operation and Functional Description Manual, form number 1152303.

B 1000 Systems RJE Host Installation, Operation and Functional Description Manual, form number 1152329.

B 1000 Systems RJE Terminal Installation and Operation Manual, form number 1152311.

B 1000 Systems RJE3780 Installation, Operation and Functional Description Manual, form number 1152295.

B 1000 Systems SMCS Installation, Operation and Functional Description Manual, form number 1152279.

B 1000 Systems SYCOM Installation, Operation and Functional Description Manual, form number 1152287.

B 1000 Systems Burroughs Network Architecture (BNA) Installation and Operation Manual, form number 1151974.

B 1000 Systems Job Spawning Functional Description Manual, form number 1152261.

B 1000 Systems Interactive SDL2 System Analyzer (ISSA), form number 1152055.

B 1000 Systems System Software Capabilities and Features Manual, form number 1138229.

B 1000 Systems System Development Language (SDL) and User Programming Language (UPL) Language Manual, form number 1137833.

SECTION 2
NETWORK CONTROLLER OVERVIEW

General

This section provides a description of the B 1000 network controller. The network controller is generated from the network attributes defined in the network controller symbolic. The network controller symbolic describes the physical devices in the network, the line disciplines to be used, the order and priority of line use, and the grouping of stations into files. A B 1000 network controller is generated by compiling the symbolic network attributes with the Network Definition Language (NDL) compiler.

If the physical network is reconfigured, the user can easily change the attributes that describe the network and can regenerate the network controller by recompiling with the NDL compiler.

The network controller can accommodate one or more optional message control systems. A Message Control System (MCS) can be a Burroughs-written MCS, such as GEMCOS or SMCS, or a user-written MCS. A Message Control System is used when certain functions and system decisions need to be controlled.

It is assumed that the reader has a working knowledge of the basic hardware and software functions of the B 1000 Data Processing System.

Network Controller Concepts

The network controller is the heart of the data communications system, and its function is to process and supervise the flow of messages between application programs and remote network stations. Through the NDL-generated network controller, the application program can handle communications devices in the same manner as more conventional peripheral devices (card readers, printers, disk, or tape).

It is the function of the network controller to

1. Provide the data communications line disciplines for all stations.
2. Construct all of the input or output descriptors B 1000 System.
3. Initiate I/O operations and service I/O completions.
4. Provide any optional Message Control System (MCS) with appropriate information for it to perform its functions.

Message Flow

When a message is received from a station, the network controller queues it for the remote file of the program to which the station is attached. The number of possible queued messages for the remote file of the program is determined by the file attribute Q.MAX.MESSAGES for the remote file. When the remote file queue is full, the network controller cannot deliver any more messages until the receiving program executes a read operation to take a message out of the queue. When the program executes a read operation on its remote file, the program receives the messages in the order in which they were queued to the remote file by the network controller.

When a program executes a remote file write operation to send a message to a station, the MCP queues the message to the appropriate entry in the station queue family (output part of a remote file). There is an entry in the station queue family for every station declared in the network. The number of messages that can be queued for any one station is determined by the attribute MAX MESSAGES declared

in the network controller symbolic code. The number of those queued messages kept in memory is determined by the attribute MAX BUFFERS declared in the network controller symbolic code. Once a message has been queued for a station, the network controller initiates the appropriate procedures to transmit the message to the station.

Figure 2-1 illustrates in block form the basic message flow in the data communications system.

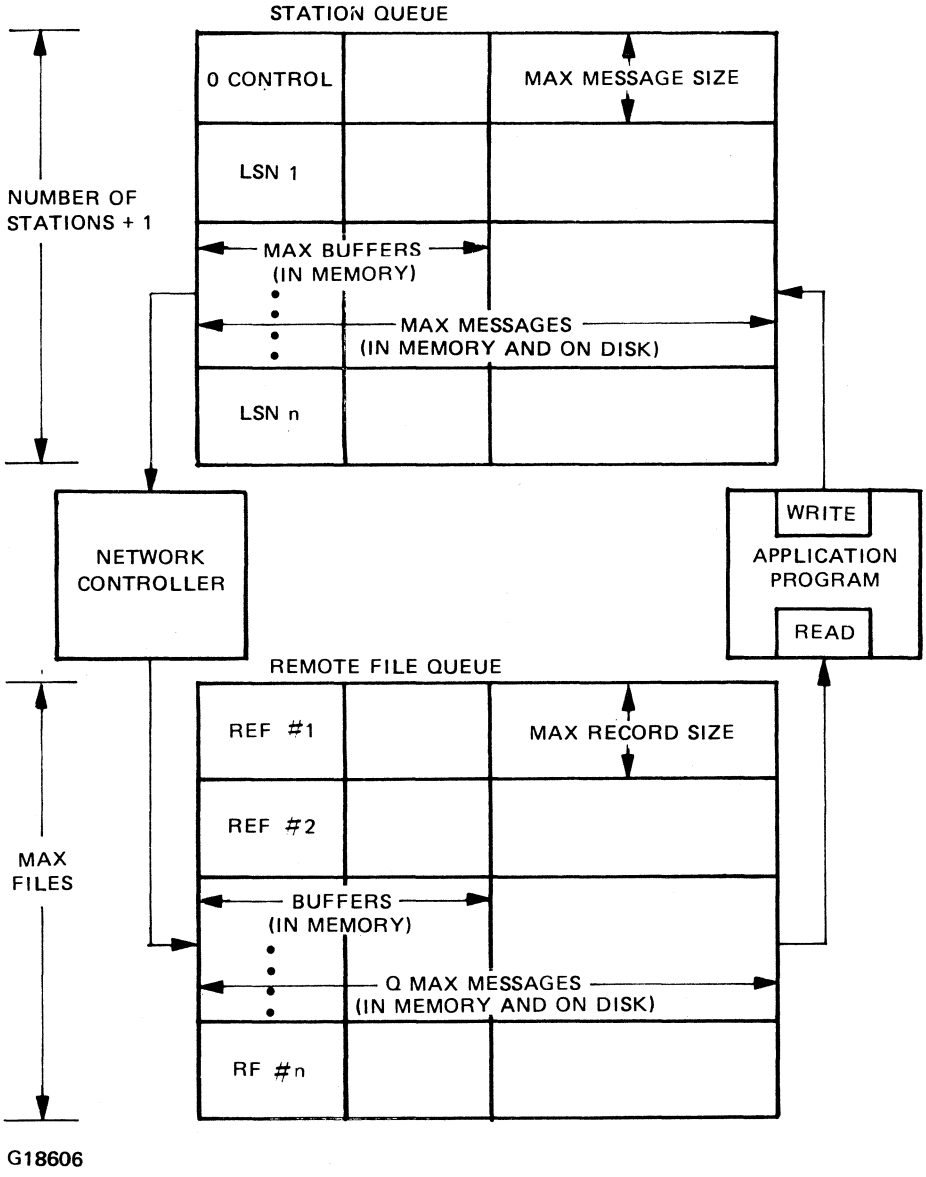


Figure 2-1. Basic Message Flow

Network Controller Symbolic Code Organization

Every network controller symbolic or source code contains the seven sections listed below in the order given:

Declaration (optional)
Request (optional)
Line Control (optional)
Terminal
Station
Line
File

A brief description of each section with its functions follows.

The Declaration Section provides a means of specifying global definitions for the network controller. The Declaration Section is optional if the standard global definitions in the SYSTEM/CONTROLLER standard network controller are used.

The Request Section defines the line discipline routines (Request procedures) used by the various remote devices in the network. The Request Section is optional if the standard Request procedures are used.

The function of the line Control Section (or Control Section) is to drive the line and allocate the line to various tasks. The line Control Section is optional if the standard line Control procedures are used.

The Terminal Section is used to describe some of the physical hardware characteristics of the various types of remote devices in the network.

The Station Section specifies the attributes for every remote device in the network.

The Line Section lists the attributes for each line in the network.

The File Section groups stations into logical remote files.

When the network controller symbolic code is compiled, any user-written Request and line Control procedures are bound with the standard network controller (SYSTEM/CONTROLLER) to produce the specific network controller. The standard Request and line Control procedures are already contained within the standard network controller. The system-supplied standard network controller (SYSTEM/CONTROLLER) provides for I/O initiation, I/O completion, entrance into line Control and Request line disciplines (the Request and Control procedures are incorporated as subroutines), communications with the application programs and Message Control Systems, and the general supervision of all lines and stations.

The descriptive sections of the network controller symbolic code are used to provide information for the Line, Station, Terminal, and File tables of the network controller. The initialization values for the Line, Station, Terminal, and File tables are contained in the network controller code file.

Figure 2-2 shows the network controller generation process.

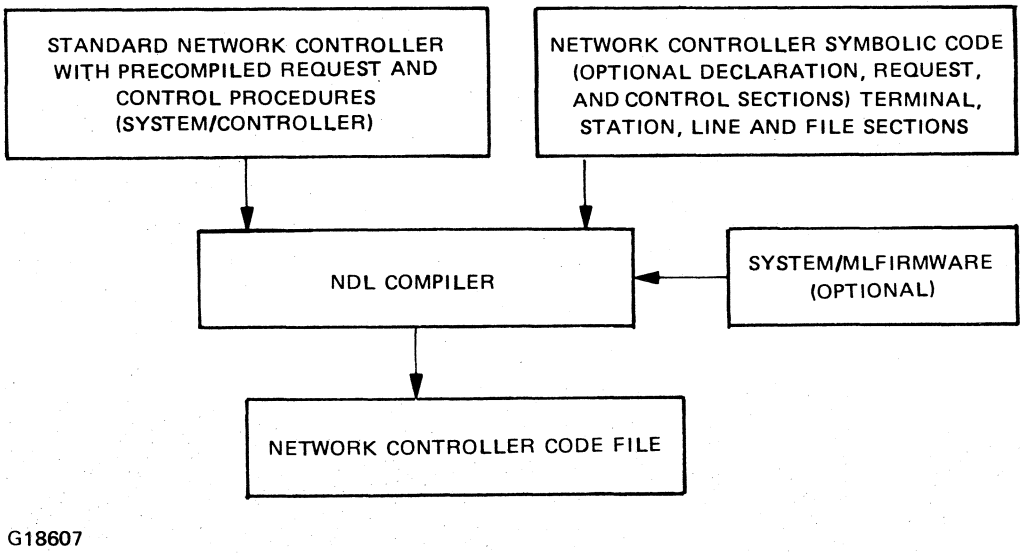


Figure 2-2. Network Controller Generation Process

Network Controller Resource Requirements

The hardware, software, and memory requirements for an NDL-generated network controller are described in the following paragraphs.

Hardware Requirements

The following hardware is required to run an NDL-generated network controller.

- B 1800/B 1900 Central System (except B 1830).
- Multiline or Single-Line Control.
- Line Adapters (as required).
- Cabling (as required).
- Terminals (as required).

Software Requirements

The following software is required to generate a network controller.

- Master Control Program II (MCPII).
- Network Definition Language Compiler (NDL).
- Standard Network Controller (SYSTEM/CONTROLLER).
- SYSTEM/CONFIGURE program (for B 1990 systems).

The following software is used to debug the network controller.

- DC/AUDIT program.
- ISSA program.

The following software requires an NDL-generated network controller in order to run.

- Command AND Edit MCS (CANDE).
- DMS/INQUIRY program.
- GEMCOS MCS.
- IBASIC program.
- Remote Display program (RD).
- REMOTE/PRINT program.
- RJE3780 program.
- RJE/CONTROLLER program.
- Supervisory Message Control System MCS (SMCS).
- System Communication Module program (SYCOM).
- EM3270/BSC program.
- EM3270/SNA program.

Memory Requirements

The following are the memory requirements for an NDL-generated network controller.

1. The following are the memory requirements for the SYSTEM/CONTROLLER standard network controller. The SYSTEM/CONTROLLER standard network controller contains the precompiled standard Request and Control procedures, two terminal declarations, two station declarations, two line declarations, and two file declarations.

Basic	8822 bytes
Line buffers	4502 bytes
General buffer	2306 bytes
Autopoll buffer	20 bytes
Tables	1000 bytes
Total	15650 bytes

2. For each additional terminal declaration, add 212 bits.
3. For each additional station declaration, add 665 bits.
4. For each additional line declaration, add 2669 bits plus the line buffer size.
5. For each additional file declaration, add 206 bits.

The values in items 1 through 5 are subject to change with each release of the B 1000 system software.

Statistics

The network controller possesses the capability to create a file containing statistics information pertaining to every message that passes through the network controller. Both the data and control messages are logged in the statistics file. Data messages are logged when the message is written to the remote file queue and when the network controller has delivered the message to the station. Control messages are logged when the message is read by the network controller and when a reply, if any, is written.

The statistics gathering function is initiated by one of the following:

1. The inclusion of the STATISTICS = TRUE declaration in the network controller symbolic file.
2. Setting program switch 3 at network controller beginning of job.
3. The explicit network controller command, STATISTICS ON.

When the statistics function is initiated, the network controller opens a file with the name STATISTICS/<yymmddhhmm>, where yymmdd is the year, month, and day and hhmm are the hours and minutes that the file was created.

The statistics file contains 900 byte records with 30 statistic entries per record. The reason for this blocking is to avoid performing a logical write to disk for every message, keeping the network controller overhead to a minimum.

The format of the statistics file is as follows:

```
RECORD STATISTICS_ENTRY_LAYOUT
    MESSAGE_TYPE                        CHARACTER (2),
    LSN                                 BIT (16),
    REMOTE_FILE_NO                      BIT (08),
    JOB_NO                              BIT (16),
    LINE_NO                             BIT (08),
    MESSAGE_LENGTH                      BIT (16),
    TIME_STAMP                          BIT (20),
    CPU_TIME_STAMP                      BIT (20),
    FIRST_15_TEXT_CHARACTERS            CHARACTER (15);
RECORD STATISTCS_FILE_RECORD_LAYOUT
    ENTRY (30)                          STATISTICS_ENTRY_LAYOUT;
```

If the last record is incompletely filled, the empty entries are set to zero.

