Burroughs

B 1000

Systems

Network

Controller

Installation

Guide

Comments or suggestions regarding this document should be submitted on a Field Com-
munication Form (FCF) with the CLASS specified as 2(S.W.:System Software), and the
Type specified a 1(F.T.R.), and the product specified as the 7-digit form number of the
manual (for example, 5025257).

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

GENERAL

This manual describes the B 1000 Network Controller installation and the operation of the SYSTEM/QCONFIGURE program which creates line configurations for the B 1990 systems.

AUDIENCE LEVEL

This manual 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: Controls, Line Adapters, Firmware, and Line Configurations
  Describes the B 1000 hardware used for data communications, firmware and line configurations loaded into line adapters. The selection of firmware and configurations files at network controller compile and run time is described. The set of standard line configurations is listed.

Section 7: SYSTEM/QCONFIGURE Program
  Describes the use of the SYSTEM/QCONFIGURE 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 this document.

RELATED DOCUMENTS

The following documents are referenced in this document:

B 1000 Software Manuals

B 1000 Systems NDL/LIBRARY Functional Description Manual, form number 1168937.

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

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

B 1000 Systems System Software Operation Guide, Volume 1, form number 5024508.

B 1000 Systems System Software Operation Guide, Volume 2, form number 1169091.

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, 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 for the data communications hardware controls.
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.
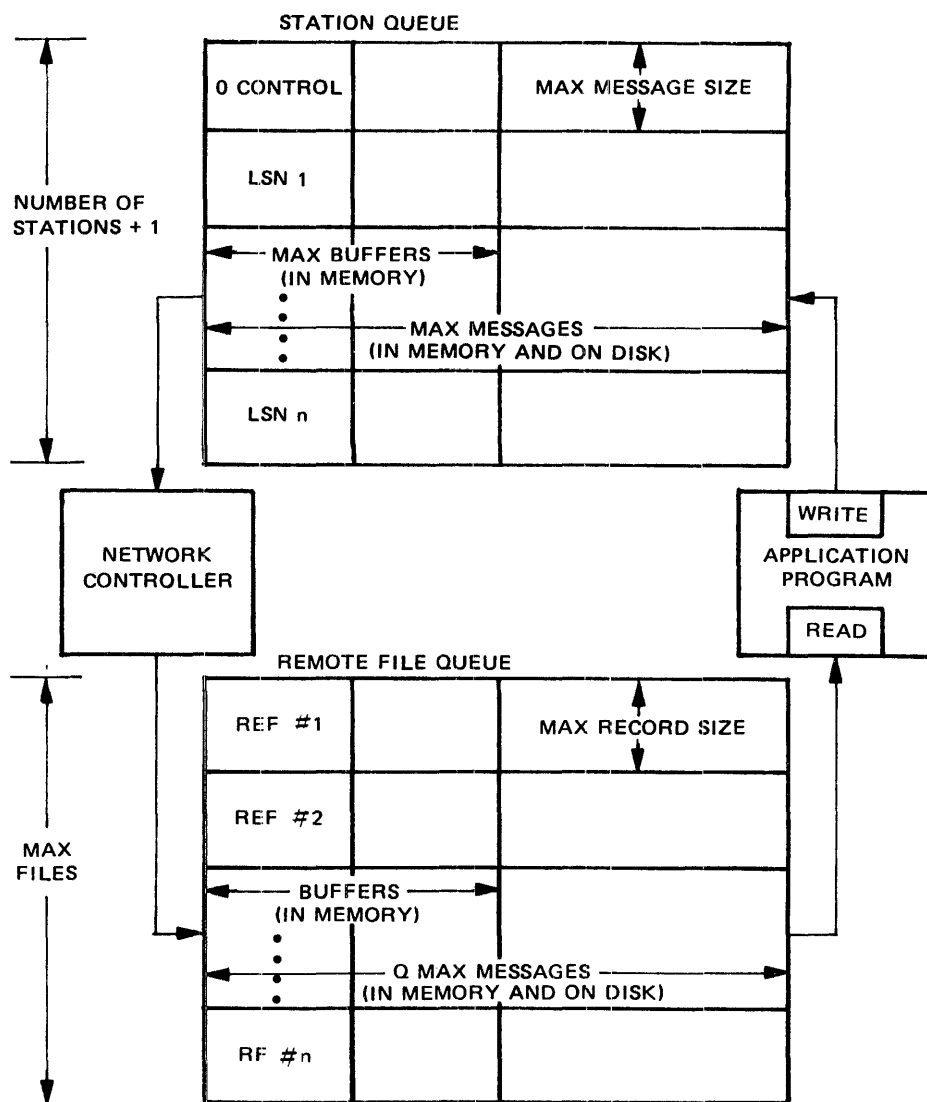
Figure 2-1. Basic Message Flow

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.
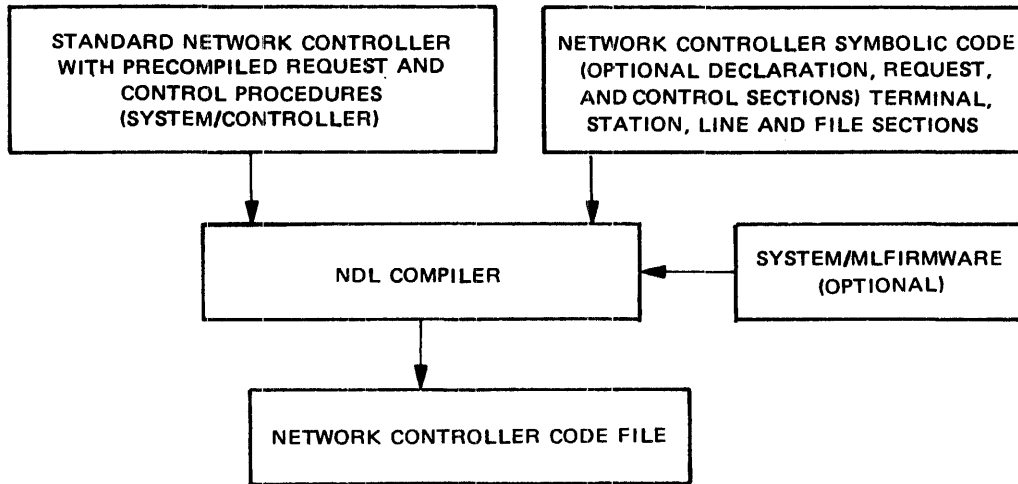
Figure 2-2. Network Controller Generation Process

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/QCONFIGURE program (for B 1990 systems).

The following software is used to debug the network controller.

- DC/AUDIT program.

- SYSTEM/IDA program.

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

- BNA (Station Transfer).

- Command AND Edit MCS (CANDE).

- DMS/INQUIRY program.

- EM3270/BSC program.

- EM3270/SNA program.

- GEMCOS MCS.

- IBASIC program.

- MCS74 program.

- ODESY.

- Remote Display (RD) program.

- REMOTE/LISTER program.

- REMOTE/PRINT program.

- RJE3780 program.

- RJE/CONTROLLER program.

- Supervisory Message Control System MCS (SMCS).

- System Communication Module program (SYCOM).

- SYSTEM/QCONFIGURE program.

- TEXT/EDITOR 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	12504 bytes
Line buffers	4502 bytes
General buffer	2306 bytes
Autopoll buffer	20 bytes
Tables	1000 bytes
Total	20332 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.
6. For each station, add MAXMESSAGESIZE * MAXBUFFER bytes.

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.

SECTION 3
NETWORK CONTROLLER SYMBOLIC SPECIFICATIONS

GENERAL

A network controller symbolic file contains seven sections, three of which are optional. These sections are:

1. Declaration Section (optional)
2. Request Section (optional)
3. Control Section (optional)
4. Terminal Section
5. Station Section
6. Line Section
7. File Section

Each of these sections are described in detail after the description of the common syntax.

COMMON SYNTAX

The following syntax is common to a large part of the network controller symbolic specifications and is summarized here. This syntax is not repeated throughout this section.

Identifier

An identifier is a string of characters used to represent some entity, such as an item name. An identifier can be up to 72 characters in length. An identifier must begin with a letter and cannot have special characters except the underscore (_) character embedded within the identifier. An identifier cannot be an NDL reserved word.

Syntax:
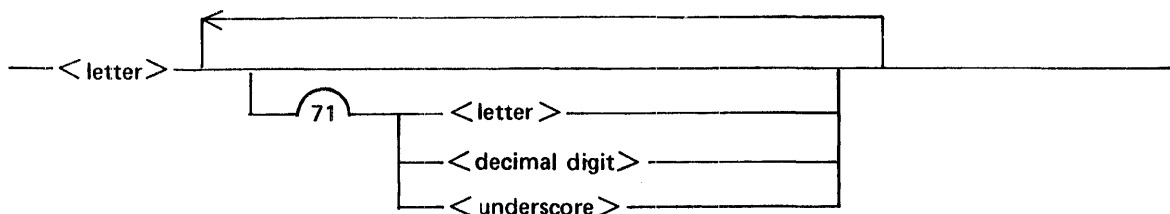
Semantics:

\<letter\>
   A \<letter\> can be any of the upper-case letters A through Z or any of the lower-case letters a through z.

\<decimal digit\>
   A \<decimal digit\> can be any of the decimal digits 0 through 9.

\<underscore\>
   An \<underscore\> consists of the single character "_".

Pragmatics:

Identifiers cannot be split across symbolic code line boundaries.

Symbolic Name

Symbolic names are names that are assigned to terminals, stations, lines, and files. Symbolic names follow the same rules as identifiers with one exception: they can be up to ten characters in length.

Syntax:
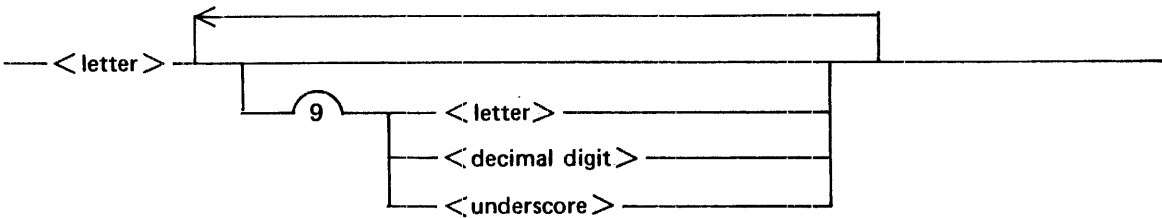
Semantics:

Each of the items in the symbolic name syntax has the same meaning as in the identifier syntax.

Pragmatics:

Symbolic names cannot be split across symbolic code line boundaries.

File Identifier

A <file identifier> is used to identify and reference a file.

Syntax:
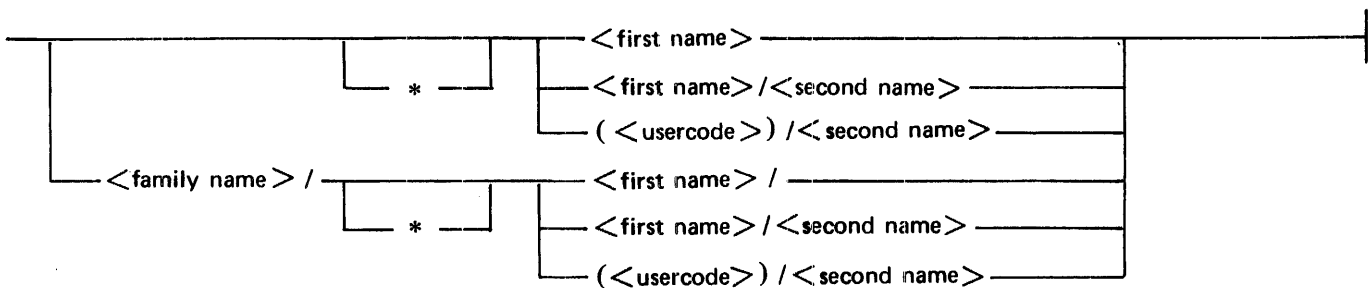
Semantics:

<family name>
    A <family name> is the name of a disk or a pack. A <family name> consists of one to ten characters.

*

    The asterisk character is used in a file name to override the default usercode and the default pack <family name> associated with that usercode.

<first name>
    A <first name> is the first part of a file name. A <first name> consists of one to ten characters. When the asterisk syntax is used, the asterisk character counts as one of the ten characters in the <first name>.

<second name>
    A <second name> is the second part of a file name. A <second name> consists of one to ten characters.

<usercode>
    A <usercode> is a name assigned to a user to secure system and file access. When a < usercode> in parentheses is the first part of a file name, the file belongs to the user represented by < usercode>. A <usercode> consists of one to eight characters. When the asterisk syntax is used with a <usercode>, the asterisk character counts as one of the eight characters in the <usercode> .

Integer

An integer is a string of adjacent decimal digits representing the decimal value of the integer. Only positive integers are allowed in the NDL language, and they are treated as 24-bit binary numbers. An integer cannot exceed 16777215.

Syntax:
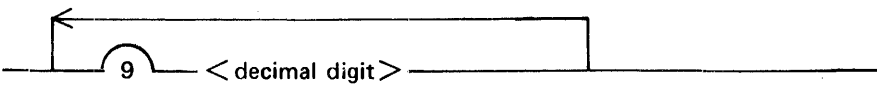
Semantics:

<decimal digit>
    A <decimal digit> can be any of the decimal digits 0 through 9.

Pragmatics:

Integers cannot be split across symbolic code line boundaries.

String

A string is either an EBCDIC string or a hexadecimal string.

EBCDIC String

An EBCDIC string consists of one or more adjacent EBCDIC characters enclosed within quotation marks. The maximum length of an EBCDIC string is 80 bytes. An internal quotation mark is represented by two consecutive quotation marks. For example, "" "" represents an EBCDIC string of one single quotation mark.

Syntax:
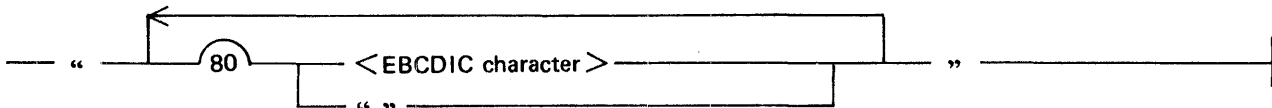
Semantics:

<EBCDIC character>
    An <EBCDIC character> is any 8-bit character in the EBCDIC character set.

Hexadecimal String

A hexadecimal string consists of one or more hexadecimal digits enclosed in quotation marks and preceded by the number 4. The maximum length of a hexadecimal string is 80 bytes or 160 hexadecimal digits.

Syntax:
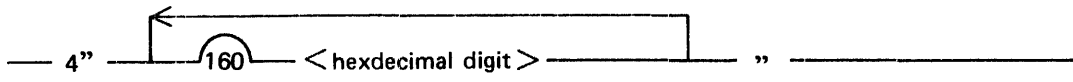
Semantics:

<hexadecimal digit>
    A hexadecimal digit is a 4-bit character whose binary value is represented by one of the numeric digits 0 through 9 or one of the upper-case letters A through F.

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.

Refer to the B 1000 Systems NDL/LIBRARY Functional Description Manual for a complete description of the NDL/LIBRARY file.