The BIT fields contain binary information. The CHARACTER fields contain EBCDIC character information. The TIME_STAMP and CPU_TIME_STAMP are in tenths of a second. The MESSAGE_LENGTH is in units of characters.

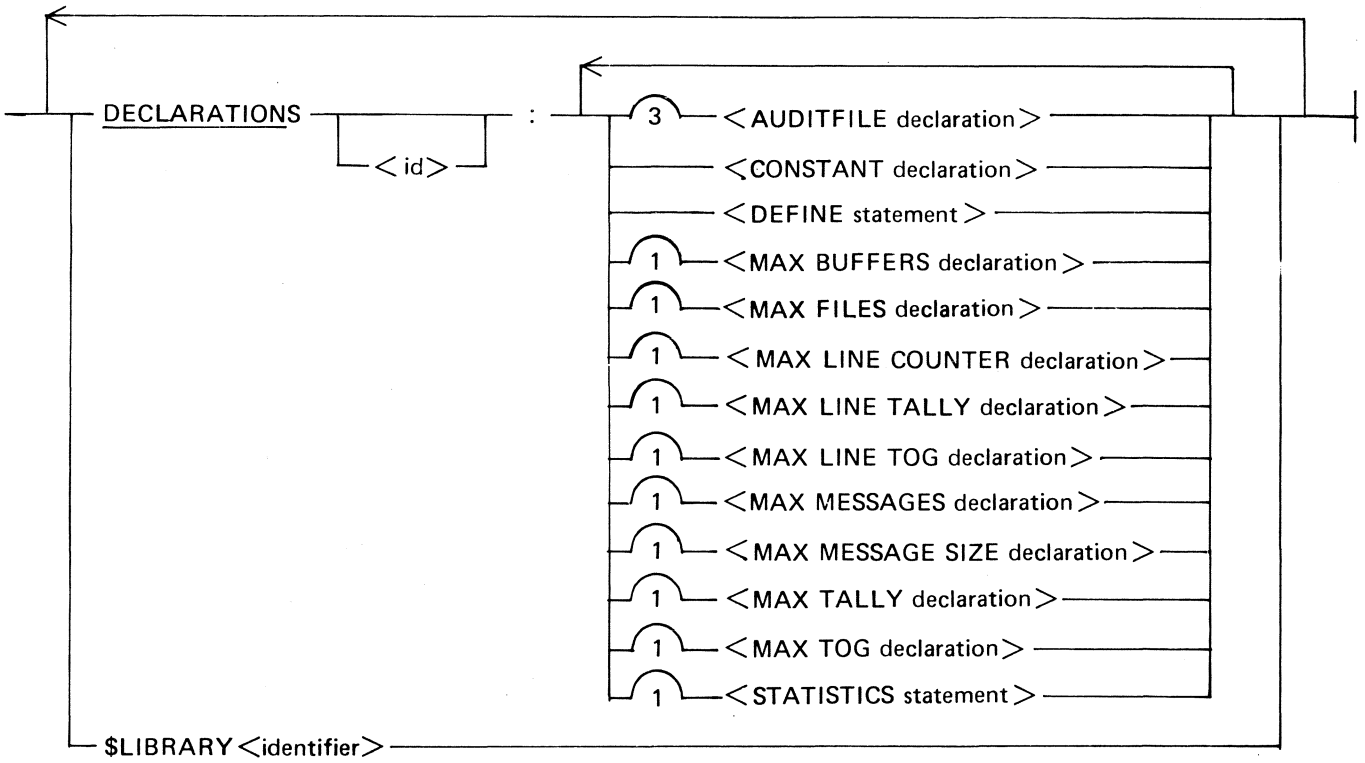
Some of the information is stored in binary and is not converted to COBOL decimal format for two reasons. First, such a conversion requires processor overhead and degrades the data communications throughput. Second, the reformatted data has greater disk and memory requirements.

Declaration Section

The Declaration Section is optional and provides a means of specifying global definitions for the network controller. All Declaration statements must appear before the first Request procedure.

Some Request and Control procedures in the NDL/LIBRARY file require the use of a Declaration statement file from the NDL/LIBRARY file. These are included in the symbolic file through the use of the $LIBRARY <symbolic name> statement. Refer to the B 1000 Systems NDL/LIBRARY Functional Description Manual for a complete description of the NDL/LIBRARY file.

Syntax:



Semantics:

<id>
<id> is a name optionally assigned to a Declaration statement.

<identifier>
<identifier> is the name of a standard declaration in the NDL/LIBRARY file.

The following standard declarations for the precompiled Request and Control procedures are included in the SYSTEM/CONTROLLER standard network controller.

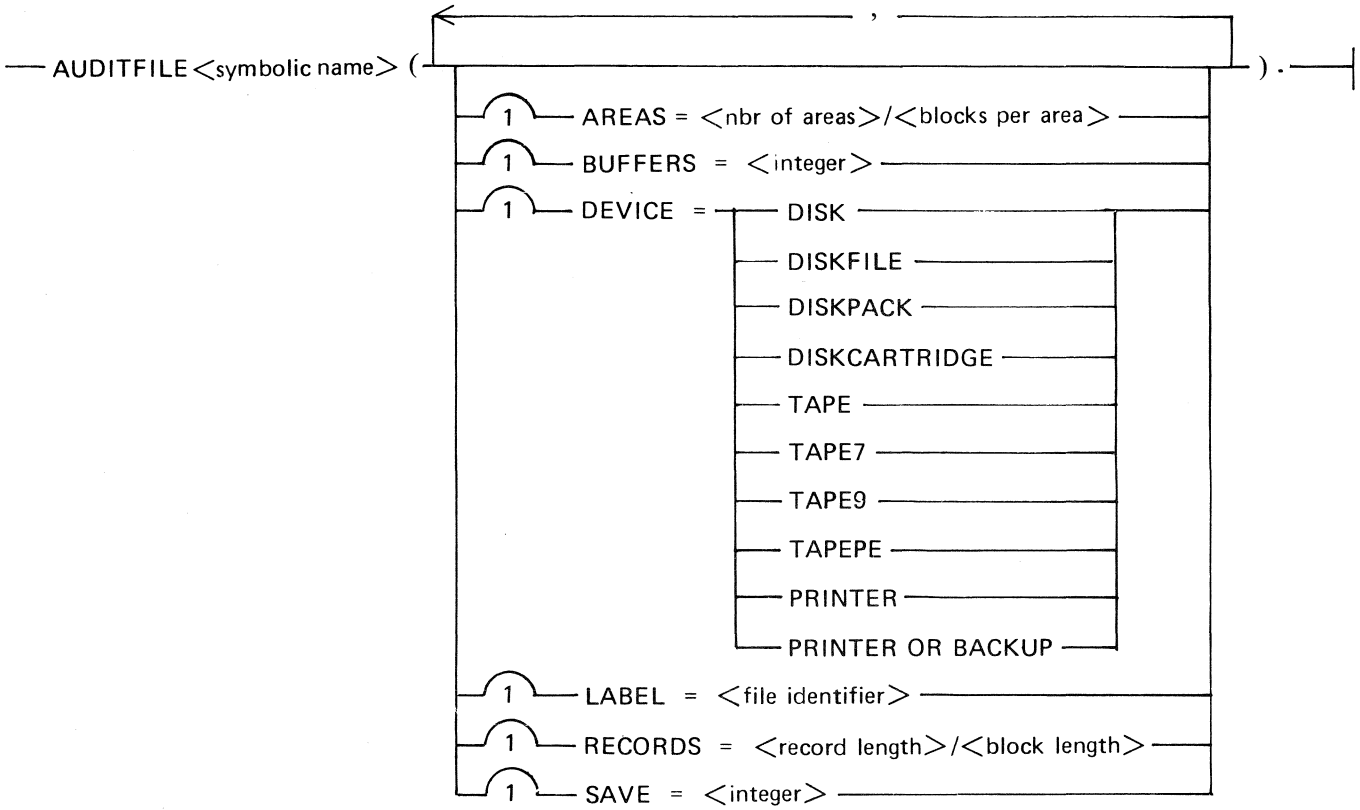
CANDEDEC
SYCOMDEC
RJE3780DCLS
BISYNC3270DECLARATION

The Declaration statements that can be included in the Declaration Section are described in the following paragraphs.

AUDITFILE Declaration

Audit files are used in the AUDIT statement in a Request or Control procedure and any audit file can be referenced in any Request or Control procedure. A maximum of three audit files can be declared. Audit files are sequential disk files that can be opened for output only by the network controller.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the audit file.

AREAS
<nbr of areas> specifies the maximum number of areas on disk that can be allocated for the audit file. <blocks per area> is the maximum number of blocks in each disk area.

BUFFERS
<integer> specifies the number of buffers for the audit file.

DEVICE
The default values are DEVICE = DISK, LABEL = <file identifier>, BUFFERS = 1, and SAVE = 30. If DEVICE = DISK is specified, RECORDS = 180 (unblocked) and AREAS = 40 (100 records per area) is used. If DEVICE = TAPE is specified, RECORDS = 180 (unblocked) is used. If DEVICE = PRINTER is specified, RECORDS = 132 is used and the file is written to backup disk or tape DISKFILE refers to head-per-track disk, TAPE7 refers to 7-track tape, TAPE9 refers to 9-track tape, and TAPEPE refers to phase-encoded tape.

LABEL

<file identifier> is the external file name assigned to the audit file, such as TESTPACK/(DATACOMM)/AUDITFILE1.

RECORDS

<record length> specifies the length of each record in bytes <block length> specifies the length of each block in bytes and is an even multiple of <record length>.

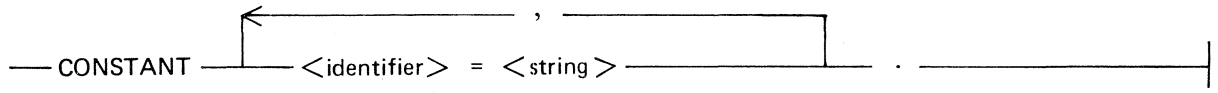
SAVE

<integer> is the number of days the file is saved on disk before it is marked as an expired file.

CONSTANT Declaration

The CONSTANT declaration provides the user with a way to equate identifiers with strings of characters and control characters for which no EBCDIC graphic exists.

Syntax:



Twenty-eight common control character identifiers have been defined within the NDL compiler to equal the strings shown in table 3-1. The identifiers can be redefined to equal other strings but cannot be used as non-constant identifiers.

Table 3-1. NDL Compiler-Defined Constants

Identifier	String	Identifier	String
ACK	4"2E"	FSL	4"A2"
ACK0	4"70"	GS	4"1D"
ACK1	4"61"	HT	4"05"
BEL	4"2F"	LF	4"25"
CAN	4"18"	NAK	4"3D"
CR	4"0D"	NUL	4"00"
DC1	4"11"	POL	4"97"
DC2	4"12"	RS	4"1E"
DC3	4"13"	RVI	4"7C"
DC4	4"3C"	SI	4"0F"
DEL	4"07"	SEL	4"98"
DLE	4"10"	SO	4"0E"
ENQ	4"2D"	SOH	4"01"
EOT	4"37"	STX	4"02"
ESC	4"27"	SYN	4"32"
ETB	4"26"	US	4"1F"
ETX	4"03"	VT	4"0B"
FF	4"0C"	WACK	4"6B"
FS	4"1C"		

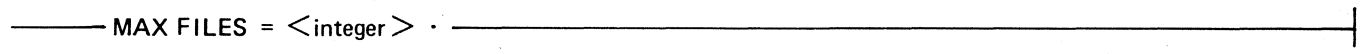
Examples:

```
CONSTANT    BAD1 = "ERROR",
            RETRANSMIT = "PLEASE SEND AGAIN".
CONSTANT    QUOTE = """".
CONSTANT    CRLF = 4"OD250000".
```

MAX FILES Declaration

The MAX FILES declaration specifies the number of remote files in the network controller remote file table that can remain in memory at one time.

Syntax:



G18196

Semantics:

<integer>
The default value for <integer> is the number of files declared in the File Section of the network controller symbolic file. A minimum of 5 is used for <integer> if the value of MAX FILES is less than 5.

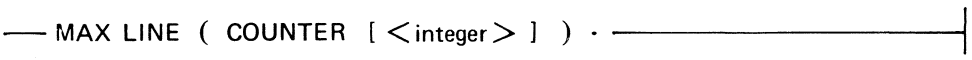
Example:

```
MAX FILES = 30.
```

MAX LINE COUNTER Declaration

The MAX LINE COUNTER declaration specifies the maximum number of line counters that are available for use in the Request and Control procedures of the network controller being generated. The standard SYSTEM/CONTROLLER network controller contains four line counters per line.

Syntax:



Semantics:

<integer>
The value of <integer> can be in the range from 0 to 3 inclusive; for any other values, a syntax error is generated.

Pragmatics:

The standard CANDE line Control procedure, CANDETDCTL, requires four line counters.

Example:

```
DECLARATION:
  MAX LINE (COUNTER [3]).
```

Table 3-2. Associated Request and Control Procedures

Declaration	Input Request	Output Request	Control	Usage
CANDEDEC	CANDEPOLTD	CANDESELTD CANDEFSLTD	CANDETDCTL	Data entry CANDE, SMCS Remote printers
CANDEDEC	CANDEPOLTD	C74CANDSEL C74CANDFSL	CANDETDCTL	COBOL74 programs using SKIP/SPACE, SMCS, CANDE
SYCOMDEC	SYCOMREQ	SYCOMREQ	SYCOMCTL	SYCOM
– –	POLTCTDDYN	SELTCTDDYN	AUTODYNCTL	Line groups
– –	RJEHOST	RJEHOST	RJECTL	RJE/CONTROLLER
RJE3780DCLS	RJE3780RQST	RJE3780RQST	RJE3780CNTRL	RJE3780 or user-written MCS
– –	TCTUPTRCV	TCTUPTXMT	PTPTCONCTL	Point-to-Point applications
– –	TDPTRCV	TDPTXMT	PTPTCONCTL	
– –	– –	TDBATCHXMT	PTPTCONCTL	
– –	TCTUPTIO	TCTUPTIO	CONVERCTL	
– –	READTTY	WRITETTY	PTPTCONCTL	Teletype
– –	CANDEIOTTY	CANDEIOTTY	CONVERCTL	CANDE on teletype
BISYNC3270DECLARATION	BISYNC3270INPUT	BISYNC3270OUTPUT	BISYNC3270CONTROL	EM3270/SNA program

Table 3-3. Request Procedures and Terminals Currently Supported

Input Request	Output Request	Control	Terminals Supported
CANDEPOLTD	CANDESELTD CANDEFSLTD C74CANDSEL C74CANDFSL	CANDETDCTL	ET 1000, MT 985, MT 983, TD 830, TD 820, TD 800, TD 730, TD 700, AP 300, B 9246, AP 1301, AP 1340 ET 2000, S 3000, S 1000, TT 102, TT 602, TC 500, TC 700, TC 1700, TC 3500, TC 4000, TC 5100
POLTCTDDYN	SELTCTDDYN	AUTODYNCTL	Same as for CANDEPOLTD
SYCOMREQ	SYCOMREQ	SYCOMCTL	B 1000 as local host, B 1000, CMS system, or B 5000/B 6000/B 7000 as remote host.
RJEHOST	RJEHOST	RJECTL	B 1000 as local host.
RJE3780RQST	RJE3780RQST	RJE3780CNTRL	B 1000 as local system using IBM 3780/2780 protocol to a remote host.
TCTUPTRCV	TCTUPTXMT	PTPTCONCTL	TC and TU series
TDPTRCV	TDPTXMT	PTPTCONCTL	TD series
– –	TDBATCHXMT	PTPTCONCTL	TC series
TCTUPTIO	TCTUPTIO	CONVERCTL	TC and TU series
READTTY	WRITETTY	PTPTCONCTL	Teletype
CANDEIOTTY	CANDEIOTTY	CONVERCTL	Teletype
BISYNC3270INPUT	BISYNC3270OUTPUT	BISYNC3270CONTROL	IBM 3270 terminals with the EM3270/SNA program.

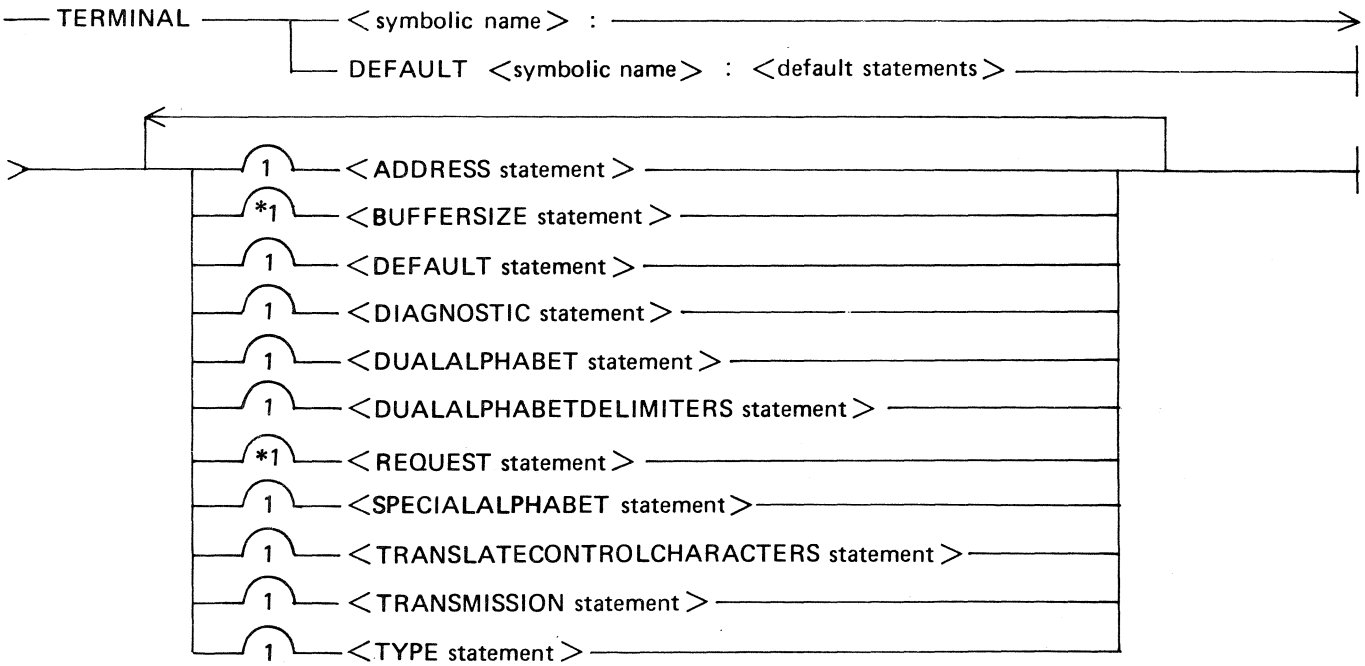
Terminal Section

The purpose of the Terminal Section is to describe each type of remote device in the data communications system. For example, if the system consists of TD 830 and MT 985 devices only, then only two terminal definitions are required in the Terminal Section, one for each type of device.

Of the options in each Terminal definition, only the REQUEST and the BUFFERSIZE statements are required. However, the NDL compiler issues a warning if the ADDRESS, TRANSMISSION, or TYPE statements are omitted.

The TERMINAL DEFAULT definition is used for terminals which have several common attributes.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the terminal definition or TERMINAL DEFAULT definition.

<default statements>
<default statements> are terminal statements used in the TERMINAL DEFAULT definition.

Predefined Terminals

A default terminal has been declared for use as the system Operator Display Terminal (ODT) and for remote diagnostics on the B 1965/B 1995 systems.

The name for the ODT terminal declaration is ODT and it is declared as follows:

```
TERMINAL ODT:
      ADDRESS          =  2.
      TRANSMISSION     =  0.
      TYPE             =  46.
      BUFFERSIZE       =  2250.
      REQUEST          =  CANDEPOLTD: RECEIVE,
                          CANDESELTD: TRANSMIT.
      DIAGNOSTIC       =  CANDEPOLTD: RECEIVE,
                          C74CANDSEL: TRANSMIT.
```

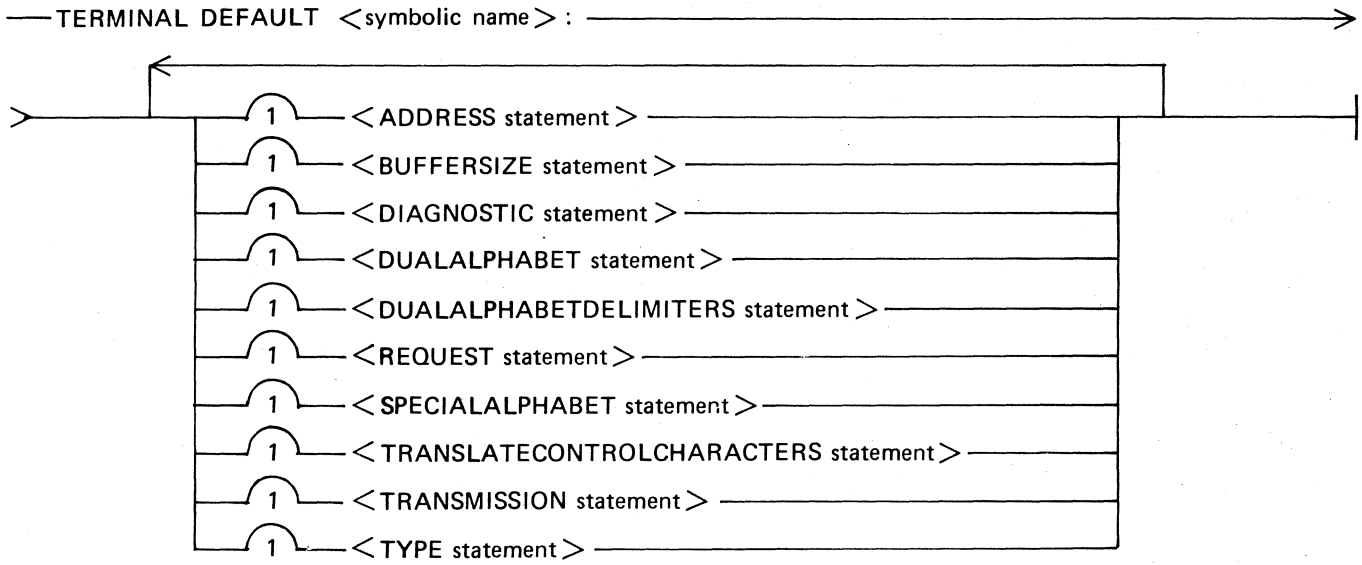
The name for the remote diagnostic terminal declaration is RDTERMINAL and it is declared as follows:

```
TERMINAL RDTERMINAL:
      ADDRESS          =  2.
      TRANSMISSION     =  0.
      TYPE             =  62.
      BUFFERSIZE       =  2250.
      REQUEST          =  CANDEPOLTD: RECEIVE,
                          CANDESELTD: TRANSMIT.
      DIAGNOSTIC       =  CANDEPOLTD: RECEIVE,
                          C74CANDSEL: TRANSMIT.
```

TERMINAL DEFAULT Definition

The TERMINAL DEFAULT definition is used for terminals that have several common attributes. The common attributes are grouped in a TERMINAL DEFAULT definition. That TERMINAL DEFAULT definition is then referenced in the Terminal definition using the DEFAULT = <symbolic name> statement. A TERMINAL DEFAULT definition must precede any reference to it in a Terminal definition.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the TERMINAL DEFAULT definition.

Pragmatics:

1. If a default attribute conflicts with an explicit attribute, the explicit attribute is given precedence.
2. A TERMINAL DEFAULT definition must precede any reference to it in a DEFAULT statement of a Terminal definition.

Example:

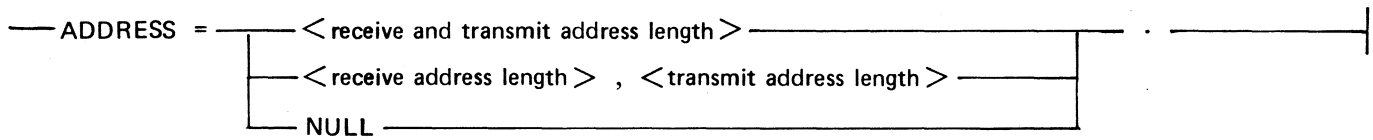
Assume that terminals TTY1, TTY2, and TTY3 have the same attributes. A DEFAULT statement could be used to simplify the coding of the descriptions of these three terminals as follows:

```
TERMINAL DEFAULT TTYDEF:
    BUFFERSIZE = 72.
    ADDRESS = 0.
    REQUEST = READTTY: RECEIVE,
              WRITETTY:TRANSMIT.
TERMINAL TTY1:
    DEFAULT = TTYDEF.
    BUFFERSIZE = 95.
TERMINAL TTY2:
    DEFAULT = TTYDEF.
    ADDRESS = 2.
TERMINAL TTY3:
    DEFAULT = TTYDEF.
```

ADDRESS Statement

The ADDRESS statement specifies the number of characters in the terminal address.

Syntax:



If the <receive address length> and the <transmit address length> of a terminal are different, they must both be specified and separated by a comma. If the ADDRESS statement is omitted, or if ADDRESS = NULL is specified, the address length is assumed to be 0. The possible values that can be specified are 0, 1, 2, or 3.

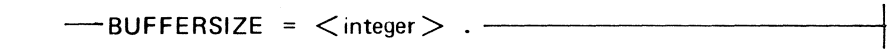
Example:

```
ADDRESS = 2.
```

BUFFERSIZE Statement

The BUFFERSIZE statement is required in the Terminal Section and defines the maximum size of a buffer in bytes. The largest buffer size that can be declared is 8191 bytes. The largest buffer size declared for stations on a line is used to determine the length of the line buffer used by the network controller to receive and transmit messages to the terminal.

Syntax:



Semantics:

<integer>
The value of <integer> specified for BUFFERSIZE must be large enough to hold the largest message transmitted to or received from the terminal and the acknowledgement of that message.

Pragmatics:

For MT985/TD830 type terminals, the BUFFERSIZE must be set to 2250 bytes; otherwise, messages can be truncated.

For the SYCOM program, the BUFFERSIZE must be at least 2 * (the value of the SYCOM BUFFER parameter + 7). For example, if the SYCOM BUFFER parameter is equal to 2500, the BUFFERSIZE statement must be at least 5014 and MAX MESSAGE SIZE must be at least 2500. For more information, refer to the B 1000 Systems SYCOM Installation, Operation and Functional Description Manual.

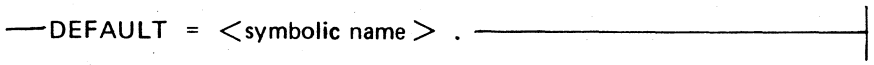
Example:

```
BUFFERSIZE = 2250.
```

DEFAULT Statement

The DEFAULT statement causes the attributes specified in the TERMINAL DEFAULT definition named by <symbolic name> to be included in the terminal declaration.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name of the TERMINAL DEFAULT definition being referenced.

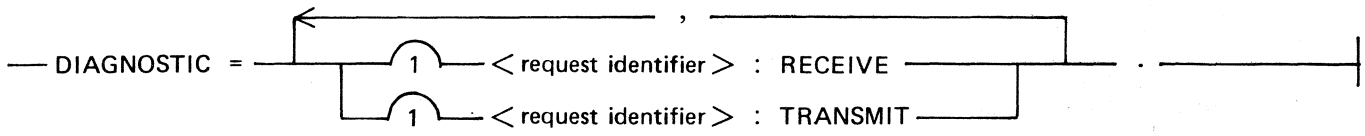
Example:

```
DEFAULT = MT985DEF.
```

DIAGNOSTIC Statement

The DIAGNOSTIC statement is used to declare an alternate Request procedure that can be invoked with a Change Request initiated by a Message Control System (MCS). This statement is optional.