Syntax:
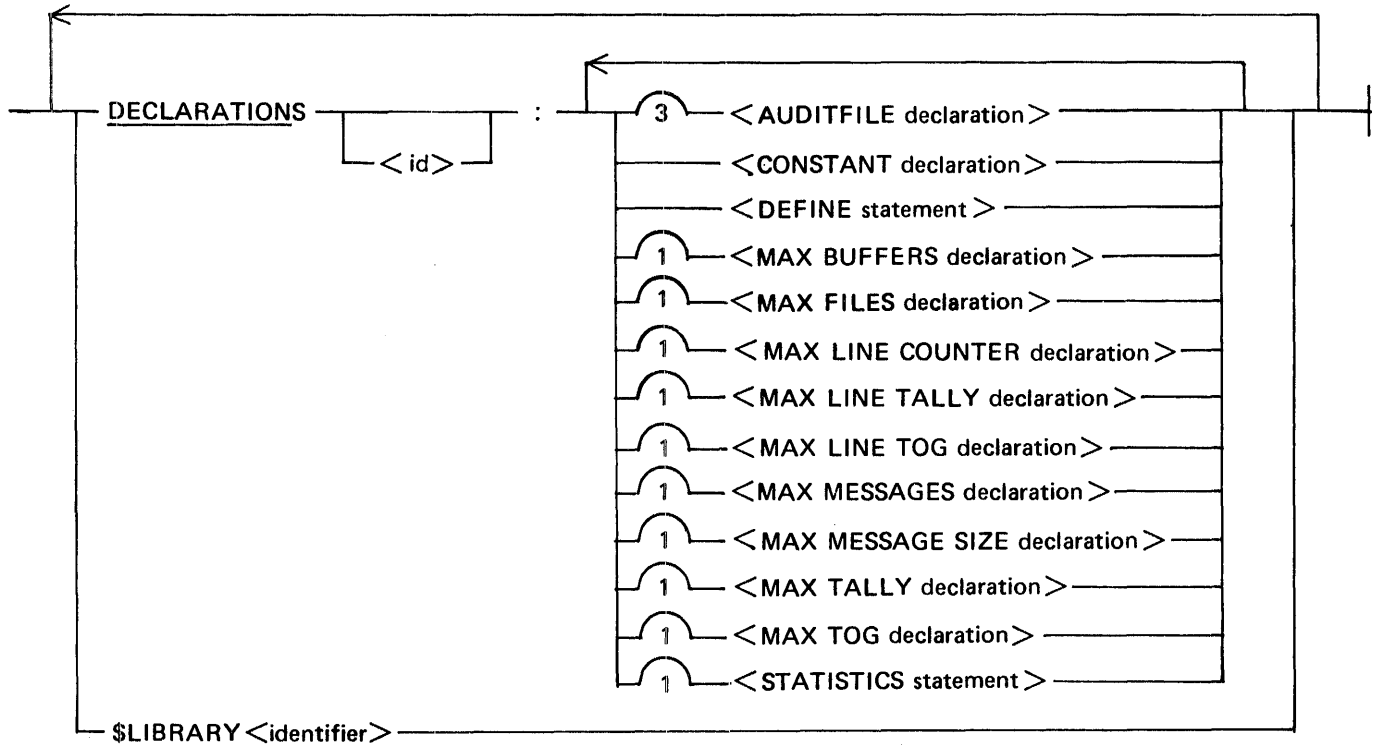
Semantics:

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

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:
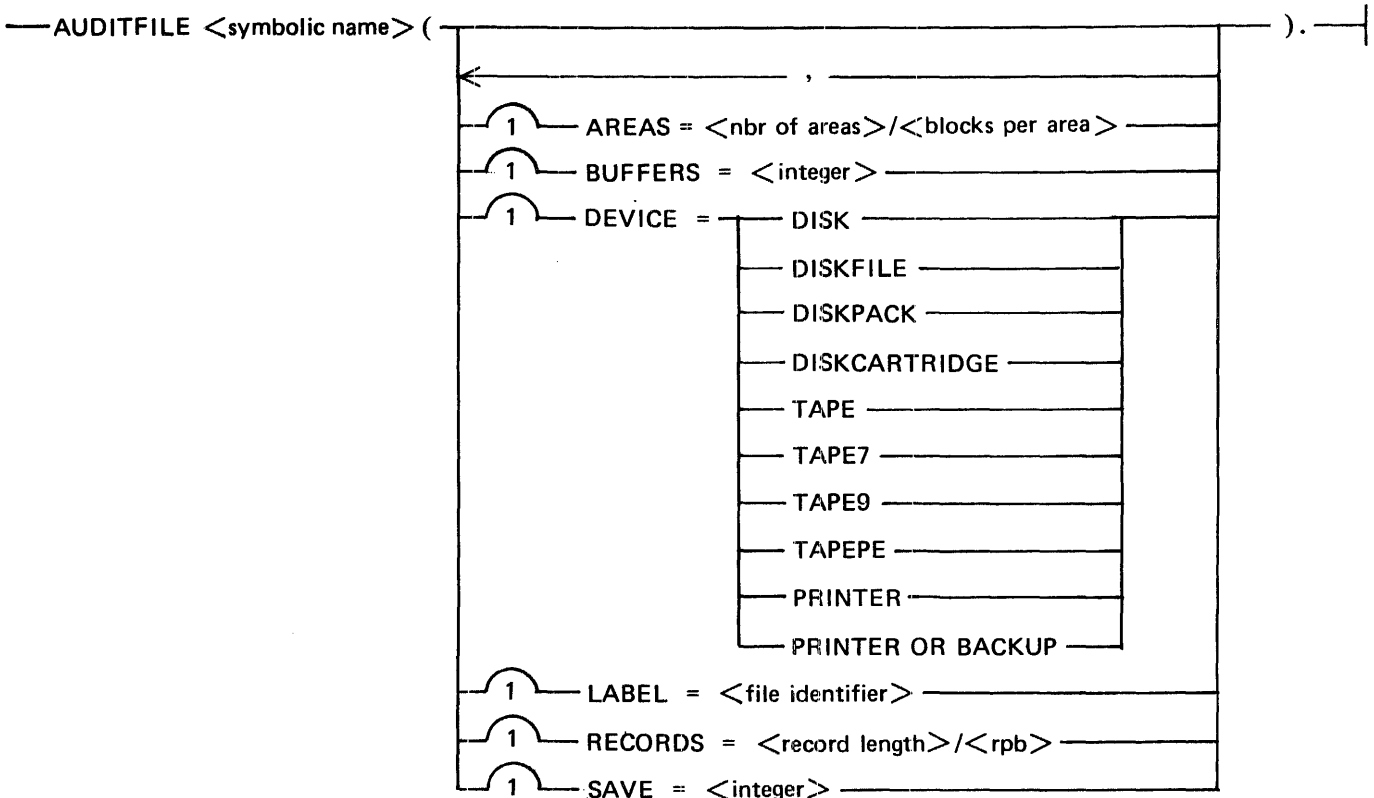
Semantics:

<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. <rpb> (records per block) specifies the
>   number of records per block.

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:

```
                           ┌───────────── , ──────────┐
  ── CONSTANT ───┬──── <identifier> = <string> ───────┴──── . ──────────────┤
```
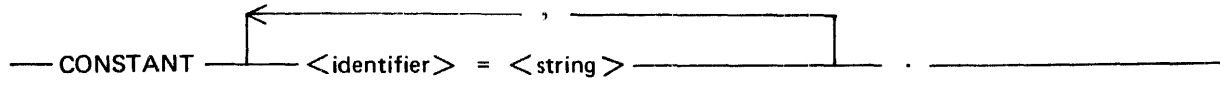
Twenty-eight common control character identifiers have been defined within the NDL compiler to equal the strings shown in table 3-1. The identifiers can be redefined to equal other strings but cannot be used as non-constant identifiers.

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".
```

DEFINE Statement

The DEFINE statement allows a token or tokens to be redefined as another token. Beginning with the first Request procedure, any define <identifier> found by the compiler is replaced by its <definition>. Nested DEFINE statements are permitted with no restriction on the number of levels.

Syntax:

```
                          ┌─────────────── , ───────────────┐
   ── DEFINE ──────┌──────  < identifier >  = #  < definition > #  ──────┘──── . ──────────────────┤
```

Pragmatics:

1. A define <identifier> cannot be an NDL reserved word or an identifier which is already a constant.
2. Compiler control ($) records must not contain defined symbols.
3. Any control record detected while processing a DEFINE statement is ignored.
4. Comments are allowed within the < definition>, but are discarded during processing. Also, all excess blanks are removed.
5. A define <identifier> must not define itself.
6. A <definition> must not exceed 327 characters.
7. Nested defines are expanded without regard to the number of levels until the total size of the expansion and remaining symbolic image exceeds 400 characters.
8. All define statements are global.

Example:

```
DEFINE CHECKRETRY= # IF RETRY GE 0
            THEN DO.
                    RETRY := RETRY - 1.
                    TERMINATE NOINPUT.
            END.#
```

MAX BUFFERS Declaration

The MAX BUFFERS declaration specifies the maximum number of buffers for the station queue (the output part of a remote file). MAX MESSAGES declares the number of messages that can be queued for a station in the station queue, and MAX BUFFERS declares the number of those messages that can reside in memory.

Syntax:

```
──────── MAX BUFFERS = <integer> · ──────────────────────────────────────────────┤
```

G18195

Semantics:

\<integer\>

    \<integer\> can be in the range from 1 to 255 inclusive. The default value for \<integer\> is two buffers.

Example:

    MAX BUFFERS = 5.

MAX FILES Declaration

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

Syntax:

——————MAX FILES = <integer> · ———————————————————————————————————————|

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:

```
── MAX LINE  (  COUNTER  [ <integer> ]  )  · ──────────────────┤
```

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]).

MAX LINE TALLY Declaration

The MAX LINE TALLY declaration specifies the maximum number of line tallies that are available for use by the Request and Control procedures of the network controller being generated. The standard SYSTEM/CONTROLLER network controller contains 30 line tallies per line.

Syntax:

```
——MAX LINE  ( TALLY  [ <integer> ]  )  · —————————————————————|
```

Semantics:

<integer>
   The value of <integer> can be in the range from 0 to 29 inclusive; for any other values, a syntax error is generated.

Example:

    DECLARATION:
       MAX LINE (TALLY [22]).

MAX LINE TOG Declaration

The MAX LINE TOG declaration specifies the maximum number of line toggles that are available for use in the Request and Control procedures of the network controller being generated. The standard SYSTEM/CONTROLLER network controller contains 30 line togs per line.

Syntax:

```
── MAX LINE  (  TOG  [ < integer > ]  )  ·  ─────────────────────┤
```

Semantics:

<integer>
  The value of <integer> can be in the range from 0 to 29 inclusive; for any other values, a syntax error is generated.

Example:

    DECLARATION:
        MAX LINE (TOG [22]).

MAX MESSAGES Declaration

The MAX MESSAGES declaration specifies the maximum number of messages that can be held in the station queue (the output part of a remote file) for a station. The difference between the MAX MESSAGES and MAX BUFFERS declaration is the number of messages in the station queue that are written to disk.

Syntax:

```
————— MAX MESSAGES = <integer> · ————————————————————————————————————————|
G18197
```

Semantics:

<integer>
    The default value <integer> is 20.

Example:

    MAX MESSAGES = 40.

MAX MESSAGE SIZE Declaration

The MAX MESSAGE SIZE declaration specifies the maximum number of characters that can be received by the network controller from a user program and the MCP. This declaration defines the message size for the station queue (output part of a remote file).

If the SYCOM program is used, the MAX MESSAGE SIZE declaration must be at least (BUFFERSIZE / 2) − 7 where BUFFERSIZE is the BUFFERSIZE declaration in the Terminal Section. The value of the SYCOM BUFFER parameter must be no greater than (BUFFERSIZE / 2) − 7. For more information, refer to the B 1000 Systems SYCOM Installation, Operation and Functional Description Manual.

If the MAX MESSAGE SIZE declaration is omitted, the NDL compiler assigns a MAX MESSAGE SIZE sufficient to hold the largest of the following:

1. A maximum size REMOTE_FILE_INFO_REPLY message.
2. A maximum size file OPEN message.
3. The largest BUFFERSIZE declaration in the Terminal Section.

Syntax:

```
——— MAX MESSAGE SIZE = <integer> · ——————————————————————————|
```
G18198

Semantics:

<integer>
    <integer> specifies the maximum message size in bytes.

Pragmatics:

A MAX MESSAGE SIZE that is too small can cause truncation of messages from the data communications system to the remote file with no warning.

Example:

    MAX MESSAGE SIZE = 2500.

MAX TALLY Declaration

The MAX TALLY declaration specifies the maximum number of station tallies available for use by the Request and Control procedures of the network controller being generated. The standard SYSTEM/CONTROLLER program contains 30 counters per station.

Syntax:

```
—— MAX TALLY  [ < integer > ]  ·  ————————————————————————|
```

Semantics:

<integer>
    The value of <integer> can be in the range from 0 to 29 inclusive; for any other values, a syntax error is generated.

Example:

    DECLARATION:
      MAX TALLY [22].

MAX TOG Declaration

The MAX TOG declaration specifies the maximum number of station toggles available for use in the Request and Control procedures of the network controller being generated. The standard SYSTEM/CONTROLLER program contains 30 counters per station.

Syntax:

```
──MAX TOG  [ <integer> ]  · ─────────────────────────────┤
```

Semantics:

<integer>
    The value of <integer> can be in the range from 0 to 29 inclusive; for any other values, a syntax error is generated.

Example:

    DECLARATION:
        MAX TOG [22].

STATISTICS Statement

The STATISTICS statement specifies whether the statistics gathering process is to begin at network controller beginning of job. This process is also controlled by the setting of program switch 3. The statistics function is disabled by default.

Syntax:

```
── STATISTICS  =  ──┬── FALSE ──────┬──────────────────┤
                    └── TRUE ───────┘
```

Example:

```
    DECLARATION:
       STATISTICS = TRUE.
```

REQUEST SECTION

The Request Section is optional and consists of one or more Request procedures, which are line discipline routines that are used by the network controller in communicating with the remote devices of the data communications network. A Request procedure can be defined for each type of I/O capability for a terminal. If a terminal is capable of sending input to the system and receiving output from the system, a receive and transmit Request procedure must be defined for that terminal. For each of these capabilities, the Request procedure must be specified in the Terminal Section of the network controller symbolic file.

Syntax:

Semantics:

<identifier>
    The $LIBRARY <identifier> is the name of a standard Request procedure. The REQUEST <identifier> is the name assigned to a user-written Request procedure.

<Request statement list>
    <Request statement list> is a group of user-written Request statements which constitute the Request procedure.

The standard library Request procedures to handle common line disciplines have been precompiled into the SYSTEM/CONTROLLER standard network controller. The precompiled Request procedures, grouped by category, are:

The CANDE Request procedures

    CANDEPOLTD
    CANDEFSLTD
    C74CANDFSL
    CANDESELTD
    C74CANDSEL

The Point-to-Point Request procedures

    TCTUPTXMIT
    TCTUPTRCV
    TCTUPTIO
    TDPTXMIT
    TDPTRCV
    TDBATCHXMT

The Multipoint Request procedures

    POLTCTDDYN
    SELTCTDDYN

The Bisync 3270 Host Multipoint Request procedures

    BISYNC3270INPUT
    BISYNC3270OUTPUT

The SYCOM Request procedure

SYCOMREQ

The Teletype Request procedures

WRITETTY
READTTY
CANDEIOTTY

The RJE3780 Request procedure

RJE3780RQS

The RJE Host Request procedure

RJEHOST

Refer to the *B 1000 Systems NDL/LIBRARY Functional Description Manual* for a description of each of these NDL/LIBRARY Request procedures. For those users wishing to code their own Request procedures, refer to section 4, NDL Request Procedure Statements, of the *B 1000 Systems Network Definition Language (NDL) Language Manual* for an explanation of the Request statements.

Refer to table 3-2 for a list of Request procedures and their associated Control procedures. Table 3-3 shows the terminals used with the various Request and Control procedures.

Examples:

$LIBRARY CANDEPOLTD
$LIBRARY CANDESELTD

CONTROL SECTION

The line Control Section is optional and consists of one or more line Control procedures. A line Control procedure is invoked by the network controller to decide which I/O function should get use of the line. Each line must have a line Control procedure associated with it. The purpose of a line Control procedure is to assign a station for activity and initiate the proper Request procedure to perform a communication procedure.

Syntax:

Semantics:

<identifier>
    The $LIBRARY <identifier> is the name of a standard Control procedure. The CONTROL <identifier> is the name assigned to a user-written Control procedure.

<Control statement list>
    <Control statement list> is the group of user-written Control statements that constitute the Control procedure.

The standard library Control procedures to handle common line disciplines have been precompiled into the SYSTEM/CONTROLLER standard network controller. The precompiled Control procedures are:

    AUTODYNCTL
    BISYNC3270CONTROL
    CANDETDCTL
    CONVERCTL
    PTPTCONCTL
    RJECTL
    RJE3780CNTRL
    SYCOMCTL

Refer to the *B 1000 Systems NDL/LIBRARY Functional Description Manual* for a description of these NDL/LIBRARY Control procedures. For those users wishing to code their own Control procedures, refer to section 5, NDL Control Procedure Statements, of the *B 1000 Systems Network Definition Language (NDL) Language Manual* for an explanation of Control statements.

Refer to table 3-2 for a list of Requests and their associated Control Procedures. Table 3-3 shows the terminals used with the various Request and Control procedures.

Example:

    $LIBRARY CANDETDCTL

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
– –	TDPTRCV	TDPTXMT	PTPTCONCTL	applications
– –	– –	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 AP 1350, 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 TDPTRCV − − TCTUPTIO	TCTUPTXMT TDPTXMT TDBATCHXMT TCTUPTIO	PTPTCONCTL PTPTCONCTL PTPTCONCTL CONVERCTL	TC and TU series TD series TC series TC and TU series
READTTY	WRITETTY	PTPTCONCTL	Teletype
CANDEIOTTY	CANDEIOTTY	CONVERCTL	Teletype
BISYNC3270INPUT	BISYNC3270OUTPUT	BISYNC3270CONTROL	IBM 3270 terminals

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:

```
── TERMINAL ──┬── < symbolic name > : ──────────────────────────────────────────>
              └── DEFAULT  < symbolic name >  :  < default statements > ──────────┤