Syntax:



Semantics:

<Request identifier>
<Request identifier> is the name of the Request procedure to be invoked.

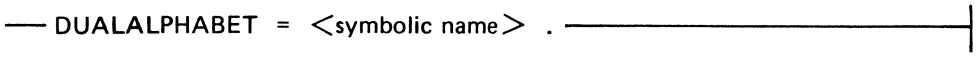
Example:

```
DIAGNOSTIC      =   CANDEPOLTD: RECEIVE,
                    C74CANDSEL: TRANSMIT.
```

DUALALPHABET Statement

The DUALALPHABET statement defines what translate table to use on messages to and from a dual alphabet terminal. There are currently two predefined translate tables, ASCII8 and V20. For the Arabic MT 985, the V20 translate table must be invoked. The ASCII8 translate table must be invoked for special MT 983 versions of the Arabic dual alphabet terminals. If the DUALALPHABET statement is invoked, the DUALALPHABETDELIMITERS statement must also be invoked.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name of the translate table.

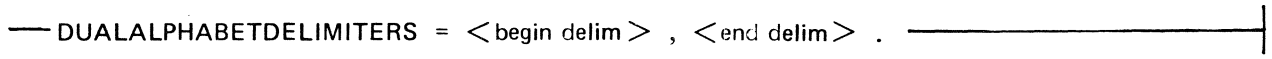
Example:

```
DUALALPHABET = V20.
```

DUALALPHABETDELIMITERS Statement

The DUALALPHABETDELIMITERS statement defines the delimiters enclosing an alternate language field in a message. Two character strings of a maximum three characters are defined in the DUALALPHABETDELIMITERS statement. The first character string is the beginning alternate language delimiter and the second is the ending delimiter. For the Arabic MT 985, the delimiters are either 4″0E″,4″0F″ or 4″270E″,4″270F″. If the DUALALPHABETDELIMITERS statement is specified, the DUALALPHABET statement must be specified.

Syntax:



Semantics:

<begin delim>
<begin delim> is the beginning alternate language delimiter.

<end delim>
<end delim> is the ending alternate language delimiter.

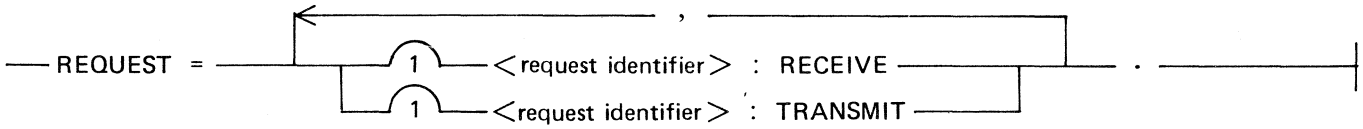
Example:

```
DUALALPHABETDELIMITERS = 4″270E″,4″270F″.
```

REQUEST Statement

The REQUEST statement is required and links the terminal with one or two line disciplines. A receive Request procedure, a transmit Request procedure, or both can be specified. The same procedure can be used for both receiving and transmitting. When input is to be received from a station, the Terminal table entry of the station is checked, and the Request procedure identified the receive Request procedure is invoked. For output, the Request procedure identified as the transmit Request procedure is invoked.

Syntax:



Semantics:

<Request identifier>
<Request identifier> is the name of the Request procedure to be used.

Pragmatics:

For Control procedures that are designed to use both an input and output Request procedure, such as CANDETDCTL, both the input and output Request procedure must be declared. Both procedures must be declared, even if the terminal device is an output only device such as a remote printer.

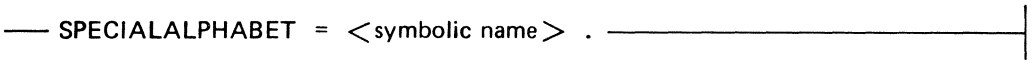
Example:

```
REQUEST     =   CANDEPOLTD: RECEIVE,
                CANDESELTD: TRANSMIT.
```

SPECIALALPHABET Statement

The SPECIALALPHABET statement specifies that messages to and from the terminal are to be processed by a special handling routine. Currently, only one special handling routine is implemented: KANJI. This statement is intended for use with the MT 687K terminal. If this routine is specified, the DUALALPHABET, DUALALPHABETDELIMITERS, and TRANSLATECONTROLCHARACTERS statements cannot be specified.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name of the special handling routine.

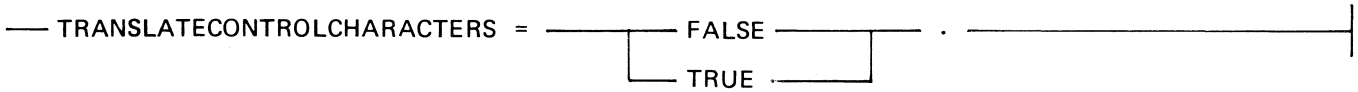
Example:

```
SPECIALALPHABET = KANJI.
```

TRANSLATECONTROLCHARACTERS Statement

The TRANSLATECONTROLCHARACTERS statement is used in conjunction with the DUALALPHABET and DUALALPHABETDELIMITERS statements.

Syntax:



Semantics:

FALSE
By default, this parameter is FALSE and the control characters maintain their representation wherever they are found.

TRUE
If TRANSLATECONTROLCHARACTERS is specified as TRUE, the TRANSLATECONTROLCHARACTERS statement causes the ASCII control characters 4"00" through 4"1F" found in alternate language fields to be translated using the translate table specified by the DUALALPHABET statement.

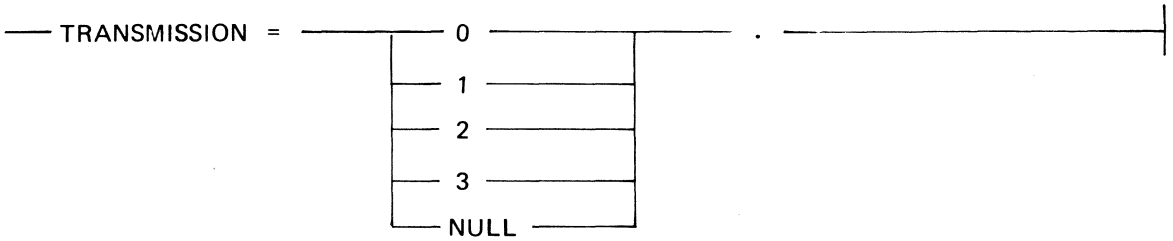
Example:

```
TRANSLATECONTROLCHARACTERS = FALSE.
```

TRANSMISSION Statement

The TRANSMISSION statement is used to set the length of the transmission number. The possible values are 0, 1, 2, 3, or NULL. The number 0 and the symbol NULL are equivalent, and indicate that no transmission number is to be used. The use of transmission numbers provides a method of checking for duplicate transmissions. When transmission number checking is done in a Request procedure, a message with the same transmission number as the previous message is considered to be a duplicate message and is ignored.

Syntax:



Pragmatics:

The standard CANDE Request procedures use alternating transmission numbers for terminal types greater than or equal to 28. All others are incremented depending on the value of TRANSMISSION.

Example:

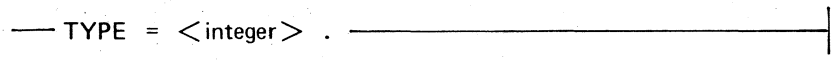
```
TRANSMISSION = 1.
```

TYPE Statement

The TYPE statement is used to specify the terminal type. If the TYPE statement is not specified, a compiler warning is printed and the TYPE is set to 0.

Use of the TYPE statement is dependent upon the Request and line Control procedures being used. Users who write their own Request and line Control procedures can define TYPE as they see fit.

Syntax:



Semantics:

<integer>
The value of <integer> is used to specify the terminal type and can be in the range from 0 to 99 inclusive.

Table 3-4 shows the values used for the TYPE statement in the standard Request and Control procedures.

Table 3-4. Terminal Types

Terminal Type Value	Terminal	Comments
00	B 9350	Teletype
01	B 9353	Bids
02	B 9352	Wide Screen
03	B 9352	Narrow Screen
04	(not used)	
05	(not used)	
06	RT 200 S 1000/S 3000	
07	TT 102	
08	TT 602	
09	(not used)	
10	(not used)	
11	(not used)	
12	(not used)	
13	DC 110	
14	DC 140	
15	TU 300	
16	TU 400	
17	TU 500	
19	TU 700	
20	TU 800	
21	TC 500	
23	TC 700	

Table 3-4. Terminal Types (Cont)

Terminal Type Value	Terminal	Comments
24	TC 1700	
25	TC 3500	
26	TC 4000	3-character transmission numbers
27	TC 5100	
28	TC 4000	Alternate transmission numbers (0/1)
29	(not used)	
30-31	(not used)	
32	TD 700	
33	TD 730A	
34	TD 730B	
35-39	(not used)	
40	(not used)	
41	TD 801	
42	TD 802	
43	TD 821	
44	TD 822	
45	TD 831	
46	TD 832/MT 983/MT 985	
47-49	(not used)	
50	MT 686/MT 687	Not qualified
51	Bisync 3270	EM3270/SNA
52-57	(not used)	
58	Real Virtual Program	SYCOM
59	Virtual Program Terminal	SYCOM internal use
60	Virtual ODT	SYCOM internal use
61	Virtual Terminal	SYCOM
62	B 1000 Series	
63-69	(not used)	
70-79	Remote Printers	(As a group)
70	B 9246 (band)	
71	AP 300	
72	AP 1301	Daisy Wheel
73	AP 1340	Serial Matrix
74-79	(reserved for future use)	
80-89	Screen Terminals	(As a group)
80	SR 100/SR 110	
81	ET 1000	
82-89	(reserved for future use)	
90-99	(reserved for future use)	

NOTE

1. Terminal types 59 and 60 are reserved for SYCOM program use only. They should not be declared in network controller symbolic files.
2. Terminal types 28-99 use alternating transmission numbers; terminal types 0-27 are incremented depending on the value of the TRANSMISSION statement.
3. Terminal types 80-89 refer to screen terminals that do not emulate the TD 830/MT 983/MT 985 terminals. If for example, an SR 110 terminal is being used as a TD 830 look-alike, then use terminal type 46.

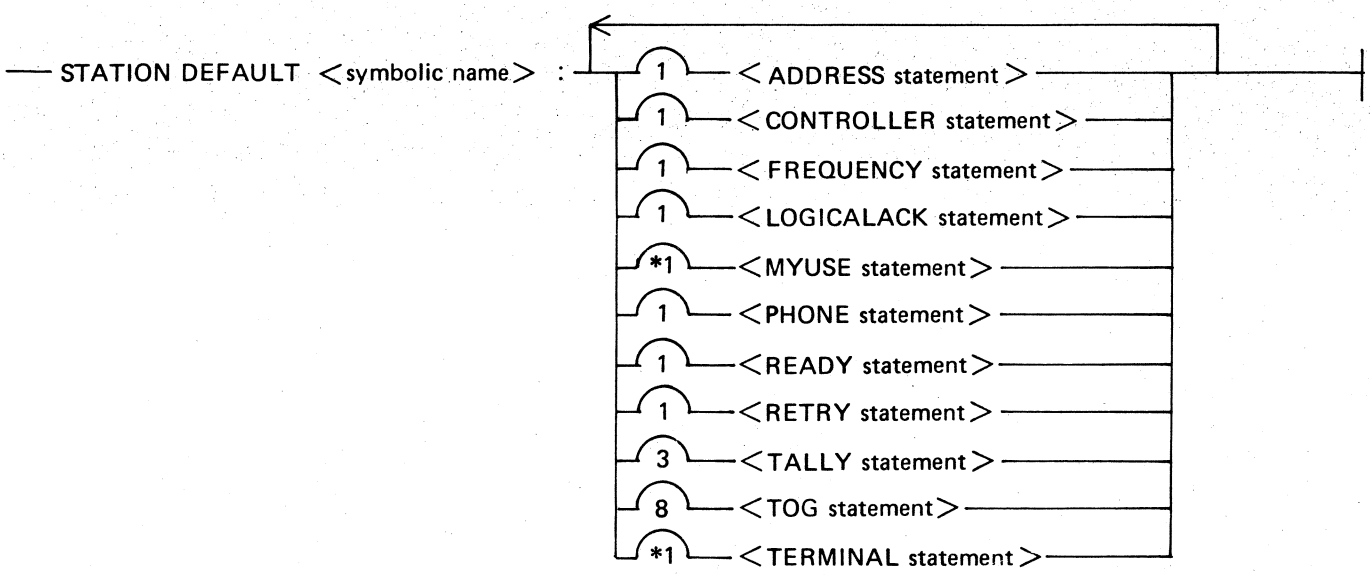
Example:

```
TYPE = 81.
```

STATION DEFAULT Definition

The STATION DEFAULT definition is used for stations that have several common attributes. The common attributes are grouped in a STATION DEFAULT definition. That STATION DEFAULT definition is then referenced in the Station definition using the DEFAULT = <symbolic name> statement.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the STATION DEFAULT definition.

Pragmatics:

1. If a default attribute conflicts with an explicit attribute, the explicit attribute is given precedence.

2. A STATION DEFAULT definition must precede any reference to it in a DEFAULT statement in a Station definition.

Example:

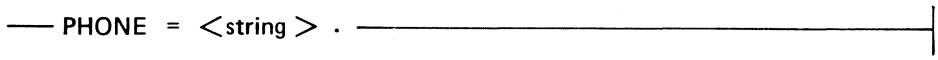
Assume that the stations SMITH, JONES, and ROOM273 have the same station attributes. A DEFAULT statement could be used to simplify the coding of the descriptions of these three stations as follows:

```
STATION DEFAULT REMOTEODT:
      RETRY = 25.
      MYUSE = INPUT,OUTPUT.
      TERMINAL = MT985.
      FREQUENCY = 250,250.

STATION SMITH:
      DEFAULT = REMOTEODT.
      ADDRESS = "R1".

STATION JONES:
      DEFAULT = REMOTEODT.
      ADDRESS = "R2".

STATION ROOM273:
      DEFAULT = REMOTEODT.
      ADDRESS = "R3".
```

PHONE Statement

The PHONE statement is optional and specifies the phone number used by the Automatic Calling Unit (ACU). If an ACU is present and a PHONE number is provided and a message is queued for the station, the dialing function is performed on the first INITIATE TRANSMIT statement in the Request procedure. An INITIATE TRANSMIT statement is executed to begin a tranmission to a station.

Syntax:



Semantics:

<string>
The <string> can consist of from 1 to 20 EBCDIC characters. All characters except 0 through 9 are changed to a time-fill character @FF@. Time-fill characters cause the dial hardware to pause in the dialing sequence.

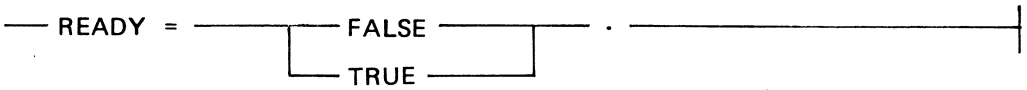
Example:

```
PHONE = "1-805-9646881".
```

READY Statement

The READY statement is used to initially mark a station as ready or not ready for input and output operations. If READY = FALSE, the Request and Control procedures do not perform input and output operations to and from this station until the station is explicitly made ready, either by a Message Control System (MCS) or by the network controller MAKE STATION READY ODT command. READY defaults to TRUE if the statement is omitted.

Syntax:



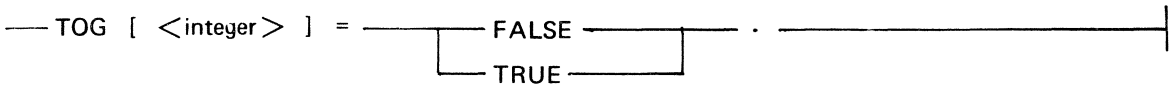
Example:

```
READY = FALSE.
```

TOG Statement

The TOG statement causes an initial value to be put into the specified station toggle at the network controller beginning of job. The default value for a TOG is false.

Syntax:



Semantics:

<integer>
<integer> is the toggle index and must have a value from 0 through 7. Only toggles 0 through 7 can be initialized by the TOG statement.

Pragmatics:

For the CANDE Request procedures, TOG[2] and TOG[4] can be set to cause the following effects:

TOG[2] = FALSE
No scrolling.

TOG[2] = TRUE
Scrolling.

TOG[4] = FALSE
Clear messages queued for output when a station retry number for timeouts occurs, that is, when a station becomes off line.

TOG[4] = TRUE
Do not clear messages queued for output when a station retry number for time-outs occurs.

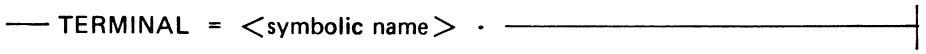
Example:

```
TOG[2] = TRUE.
```

TERMINAL Statement

The TERMINAL statement specifies the name of the Terminal definition that describes the physical characteristics of the station being described. The TERMINAL statement must be included in the Station definition. More than one station definition can reference the same Terminal definition.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the Terminal definition.

Example:

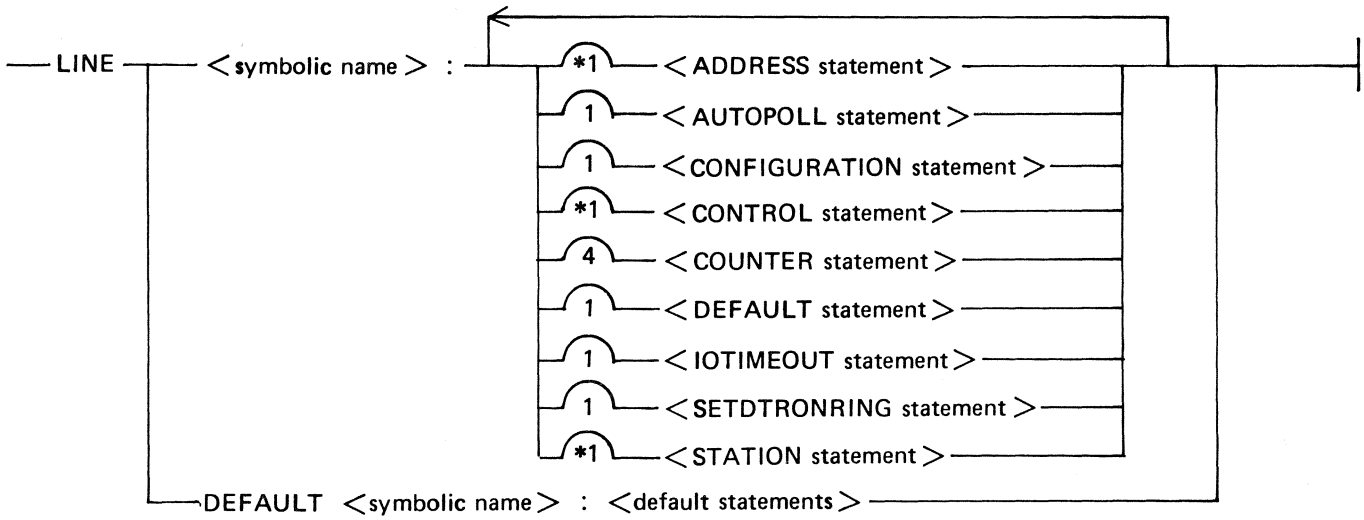
```
TERMINAL = SMITH.
```

Line Section

At least one Line definition must appear in the Line Section of every network controller symbolic file. Each line in the data communications subsystem must be described in the Line Section.

LINE DEFAULT definitions are used for lines which have several common attributes.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the line definition or to the LINE DEFAULT definition.

<default statements>
<default statements> are the statements in the LINE DEFAULT definition.

Predefined Lines

Two lines are predefined in the SYSTEM/CONTROLLER minimal network controller for use on B 1965/B 1995 systems for the Operator Display Terminal (ODT) and for remote diagnostics.

The ODT line is named ODTLINE and the default declaration is as follows:

```
LINE ODTLINE:
ADDRESS       = 1:0:0.
CONTROL       = CANDETDCTL.
CONFIGURATION = ODTDI96.
STATION       = SYSTEMODT.
```

NOTE

The line ADDRESS on a B 1965/B 1995 system is automatically changed by the network controller at beginning of job to adapter 0 of the lowest-numbered multiline control. It is mandatory that ODTs on a system with a SPO-control be located on adapter 0 of the lowest-number multiline control.

Because of the overhead with the line declarations and the existence of systems with a SPO-control, the ODT line is not declared by default.

To include the ODT line in the network controller, the following declaration is the minimum that need be specified:

```
LINE ODTLINE:
```

To include other stations on the line with the ODT, the following declaration is used:

```
LINE ODTLINE:
  STATION = <list of additional stations on the line> .
```

Example:

```
LINE ODTLINE:
     STATION = SMITH,
               JONES,
               ROOM273,
               EXT11.
```

The default setting of the line counters can be changed by specifying the COUNTER statement.

The default address of the ODTLINE can be changed by the ADDRESS statement. In this instance, the channel and adapter are set to zero (0), and line groups are not allowed. This permits the changing of the default port, but not the adapter which also must be in position zero (0).

The remote diagnostics line is named RDLINE and the default declaration is as follows:

```
LINE RDLINE:
     ADDRESS          =  1:0:3.
     CONTROL          =  CANDETDCTL.
     CONFIGURATION    =  ASYNC12SW.
     STATION          =  RDSTATION.
```

NOTE

The ADDRESS used by remote diagnostics must be 1:0:3 and cannot be changed. This is a hardware restriction.

Because of the overhead with the line declarations and the existence of systems that are not capable of supporting remote diagnostics, the remote diagnostic line is not declared by default.

To include the remote diagnostics line in the network controller, the following declaration is the minimum that need be specified:

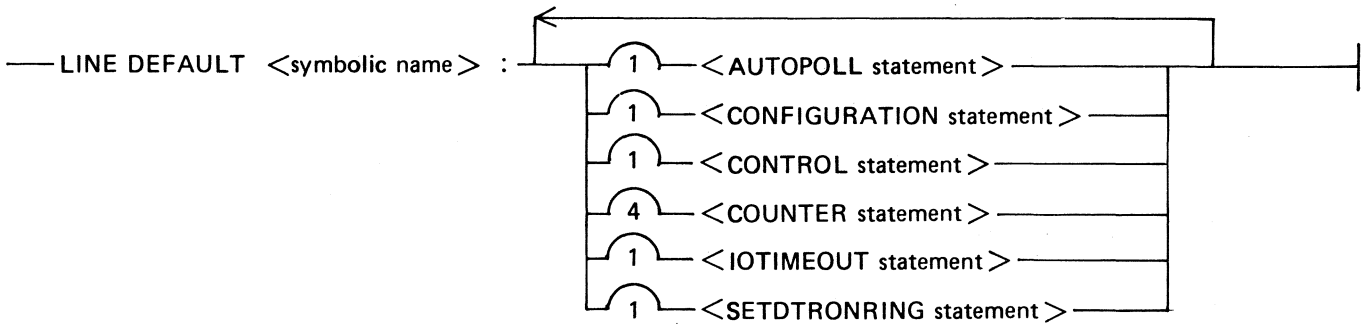
```
LINE RDLINE:
```

The default setting of the line counters can be changed by specifying the COUNTER statement, the default line configuration, ASYNC12SW, can be changed by the CONFIGURATION statement, and the SETDTRONRING statement can be specified.

LINE DEFAULT Definition

LINE DEFAULT definitions are used for lines which have several common attributes. The common attributes are grouped in a LINE DEFAULT definition. That LINE DEFAULT definition is then referenced in the Line definition using the DEFAULT = <symbolic name> statement. A LINE DEFAULT definition must precede any reference to it in a Line definition.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the LINE DEFAULT definition.

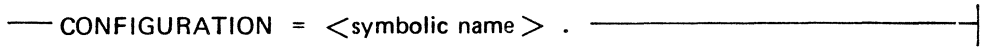
Pragmatics:

1. If a LINE DEFAULT attribute conflicts with an explicit attribute, the explicit attribute is given precedence.
2. Any reference to a LINE DEFAULT definition must be preceded by that definition, which occurs in a DEFAULT statement of a Line definition.

CONFIGURATION Statement

The CONFIGURATION statement specifies what line configuration (as defined by the SYSTEM/CONFIGURE program) is to be loaded into the MLC-4 quad adapter. The CONFIGURATION statement is required for use with the MLC-4 quad adapters so that the network controller can load firmware from them into the MLC-4 quad adapter. If a CONFIGURATION statement is omitted for a MLC-4 quad adapter, a default configuration titled DEFAULT is assumed. If a CONFIGURATION statement is specified for an adapter on a multiline control other than the MLC-4, this statement has no effect.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is one of the standard line configuration names. A list of the standard line configurations is provided in section 6, SYSTEM/CONFIGURE.

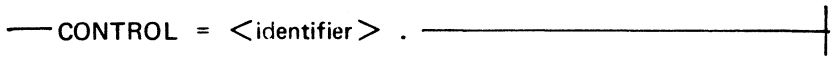
Example:

```
CONFIGURATION = TDI192.
```

CONTROL Statement

The CONTROL statement is required and specifies the line Control procedure to be used with this line, and associates a Control procedure named in the line Control Section with the line being described.

Syntax:



Semantics:

<identifier>

<identifier> is the name of the Control procedure.

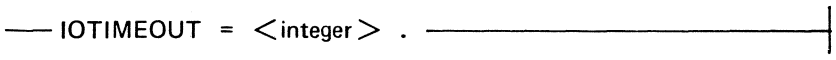
Example:

```
CONTROL = CANDETDCTL.
```

IOTIMEOUT Statement

The IOTIMEOUT statement defines the time in tenths of a second that the network controller waits for an I/O operation to complete. The IOTIMEOUT statement is only effective on I/O operations that should complete in a finite amount of time. A default value of 300 (30 seconds) is assumed if this statement is omitted.

Syntax:



Semantics:

<integer>
The values for <integer> can be in the range from 0 to 864000 inclusive.

Pragmatics:

Very low values for <integer> can cause erroneous indications of line hangs on very slow lines when large buffer sizes are used.

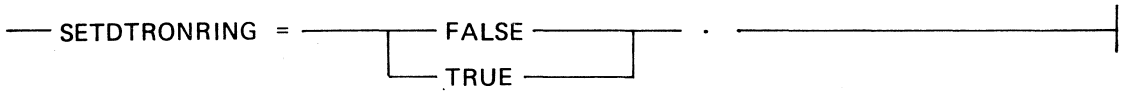
Example:

```
IOTIMEOUT = 600.
```

SETDTRONRING Statement

The SETDTRONRING statement affects the status of the data terminal ready level when a switched line is waiting for a ring condition. Some modems, such as the TA 2403, require the data terminal ready level to be FALSE for ringing to be detected. By default, the data terminal ready level is TRUE.

Syntax:



Example:

```
SETDTRONRING = FALSE.
```

Predefined Files

Two files are predefined for use with the SYSTEMT/ODT program and remote diagnostics on B 1965/B 1995 systems.