>──┬──┬──⌒1──── < ADDRESS statement > ──────────────────────────────┬──┤
      ├──⌒*1──── < BUFFERSIZE statement > ─────────────────────────────┤
      ├──⌒1──── < DEFAULT statement > ──────────────────────────────┤
      ├──⌒1──── < DIAGNOSTIC statement > ──────────────────────────┤
      ├──⌒1──── < DUALALPHABET statement > ──────────────────────┤
      ├──⌒1──── < DUALALPHABETDELIMITERS statement > ────────────┤
      ├──⌒*1──── < REQUEST statement > ──────────────────────────┤
      ├──⌒1──── < SPECIALALPHABET statement > ──────────────────┤
      ├──⌒1──── < SPECIALALPHABETOPTION statement > ────────────┤
      ├──⌒1──── < TRANSLATECONTROLCHARACTERS statement > ───────┤
      ├──⌒1──── < TRANSMISSION statement > ─────────────────────┤
      └──⌒1──── < TYPE statement > ─────────────────────────────┤
```
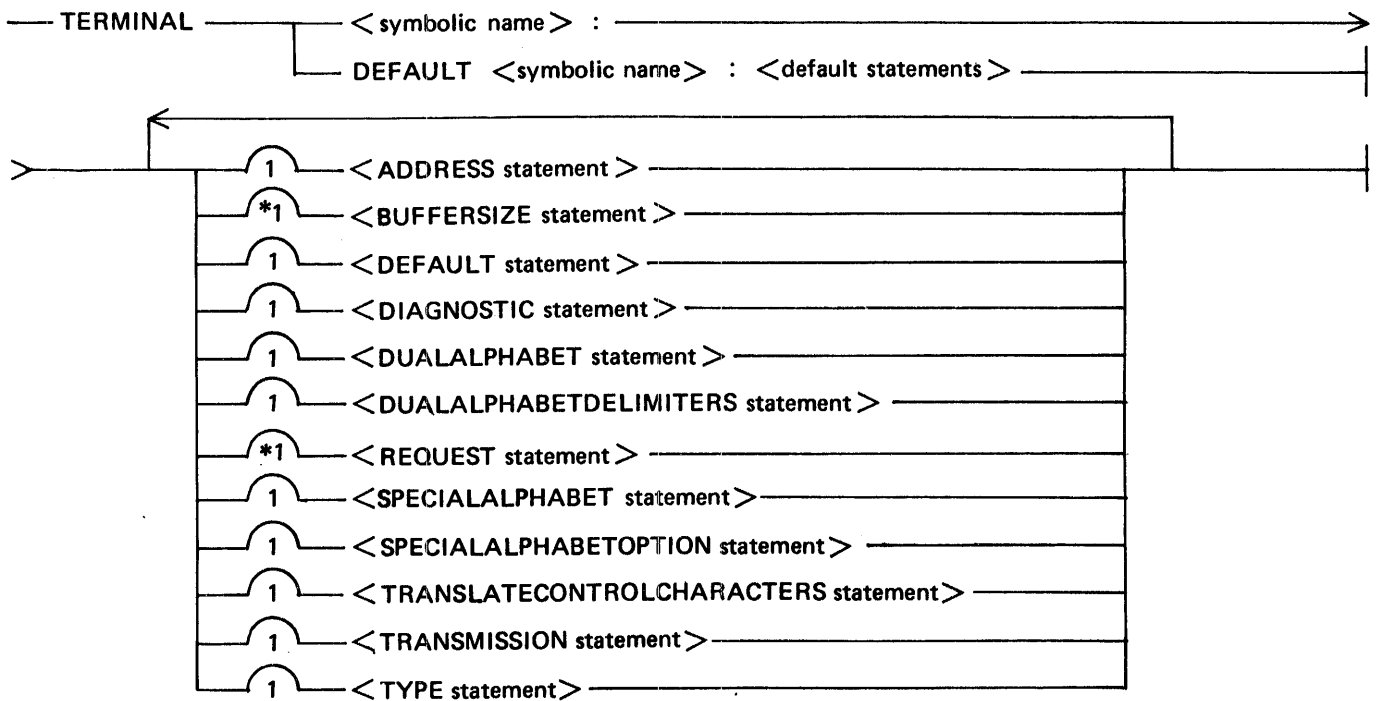
Semantics:

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

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

Predefined Terminals

Two predefined terminal types have been declared for use as the system Operator Display Terminal (ODT) and as a remote diagnostics terminal 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.
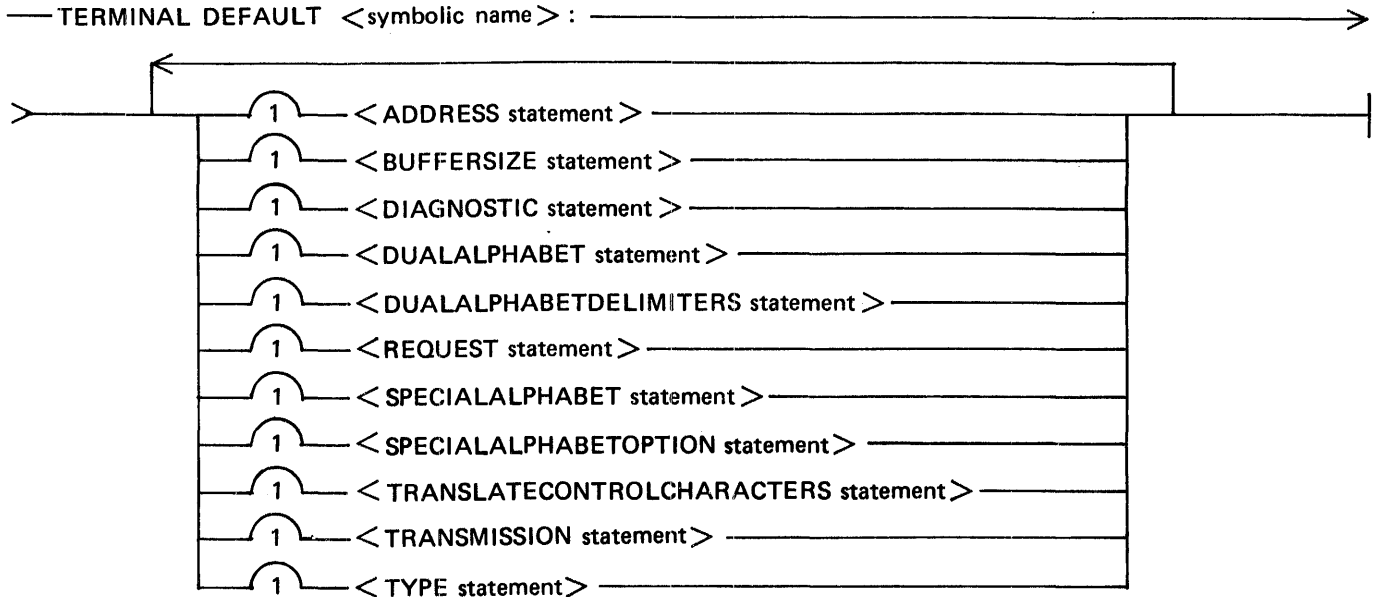
Syntax:

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:

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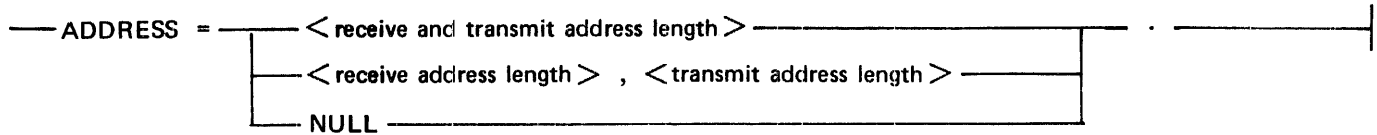
If the <receive address length> and the <transmit address length> of a terminal are different, they must both be specified and separated by a comma. If the ADDRESS statement is omitted, or if ADDRESS = NULL is specified, the address length is assumed to be 0. The possible values that can be specified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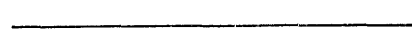
Example:

    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:

```
—BUFFERSIZE = < integer > . ————————————————|
```

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 at least 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:

——DEFAULT = <symbolic name > . ————————————————————|

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:
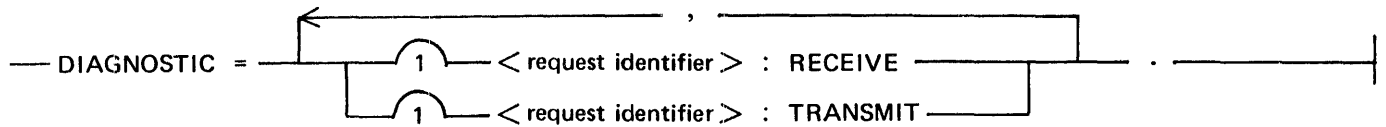
Semantics:

<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.

Syntax:

```
── DUALALPHABET = ──┬── ASCII8 ──┬── . ──────────────┤
                    ├── V20 ──────┤
                    └── V16A ─────┘
```
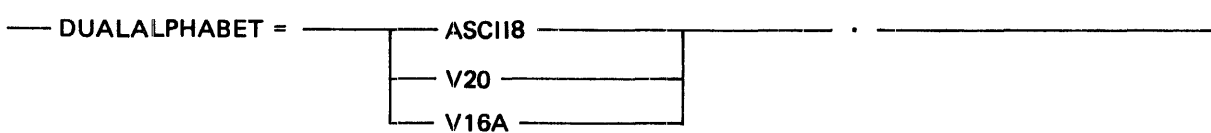
Semantics:

ASCII8
    The ASCII8 translation is used for special MT983 versions of the Arabic dual alphabet terminals.

V20
    The V20 translation is used for the Arabic MT985.

V16A
    The V16A translation is used for the Icelandic accented characters on an ET1100. In this method each accented character is represented by either three or five characters on the data communications line according to the delimiter option chosen.

Pragmatics

If the DUALALPHABET statement is invoked, the DUALALPHABETDELIMITERS statement must also be invoked.

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:

```
──DUALALPHABETDELIMITERS  =  <begin delim> ,  <end delim> .  ─────────────────┤
```

Semantics:

    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 Request 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 as the receive Request procedure is invoked. For output, the Request procedure identified as the transmit Request procedure is invoked.

Syntax:

```
                                   ,
  ── REQUEST  =  ─────────┬──────────────────────────────────────┬──── . ──────────────┤
                          │      ┌─1─┐── <request identifier>  :  RECEIVE ────────┐│
                          └──────┴─1─┴── <request identifier>  :  TRANSMIT ───────┘
```
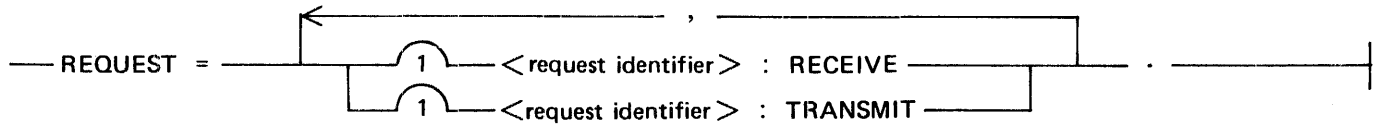
Semantics:

<Request identifier>
     <Request identifier> is the name of the Request procedure to 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 should be declared, even if the terminal device is an output only device such as a remote printer. If a request is not declared and the control section attempts to INITIATE the missing request, a run error 108 occurs.

Example:

    REQUEST = CANDEPOLTD: RECEIVE,
              CANDESELTD: TRANSMIT.

SPECIALALPHABET Statement

The SPECIALALPHABET terminal parameter controls special message translation for terminals. The default option is NULL.

Syntax:

```
── SPECIALALPHABET = ──┬── NULL ──────────────┬──── . ─────────────┤
                       ├── KATAKANAV12 ──────┤
                       ├── KANJIV24 ─────────┤
                       ├── CHINESE ──────────┤
                       └── ICELANDIC ────────┘
```
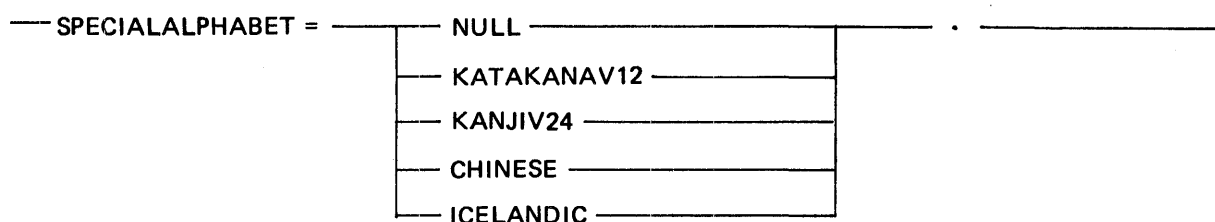
Semantics:

NULL
> The NULL option is the default option and no special message translation occurs for terminals. For standard terminals, this option should either be explicitly set to NULL or omitted completely.

KATAKANAV12
> The KATAKANAV12 option is applicable to a Japanese version of a TD830 terminal which handles Katakana characters within the ordinary terminal character set.

KANJIV24
> The KANJIV24 option is applicable to a Japanese terminal which handles a full English character set, Katakana, and Kanji characters. There are three different character sets that may coexist in a single data communication message to or from a terminal. A very specific interface to a Kanakana to Kanji conversion is also included in this option. The general structure and codes used for this option are the current standard interface for the data communication devices made and marketed for the Japanese subsidiary.

CHINESE
> The CHINESE option is applicable to a terminal that can enter and display Chinese ideographs. There are about 13,000 different characters defined and hence a Chinese character is represented by two 8-bit characters within the terminal and application-program and is transmitted as two or three characters on the communication line.

ICELANDIC
> The ICELANDIC option is applicable to a device that handles the accented upper-case and lower-case A, E, I, O, U, and Y characters. These characters are represented by one 8-bit character within the terminal and application program and by two characters on the line.

SPECIALALPHABETOPTION Statement

The SPECIALPHABETOPTION inhibits translation of certain ESC control sequences, characters following ESC, and ESC $ sequences. The value can range from 0 to 99 inclusive, but only the values 1 or 2 have an effect.

Syntax:

```
—— SPECIALALPHABETOPTION = <number> · ——————————————————|
```

Semantics:

<number>
    The <number> field has a range of 0 to 99 inclusive.

Pragmatics:

The SPECIALALPHABETOPTION = 1 has an effect on terminals defined to use SPECIALALPHABET = KATAKANAV12 or KANJIV24. The effect is to inhibit translation of the ESC " (cursor positioning) characters and ESC $ (page direction) sequences. If this option is used with SPECIALALPHABET = KANJIV24, then any ESC 3 or ESC 4 sequences from the terminal are converted to SI and SO.

The SPECIALALPHABETOPTION = 2 applies to a terminal defined to use SPECIALALPHABET = KANJIV24 only. The effect is to inhibit translation of certain ESC control sequences.

TRANSLATECONTROLCHARACTERS Statement

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

Syntax:

```
—— TRANSLATECONTROLCHARACTERS  =  ————┬— FALSE ————┬— . ———————————————————┤
                                      └— TRUE ·————┘
```

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 TRANSLATE-CONTROLCHARACTERS 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:

```
— TRANSMISSION  =  ———┬— 0 ——————┬—     .  ——————————————|
                      ├— 1 ——————┤
                      ├— 2 ——————┤
                      ├— 3 ——————┤
                      └— NULL ———┘
```
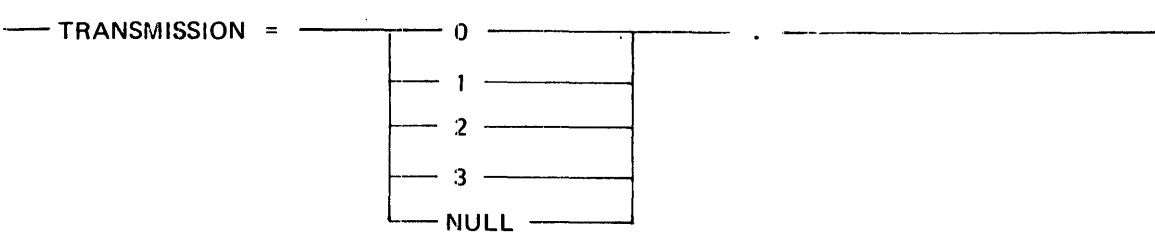
Pragmatics:

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:

```
—— TYPE = <integer> . ———————————————————————|
```

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, TTY 33, TTY 35	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, TU 910	
21	TC 500	
23	TC 700	
24	TC 1700	
25	TC 3500	

Table 3-4. Terminal Types (continued)

Terminal Type Value	Terminal	Comments
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/SR 100/SR 110/B20 EMUL/ ET 1000/ET 2100EMUL	
47-49	(not used)	
50	MT 686	
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) (with B9949-10 INTERFACE)	
71	AP 300	Design Levels 1 and 2
72	AP 1301	Daisy Wheel
73	AP 1340	Serial Matrix
74	AP 1350	Color Dot Matrix
75-78	(reserved for future use)	
79	User Remote Printer	CR, LF, FF, and Buffered
80-99	(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 = 46.

STATION SECTION

The purpose of the Station Section is to define each individual remote device in the data communications system. The Station Section must contain at least one Station definition.

A maximum of 999 stations can be declared in the network controller symbolic file. Each station defined in the Station Section must be associated with a line defined in the Line Section.

The STATION DEFAULT definition is used for stations which have several common attributes.

Syntax:
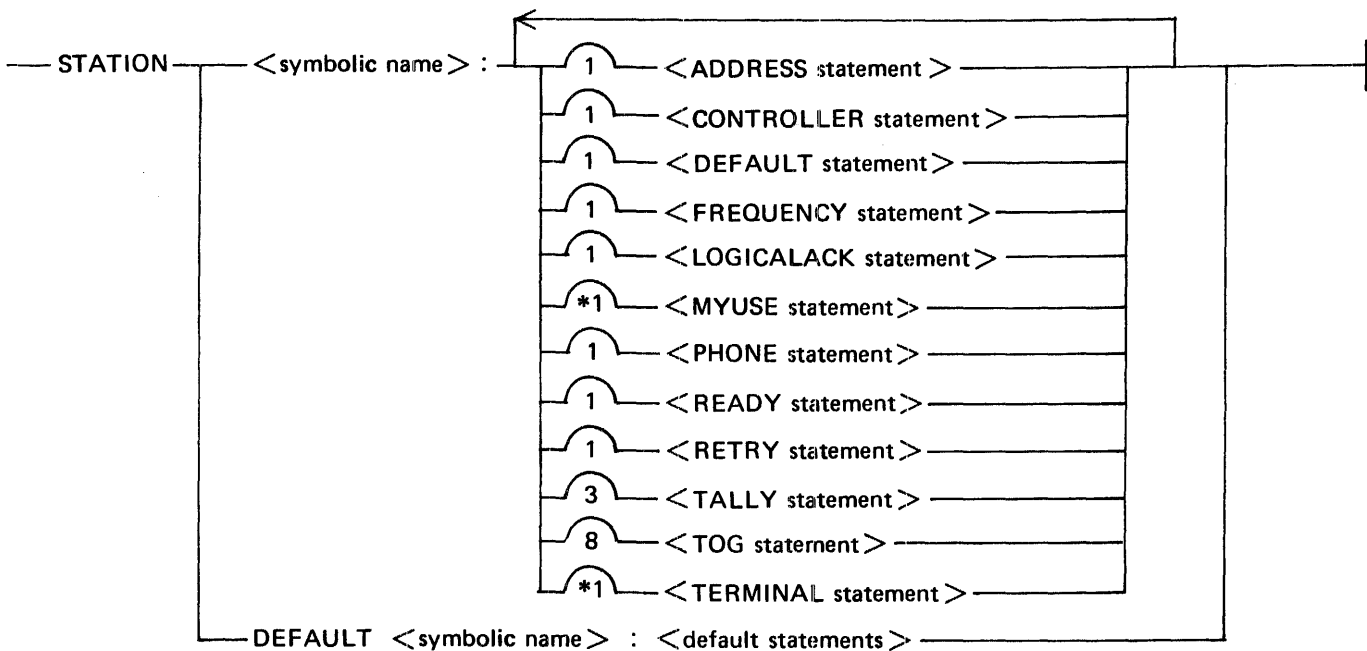
Semantics:

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

<default statements>
    <default statements> are the statements used in the STATION DEFAULT definition.

Predefined Stations

Two predefined stations are provided for use as the system ODT and for remote diagnostics on B 1965/ B 1995 systems. These stations are logical station numbers (LSN) 1 (ODT station) and 2 (remote diagnostics station).

The name for the ODT station is SYSTEMODT and it is declared as follows:

```
STATION SYSTEMODT:
                TERMINAL       = ODT.
                ADDRESS        = "$$".
                FREQUENCY      = 255,255.
                RETRY          = 255.
                MYUSE          = INPUT, OUTPUT.
```

3-42

The name for the remote diagnostic station is RDSTATION and it is declared as follows:

```
STATION RDSTATION:
                TERMINAL        = RDTERMINAL.
                ADDRESS         = "$$".
                FREQUENCY       = 255,255.
                RETRY           = 255.
                MYUSE           = INPUT, OUTPUT.
```

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:
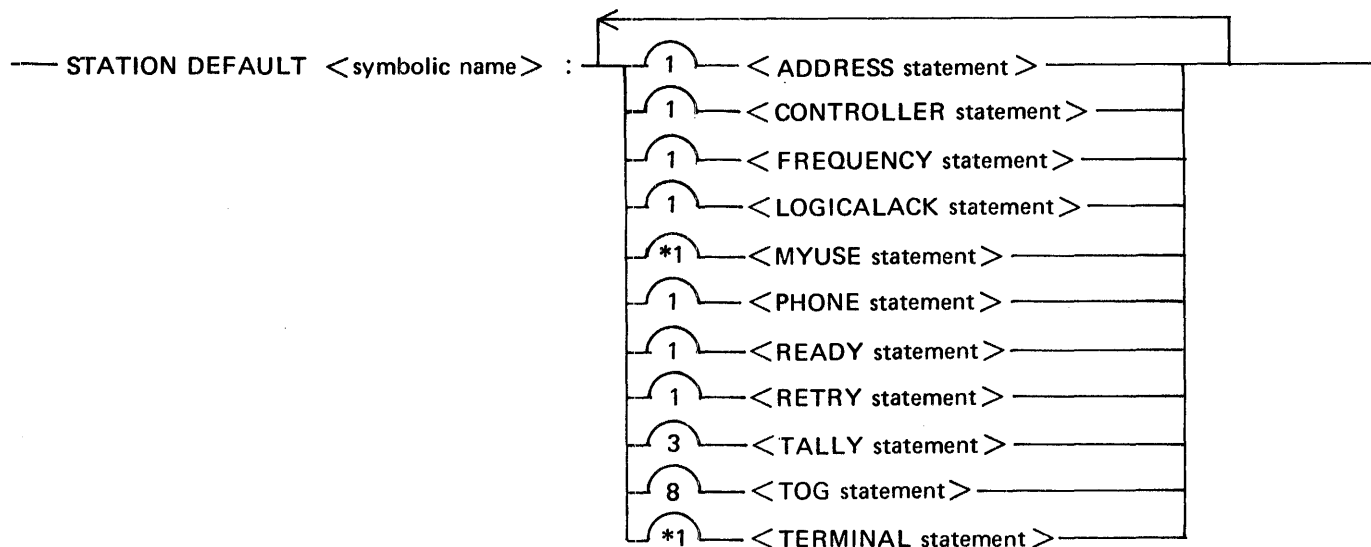
Semantics:

<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 by a DEFAULT statement in a Station definition.

Example:

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

```
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".
```

ADDRESS Statement

The address of a station is specified by the ADDRESS statement and is used for operations such as polling and selecting a station.

Syntax:

```
── ADDRESS = ──┬── < receive and transmit address > ──────────┬── · ─────────────────────┤
               └── < receive address > , < transmit address > ──┘
```

The length of the address must be equal to the length as defined in the ADDRESS statement of the associated Terminal definition in the Terminal Section. If the <receive address> and <transmit address> of a station are different, they must both be specified and separated by a comma. The <receive address> must be specified first, the <transmit address> second.

Example:

        ADDRESS = "01".
        ADDRESS = "AB", " BA".

CONTROLLER Statement

The CONTROLLER statement is required only for stations that are a part of a controller/slave station configuration, as illustrated in figure 3-1. In a controller/slave configuration, a station can reply with an address that differs from the polled address.

Syntax:

```
── CONTROLLER = ──┬── FALSE ──────────────────┬── . ──────────────────┤
                  ├── TRUE ────────────────────┤
                  └── < symbolic name > ───────┘
```
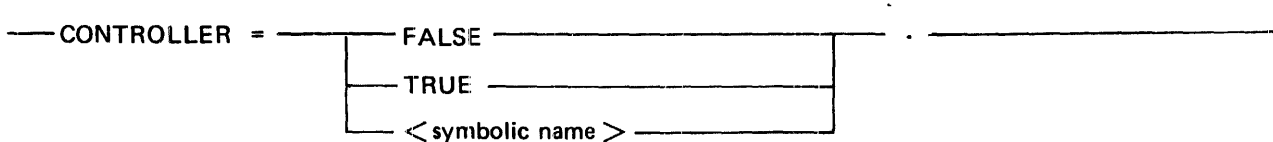
Semantics:

FALSE
  CONTROLLER = FALSE is the default value.

TRUE
  CONTROLLER = TRUE designates the station as a controller, such as the TU910 in figure 3-1.

<symbolic name>
  CONTROLLER = <symbolic name> indicates that the station is a slave station, such as the TU510s in figure 3-1, and <symbolic name> is its controller station. <symbolic name> must refer to a preceding STATION definition.

In figure 3-1, the TU910 is designated as a controller for the four TU510 slave stations.

G18204
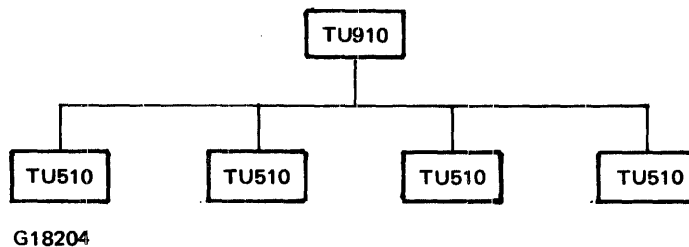
Figure 3-1. Controller-Slave Station Configuration

Example:

```
CONTROLLER = TRUE.
CONTROLLER = FALSE.
CONTROLLER = SMITH.
```

DEFAULT Statement

The DEFAULT statement causes the station attributes declared in the STATION DEFAULT definition named by <symbolic name> to be added to the station attributes. If any of the station attributes are explicitly defined in the Station definition and are also present in the STATION DEFAULT definition, the attribute in the Station definition supercedes the attribute in the STATION DEFAULT definition.

Syntax:

—— DEFAULT = <symbolic name > · ——————————————————|

Semantics:

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

Example:

     DEFAULT = REMOTEODT.

FREQUENCY Statement

The FREQUENCY statement is optional and specifies a priority for input and output for use by the station in the line Control procedure or for use as a possible classification of stations by an MCS.

Syntax:

```
——FREQUENCY = <integer1>————————————————————————. ———————————————————————|
                          └————, <integer2>————————┘
```
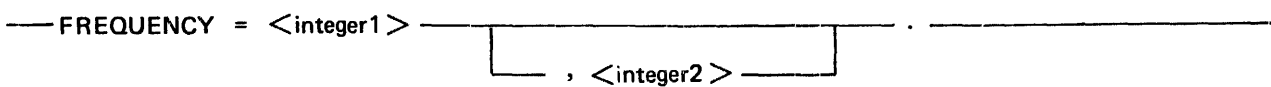
Semantics:

<integer1>
    <integer1> specifies the input priority and its value can be in the range from 0 to 255 inclusive. The default value is zero.

<integer2>
    <integer2> specifies the output priority and its value can be in the range from 0 to 255 inclusive. The default value is zero.

The FREQUENCY statement is currently used by the SMCS program to specify whether a station is a workstation, remote ODT, a controller station, or both a controller station and remote ODT as shown in table 3-5.

Table 3-5. SMCS Station Type Frequency Values

Station Type	Input FREQUENCY	Output FREQUENCY
Workstation	0 - 249	0 - 249
Controller	250 - 255	0 - 249
Remote ODT	0 - 249	250 - 255
Controller/ODT	250 - 255	250 - 255

Example:

    FREQUENCY = 250, 250.
    FREQUENCY = 0.

LOGICALACK Statement

A TERMINATE LOGICALACK statement in an input Request procedure allows an input Request procedure to be interrupted.

Syntax:

```
── LOGICALACK = ──┬── FALSE ──┬── . ──────────────────────────┤
                  └── TRUE ───┘
```
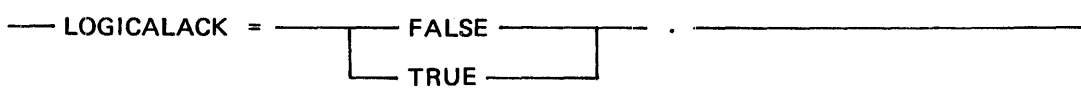
Semantics:

FALSE
> If LOGICALACK = FALSE is specified and the Request procedure does a TERMINATE LOGICALACK operation, the input message is queued for the appropriate application program and the Request procedure continues.

TRUE
> If TRUE is specified in the LOGICALACK statement and the Request procedure does a TERMINATE LOGICALACK operation, the input message received on a RECEIVE TEXT operation is queued for the MCS. The MCS must reply to the input message with a LOGICALACK__REPLY message before the Request procedure continues.

MYUSE Statement

The MYUSE statement indicates the functional use of a station: INPUT, OUTPUT, or INPUT and OUTPUT. MYUSE must be defined for every STATION definition.

Syntax:

```
— MYUSE = ———┬— INPUT ————————┬— . ———————————————|
             ├— OUTPUT ———————┤
             └— INPUT , OUTPUT —┘
```
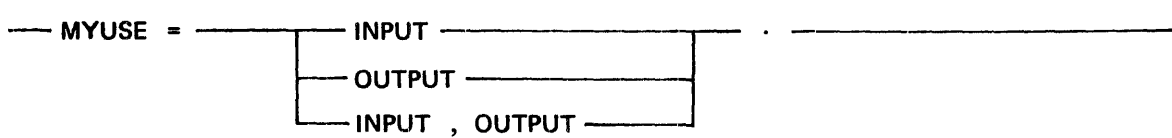
Example:

```
MYUSE = INPUT, OUTPUT.
MYUSE = INPUT.
MYUSE = OUTPUT.
```

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:

```
—— PHONE  =  <string> . ——————————————————————————|
```

Semantics:

<string>
    The <string> can consist of from 1 to 20 EBCDIC characters. The string must consist of only the characters 0 through 9, *, #, E, S, and –. Characters 0 through 9, *, and # are dialed. The letter E represents the end of number code. The letter S represents a wait for second dial tone. The hyphen (-) character is changed to a time-fill character @FF@. Time-fill characters cause the dial hardware to pause for three seconds in the dialing sequence.

Example:

    PHONE = "9S1-800-5551234".

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:

```
── READY = ──────┬── FALSE ──────┬── . ──────────────────────────┤
                 └── TRUE ───────┘
```
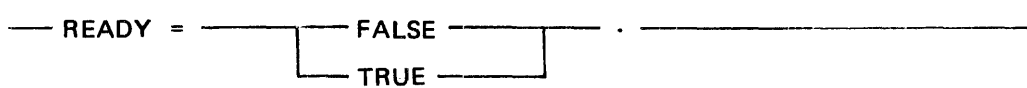
Example:

    READY = FALSE.

RETRY Statement

The RETRY statement provides the initial retry count for the station. The RETRY statement counts the receive and transmit attempts in Request procedures. Refer to B 1000 Systems NDL/LIBRARY Functional Description Manual for a description of how the RETRY count is used in each Request procedure.

Syntax:

—— RETRY = <integer> · ——————————————————|

Semantics:

<integer>
    <integer> must be in the range from 0 to 255 inclusive.

Pragmatics:

When using the CANDEPOLTD Request procedure, a value of 255 for the RETRY statement is considered infinite by the CANDEPOLTD Request procedure.

Example:

    RETRY = 25.
    RETRY = 255.

TALLY Statement

The TALLY statement causes an initial value to be put into the specified station tally at the network controller beginning of job. Only TALLY[0], TALLY[1], TALLY[2] can be initialized by the TALLY statement.

Syntax:

```
——TALLY  [ ——┬—— 0 ——┬—— ]  =  <integer> · ————————————————————|
              ├—— 1 ——┤
              └—— 2 ——┘
```

Semantics:

<integer>
   <integer> must be in the range from 0 to 255 inclusive in a TALLY statement. The default value is zero.

Pragmatics:

For the CANDE Request procedures, TALLY[1] can be set to condition the reporting of a station becoming on line and off line to the controlling remote file as follows:

TALLY[1] = 0
   On line is not reported; off line is reported by the TERMINATE ERROR mechanism. The TERMINATE ERROR mechanism is described in the *B 1000 Systems Network Definition Language (NDL) Language Manual.*

TALLY[1] = 1
   On line is reported by a zero length data message with TALLY[0] set a value of 10.

TALLY[1] = 2
   On line is not reported; off line is reported by a zero length data message with TALLY[0] set to a value of 11.

TALLY[1] = 3
   On line and off line are reported by a zero length data message with TALLY[0] set to a value of 10 for on line and 11 for off line.
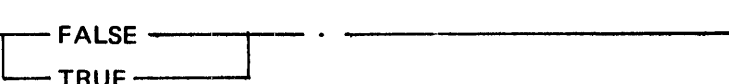
Example:

   TALLY [2] = 125.

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:

```
── TOG  [  <integer>  ]  =  ──┬── FALSE ──┬──  .  ──────────────────────┤
                              └── TRUE ───┘
```

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 timeouts 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 declared. The TERMINAL statement must be included in the Station definition. More than one station definition can reference the same Terminal definition.

Syntax:

```
── TERMINAL  =  <symbolic name>  ·  ─────────────────────────┤
```

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:
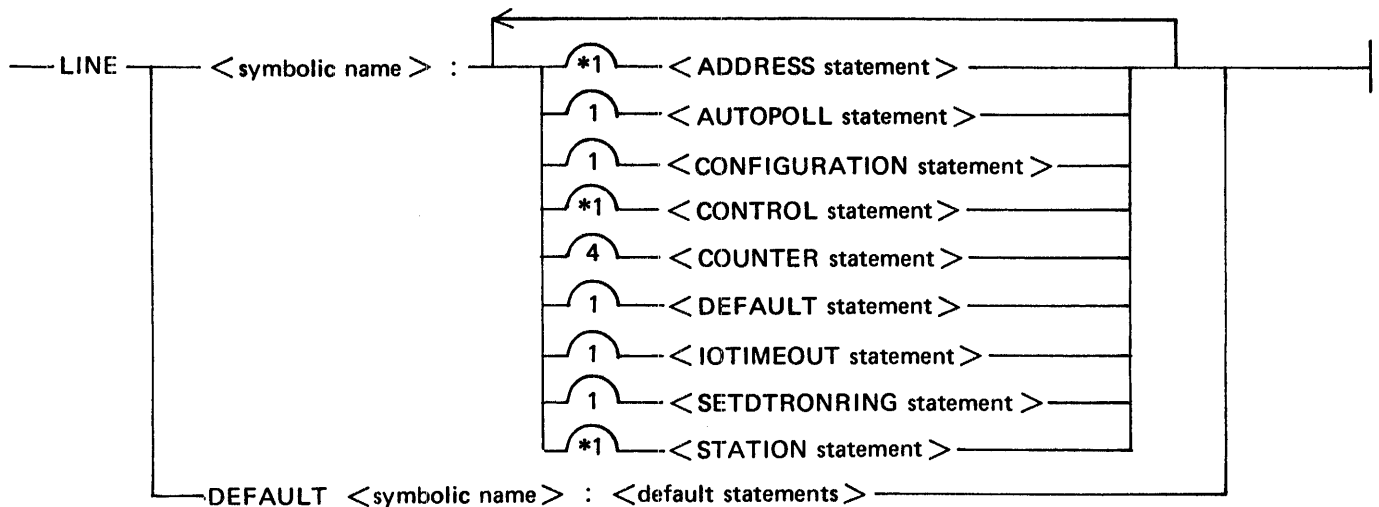
Semantics:

<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 1990 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 ODT devices on a system without a SPO-control be located on adapter 0 of the lowest-numbered 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:
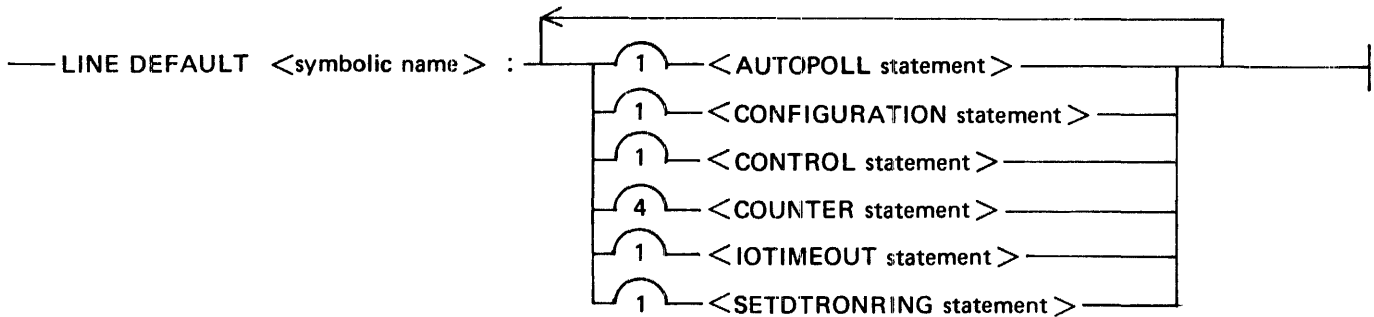
Semantics:

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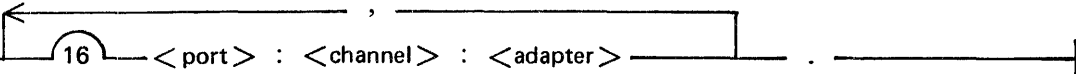
Pragmatics:

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 that occurs in a DEFAULT statement of a line definition must be preceded by that definition.

ADDRESS Statement

The ADDRESS statement is used to specify the port number, the channel number, and the adapter number to which a data communications line is connected. A line becomes active when either of the following conditions is present: 1) when a remote file is opened that contains stations declared on that line or 2) when a COBOL74 program executes a SEND command to a station declared on that line. Two active lines cannot have the same address.

Syntax:

Semantics:

<port>
  <port> is the number of the port being used to access memory for the line.

<channel>
  <channel> is the number of the I/O channel used by the line.

<adapter>
  <adapter> is the number of the adapter to which the line is connected.

Pragmatics:

If the ADDRESS statement is used for the special ODTLINE, < channel> and <adapter> must be zero and line groups are not allowed.

Example:

    ADDRESS = 1:0:1.

In a switched-line environment, there can be no predetermined hardware connections as is the usual case with leased lines. Several different station configurations can be alternately connected to several different line adapters. This is accomplished through the use of line groups. A line group is a set of addresses possessing common line attributes that are grouped together under one Line definition. A line group has a station list that comprises all of the stations that can dial in on any of the lines.

Example:

    LINE L1:
        ADDRESS = 1:0:0, 1:0:1, 1:0:2.

Line groups are not necessary if a station can dial in on one line only.

AUTOPOLL Statement

The AUTOPOLL statement is used to define the maximum size in characters of an autopoll buffer. This buffer contains the autopoll string for the line. If the line Control procedure performs no autopoll operation, the AUTOPOLL statement is not required. For lines using the standard CANDETDCTL, AUTOPOLCTL, or AUTODYNCTL line Control procedures, this statement is optional. If present, the value specified is checked against the calculated minimum value and a syntax error is emitted if the value specified is too small. If the AUTOPOLL statement is not present, the autopoll buffer is set to a calculated minimum.

Syntax:

— AUTOPOLL = <integer> . ——————————————————|

Semantics:

<integer>
    <integer> is the maximum size in characters of the autopoll buffer.

Pragmatics:

For the CANDETDCTL Control procedure, the calculated minimum is set to:

    10 * <number of stations on the line>

For the AUTOPOLCTL Control procedure, the calculated minimum is set to:

    5 * <number of stations on the line>

For the AUTODYNCTL Control procedure, the calculated minimum is set to:

    5 * <number of stations on the line> + 5

Example:

    AUTOPOLL = 60.

CONFIGURATION Statement

The CONFIGURATION statement specifies the name of the line configuration to be loaded into a Quad Line Adapter-1 or 2. This statement specifies the configuration name that the network controller downloads into the adapter. If the CONFIGURATION statement is omitted, a configuration named DEFAULT is used. A CONFIGURATION statement specified for a line adapter other than a Quad Line Adapter-1 or 2 has no effect. For more information, refer to the headings entitled Multiline Control-4, Firmware and Configuration Loading, and Line Configuration in section 6.

Syntax:

──── CONFIGURATION = <symbolic name > . ─────────────────────────┤

Semantics:

<symbolic name>
   <symbolic name> is one of the line configuration names defined in the configurations file. A list of the standard line configurations is provided in section 6 under the heading entitled Standard Line Configurations.

Example:

   CONFIGURATION = TDI192.

CONTROL Statement

The CONTROL statement is required and specifies the line Control procedure to be used with this line.

Syntax:

```
── CONTROL = <identifier> . ──────────────────────┤
```

Semantics:

<identifier>
    <identifier> is the name of the Control procedure.

Example:

    CONTROL = CANDETDCTL.

COUNTER Statement

The COUNTER statement specifies an initial value for a line counter at network controller beginning of job. Only line counters with an index of 0, 1, 2, or 3 can be initialized in this manner.

Syntax:

```
── COUNTER  [ ──────┬──── 0 ────┬──── ]  =  < signed integer > .  ─────────────────────┤
                    ├──── 1 ────┤
                    ├──── 2 ────┤
                    └──── 3 ────┘
```

Semantics:

<signed integer>
    The value of the <signed integer> can be in the range from -8388607 to $+8388607$ inclusive. If the sign is omitted, the value is assumed to be positive. The default value is zero.

Pragmatics:

For the CANDETDCTL Control procedure, the LINE COUNTERS are used to replace the SOFTIMEPOL, RETRYDELAY, and IDLEDELAY defines as follows:

COUNTER [0]
    SOFTIMEPOL. COUNTER[0] controls the soft timeout feature of the CANDETDCTL Control procedure and must be increased for certain types of modems. The value for COUNTER[0] is in tenths of a second.

COUNTER [1]
    RETRYDELAY. Off line stations are retried every COUNTER[1] tenths of a second.

COUNTER [2]
    IDLEDELAY. If all stations are off line, the line is idled for COUNTER[2] tenths of a second.

COUNTER [3]
    AUTOPOLLINTERRUPT. The AUTOPOLL I/O operation is interrupted every COUNTER[3] tenths of a second.

Example:

```
COUNTER [0] = 4.        % Cancel single-poll after 4/10 seconds.
COUNTER [1] = 300.      % Retry off line stations every 30 seconds.
COUNTER [2] = 600.      % Idle line for 60 seconds if all
                        % stations off line.
COUNTER [3] = 200.      % Interrupt autopoll every 20 seconds.
```

DEFAULT Statement

The DEFAULT statement causes the line attributes declared in the LINE DEFAULT definition named by <symbolic name> to be added to the line attributes. If any of the line attributes are explicitly defined in the Line definition and are also present in the LINE DEFAULT definition, the attribute in the Line definition supercedes the attribute in the LINE DEFAULT definition.

Syntax:

```
—— DEFAULT = <symbolic name> . ————————————————|
```

Semantics:

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

Example:

    DEFAULT = TDILINE.

IOTIMEOUT Statement

The IOTIMEOUT statement defines the time that the network controller waits for an I/O operation to complete before assuming the line is "hung" or not responding properly. The IOTIMEOUT statement is only effective on I/O operations that should complete in a finite amount of time. Refer to Network Controller Program Switches in section 5 of this manual for the action taken when a line appears to be hung.

Syntax:

```
—— IOTIMEOUT  =  <integer> . ———————————————————|
```

Semantics:

<integer>
> <integer> is the time in tenths of a second. A default value of 300 (30 seconds) is assumed if the IOTIMEOUT statement is omitted.

Pragmatics:

The IOTIMEOUT value should be set large enough to allow I/O operations to complete without an erroneous indication of a line hang, but small enough to recognize a hung line without an excessive delay. The amount of time for an I/O operation to complete is dependent upon the time to send the message, receive the reply, and the INITIAL-TIMEOUT value for the line.

The IOTIMEOUT value should be set based upon the line buffersize. Note that the line discipline might add some characters to the message, for example, the multipoint poll/select protocol adds about eight characters. The time to send a message is based on the size of the message and the speed of the line. The time can be calculated by multiplying the size of the message in characters by the number of bits in a character (usually 10 for ASYNC, TDI or TTY disciplines or eight for SYNC disciplines) and dividing by the speed of the line in bits per second (baud rate). For example, an 8191 character line buffer on a 1200 baud ASYNC line would take 68.4 seconds (8191 + 6) characters * 10 bits/character / 1200 bit/second). The INITIALIOTIMEOUT value can be as large as 65 seconds but is typically 5 seconds. The INITIALIOTIMEOUT value can be set to several times the expected time to complete an I/O operation without an excessive delay in recognizing a hung line. The IOTIMEOUT value for the line in the example above should be set to 750 (75.0 seconds).

The longer an I/O operation takes, the greater the chances of a transmission error. Reducing the size of messages sent on slow lines is recommended instead of increasing the IOTIMEOUT value to more than 600 (60.0 seconds).
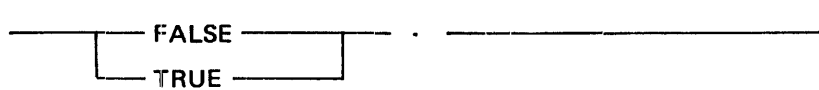
Example:

    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:

```
── SETDTRONRING = ──┬── FALSE ──┬── . ─────────────────────┤
                    └── TRUE ───┘
```
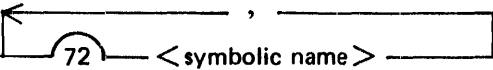
Example:

    SETDTRONRING = FALSE.

STATION Statement

The STATION statement is required and is used to identify the stations on the line.

Syntax:

```
                              ┌─────────── , ───────────┐
── STATION = ────────┌───────┴─⟨72⟩── <symbolic name> ──┴──── . ──────────────────────┤
```

Semantics:

<symbolic name>
> <symbolic name> is the name of a station.

Pragmatics:

The <symbolic name> station must have been defined in the Station Section. A station can be associated with only one line and cannot be listed twice for the same line.

Example:

```
STATION = SMITH,
          JONES,
          ROOM273,
          EXT11.
```

FILE SECTION

The File Section is used to associate stations that are part of an NDL-generated network controller with the remote files of application programs. File definitions allow sets of stations to be grouped into logical entities called files. The names of stations that are members of the file being described are listed in the File definition. A station can be associated with any number of files.

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.

Syntax:
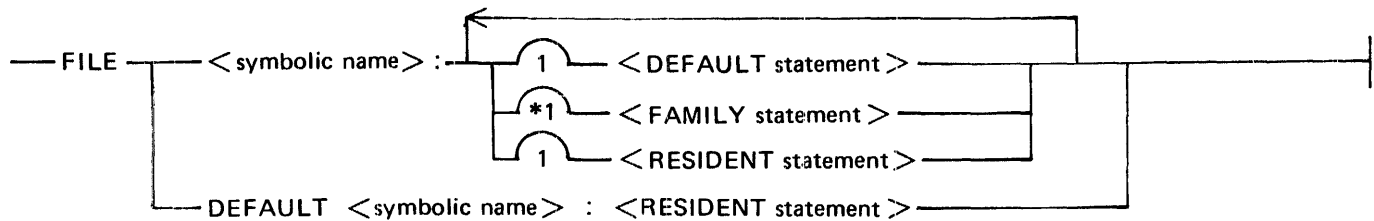
Semantics:

\<symbolic name\>
   \<symbolic name\> is the name of the File definition or the FILE DEFAULT definition.

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.

Syntax:

```
——FILE DEFAULT  <symbolic name>  :  RESIDENT  = ————┬—— DISK ————————┬—— · ————————————|
                                                    └—— MEMORY ———————┘
```

Semantics:

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

DEFAULT Statement

The DEFAULT statement causes the file attribute specified in the FILE DEFAULT definition named by <symbolic name> to be added to the file attributes. If the file attribute is explicitly defined in the File definition and is also present in the FILE DEFAULT definition, the attribute in the File definition supercedes the attribute in the FILE DEFAULT definition.

Syntax:

```
—DEFAULT  =  <symbolic name > . ─────────────────────────┤
```

Semantics:

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

Example:

    DEFAULT = FILEDEF.

FAMILY Statement

The names of the stations that are members of the file being described are listed in the FAMILY statement. A station can be associated with any number of files.

Syntax:

```
── FAMILY = ──┬──── ALL ───────┬── . ──────────────────────┤
              ├──── DUMMY ──────┤
              │   ┌───── , ─────┐ │
              └───┴─< symbolic name >─┴──┘
```
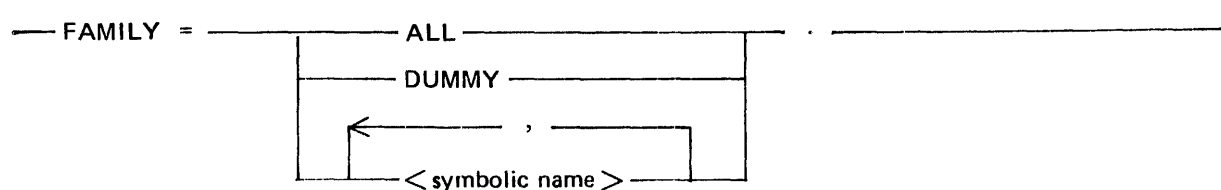
Semantics:

ALL
   IF FAMILY = ALL is specified, all stations declared in the Station Section, except the predefined stations SYSTEMODT and RDSTATION, are included in the remote file.

DUMMY
   If FAMILY = DUMMY is specified, no stations are attached when the file is opened. This statement is intended to be used by an MCS that assigns stations to be associated with the file at file open time, or to attach stations at a later time up to the maximum number of stations declared for that remote file in the application program. The end result obtained from declaring FAMILY = DUMMY is identical to that achieved by having the application program open a remote file that is not declared in the File Section. However, an MCS must be present, and must assign the stations to be associated with the file at open time. The MCS must have zip-executed the program opening this file.

<symbolic name>
   <symbolic name> is name of a station.

Example:

```
FAMILY = SMITH,
         JONES,
         ROOM273,
         EXT11.
```

RESIDENT Statement

The RESIDENT statement is optional. RESIDENT = DISK indicates that the program that opened this remote file can be rolled out to disk while waiting for input for this file. If the RESIDENT statement is omitted, the default RESIDENT = MEMORY is used to indicate that the program is not to be rolled out to disk when reading this file.

Syntax:

```
──RESIDENT = ──┬── DISK ──────┬── . ──────────────────────┤
               └── MEMORY ─────┘
```
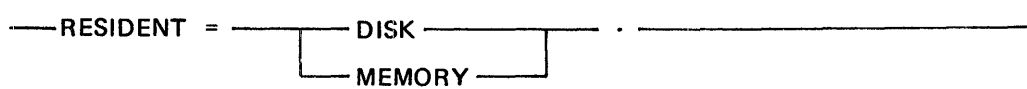
NETWORK CONTROLLER EXAMPLE

The following is an example of a network controller for use with SMCS, SYCOM, a Remote Print program, and RJE3780. For specific information relating to each program, refer to the appropriate reference manual.

Network Controller Name

The name of the network controller program is specified by the user.

Declaration Section

The following is an example of the Declaration Section for a network controller.

Example:

```
DECLARATION:
    MAX  BUFFERS          =    5.
    MAX  MESSAGES         =    20.
    MAX  MESSAGE  SIZE    =    4088.    % Maximum  possible  for  SYCOM.  The
                                       % SYCOM  BUFFER  parameter  can  be
                                       % up  to  this  size.
MAX  FILES                =    40.
```

Terminal Section

The various terminals used in the network are described in the Terminal Section.

SMCS Terminal Declaration

```
TERMINAL  MT985:
    TRANSMISSION      =    1.
    ADDRESS           =    2.
    TYPE              =    46.
    BUFFERSIZE        =    2250.
    REQUEST           =    CANDEPOLTD:  RECEIVE,
                           CANDEFSLTD:  TRANSMIT.
```

SMCS COBOL74 Terminal Declaration

```
TERMINAL  C74MT985:
    TRANSMISSION      =    1.
    ADDRESS           =    2.
    TYPE              =    46.
    BUFFERSIZE        =    2250.
    REQUEST           =    CANDEPOLTD:  RECEIVE,
                           C74CANDFSL:  TRANSMIT.
```

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.
```

IBM 3270 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.
```

IBM 3270 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.
```

IBM 3270 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.
```

NETWORK CONTROLLER EXAMPLE

```
DECLARATION:
    MAX BUFFERS        = 5.
    MAX MESSAGES       = 20.
    MAX MESSAGE SIZE   = 4098.          % Maximum possible for SYCOM. The
                                        % SYCOM BUFFER parameter can be
                                        % up to this value.
    MAX FILES          = 40.