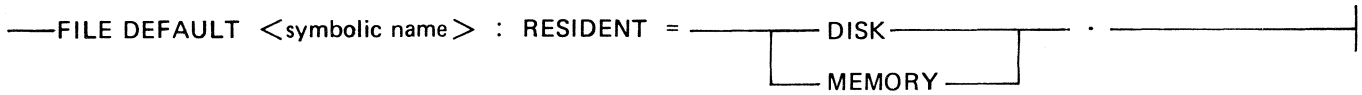
These files are defined as follows:

```
FILE OPERATOR:
    FAMILY = SYSTEMODT.
FILE RDHOST:
    FAMILY = RDSTATION.
```

FILE DEFAULT Definition

The FILE DEFAULT definition is similar to the TERMINAL DEFAULT, STATION DEFAULT, and LINE DEFAULT definitions, but the only statement allowed is the RESIDENT statement. The FILE DEFAULT definition is referenced in a File definition using the DEFAULT statement. A FILE DEFAULT definition must precede any reference to it in a File definition. The attribute in the FILE DEFAULT definition is added to the explicit attributes specified in the File definition in which this statement appears.

Syntax:



Semantics:

<symbolic name>
<symbolic name> is the name assigned to the FILE DEFAULT definition.

SYCOM Control Terminal Declaration

```
TERMINAL SYCOM:
    TRANSMISSION    =   1.
    ADDRESS         =   2.
    TYPE            =   62.
    BUFFERSIZE      =   8191.    %Maximum possible.
    REQUEST         =   SYCOMREQ: RECEIVE,
                        SYCOMREQ: TRANSMIT.
```

SYCOM Virtual Terminal Declaration

```
TERMINAL VIRTUALTRM:
    TRANSMISSION    =   1.
    ADDRESS         =   2.
    TYPE            =   61.
    BUFFERSIZE      =   4500.
    REQUEST         =   SYCOMREQ: RECEIVE,
                        SYCOMREQ: TRANSMIT.
```

Remote Printer Terminal Declaration

```
TERMINAL AP310:
    TRANSMISSION    =   1.
    ADDRESS         =   2.
    TYPE            =   71.% This implies alternating transmission numbers.
    BUFFERSIZE      =   512.
    REQUEST         =   CANDEPOLTD: RECEIVE,
                        CANDEFSLTD: TRANSMIT.
```

RJE3780 Terminal Declaration

```
TERMINAL RJE3780:
    BUFFERSIZE      =   516.
    TRANSMISSION    =   0.
    ADDRESS         =   0.
    TYPE            =   62.
    REQUEST         =   RJE3780RQST: RECEIVE,
                        RJE3780RQST: TRANSMIT.
```

EM3270/SNA Terminal Declaration

```
TERMINAL BISYNC3270:
    TRANSMISSION    =   0.
    ADDRESS         =   2.
    TYPE            =   51.
    BUFFERSIZE      =   2250.
    REQUEST         =   BISYNC3270INPUT: RECEIVE,
                        BISYNC3270OUTPUT: TRANSMIT.
```

Station Section

The stations in the network are described in the Station Section.

SMCS Station Declaration

```
STATION DEFAULT WORKSTATION:
    RETRY               =   25.
    FREQUENCY           =   0,0.
    MYUSE               =   INPUT,OUTPUT.
STATION DEFAULT REMOTEODT:
    RETRY               =   25.
    FREQUENCY           =   0,250.
    MYUSE               =   INPUT,OUTPUT.
STATION DEFAULT CTRLRSTATION:
    RETRY               =   25.
    FREQUENCY           =   250,0.
    MYUSE               =   INPUT,OUTPUT.
STATION DEFAULT CTRLRODT:
    RETRY               =   25.
    FREQUENCY           =   250,250.
    MYUSE               =   INPUT,OUTPUT.

STATION SMITH:
    TERMINAL            =   MT985.
    DEFAULT             =   REMOTEODT.
    ADDRESS             =   "S1".
STATION JONES:
    TERMINAL            =   C74MT985.
    DEFAULT             =   REMOTEODT.
    DDRESS              =   "S2".
STATION ROOM273:
    TERMINAL            =   MT985.
    DEFAULT             =   CTRLRSTATION.
    ADDRESS             =   "S3".
STATION EXT11:
    TERMINAL            =   MT985.
    DEFAULT             =   CTRLRODT.
    ADDRESS             =   "S4".
```

SYCOM Control Stations

```
STATION DEFAULT SYCOMTERM:
    RETRY              =    250.
    MYUSE              =    INPUT,OUTPUT.
    TERMINAL           =    SYCOM.
STATION SYCOMCTRL:
    DEFAULT            =    SYCOMTERM.
    ADDRESS            =    "00".
STATION SYCOMSPO:
    DEFAULT            =    SYCOMTERM.
    ADDRESS            =    "01".
STATION SYCOMCRD:
    DEFAULT            =    SYCOMTERM.
    ADDRESS            =    "02".
STATION SYCOMPTR:
    DEFAULT            =    SYCOMTERM.
    ADDRESS            =    "03".
STATION SYCOMFTP:
    DEFAULT            =    SYCOMTERM.
    ADDRESS            =    "04".
```

SYCOM Virtual Stations

```
STATION DEFAULT VIRTUALTERM:
    RETRY              =    250.
    FREQUENCY          =    250,250.
    MYUSE              =    INPUT,OUTPUT.
    TERMINAL           =    VIRTUALTRM.

STATION USER1:
    DEFAULT            =    VIRTUALTERM.
    ADDRESS            =    "AB" , "BA".
STATION USER2:
    DEFAULT            =    VIRTUALTERM.
    ADDRESS            =    "AC" , "CA".
STATION SERVER1:
    DEFAULT            =    VIRTUALTERM.
    ADDRESS            =    "BA" , "AB".
STATION SERVER2:
    DEFAULT            =    VIRTUALTERM.
    ADDRESS            =    "CA" , "AC".
```

Remote Printer Station Declaration

```
STATION DEFAULT APDEF:
   RETRY          =   10.
   FREQUENCY      =   0,0.
   MYUSE          =   INPUT,OUTPUT.
   TERMINAL       =   AP310.
 STATION OFFICE:
   TERMINAL       =   AP310.
   DEFAULT        =   APDEF.
   ADDRESS        =   "AP".
```

RJE3780 Station Declaration

```
STATION DEFAULT RJE3780DEF:
   MYUSE          =   INPUT, OUTPUT.
   RETRY          =   15.
   CONTROLLER     =   FALSE.
   TERMINAL       =   RJE3780.

STATION RJE3780ST1:
   DEFAULT        =   RJE3780DEF.
STATION RJE3780ST2:
   DEFAULT        =   RJE3780DEF.
```

EM3270/SNA Station Declaration

```
STATION DEFAULT BSC3270DEF:
   TERMINAL       = BISYNC3270.
   MYUSE          = INPUT, OUTPUT.
   RETRY          = 5.
   FREQUENCY      = 0, 250.          % SMCS Remote ODT
   TOG[0]         = TRUE.            % Invoke all the
   TOG[1]         = TRUE.            % Glass TTY' routines
   TOG[2]         = TRUE.            % initially.
   TOG[3]         = TRUE.

STATION C1D1:                         % Control Unit 1, Device 1
   ADDRESS        = "  ".
STATION C1D2:                         % Control Unit 1, Device 2
   ADDRESS        = " A".
STATION C2D1:                         % Control Unit 2, Device 1
   ADDRESS        = "A ".
STATION C2D2:                         % Control Unit 2, Device 2
   ADDRESS        = "AA".
```

Line Section

Each line in the network is described in the Line Section.

SMCS Line Declaration

```
LINE TDLINE:
    ADDRESS   =   2:0:0.
    CONTROL   =   CANDETDCTL.
    STATION   =   SMITH,
                  JONES,
                  ROOM273,
                  EXT11.
```

SYCOM Line Declaration

```
LINE SYCOMLINE:
    ADDRESS   =   2:0:1.
    CONTROL   =   SYCOMCTL.
    STATION   =   SYCOMCTRL,      % control stations
                  SYCOMSPO,
                  SYCOMCRD,
                  SYCOMPTR,
                  SYCOMFTP,
                  USER1,          % user virtual stations
                  USER2,
                  SERVER1,        % server virtual stations
                  SERVER2.
```

Remote Printer Line Declaration

```
LINE APLINE:
    ADDRESS   =   2:0:2.
    CONTROL   =   CANDETDCTL.
    STATION   =   OFFICE
```

RJE3780 Line Declaration

```
LINE RJE3780LINE:
    ADDRESS   =   2:0:3.
    CONTROL   =   RJE3780CNTRL.
    STATION   =   RJE3780ST1,
                  RJE3780ST2.
```

EM3270/SNA Line Declaration

```
LINE BSC3270LINE:
    ADDRESS     =   2:0:4.
    AUTOPOLL    =   22.                 %   11 * number of control units
    CONTROL     =   BISYNC3270CONTROL.
    STATION     =   C1D1,
                    C1D2,
                    C2D1,
                    C2D2.
```

File Section

Stations are associated with remote files in the File Section.

SMCS File Declaration

```
FILE MCSREMOTE:
    RESIDENT    =   MEMORY.
    FAMILY      =   SMITH,          %   physical stations
                    OFFICE,
                    JONES,
                    EXT11,
                    C1D1,           %   EM3270/SNA Bisync 3270 stations
                    C1D2,
                    C2D1,
                    C2D2,
                    SERVER1,        %   SYCOM server virtual stations
                    SERVER2.
```

SYCOM File Declaration

```
FILE LINK:
    RESIDENT    =   MEMORY.
    FAMILY      =   SYCOMCTRL,      %   control stations
                    SYCOMSPO,
                    SYCOMCRD,
                    SYCOMPTR,
                    SYCOMFTP,
                    USER1,          %   user virtual stations
                    USER2.
```

Remote Printer File Declaration

```
FILE APFILE:
    RESIDENT    =   MEMORY.
    FAMILY      =   OFFICE.
```

RJE3780 File Declaration

```
FILE RJE3780Q:
    RESIDENT    =   MEMORY.
    FAMILY      =   RJE3780ST1,
                    RJE3780ST2.
```

Patch Identification

Record positions 81 through 90 inclusive can be used for patch record identification.

Compiler Control Statements

Various options are available during compilation and can be activated by $ control statements. The options cover the areas of list format, error messages, warning messages, symbolic maintenance, and merging symbolic statements.

Each $ control statement must have the $ symbol in record position 1. The options to be included follow the $ with one or more spaces separating each option specified. The various options can be grouped on one or more symbolic statements, with the exception of the LIBRARY options which must be on a separate symbolic statement. The options cannot be split across statement boundaries. All options except LIST, CHECK, and DOUBLE are reset by default. The available options are described in the following paragraphs.

CHECK

This option causes the compiler to print warning messages for sequence errors in the network controller symbolic input. A sequence error occurs when the sequence number of the previous symbolic statement is greater than or equal to the current sequence number.

CODE

The generated SDL2 object code is listed on the line printer when this option is specified.

DOUBLE

The DOUBLE option causes the printer listing of the network controller symbolic code to be double-spaced.

LIBINFO

This option lists information about Request and Control procedures included in the SYSTEM/CONTROLLER standard network controller.

LIBRARY <library identifier>

The Declaration, Request, or Control symbolic code specified by <library identifier> is retrieved from the NDL/LIBRARY and inserted in the network controller symbolic code following the $LIBRARY statement.

The LIBRARY option cannot be included in a statement containing other options.

When the LIBRARY option is specified to access Declaration, Request, and Control procedures, the Declaration must precede the first Request procedure, and the Request procedures must precede the first Control procedure.

LIST

The network controller symbolic code is listed on the line printer.

LISTDOLLAR

All $ control statements are listed on the network controller symbolic code output listing.

MERGE

This option is used to merge the primary input with the secondary input.

NEW

A new symbolic file is created when this option is specified.

NO

Options can be reset by specifying NO followed by the name of the option to be reset. NO does not affect the VOID or LIBRARY options. The NO $ compile option is the same as the RESET $ compile option.

PAGE

This option causes the compiler to skip to the top of a new page when printing.

RESET

Options can be reset by specifying RESET followed by the name of the option to be reset. RESET does not affect the VOID or LIBRARY options. The RESET $ compile option is the same as the NO $ compile option.

SEQ <seq base> + <seq increment>

The network controller symbolic code can be sequenced by supplying a beginning sequence number and an increment. The numbering begins at <seq base> and is incremented by <seq increment>.

If only $ SEQ is specified (<seq base> and <seq increment> omitted) the numbering starts with 00000000 and each succeeding number is incremented by 100. Sequence numbers for $ control statements are printed when $ SEQ is specified.

SET

Options can be set by specifying SET followed by the name of the option to be reset. SET does not affect the VOID or LIBRARY $ compile options.

SINGLE

The listing of the network controller symbolic code on the line printer is single-spaced when this option is specified.

VOID

This option can be used in conjunction with $ MERGE to eliminate certain secondary symbolic records from the new symbolic file being created. By specifying $VOID, the secondary symbolic record with the current sequence number is skipped by the compiler.

VOID <integer>

$ VOID can optionally be followed by an 8-character integer, which instructs the compiler to skip all secondary symbolic records beginning at the current sequence number and continuing until the reading of a secondary symbolic record that has a sequence number higher than the 8-character integer specified.

WARNSUPR

Syntax warnings are not printed on the compile listing when the WARNSUPR option is specified.

Network Analysis

An analysis of the network controller code file Network Information Tables is produced automatically with the compile listing if the LIST $ compile option is set. This listing provides a summary of the Declarations, the Request and Control procedures specified, the terminal declarations, the station declarations, the line declarations, and the file declarations.

An analysis of the Network Information Tables can also be produced by explicitly executing the NDL compiler and file equating the file CODE to the previously compiled network controller. For example:

```
EXECUTE NDL FI CODE TITLE <network controller title>;
```

SECTION 5
NETWORK CONTROLLER INSTALLATION

Network Controller Execution

After the network controller program is compiled, it can be executed in two ways: explicitly or automatically.

The network controller is executed explicitly using the following MCP control instruction:

EXECUTE <network controller program identifier>

In order to have the network controller program automatically executed by the MCP, it is necessary to identify to the MCP the name of the network controller. This is done by placing the identifier of the network controller program into the C entry, or slot, of the MCP Name Table using the following MCP command:

CM C <network controller program identifier>

An experimental network controller can be placed in the CX entry, or slot, of the MCP Name Table using the following MCP command:

CM CX <network controller program identifier>

Network Controller Priority

For optimum response time for the remote programs, it is strongly suggested that the network controller run at a higher priority than other jobs in the mix. The network controller is automatically assigned a priority of 15 by the NDL compiler and the invisible bit is set.

Network Controller Program Switches

Program Switch 2

Program switch 2 controls the network controller handling of data communications line that is "hung" or not responding properly. In all cases of line "hangs," a message is displayed on the operator display terminal (ODT). The following action is then taken depending on the value of program switch 2.

SW 2 Value	Action
0	A program dump is taken and the line is restarted.
1	The line is restarted.
2	The line is left in a "hung" status.
3-15	The network controller is suspended until the operator responds using the NC ODT command.

Program Switch 3

Program switch 3 controls the initial accumulation of statistics by the network controller. The following action is taken depending on the value of program switch 3.

SW 3 Value	Action
0	The statistics function is controlled by the STATISTICS declaration in the network controller symbolic code.
1	Unconditionally initiate accumulation of statistics from beginning of job.
2	Do not initiate accumulation of statistics regardless of the STATISTICS declaration in the network controller symbolic code.
3-15	Same as for program switch 3 equal to zero.

Program Switch 7

If the network controller program switch 7 is set to 1, the IOLOG debugging facility is initiated at beginning of job. Refer to the Diagnostic Aids Section of the B 1000 Systems Network Definition Language (NDL) Language Manual for more information on the IOLOG debugging facility.

Network Controller Run Errors

To ensure proper functioning of the data communications system, the network controller does not continue execution after certain errors occur. When a run error occurs, one of the following messages is displayed on the operator display terminal (ODT):

RUN ERROR 1nn <text> AT <sequence number>
RUN ERROR 8nn AT <sequence number>
RUN ERROR 9nn AT <sequence number>

The <sequence number> is the place in the SYSTEM/CONTROLLER symbolic code where the run error occurred, and the error number is described as follows:

User Environment Errors

Error Number	Description
100	Memory parity error reported in an I/O descriptor.
101	Too many remote files have gone into overflow mode.
102	Overflow queue count error on a close of a remote file.
103	Overflow queue count error on a retrieval of a message for a remote file.
104	Overflow queue message not found while trying to retrieve a message for a remote file.
105	Invalid assignment to the STATION variable in the Control procedure.
106	The STATION variable was not set to a valid station on the line before exiting the Control procedure.
107	A station was not found waiting dialout when expected.
108	Invalid Request number when entering the Request procedure.
109	The Request number would have changed after a RECEIVE ADDRESS(STATION) statement.
110	The Request number would have changed after a TERMINATE OUTPUT(RETURN) statement.

Error Number	Description
111	A station was not found on the line after a RECEIVE ADDRESS(STATION) statement.
112	A station was not found on the line after an autopoll operation had completed.

Network Information Encoding Errors

Error Number	Description
800	Missing declaration record.
801	Declaration record version mismatch.
802	Terminal information record bad.
803	Translate table information record bad.
804	Station information record bad.
805	Line information record bad.
806	File information record bad.

NDL System Errors

Error Number	Description
993	Complex wait return value out of range.
994	Line buffer memory layout problem.
995	Autopoll buffer memory layout problem.
996	Static memory layout problem.
997	Too much static memory requested for line and autopoll buffers.
998	Bad code file structure.
999	MCP and network controller incompatibility.

SECTION 6
SYSTEM/CONFIGURE

Introduction

The SYSTEM/CONFIGURE program is provided to program the multiline control-4 (MLC-4), which is a programmable version of the original hard-wired multiline control. The multiline control-4 (MLC-4) consists of a single base-card and up to four quad adapter cards each capable of driving up to four data communications lines. Before it is capable of driving a line, each quad adapter must be loaded by a program with the firmware necessary to run. In addition, each of the four lines must be loaded with the proper configuration before it can run. This is also done by a program but after the firmware has been loaded to the quad adapter.

Previously there were a number of adapters available for the hard-wired multiline control. There exists individual line adapters for the asynchronous, binary synchronous (bisync), synchronous, Teletype, and two-wire direct interface (TDI) line disciplines. In addition, each adapter had a number of parameters (timeout values for example) which were set into the adapter by means of hardware straps.

For the MLC-4, this multitude of line adapters is replaced by a single-line adapter called a quad line adapter. The line discipline and the associated parameters are stored in line configurations. This line configuration, which is loaded to each line of a quad adapter, serves to configure each line to run a potentially different discipline as well as supply the values for what was previously entered by means of hardware straps on older adapters. The only strapping that is required for the quad line adapters is to specify whether the interface is RS232 or TDI.

Therefore, before the MLC-4 can be used by software, the firmware must be loaded into the multiline control and the appropriate line configurations into the quad adapter. The firmware and line configuration must be reloaded if one of the following occurs:

1. A parity error is encountered in the firmware.
2. If the line discipline is to be changed from what was originally loaded into the adapter.

Included in the system software release is a file labeled SYSTEM/MLFIRMWARE which contains the firmware to be loaded to a MLC-4 and a series of standard line configurations. This firmware is compiled into the standard network controller, SYSTEM/CONTROLLER.

All other programs that interface directly with a MLC-4 are capable of loading the firmware and line configurations to the MLC-4. A similar routine has been included in each of these programs to do the load and to do a reload when necessary. However, all of these programs, other than the network controller, obtain their firmware and line configurations from the standard system file SYSTEM/MLFIRMWARE.

SYSTEM/MLFIRMWARE File

The SYSTEM/MLFIRMWARE file is created either by the SYSTEM/CONFIGURE program or by using the network controller NC CREATE FIRMWARE ODT command. The SYSTEM/MLFIRMWARE file is modified using the SYSTEM/CONFIGURE program. It contains both the firmware, which must be loaded to the quad adapters, as well as the line configurations. The SYSTEM/MLFIRMWARE file is capable of storing 50 line configurations.

The file is composed of a header record, followed by the firmware, and a space to hold the line configurations. The file is composed of one area, and contains 90 character records, blocked 20.

All configurations accessed by data communications programs are referenced by the name of the configuration. The name of the configuration is defined within the program and at load time the SYSTEM/MLFIRMWARE file is opened and a search of the linked list of entries is performed to find the configuration of that name.

SYSTEM/CONFIGURE Operation

The SYSTEM/CONFIGURE program is used to create and modify the MLC-4 firmware in the SYSTEM/MLFIRMWARE file, the current network controller, the experimental network controller, or a user-specified firmware file, which all contain the code for the microprocessor in the MLC-4.

When modifying the firmware file, input to the program is made through a remote file opened by the program.

Program Switch

Only program switch 9 is used by the SYSTEM/CONFIGURE program. If program switch 9 is set, the SYSTEM/MLFIRMWARE file is created and the SYSTEM/CONFIGURE program goes to end of job.

Program Execution

The SYSTEM/CONFIGURE program can be executed from the Operator Display Terminal (ODT) or through the SMCS program.

If the SYSTEM/CONFIGURE program is executed through the SMCS program, the following control string is used:

```
EXECUTE SYSTEM/CONFIGURE
```

The SYSTEM/CONFIGURE can be executed through the ODT running the SYSTEM/CONTROLLER standard network controller using the following control string:

```
EXECUTE SYSTEM/CONFIGURE FILE REMOTE STATIONS SYSTEMODT;
or
EXECUTE SYSTEM/CONFIGURE FILE REMOTE TITLE OPERATOR;
```

The SYSTEM/CONFIGURE program is designed to function only with Burroughs screen terminals such as a TD 830 or MT 983/MT 985 device. The use of any other terminals can produce undefined results.

Program Operation

The following paragraphs describe the commands entered to the SYSTEM/CONFIGURE program using the remote file interface. These commands are used to modify the SYSTEM/MLFIRMWARE file, the current network controller, the experimental network controller, or a user-specified firmware file.

All input to the program is expected to be in upper-case letters. The only exception to this is when a NAME of a configuration is entered. The name may be any series of non-blank characters. Also, within the fields presented on the various screens, input may be in free-form format. However, embedded blanks within tokens are not permitted.

In the examples of output from the SYSTEM/CONFIGURE program that follow, the "> and <" characters are used to represent the left and right forms characters.

BAUD-RATE

The line speed of the configuration is to be entered for the BAUD-RATE parameter. Line speeds from 110 to 56000 can be entered. If a line speed greater than 19200 baud is selected, the high speed option of the quad adapter is invoked. The high speed option is valid only for the first line of the quad adapter and invalidates all other lines declared for that quad adapter. Hence, the high speed option can only be specified for quad adapters 0, 4, 8, and 12 for a particular multiline control. The high speed option is only valid for the BISYNC-A, BISYNC-E, SYNC, and SYNC-J.

DISCIPLINE

Allowable line disciplines that can be entered for the DISCIPLINE parameter follow:

Mnemonic	Line Discipline
ASYNC	Standard Asynchronous.
BISYNC-A	Binary Synchronous (Bisync) ASCII.
SYNC-E	Binary Synchronous (Bisync) EBCDIC.
SYNC	Standard Synchronous.
TDI	Two-wire Direct Interface.
TTY	Teletype Interface.
ASYNC-J	Standard Asynchronous, Japanese.
SYNC-J	Standard Synchronous, Japanese.
TDI-J	Two-wire Direct Interface, Japanese.

NOTE

For Japanese users, the ASYNC-J, SYNC-J, and TDI-J values for the DISCIPLINE parameter are valid. The specification of one of these values is equivalent to declaring SPECIALALPHABET = KANJI in the Terminal Section of the network controller symbolic code. Both the Japanese variants for the DISCIPLINE parameter and the KANJI option in the network controller symbolic code should not be declared at the same time.

RECEIVE-DELAY

Any numeric value from 0 to 99999 can be entered for the RECEIVE-DELAY parameter to specify the write-to-read delay in milliseconds.

TRANSMIT-DELAY

Any numeric value from 0 to 99999 can be entered for the TRANSMIT-DELAY parameter to specify the read-to-write delay in milliseconds.

INITIAL-TIMEOUT

Any numeric value from 0 to 99999 inclusive can be entered for the INITIAL-TIMEOUT parameter to specify the timeout before the first character read in milliseconds.

TEXT-TIMEOUT

Any numeric value from 0 to 99999 inclusive can be entered for the TEXT-TIMEOUT parameter to specify the timeout between characters in milliseconds.

STOP-BITS

Values of 0, 1, 1.5 or 2 can be entered for the STOP-BITS parameter to specify the number of stop bits.

VERTICAL-PARITY
Either NONE, ODD or EVEN can be entered for the VERTICAL-PARITY parameter to specify the vertical parity.

CHARACTER-SIZE
Either 5, 6, 7 or 8 can be entered for the CHARACTER-SIZE parameter to specify the character size in bits.

CONTINUOUS-CARRIER
Either a Y (yes) or N (no) can be entered for the CONTINUOUS-CARRIER parameter to specify the presence or absence of a continuous carrier field.

BCS
Either NONE, EVEN, ODD or CRC can be entered for the BCS parameter to specify the block check sequence.

PAD
Either a Y (yes) or N (no) can be entered for the PAD parameter to indicate the presence or absence of pad characters.

TRANSPARENCY
Either a Y (yes) or N (no) can be entered for the TRANSPARENCY parameter to indicate the presence or absence of transparency.

DIAL-MODE
Either a Y (yes) or N (no) can be entered for the DIAL-MODE parameter to indicate the presence or absence of dial mode.

SWITCHED
Either a Y (yes) or N (no) can be entered for the SWITCHED parameter to indicate switched or leased lines.

BREAK-ENABLE
Either a Y (yes) or N (no) can be entered for the BREAK-ENABLE parameter to indicate the presence or absence of a break enable.

LONG-BREAK
Either a Y (yes) or N (no) can be entered for the LONG-BREAK parameter to indicate the presence or absence of a long break.

PSEUDO-CARRIER-DETECT
Either a Y (yes) or N (no) can be entered for the PSEUDO-CARRIER-DETECT parameter to indicate the presence or absence of a pseudo carrier detection.

NEW-SYNC
Either a Y (yes) or N (no) can be entered for the NEW-SYNC parameter to indicate the presence or absence of the NEW-SYNC option.

EOT-DISCONNECT
Either a Y (yes) or N (no) can be entered for the EOT-DISCONNECT parameter to specify disconnect upon receipt of an EOT character.

CCITT
Either a Y (yes) or N (no) can be entered for the CCITT parameter to specify that the CCITT option is to be used.

SELECT-RATE
Either a Y (yes) or N (no) can be entered for the SELECT-RATE parameter to specify that the CCITT select rate option is to be used.

SELECT-STANDBY
Either a Y (yes) or N (no) can be entered for the SELECT-STANDBY parameter to specify that the CCITT select standby option is to be used.

A Burroughs Field Engineer should be consulted for an explanation of parameters to the SYSTEM/CONFIGURE program and the proper values for these parameters.

When entering the configuration, the screen should be transmitted from the home position so that the entire screen may be entered.

The value of each parameter is checked. In addition to the value entered for each parameter, two other checks are made. It is not possible to set both the DIAL-MODE and NEW-SYNC parameters or the EOT-DISCONNECT and CCITT parameters in the same configuration.

After entering the screen, the following message is returned if no errors were detected:

CONFIGURATION NUMBER <number> – <name> ACCEPTED.

A prompt to the program returns the original menu again.

If any errors are detected, the configuration is not accepted and may be re-entered. In addition to errors as a result of bad values entered, an error is indicated if either the NAME entered was previously defined for another configuration or if the firmware file is full of configurations. If the firmware file is full, it is necessary to delete a configuration before another can be entered.

If any of the parameter fields are in error, the screen is returned with the erroneous fields highlighted.

Examples of the three types of error messages returned are:

CONFIGURATION NAME IS NOT DEFINED - RECORD_NUMBER WILL BE USED
The configuration name was not defined, so the record number is used instead of the name.