%%% Terminal Section

TERMINAL MT985:
    TRANSMISSION = 1.
    ADDRESS      = 2.
    TYPE         = 46.
    BUFFERSIZE   = 2250.
    REQUEST      = CANDEPOLTD: RECEIVE,
                   CANDEFSLTD: TRANSMIT.


TERMINAL C74MT985:
    TRANSMISSION = 1.
    ADDRESS      = 2.
    TYPE         = 46.
    BUFFERSIZE   = 2250.
    REQUEST      = CANDEPOLTD: RECEIVE,
                   C74CANDFSL: TRANSMIT.


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


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

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


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


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


%%% Station Section

        % >> SMCS Declarations <<
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             = REMOTEUDT.
    ADDRESS             = "S1".
STATION JONES:
    TERMINAL            = C74MT985.
    DEFAULT             = REMOTEUDT.
    ADDRESS             = "S2".
STATION ROOM273:
    TERMINAL            = MT985.
    DEFAULT             = CTRLFSTATION.
    ADDRESS             = "S3".
STATION EXT11:
    TERMINAL            = MT985.
    DEFAULT             = CTRLROOT.
    ADDRESS             = "S4".


        % >> SYCOM Control Declarations <<
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             = "AE" , "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 DEFAULT APDEF:
    RETRY          = 10.
    FREQUENCY      = 0,0.
    MYUSE          = INPUT,OUTPUT.
    TERMINAL       = AP310.
STATION OFFICE:
    TERMINAL       = AP310.
    DEFAULT        = APDEF.
    ADDRESS        = "AP".


        % >> RJE3780 Stations <<
STATION DEFAULT RJE3780DEF:
    MYUSE          = INPUT, OUTPUT.
    RETRY          = 15.
    CONTROLLER     = FALSE.
    TERMINAL       = RJE3780.
STATION RJE3780ST1:
    DEFAULT        = RJE3780DEF.
STATION RJE3780ST2:
    DEFAULT        = RJE3780DEF.
```

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


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.


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

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

LINE BSC3270LINE:
    ADDRESS            = 2:0:4.
    AUTOPOLL           = 22.
    CONTROL            = BISYNC3270CONTROL.
    STATION            = C1D1,
                         C1D2,
                         C2D1,
                         C2D2.


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

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

FILE APFILE:                                  % For Remote Printer
    RESIDENT           = MEMORY.
    FAMILY             = OFFICE.

FILE RJE3780Q:                                % For RJE3780
    RESIDENT           = MEMORY.
    FAMILY             = RJE3780ST1,
                         RJE3780ST2.
```

SECTION 4
NETWORK CONTROLLER GENERATION

GENERAL

The network controller is generated by using the Network Definition Language (NDL) compiler. The NDL compiler takes the symbolic specifications described in section 3, Network Controller Symbolic Specifications, and binds them with the SYSTEM/CONTROLLER basic network controller to create a new network controller.

COMPILER INPUT FILES

The primary symbolic input file to the B 1000 NDL Compiler is a card file with the internal name CARD. This file is required for compilation and can be file-equated to another file assigned to a different hardware device.

The secondary symbolic input file to the NDL compiler is a disk file with the internal name SOURCE. This file is optional and is normally a disk file, although it can be assigned to a different device. This file is accepted as input when the $MERGE compiler option is specified. The card images in this file (SOURCE) are merged with the card images in the primary file (CARD) according to sequence number.

COMPILER OUTPUT FILES

The NDL compiler processes the input data and produces several output files, one being the line printer listing of the network controller symbolic code. The internal name of this file is LINE. The name of the compiled network controller is printed on the first line of the printer listing. The code segment dictionaries for the Request and Control sections are printed at the end of the listing: Segment 0 contains a CASE statement driven by the value of the Request or Control procedure index. Segments 1 through n are the compiled Request and Control procedures, with one segment per Request or Control procedure. A summary of the Request and Control procedures, the terminals, stations, lines, and files declared follows.

When the $MERGE and $NEW options are specified to produce a new symbolic file, the resulting output file has the internal and external name NEWSOURCE.

NDL CONTROL CARDS

The following control statements must be specified to compile the network controller:

```
        ?    COMPILE USERNC NDL LIBRARY;
        ?    DATA CARD
             <network controller symbolic code>
        ?    END

 or
        ?    COMPILE USERNC NDL LIBRARY;
             FILE CARD DISK TITLE NC/SOURCE ON USER;
```

In the above examples, USERNC was chosen as the name of the network controller object file.

If COMPILE FOR SYNTAX is specified, the only NDL compiler output is a printer listing of the symbolic code which includes any syntax errors.

If the user has non-standard Request and Control procedures in the network controller and no changes are to be made to these Request and Control procedures, compile time can be reduced by using the old network controller in place of SYSTEM/CONTROLLER as the skeletal network controller. In this case, the following control statements are used:

```
?    COMPILE NEWNC WITH NDL TO LIBRARY;
?    FILE MACRO TITLE OLDNC;
?    DATA CARD
         <network controller symbolic code>
?    END
```

In this example, OLDNC is the name of the old network controller and NEWNC is the network controller that is to be created using OLDNC as the skeletal network controller.

NDL SYMBOLIC CODE STATEMENT FORMAT

The symbolic records are contained in record positions 1 through 72 inclusive, and can contain NDL statements or comments in free-form format. Identifiers cannot be split across records; an identifier must be entirely contained on a record.

Comments

A percent sign (%) in any record position from 1 through 72 inclusive results in the compiler skipping to position 1 of the next symbolic record. The programmer can enter any desired remarks or comment following the percent sign.

Example:

STATION AARDVARK: % Computer room terminal

Percent signs are considered to be included in the NDL symbolic code if they are within an EBCDIC string, as in the following example.

Example:

ADDRESS = "1%".

If the "%" is in position 1, the remaining 71 positions of the free-form field are treated as a comment.

Example:

%THIS ENTIRE CARD CONTAINS A COMMENT

Sequence Numbers

Record positions 73 through 80 inclusive can be used for sequence numbers. The use of sequence numbers is optional.

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

This option causes the compiler to list on the LINE file the object code generated for the REQUEST procedure statements and the CONTROL procedure statements. The CODE option does not list the code in the precompiled REQUESTS and CONTROLS.

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 set. 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 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

DATA COMMUNICATIONS ODT

The SYSTEM/ODT program can operate with either the multiline control or the ODT control. This feature of an ODT running through the multiline control is referred to as a data communication ODT.

The data communication ODT implementation is mandatory for the B 1990 systems and optional for all other B 1000 systems.

To run a data communication ODT, it is required that an MT 985/MT 983 terminal be configured to run two pages of 1920 characters each.

For user programs that utilize forms mode to function correctly on a data communication ODT, the exit forms mode sequence should precede each message.

DIFFERENCES BETWEEN DATA COMMUNICATION AND ODT-CONTROL ODT DEVICE

The data communication ODT behaves functionally the same as an ODT device connected to an ODT control with the following exceptions:

1. All ODT output is directed to page 2 of the ODT. Page 1 is used for register information on the B 1990 systems when the system is interrupted or halted.

2. The SPCFY key is used to change pages on the data communication ODT. The CTRL → (right arrow) key sequence should never be used to change pages on a data communication ODT; unpredictable results can occur as a result of using the CTRL → (right arrow) key sequence to change pages. At network controller beginning of job, the SPCFY key must be depressed three times to initially change pages; after the initial change of pages, the SPCFY key need only be depressed once.

Using the Data Communications ODT as a Terminal

As a consequence of the implementation of the data communication ODT, it is now possible to run user programs on the ODT with the following restrictions:

1. Only page 1 of the terminal is available for use.

2. The SPCFY key cannot be used by the program.

3. If forms mode is to be used, the Mark 3.8 (or later) firmware in the MT 985/MT 983 terminal must be used. In addition, firmware bit 1 of location A0 must be set.

NOTE
Due to changes in the Mark 3.8 firmware in the MT 985/MT 983 terminal, an explicit exit forms mode sequence (4"27" X) must be sent for every message intended to clear a screen that is already in forms mode. Prior to the Mark 3.8 firmware, or if firmware bit 1 of location A0 is reset, the receipt of a SOH character that precedes a data communication message takes the station out of forms mode.

The following procedures are used to run the CANDE program on the data communication ODT. For the purposes of this discussion, it is assumed that the SMCS program is the Message Control System (MCS) and that the SMCS program is currently running in the mix.

1. Attach the data communication ODT to the SMCS program using the SMCS ATTACH command as follows:

    <SMCS mix number> AX ATTACH SYSTEMODT

2. Change to page 1 of the data communication ODT by depressing the SPCFY key on the ODT keyboard. All user programs are run on page 1. When the data communication ODT is attached to a user program such as the CANDE program, all input from page 1 of the data communication ODT is directed to the user program and all input from page 2 is directed to the SYSTEM/ODT program. If the CTRL → (right arrow) or CTRL ← (left arrow) control sequences are used to change pages, the network controller might direct the input to the wrong program.

3. Log on to the SMCS program by entering the following.

    USER <usercode>/< password>

4. Sign on to the CANDE program by entering the following.

    SIGN ON CANDE

The operator can now edit a work file and perform other CANDE functions.

The operator can also execute programs that open a remote file that is file-equated to the system ODT as follows. The SYSTEM/QCONFIGURE program is used as an example:

1. Execute the SYSTEM/QCONFIGURE program as follows:

    EXECUTE SYSTEM/QCONFIGURE FILE REMOTE STATIONS SYSTEMODT;

2. Change to page 1 of the data communication ODT by depressing the SPCFY key on the ODT keyboard. All application programs are run on page 1.

3. Enter the appropriate specifications to the SYSTEM/QCONFIGURE program.

The above examples can be applied to any application program. If it is desired to enter ODT commands, the operator can switch to page 2 (using the SPCFY key) and enter the ODT command. All ODT output is directed to page 2.

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 a 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 statement in the network controller symbolic code.
1	Unconditionally inititate accumulation of statistics from beginning of job.
2	Do not initiate accumulation of statistics regardless of the STATISTICS statement 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
CONTROLS, LINE ADAPTERS, FIRMWARE,
and LINE CONFIGURATIONS

INTRODUCTION

The B 1000 product line supports Multiline Control-1, 2, and 4 and Single-Line Control-1, 2, 3A, and 3B. Each of these controls support a set of individual or built-in line adapters. Some of the adapters must be downloaded with firmware, or a configuration (set of parameters), or both before they can drive a data communications line.

MULTILINE CONTROL-1 AND 2 AND SINGLE-LINE CONTROL-1 AND 2

Multiline Control-1 and 2 and Single-Line Control-1 and 2 support individual hardware line adapters. Each type of hardware line adapter is specifically designed for one of the following disciplines:

- **Standard Direct Asynchronous (TDI)**

- Standard Burroughs Direct Interface (BDI)

- Teletype Asynchronous Data Set (TTY)

- Teletype Asynchronous Direct (TTY Current Loop)

- Standard Asynchronous (ASYNC)

- Standard Synchronous (SYNC)

- Binary Synchronous EBCDIC (BISYNC-E)

- Wideband Binary Synchronous (WIDEBAND)

- Burroughs Data Link Control (BDLC)

- Automatic Calling Unit (ACU)

Each adapter type has a unique set of options controlled by hardware straps.

Single-Line Control-1 supports both a line adapter and an Automatic Calling Unit (ACU) adapter. Single-Line Control-2 supports one hardware line adapter but not the ACU adapter.

SINGLE-LINE CONTROL-3A AND 3B

Single-Line Control-3A and 3B do not require individual line adapters. Both controls can be configured with hardware straps for various disciplines and other options.

Single-Line Control-3A disciplines are:

- Standard Direct Asynchronous (TDI)

- Teletype Asynchronous Data Set (TTY)

- Standard Asynchronous (ASYNC)

- Standard Synchronous (SYNC)

Single-Line Control-3B disciplines are:

- Binary Synchronous EBCDIC (BISYNC-E)

- Binary Synchronous ASCII (BISYNC-A)

Both controls support automatic calling with Burroughs TA 1203 and TA 2403 data sets. They do not support RS366 (801) Automatic Call Units.

MULTILINE CONTROL-4

The Multiline Control-4 supports up to four adapters. The adapters can be any of the following:

- Quad Line Adapter-1

- Quad Line Adapter-2

- Dual BDLC Adapter-1

Both of the quad line adapters can support up to 4 two-way alternate (half duplex) data communications lines. The Dual BDLC adapter can support 2 two-way alternate (half duplex) or 1 two-way simultaneous (full duplex) data communications lines.

Each line of both quad line adapters can be configured for various disciplines and other options. The line discipline and options are controlled by software. The discipline type and other parameters are stored in a line configuration. For more information refer to Line Configuration in this section.

The Quad Line Adapter-1 supports automatic calling with Burroughs TA 1203 and TA 2403 data sets. It does not support RS366 (801) Automatic Call Units. The Quad Line Adapter-2 supports automatic calling with up to four lines of Burroughs TA 1203 or TA 2403 data sets, and two lines with RS366 (801) Automatic Call Units. The Dual BDLC Line Adapter-1 supports up to two RS366 (801) Automatic Call Units. It does not support dialing with Burroughs TA 1203 or TA 2403 data sets.

The Quad Line Adapter-1 must be down loaded with firmware for the adapter and a line configuration for a specific line before it can drive that line. The Quad Line Adapter-2 must be down loaded with both firmware and a line configuration for a specific line before it can drive that line. Refer to Firmware and Configuration Loading, Firmware Files, and Line Configuration in this section for more information.

FIRMWARE AND CONFIGURATION LOADING

The Quad Line Adapter-1 and 2 must be down loaded with firmware and line configuration(s) before they can drive a line.

The firmware and line configuration(s) are loaded if one of the following occurs:

1. Line status equal to released.

    This condition occurs (1) after the system is Clear/Started, (2) in a Quad Line Adapter-1 when the first line (of the four) is used, and (3) in a Quad Line Adapter-2 when each line is used. This condition also occurs in Quad Line Adapter-2 when the line is released by an operator-specified NC RELEASE LINE command, the network controller goes to end of job (EOJ), or a line becomes inactive (the last program using it goes to EOJ).

    <center>NOTE</center>

    An NC QUIT command (with a prior change of the firmware, the configurations file, or the network controller) does not necessarily cause the firmware or line configuration(s) to be loaded to a Quad Line Adapter-1.

2. Operator specifies an NC RELOAD LINE.

    This operator command causes the firmware and line configuration to be reloaded in both Quad Line Adapter-1 and 2.

    <center>NOTE</center>

    Reloading a Quad Line Adapter-1 causes a configuration reload of all lines in use on the adapter. For a Quad Line Adapter-2, only the line specified is reloaded.

3. Error conditions.

    Parity error encountered in the firmware.

All line configurations accessed by data communications programs are referenced by the configuration name. The name of the configuration is defined within the program. For information about defining the configuration name in a network controller, refer to the CONFIGURATION statement in section 3. When a configuration is loaded, the configurations file is opened and searched for the required configuration name.

FIRMWARE FILES

The firmware for both Quad Line Adapter-1 and 2 is included in the software release. The files are used by the NDL compiler, network controller, and other programs that interface directly with the quad line adapters.

Quad Line Adapter-1 Firmware

The Quad Line Adapter-1 has one set of firmware for all four lines. This firmware can be configured for any of the allowable disciplines. The firmware for the Quad Line Adapter-1 is contained in the SYSTEM/ QFQUAD1 file. (The standard network controller, SYSTEM/CONTROLLER, also contains firmware for the Quad Line Adapter-1.)

The NDL compiler binds a copy of the firmware for Quad Line Adapter-1 into network controller code files to insure that a system with a data communications Operator Display Terminal (ODT) runs even if the SYSTEM/QFQUAD1 file is removed from the system disk. If the NDL compiler cannot access the SYSTEM/QFQUAD1 file on the system disk, it prints a warning message and attempts to bind in the firmware from the MACRO file (SYSTEM/CONTROLLER).

When firmware is loaded to a Quad Line Adapter-1, the network controller attempts to obtain the firmware from the file SYSTEM/QFQUAD1 on DISK (so that the latest firmware is used). If that file is inaccessible, the network controller attempts to load firmware from inside the code file of the network controller. If the firmware is missing, the line cannot be activated.