CONFIGURATION FILE FULL
The configuration file is full and no more entries can be made.

A CONFIGURATION OF NAME <name> IS ALREADY DEFINED
The specified configuration is already defined. The CHANGE option should be specified instead.

Change Configuration

If the Change configuration option is specified, the SYSTEM/CONFIGURE program responds with the following message:

```
ENTER CONFIGURATION NAME - >          <
```

The user must then enter the name of a previously defined configuration, else the following message is returned.

THAT CONFIGURATION IS NOT DEFINED.

It is then necessary to prompt the program to have the original menu returned. If a valid name is entered, then that configuration is returned in the following form:

```
NAME ---------- >TDI96      < NUMBER 393
BAUD-RATE ---------- > 9600< 110-19200, HIGH SPEED OPTION > 19200
DISCIPLINE ----- >      TWDI< TDI,SYNC,ASYNC,TTY,BISYNC-E,BISYNC-A
RECEIVE-DELAY ------ >     0< WRITE-TO-READ DELAY (SQUELCH) (MILLI-SEC)
TRANSMIT-DELAY ----- >    32< READ-TO-WRITE DELAY IN MILLI-SEC
INITIAL-TIMEOUT ---- > 3000< TIMEOUT BEFORE FIRST CHAR READ (MILLI-SEC)
TEXT-TIMEOUT ------- >    32< TIMEOUT BETWEEN CHARACTERS (MILLI-SEC)
STOP-BITS ---------- >   1< 0, 1, 1.5, 2
VERTICAL-PARITY ----- >EVEN< NONE, ODD, EVEN
CHARACTER-SIZE ---------- >7< 5, 6, 7, OR 8 BITS PER CHARACTER
CONTINUOUS-CARRIER ----- >N< Y/N
BCS ----------------- >NONE< NONE, EVEN, ODD, CRC
PAD ------------------- >N< Y/N
TRANSPARENCY ---------- >N< Y/N
DIAL-MODE ------------- >N< Y/N
SWITCHED -------------- >N< Y/N
BREAK-ENABLE ---------- >N< Y/N
LONG-BREAK ------------ >N< Y/N
PSEUDO-CARRIER-DETECT -- >N< Y/N
NEW-SYNC -------------- >N< Y/N
DCT-DISCONNECT -------- >N< Y/N
CCITT ----------------- >N< Y/N
SELECT-RATE ----------- >N< Y/N
SELECT-STANDBY -------- >N< Y/N
```

The meanings of the parameters are the same as previously defined for the create configuration option.

The screen should be transmitted with the cursor in the home position.

The screen entered is checked for validity as if creating a new configuration.

Indications of errors occur in the same manner as when creating a line configuration.

If the screen was correctly entered the following message is returned:

CONFIGURATION NUMBER <number> – <name> CHANGED.

If the name of the configuration was changed the <number> and <name> returned reflects that of the new configuration. A prompt is then required to return the original menu.

Delete Configuration

If the Delete configuration option is specified, the SYSTEM/CONFIGURE program responds with the following message:

```
ENTER CONFIGURATION NAME OR "ALL" - >          <
```

If ALL is not entered, then the name is checked and an error is returned if the configuration named is not defined. A prompt is then necessary.

If ALL is entered then each configuration is returned to the screen one at a time. If ALL is not entered then only the configuration requested is returned. The configurations are not automatically deleted. The terminal is put in forms mode with a Y (yes) or N (no) response required. The user must respond with either of these two answers. Only a Y (yes) causes the configuration to be deleted. If an invalid response is entered then the following message is returned:

INVALID RESPONSE PLEASE ENTER "Y" OR "N".

The format of the screen is:

```
NAME ----------  DEFAULT          NUMBER --- 393
BAUD-RATE ----------   9600
DISCIPLINE -----        IWDI              SHOULD THIS BE DELETED
RECEIVE-DELAY ------       0                 ENTER "Y" OR "N"
TRANSMIT-DELAY -----       0                      > <
INITIAL-TIMEOUT ----   3000
TEXT-TIMEOUT -------      32
STOP-BITS ------------     1
VERTICAL-PARITY -----  EVEN
CHARACTER-SIZE ---------  7
CONTINUOUS-CARRIER -----  N
BCS ----------------  NONE
PAD --------------------  N
TRANSPARENCY -----------  N
DIAL-MODE --------------  N
SWITCHED ---------------  N
BREAK-ENABLE -----------  N
LONG-BREAK -------------  N
PSEUDO-CARRIER-DETECT --  N
NEW-SYNC ---------------  N
EOT-DISCONNECT ---------  N
CCITT ------------------  N
SELECT-RATE ------------  N
SELECT-STANDBY ---------  N
```

After the delete has been performed and if ALL had been entered, then the following message is returned:

END DELETE <number> CONFIGURATIONS DELETED.

If a single name was supplied then the following is returned:

CONFIGURATION NUMBER <number> [NOT] DELETED.

A prompt is then necessary.

Display Configuration

The Display configuration option permits the display of line configurations on a terminal.

The initial response to entering this option is similar to that of the Delete configuration:

```
ENTER CONFIGURATION NAME OR "ALL" - >                    <
```

If ALL is entered then all line configurations are displayed. Each configuration is displayed on the screen. A prompt is required to proceed to the next configuration. After the configurations are displayed the following message is returned:

<number> RECORDS DISPLAYED.

If the name of a configuration is entered, then the name is checked for validity, and if present, is displayed on the terminal and the following message is returned to the screen:

RECORD NUMBER <number> – <name> DISPLAYED.

After either of the two preceding messages, the program must be prompted before the original menu is returned.

Print Configuration

The Print configuration option permits the user to obtain a printed copy of the line configurations in the firmware file.

The initial response to entering this option is similar to that of delete line configuration:

```
ENTER CONFIGURATION NAME OR "ALL" - >          <
```

If ALL is entered then a copy of all line configurations is printed. Each configuration is printed on a separate page in a format identical to that of the screen. After the configurations are printed the following message is returned:

<number> RECORDS PRINTED.

If the name of a configuration is entered, then the name is checked for validity, and if present, is printed on the line printer and the following message returned to the screen:

RECORD NUMBER <number> – <name> PRINTED.

After either of the above messages the program must be prompted before the original menu is returned.

Standard Line Configurations

The following are the standard line configurations in the SYSTEM/CONTROLLER standard network controller and the SYSTEM/MLFIRMWARE file.

Configuration Name	Line Protocol
ASYNC12	1200 baud asynchronous leased.
ASYNC12SW	1200 baud asynchronous switched.
BISYNC48	4800 baud binary synchronous (Bisync) leased.
BISYNC48SW	1200 baud binary synchronous (Bisync) switched.
DEFAULT	9600 baud two-wire direct connect (TDI). Same as TDI96.
ODTDI96	9600 baud two-wire direct connect (TDI) for use with the system ODT.
SYCOM192	19200 baud two-wire direct connect (TDI) for use with the SYCOM program.
SYNC24	2400 baud synchronous leased.
SYNC24SW	2400 baud synchronous switched.
SYNC48	4800 baud synchronous leased.
SYNC48SW	2400 baud synchronous switched.
TDI96	9600 baud two-wire direct connect (TDI).
TDI192	19200 baud two-wire direct connect (TDI).
TTY300SW	300 baud Teletype switched.

APPENDIX D
SYNTAX AND NOTATION CONVENTIONS

Railroad diagrams show how syntactically valid statements can be constructed. Traversing a railroad diagram from left to right, or in the direction of the arrow heads, and adhering to the limits illustrated by bridges produces a syntactically valid statement. Continuation from one line of the diagram to an- is represented by a right arrow (→) appearing at the end of the current line and the beginning of the next line. The complete syntax diagram is terminated by the vertical bar (|).

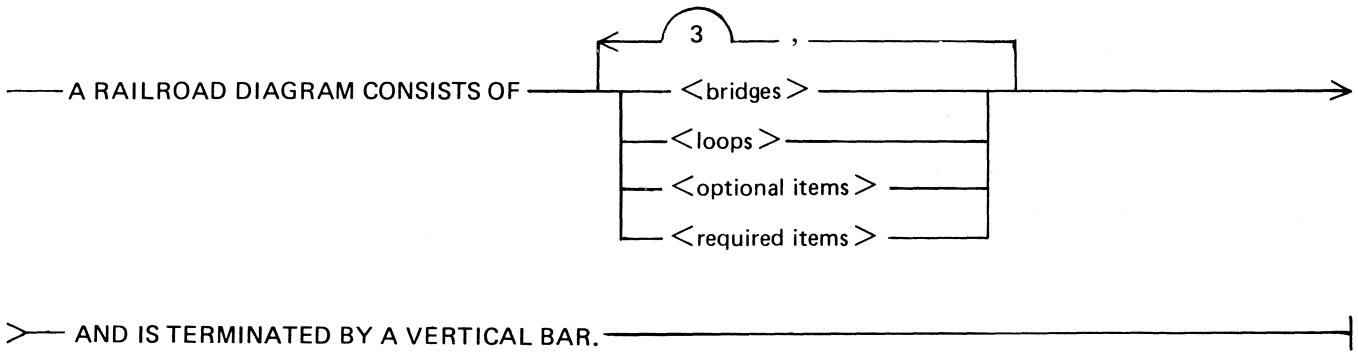
Syntactic Variables

Items contained in broken brackets (< >) are syntactic variables which must be further defined or for which the user is required to supply the information requested.

Keywords

Upper-case items are keywords and must appear literally. The minimum abbreviation for a keyword is underlined.

Example:



G50051

The following syntactically valid statements can be constructed from the above diagram:

A RAILROAD DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTI- BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items> AND IS TERMINATED BY A VERTICAL BAR.

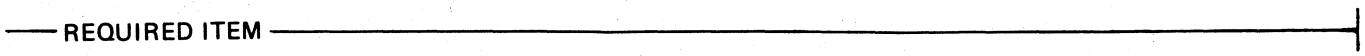
A RAILROAD DIAGRAM CONSISTS OF <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items>, <required items>, <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

Required Items

No alternate path through the railroad diagram exists for required items or required punctuation.

Example:

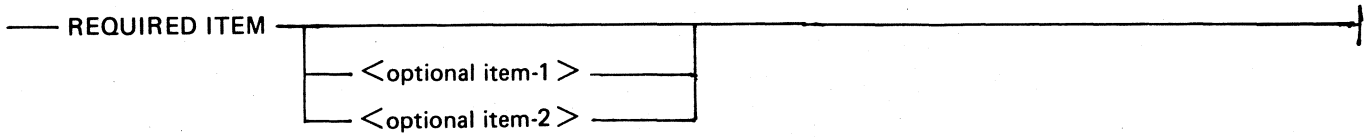


G50052

Optional Items

A vertical list of items indicates that the user must make a choice of the items specified. An empty path through the list allows the optional item to be absent.

Example:



G50053

The following valid statements can be constructed from the above diagram:

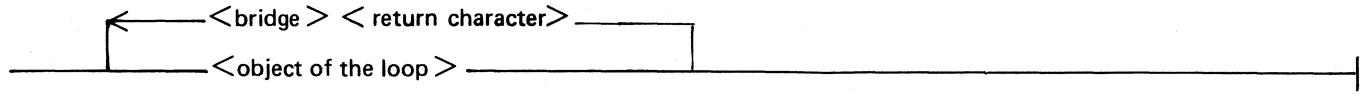
REQUIRED ITEM

REQUIRED ITEM <optional item-1>

REQUIRED ITEM <optional item-2>

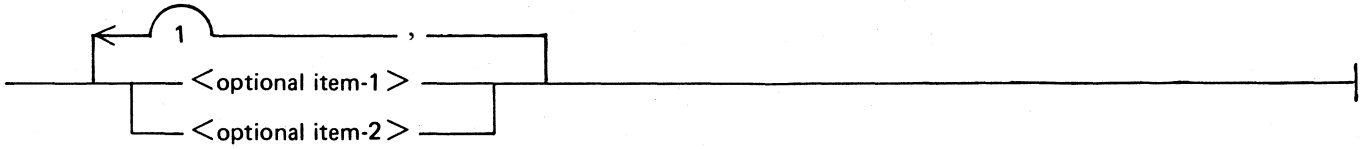
Loops

A loop is a recurrent path through a railroad diagram and has the following general format.



G50054

Example:



G50055

The following statements can be constructed from the railroad diagram in the example.

<optional item-1>

<optional item-2>

<optional item-1>,<optional item-1>

<optional item-1>,<optional item-2>

<optional item-2>,<optional item-1>

<optional item-2>,<optional item-2>

A loop must be traversed in the direction of the arrow heads, and the limits specified by bridges cannot be exceeded.

Bridges

A bridge illustrates the minimum or maximum number of times a path can be traversed in a railroad diagram.

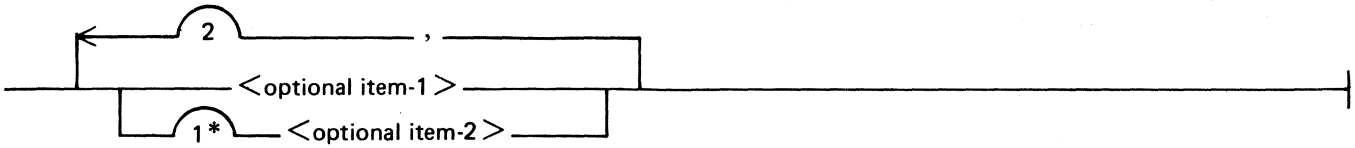
There are two forms of bridges.

n is an integer which specifies the maximum number of times the path can be traversed.

n is an integer which specifies the minimum number of times the path must be traversed.

G50056

Example:



G50057

The loop can be traversed a maximum of two times; however, the path for <optional item-2> must be traversed at least one time.

The following statements can be constructed from the railroad diagram in the example.

<optional item-1>,<optional item-2>
<optional item-2>,<optional item-2>,<optional item-1>
<optional item-2>

INDEX

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)