All other programs that interface directly with the Quad Line Adapter-1 are capable of loading the firmware and line configurations to the adapter. A similar routine is provided in each of these programs to do the loading and reloading when necessary. However, all of these programs, other than the network controller, obtain their firmware and line configurations from the standard system files SYSTEM/ QFQUAD1 and SYSTEM/QCONFIGS.

Quad Line Adapter-2 Firmware

Each line in a Quad Line Adapter-2 can be individually loaded with firmware. Firmware for the Quad Line Adapter-2 is specifically designed for the discipline type and line address. The firmware for the Quad Line Adapter-2 is contained in a set of files that have a name starting with SYSTEM/QF. The next letter of the name is "E" for even line addresses, "O" for odd line addresses, and "H" for high speed (greater than 38,400 bits per second). (High-speed firmware runs on only odd line addresses.) The suffix of the name indicates the discipline type.

Suffix	Discipline
DIRECT	Standard Direct Asynchronous (TDI)
TTY	Teletype Data Set (TTY)
ASYNC	Standard Asynchronous (ASYNC)
SYNC	Standard Synchronous (SYNC)
BISYNCE	Binary Synchronous EBCDIC (BISYNC-E)
BISYNCA	Binary Synchronous ASCII (BISYNC-A)

High-speed firmware is available for only synchronous disciplines (SYNC, BISYNCE, and BISYNCA).

It is necessary for only those firmware files used by the required configurations to be kept on the system disk.

Copies of the firmware for the Quad Line Adapter-2 are not stored in the network controller code file.

When firmware is loaded to a Quad Line Adapter-2, the network controller dynamically determines the name of the firmware file from the line address and configuration parameters. If the firmware file is not accessible, the network controller uses a minimal set of firmware resident in the line adapter for the ODT line (adapter 0 and TDI discipline) or the RD line (adapter 3 and ASYNC discipline). Other lines cannot be activated.

LINE CONFIGURATION

The hardware line adapters and Single-Line Control-3A and 3B have a set of options (such as line speed and timeout values). These adapters and controls are configured by means of hardware straps. The adapters for the Multiline Control-4 have a set of options that are configured by software. The hardware strapping requirements for the quad line adapters has been minimized. A line configuration is a collection of parameters that must be loaded into a specific line of a Quad Line Adapter-1 or 2 to control the options in the adapter.

Standard Line Configurations

The SYSTEM/QCONFIGS file contains a collection of standard line configurations. The standard network controller, SYSTEM/CONTROLLER, also contains a copy of the standard line configurations. These files are capable of containing 60 line configurations. The standard line configurations are listed in table 6-1.

Table 6-1. Standard Line Configurations

Configuration Name	Line Protocol
DEFAULT	9600 Baud Two-Wire Direct Connect (TDI)
ODTDI96	9600 Baud Two-Wire Direct Connect (TDI)
TDI96	9600 Baud Two-Wire Direct Connect (TDI)
TDI192	19,200 Baud Two-Wire Direct Connect (TDI)
TDI384	38,400 Baud Two-Wire Direct Connect (TDI)
ASYNC12	1200 Baud Asynchronous Dedicated
ASYNC12SW	1200 Baud Asynchronous Switched
ASYNC12AC	1200 Baud Asynchronous Switched with 801 ACU
ASYNC24	2400 Baud Asynchronous Dedicated
ASYNC24SW	2400 Baud Asynchronous Switched
ASYNC48	4800 Baud Asynchronous Dedicated
ASYNC48SW	4800 Baud Asynchronous Switched
ASYNC96	9600 Baud Asynchronous Dedicated
ASYNC96SW	9600 Baud Asynchronous Switched
TTY3SW	300 Baud Teletype (TTY) Switched
TTY3AC	300 Baud Teletype (TTY) Switched with 801 ACU
TTY3OR12SW	300/1200 Baud Teletype (TTY) Switched (for use with 212 modem)
TTY3OR12AC	300/1200 Baud Teletype (TTY) Switched with 801 ACU (for use with 212 modem)
SYNC12	1200 Baud Synchronous Dedicated

Table 6-1. Standard Line Configurations (continued)

Configuration Name	Line Protocol
SYNC12SW	1200 Baud Synchronous Switched
SYNC12AC	1200 Baud Synchronous Switched with 801 ACU
SYNC24	2400 Baud Synchronous Dedicated
SYNC24SW	2400 Baud Synchronous Switched
SYNC24AC	2400 Baud Synchronous Switched with 801 ACU
SYNC48	4800 Baud Synchronous Dedicated
SYNC48SW	4800 Baud Synchronous Switched
SYNC48AC	4800 Baud Synchronous Switched with 801 ACU
SYNC96	9600 Baud Synchronous Dedicated
SYNC96SW	9600 Baud Synchronous Switched
SYNC96AC	9600 Baud Synchronous Switched with 801 ACU
SYNC192	19200 Baud Synchronous Dedicated
SYNC384	38400 Baud Synchronous Dedicated
SYNC576	57600 Baud Synchronous Dedicated
SYNC658	65828 Baud Synchronous Dedicated
BSCE24	2400 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE24SW	2400 Baud Binary Synchronous (BISYNC-E) Switched
BSCE24AC	2400 Baud Binary Synchronous (BISYNC-E) Switched with 801 ACU
BSCE48	4800 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE48SW	4800 Baud Binary Synchronous (BISYNC-E) Switched
BSCE48AC	4800 Baud Binary Synchronous (BISYNC-E) Switched with 801 ACU
BSCE96	9600 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE96SW	9600 Baud Binary Synchronous (BISYNC-E) Switched
BSCE96AC	9600 Baud Binary Synchronous (BISYNC-E) Switched with 801 ACU
BSCE192	19200 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE384	38400 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE576	57600 Baud Binary Synchronous (BISYNC-E) Dedicated
BSCE658	65828 Baud Binary Synchronous (BISYNC-E) Dedicated

Customizing Line Configurations

The SYSTEM/QCONFIGURE program customizes line configurations. Line configurations can be displayed, printed, changed, copied, added, or deleted. For information about the operation of the SYSTEM/QCONFIGURE program or on the parameters in a line configuration, refer to section 7, SYSTEM/QCONFIGURE.

NDL Compiler and Network Controller Use of Line Configurations

The NDL compiler binds a copy of the configurations file for the quad line adapters into network controller code files. If the NDL compiler cannot access the SYSTEM/QCONFIGS file on the system disk, it prints a warning message and attempts to bind in the configurations from the SYSTEM/CONTROLLER file.

The NDL compiler does not insure that the configuration name in CONFIGURATION statements actually exist in the configurations file. Configurations can be added or deleted after the network controller is created.

After a network controller is created, the configurations inside the network controller can be customized without changing the SYSTEM/QCONFIGS file. However, it is highly recommended that changes be made to the SYSTEM/QCONFIGS file and a new network controller be created. This insures that any network controller created after the customization contains the customized configurations.

It is possible to customize the configurations in the standard network controller, SYSTEM/CONTROLLER. However, the configurations requires customization again when a new SYSTEM/CONTROLLER is loaded to the system.

When a line configuration is loaded to a quad line adapter, the network controller obtains the configuration from inside the code file of the network controller. If the code file does not contain configurations, the network controller obtains the line configuration from the CONFIGS_FILE (SYSTEM/QCONFIGS). If the file is not accessible, the line cannot be activated.

Once the configurations are located, the network controller searches the configurations for one with a name matching the name in the CONFIGURATION statement for the line. If a matching configuration is not found, the line cannot be activated.

For more information about when a line configuration is loaded to a quad line adapter, refer to Firmware and Configuration Loading in this section.

SECTION 7
SYSTEM/QCONFIGURE PROGRAM

INTRODUCTION

The SYSTEM/QCONFIGURE program customizes line configurations for the quad line adapters. Line configurations in a configurations file (SYSTEM/QCONFIGS) or a network controller (compiled with NDL 12.0.002 or later) can be displayed, printed, changed, copied, added or deleted. Line configurations in a firmware file (SYSTEM/MLFIRMWARE) or a network controller (compiled with NDL prior to 12.0.002) can be converted to a new format configurations file (SYSTEM/QCONFIGS).

For more information about quad line adapters, loading configurations to a line adapter, or line configurations, refer to section 6, Controls, Line Adapters, Firmware, and Line Configurations.

When running the SYSTEM/QCONFIGURE program, input is made through a remote file.

PROGRAM EXECUTION

The SYSTEM/QCONFIGURE program can be executed from the Operator Display Terminal (ODT), through the SMCS program, or from any appropriate message control system (MCS).

If the SYSTEM/QCONFIGURE program is executed through the SMCS program, the following program control instruction is used:

EXECUTE SYSTEM/QCONFIGURE

The SYSTEM/QCONFIGURE program can be executed through the ODT, when the current network controller is SYSTEM/CONTROLLER, by using either of the following program control instruction:

EXECUTE SYSTEM/QCONFIGURE FILE REMOTE STATIONS SYSTEMODT;

or

EXECUTE SYSTEM/QCONFIGURE FILE REMOTE TITLE OPERATOR;

The SYSTEM/QCONFIGURE program operates only with Burroughs screen terminals such as a ET1100, MT 983/MT 985, or TD 830 device. The use of any other terminals can produce undefined results.

PROGRAM OPERATION

The following pages describe the commands entered to the SYSTEM/QCONFIGURE program using the remote file interface.

All input to the program may be either upper-case or lower-case letters. Note that the configurations file name and the name of a configuration must be entered as shown. These names are used as entered and are not changed to upper-case letters. (The name may be any series of non-blank letters, digits, or underscore characters and it must start with a letter.) Within the fields presented on the various screens, input may be in free-form format. Embedded blanks within tokens are not permitted.

The abbreviation for any function or parameter name is displayed (where appropriate) using capital letters. The entire name or the abbreviation (capital letters) may be entered. Refer to Function Definitions for more information.

In the examples of output from the SYSTEM/QCONFIGURE program that follow, the "[" and "]" characters represent the beginning and ending unprotected field characters.

Menu Interface

The SYSTEM/QCONFIGURE program provides operator interface functions through a series of menu screens. All screens have a header block that contains the screen name (upper left-hand corner), the program name (upper right-hand corner), an action line (used to enter a desired action such as escape to a prior screen or sign off), and the allowed actions. The menu interface and the menu screens are described on the following pages.

HOME Screen

The HOME screen specifies the configurations file to be changed. This file can be a network controller, a configurations file, or a SYSTEM/MLFIRMWARE file. This screen appears as follows:

```
                1        2        3        4        5        6        7        8
            1   0        0        0        0        0        0        0        0
          ...........................................................................
          I                                                                         I
      1   I HOME (CCNFIGURATION FILE SELECTION)                     SYSTEM/QCONFIGURE I
          I                                                                         I
      2   I                                                                         I
          I                                                                         I
      3   I ACTION [               ]                                                 I
          I                                                                         I
      4   I       EYE                                                                I
          I                                                                         I
      5   I  -------------------------------------------------------------------     I
          I                                                                         I
      6   I                                                                         I
          I                                                                         I
      7   I                                                                         I
          I                                                                         I
      8   I WELCOME TO SYSTEM/QCONFIGURE                                             I
          I                                                                         I
      9   I   Enter the full file title, including any usercode and                 I
          I                                                                         I
     10   I   familyname, of the configuration file you wish to                     I
          I                                                                         I
     11   I   interrogate or modify. The choices are                                I
          I                                                                         I
     12   I       COMBINED/NC_JAN27                    (file title from C slot)      I
          I                                                                         I
     13   I       COMBINED/NC_JAN09                    (file title from CX slot)     I
          I                                                                         I
     14   I       SYSTEM/QCONFIGS or                                                 I
          I                                                                         I
     15   I       SYSTEM/MLFIRMWARE                    (used by CONVERT only)        I
          I                                                                         I
     16   I   or enter any other configuration file title.                          I
          I                                                                         I
     17   I [                                          ]                            I
          I                                                                         I
     18   I                                                                         I
          I                                                                         I
     19   I Enter the appropriate file title and press the transmit key             I
          I                                                                         I
     20   I                                                                         I
          I                                                                         I
     21   I                                                                         I
          I                                                                         I
     22   I                                                                         I
          I                                                                         I
     23   I                                                                         I
          I                                                                         I
     24   I                                                                         I
          I                                                                         I
          ...........................................................................
```

Figure 7-1. HOME Screen

FUNCTION SELECT Screen

The FUNCTION SELECT selects a desired program function. The following is displayed at the terminal after a file has been selected:

```
           1         2         3         4         5         6         7         8
      1    0         0         0         0         0         0         C         0
     .-------------------------------------------------------------------------.
     I                                                                         I
  1  I FUNCTION SELECT                                     SYSTEM/QCONFIGURE    I
     I                                                                         I
  2  I                                                                         I
     I                                                                         I
  3  I ACTION (                   )                                            I
     I                                                                         I
  4  I         BYE HOme                                                        I
     I                                                                         I
  5  I                                                                         I
     I                                                                         I
  6  I  -----------------------------------------------------------------      I
     I                                                                         I
  7  I                                                                         I
     I                                                                         I
  8  I     CONvert   Convert the configuration file, contained in             I
     I                                                                         I
  9  I               the file specified in the HOME screen,                   I
     I                                                                         I
 10  I               to SYSTEM/QCONFIGS.                                       I
     I                                                                         I
 11  I     ADd       Add a new configuration                                  I
     I                                                                         I
 12  I     CHange    Change an existing corfiguration                         I
     I                                                                         I
 13  I     COpy      Copy an existing configuration                           I
     I                                                                         I
 14  I     DELete    Delete an existing corfiguration                         I
     I                                                                         I
 15  I     DISPlay   Display an existing ccnfiguration or                     I
     I                                                                         I
 16  I               all existing configurations                             I
     I                                                                         I
 17  I     PRint     Print an existing configuration or                       I
     I                                                                         I
 18  I               all existing configurations                             I
     I                                                                         I
 19  I               (        )                                                I
     I                                                                         I
 20  I                                                                         I
     I                                                                         I
 21  I                                                                         I
     I                                                                         I
 22  I                                                                         I
     I                                                                         I
 23  I                                                                         I
     I                                                                         I
 24  I                                                                         I
     .-------------------------------------------------------------------------.
```

Figure 7-2. FUNCTION SELECT Screen

Function Definitions

To select the desired program function, enter the name of the function desired or the abbreviation and press the transmit key.

A description of each function follows:

CONVert
   The CONVERT function reformats the existing configurations in the specified file and creates a new SYSTEM/QCONFIGS file.

ADD
   The ADD function creates a new line configuration.

CHange
   The CHANGE function modifies an existing line configuration.

COPY
   The COPY function specifies an existing line configuration to be presented for subsequent change. The line configuration is presented with the name field blank, and the operator must provide a unique name prior to transmitting the configuration. After transmitting the configuration, the program treats it as an ADD.

DELete
   The DELETE function removes an existing line configuration.

DISPlay
   The DISPLAY function displays an existing line configuration, or configurations, on the terminal.

PRint
   The PRINT function writes an existing line configuration, or configurations, on the line printer.

CONFIGURATION NAME Screen

The CONFIGURATION NAME screen selects a single configuration or all configurations. The menu includes a display of up to 50 configuration names from the specified configurations file. If more than 50 configurations exist, the operator can enter NEXT to display the next 50 configurations until all names are displayed.

The following is displayed when required at the terminal after a function has been selected.

```
            1       2       3       4       5       6       7       8
        1   0       0       0       0       0       0       0       0
       ----------------------------------------------------------------------
  1  I  CONFIGURATION NAME DISPLAY FOR CHANGE                 SYSTEM/QCONFIGURE  I
  2  I                                                                          I
  3  I  ACTION [            ]                                                   I
  4  I        EYE HOME FUNCTION                                                 I
  5  I  ------------------------------------------------------------------------- I
  6  I                                                                          I
  7  I     DEFAULT        OUTOI96       TCI96         TDI192        TDI384       I
  8  I     ASYNC12        ASYNC12SW     ASYNC12AC     ASYNC24       ASYNC24SW    I
  9  I     ASYNC48        ASYNC48SW     ASYNC96       ASYNC96SW     TTY3SW       I
 10  I     TTY3AC         TTY3DR12SW    TTY3DR12AC    SYNC12        SYNC12SW     I
 11  I     SYNC12AC       SYNC24        SYNC24SW      SYNC24AC      SYNC48       I
 12  I     SYNC48SW       SYNC48AC      SYNC96        SYNC96SW      SYNC96AC     I
 13  I     SYNC192        SYNC384       SYNC576       SYNC656       BSCE24       I
 14  I     BSCE24SW       BSCE24AC      BSCE48        BSCE48SW      BSCE48AC     I
 15  I     BSCE96         BSCE96SW      BSCE96AC      BSCE192       BSCE384      I
 16  I     BSCE576        BSCE656                                                I
 17  I                                                                          I
 18  I  ENTER AN EXISTING CONFIGURATION NAME, NEW CONFIGURATION NAME,           I
 19  I  "ALL", OR "NEXT"(DISPLAY MORE EXISTING CONFIGURATIONS).                 I
 20  I     [          ]                                                          I
 21  I                                                                          I
 22  I                                                                          I
 23  I                                                                          I
 24  I                                                                          I
       ----------------------------------------------------------------------
```

Figure 7-3. CONFIGURATION NAME Screen

HELP Function

The ADD, CHANGE, DELETE, and DISPLAY screens provide a HELP function. Entering HELP or TEACH causes a display of the parameter names and abbreviations (in parentheses) that are allowed. Entering an invalid name causes the HELP or TEACH information to be displayed.

Help information for each of the parameters consists of the parameter definitions listed under Parameter Definitions in this section. The information is accessed from the SYSTEM/QPARMHELP file. If the file is not present at beginning of job (BOJ), any request for help information results in the following error message.

HELP FILE NOT PRESENT

NOTE
The file must be present at BOJ in order to be accessed.

```
           1       2       3       4       5       6       7       8
    1      0       0       0       0       0       0       0       0
    ....................................................................
 1 I  HELP FCR PARAMETER=HELP                                          I
 2 I                                                                   I
 3 I     The following parameter names or abbreviations (in parentheses) I
 4 I     car be entered in the HELP field.                             I
 5 I                                                                   I
 6 I     ACL-TIMECUT (ATI)             ACU-TYPE (AT)                   I
 7 I     ALTERNATE-BAUD-RATE (ABR)     BAUD-RATE (BR)                  I
 8 I     ECS                           BREAK-ENABLE (BE)               I
 9 I     CHARACTER-PARITY (CP)         CHARACTER-SIZE (CS)             I
10 I     CONTINUOUS-CARRIER (CC)       DATA-ENCODING (DE)              I
11 I     DISCIPLINE (DISC)             DPR-HOLD (DH)                   I
12 I     ENABLE-SE (ESB)               ENABLE-SI (ESI)                 I
13 I     ENABLE-SR-SF (ESRSF)          ENABLE-SS (ESS)                 I
14 I     EOT-DISCCNNECT (EO)           INITIAL-TIMEOUT (IT)            I
15 I     NAME (NAM)                    NEW-SIGNAL (NS)                 I
16 I     PAC-DISABLE (PAD)             RECEIVE-DELAY (RD)              I
17 I     SELECT-RATE (SR)              SELECT-STANDBY (SS)             I
18 I     STCP-BITS (SB)                SWITCHED (SW)                   I
19 I     TEXT-TIMECUT (TT)             TRANSMIT-DELAY (TD)             I
20 I     TRANSPARENCY (TR)                                             I
21 I                                                                   I
22 I                                                                   I
23 I  TRANSMIT "NEXT" FOR MORE HELP INFORMATION, ANYTHING ELSE TO RETURN (NEXT) I
24 I                                                                   I
    ....................................................................
```

Figure 7-4. HELP Function (Screen 1)

```
                1       2       3       4       5       6       7       8
          1     0       0       0       0       0       0       0       0
       .-------------------------------------------------------------------.
   1   | HELP FCR PARAMETER=SWITCHED                                        |
       |                                                                   |
   2   |                                                                   |
       |                                                                   |
   3   |       Specifies  whether  the  line  is switched or dedicated (leased).  |
       |                                                                   |
   4   |      A value cf "Y" specifies a switched  line  and  a  value  of  "N"  |
       |                                                                   |
   5   |      specifies a dedicated line.                                  |
       |                                                                   |
   6   |                                                                   |
       |                                                                   |
   7   |      Valid for all disciplines except TDI.                        |
       |                                                                   |
   8   |                                                                   |
       |                                                                   |
   9   |                                                                   |
       |                                                                   |
  10   | TRANSMIT ANYTHING TO RETURN [ ]                                   |
       |                                                                   |
  11   |                                                                   |
  12   |                                                                   |
  13   |                                                                   |
  14   |                                                                   |
  15   |                                                                   |
  16   |                                                                   |
  17   |                                                                   |
  18   |                                                                   |
  19   |                                                                   |
  20   |                                                                   |
  21   |                                                                   |
  22   |                                                                   |
  23   |                                                                   |
  24   |                                                                   |
       .-------------------------------------------------------------------.
```

Figure 7-5. HELP Function (Screen 2)

```
          1         2         3         4         5         6         7  -      8
 1        0         0         0         0         0         0         0         0
.--------------------------------------------------------------------------------.
 1 |  HELP FOR PARAMETER=HELP                                                      |
 2 |                                                                               |
 3 |  Information pertaining to the  definition of  the requested parameter        |
 4 |  and the  allowed value  will  be displayed  on one or more screens as        |
 5 |  required.  When multiple screens are required,  the user  can  transmit      |
 6 |  "NEXT" to receive the next screen of information.    Transmitting any-        |
 7 |  thing other than "NEXT" will cause a return to the original screen.           |
 8 |                                                                               |
 9 |                                                                               |
10 |  TRANSMIT ANYTHING TO RETURN [ ]                                              |
:1 |                                                                               |
:2 |                                                                               |
13 |                                                                               |
14 |                                                                               |
15 |                                                                               |
16 |                                                                               |
:7 |                                                                               |
18 |                                                                               |
19 |                                                                               |
20 |                                                                               |
21 |                                                                               |
22 |                                                                               |
23 |                                                                               |
24 |                                                                               |
.--------------------------------------------------------------------------------.
```

Figure 7-6. Example of Parameter HELP Screen

Examples of the screens produced by the ADD, CHANGE, COPY, DELETE, DISPLAY, and PRINT functions and an explanation of the various parameters are given below.

ADD Configuration Screen

The SYSTEM/QCONFIGURE program responds with the following screen when the ADD function is selected on the FUNCTION SELECT screen.

```
          1         2         3         4         5         6         7         8
:         0         0         0         0         0         0         0         0
  -------------------------------------------------------------------------------
1 | ADD (ENTRY TO FILE SYSTEM/QCONFIGS)                        SYSTEM/QCONFIGURE |
  |                                                                              |
2 |                                                                              |
3 | ACTION (              )                                                      |
4 |      EYE HOme FUNCtion                                                       |
5 | ---------------------------------------------------------------------------- |
  |                                                                              |
6 | NAMe ------------ (         )   DISCipline ----------- (         )           |
7 | SWitchec ----------------- ( )  Data-Lncoding ------------- (  )             |
8 | Baud-Rate ------------- (    )  Alternate-Baud-Rate ------- (   )            |
9 | Transmit-Delay -------- (    )  Receive-Delay ------------- (   )            |
10| Text-Timeout ---------- (    )  Iritial-Timeout ---------- (   )             |
11| Character-Parity ------- ( )    Stop-bits ----------------- (   )            |
12| Character-Size ----------- ( )  BCS --------------------- (   )              |
13| Continucus-Carrier -------- ( ) PAD-cisable ------------------- ( )          |
14| Break-Erable --------------- ( ) TRansparency ------------------ ( )         |
15| Eot-Disconnect ------------ ( ) New-Signal -------------------- ( )          |
16| Enable-SR-SF -------------- ( ) Select-Rate-switch ------------ ( )          |
17| Enable-SS ----------------- ( ) Select-Stardby-switch --------- ( )          |
18| Enable-SI ----------------- ( ) Enable-SB -------------------- ( )           |
19| Acu-Type --------------- (    ) Acu-TImeout--------------- (     )           |
20| Dcr-Hold ------------- (     )                                               |
21| ENTER "1" OR "YES" TO ADD (   ) HELP (                       )               |
22|                                                                              |
23|                                                                              |
24|                                                                              |
  -------------------------------------------------------------------------------
```

Figure 7-7. ADD New Configuration Screen

The configuration is not automatically added. The screen provides for entry of YES or Y at the bottom to verify that the configuration should be added. The operator must respond with either of these two answers to cause the configuration to be added.

The screen entered is checked for validity, and fields with errors are highlighted.

If the screen was correctly entered and the operator indicates that the configuration should be added by entering YES or Y in the lower-left screen entry field, the following message is displayed on the next screen display.

> CONFIGURATION <name> ADDED

The program then returns to the function select menu.

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 the name entered was previously defined for another configuration.

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

Examples of error messages returned are:

CONFIGURATION NAME IS NOT DEFINED
> The configuration name was not defined.

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

Parameter Definitions

The following list of valid parameters includes an explanation of the use of each parameter.

ACU-TIMEOUT (ATI)
> Specifies the timeout value in milliseconds for DIAL operations on an automatic call unit (ACU). If a DIAL operation does not complete within the specified time, a TIMEOUT error is reported.

> The default value is 20000 (20 seconds).

> Valid only when ACU-TYPE is 801, 801E, or TA and only for a Quad Line Adapter-2.

ACU-TYPE (AT)
> Specifies whether or not an automatic call unit (ACU) is connected to the line and, if so, the type of ACU. The values are NONE, 801, 801E, and TA.

> NONE
> > Specifies that the line is either manually dialed or answer only.

> 801
> > Specifies that the line has an 801 ACU (RS-366-A parallel automatic call unit) connected, and the end-of-number (EON) code should not be sent to the ACU.

> 801E
> > The same as 801, with the exception that the EON code is sent to the ACU.

> NOTE
> EON should be used only when the ACU is not capable of detecting an answer tone.

TA
>   Specifies that the line has a Burroughs autodial modem, such as a TA1203 or TA2403 connected.

ACU-TYPE is valid only when SWITCHED is YES. TA is valid only for asynchronous disciplines. 801 or 801E are valid only for a Quad Line Adapter-2. An ACU-TYPE value of TA and a NEW-SIGNAL value of YES cannot be specified at the same time.

ALTERNATE-BAUD-RATE (ABR)
>   The BAUD-RATE value used when the data set (modem) indicates (by way of the Signalling Rate Indicator signal) that it is in high-speed mode or when the data set indicates (by way of the Standby Indicator signal) that it is in standby mode. The ALTERNATE-BAUD-RATE value is used only when the line adapter has been sensitized to the appropriate indicator by way of the ENABLE-SI or ENABLE-SB parameters. If ALTERNATE-BAUD-RATE is zero, the BAUD-RATE value is used. The possible values are the same as the BAUD-RATE parameter.
>
>   The default value is zero.
>
>   Valid only for standard asynchronous or teletype disciplines and only for a Quad Line Adapter-2.

BAUD-RATE (BR)
>   The rate for any asynchronous discipline at which bits are expected by the receiver and sent by the transmitter expressed in bits per second. The rate for synchronous disciplines is determined by the data set. For a Quad Line Adapter-2 using any synchronous discipline, BAUD-RATE also specifies the rate for the Terminal Timing (DA) signal, which can be used to supply a clock signal to the data set or modem eliminator.
>
>   The value must be an integer in the range 50 to 65828. The following specific rates are supported:

50	600	3600	38400
75	1200	4800	57600
110	1800	7200	65828
150	2000	9600	
300	2400	19200	

The highest valid rate for a Quad Line Adapter-1 is 19200. The highest valid rate for asynchronous disciplines on a Quad Line Adapter-2 is 38400. A value greater than 38400 on a Quad Line Adapter-2 invokes the high-speed option and is valid only for synchronous disciplines. The high-speed option is valid for only the odd lines of the adapter. When the high-speed option is invoked, the next lower line is placed into a not present state (for example, if line 1 is a high-speed line, line 0 is not present, if line 11 is a high-speed line, line 10 is not present).

BCS

Specifies the type of horizontal block-check sequence (BCS) that is generated and checked. The values are NONE, EVEN, or CRC.

NONE

No BCS is generated or checked.

EVEN

Specifies a block-check character (BCC), which is a longitudinal redundancy check (LRC) with an initial and final value of all zeroes.

CRC

Specifies a 16-bit cyclic redundancy check, which uses a polynomial of $X^{16} + X^{15} + X^{2+1}$ with an initial and final value of zero.

EVEN is valid only for asynchronous or standard synchronous disciplines. It is the recommended value for all such disciplines except teletype. The recommended value for the Teletype discipline is NONE. CRC is valid only for binary synchronous disciplines and is the recommended value.

BREAK-ENABLE (BE)

Specifies that the adapter is to recognize the following as a break indication while processing a WRITE operation: a mark level that is equal in time to that required to receive two characters. Upon detection of a break, the adapter terminates the WRITE operation and returns a result indicating that a break was detected. The values are Y for yes and N for no.

Both the Quad Line Adapter-1 and 2 transmit a break 2.5 seconds in duration.

BREAK-ENABLE is valid for the TTY and ASYNC disciplines.

CHARACTER-PARITY (CP)

Specifies the type of vertical parity generated on each character transmitted, and checked on each character received. The values are NONE, ODD, or EVEN.

Horizontal parity is generated and checked according to the BCS parameter.

Vertical parity generation and testing is suppressed if CHARACTER-SIZE specifies 8-bit characters or when operating in binary synchronous transparent mode.

The TDI discipline forces CHARACTER-PARITY to EVEN and the binary synchronous EBCDIC discipline forces it to NONE. A value of EVEN is recommended for asynchronous disciplines, and a value of ODD is recommended for synchronous disciplines.

CHARACTER-SIZE (CS)

Specifies the number of data bits per character, but does not include the start, parity, or stop bits. The values are 5, 6, 7, and 8.

The recommended value is 7. The TDI discipline uses a value of 7 and the binary synchronous EBCDIC discipline uses 8 data bits, regardless of the setting of CHARACTER-SIZE. CHARACTER-SIZE is intended primarily for the teletype discipline.

CONTINUOUS-CARRIER (CC)

Specifies whether or not the adapter holds the request-to-send (RTS) signal TRUE. With controlled carrier, the RTS signal is TRUE only during WRITE operations. The values are Y for continuous carrier and N for controlled carrier.

Continuous carrier can be used on a full duplex communications channel to eliminate turnaround delays, even though only half duplex communication is supported.

Valid for all disciplines except TDI.

DATA-ENCODING (DE)

Specifies the type of encoding used by the transmitter and receiver. A value of NRZ specifies Non-Return to Zero encoding, NRZI specifies Non-Return to Zero Inverted encoding, MARK specifies Biphase Mark encoding, and SPACE specifies Biphase Space encoding.

Valid only on a Quad Line Adapter-2. The Quad Line Adapter-1 always uses NRZ encoding. The TDI discipline forces DATA-ENCODING to NRZ. The recommended value is NRZ.

DISCIPLINE (DISC)

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

Mnemonic	Line Discipline
ASYNC	Standard Asynchronous
BISYNC-A	Binary Synchronous ASCII
BISYNC-E	Binary Synchronous EBCDIC
SYNC	Standard Synchrounous
TDI	Two-wire Direct Interface
TTY	Teletype (data set)
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 DIS-CIPLINE 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. ASYNC-J, SYNC-J, and TDI-J are not valid for Quad Line Adapters-2.

DPR-HOLD (DH)

Specifies the time in milliseconds during autodial that the Digit Present (DPR) signal must be held TRUE, even though the Present Next Digit (PND) signal from the ACU is no longer TRUE. The DP signal is held TRUE until both the DPR-HOLD time expires and the PND signal is no longer TRUE.

This option is intended for diagnostic purposes and is normally set to zero.

Valid only when ACU-TYPE is set to 801 or 801E.

ENABLE-SB (ESB)

Specifies whether or not the line adapter is sensitized to the Standby Indicator (RS449 SB, CCITT 117) signal from the data set. The values are Y for yes and N for no.

When ENABLE-SB is set to Y and the Standby Indicator signal is TRUE, the line is reprogrammed using the ALTERNATE-BAUD-RATE parameter.

Valid only on a Quad Line Adapter-2 and for all disciplines except TDI.

ENABLE-SI (ESI)
Specifies whether or not the line adapter is sensitized to the Signalling Rate Indicator (RS449 SI, RS232 CI, CCITT 112) signal from the data set. (212 modems use the CI signal for high-speed mode.) The values are Y for yes and N for no.

When ENABLE-SI is set to Y and the Signalling Rate Indicator signal is TRUE, the line is reprogrammed using the ALTERNATE-BAUD-RATE parameter.

Valid only on a Quad Line Adapter-2 and for all disciplines except TDI.

ENABLE-SR-SF (ESRSF)
Specifies whether or not the Signalling Rate Selector (RS449 SR, RS232 CH, CCITT 111) or Select Frequency (RS449 SF, RS232 CK, CCITT 126) signal to the data set is to be controlled by the value of the SELECT-RATE parameter. The values are Y for yes and N for no.

Valid for all disciplines except TDI.

ENABLE-SS (ESS)
Specifies whether or not the Select Standby (RS449 SS, CCITT 116) signal to the data set is to be controlled by the value of the SELECT-STANDBY parameter. The values are Y for yes and N for no.

Valid for all disciplines except TDI. Either ENABLE-SS or EOT-DISCONNECT can be set to Y, but not both.

EOT-DISCONNECT (ED)
Specifies whether or not the EOT Disconnect (ED) signal to the data set is to be activated. The values are Y for yes and N for no.

EOT-DISCONNECT is valid for all disciplines except TDI. However, it is used mainly with the teletype discipline in Telex (TWX) networks. Either EOT-DISCONNECT or ENABLE-SS, but not both, can be set to yes.

HELP (TEACH)
The following parameter names or abbreviations (in parentheses) can be entered in the HELP field.

ACU-TIMEOUT (ATI)	ACU-TYPE (AT)
ALTERNATE-BAUD-RATE (ABR)	BAUD-RATE (BR)
BCS	BREAK-ENABLE (BE)
CHARACTER-PARITY (CP)	CHARACTER-SIZE (CS)
CONTINUOUS-CARRIER (CC)	DATA-ENCODING (DE)
DISCIPLINE (DISC)	DPR-HOLD (DH)
ENABLE-SB (ESB)	ENABLE-SI (ESI)
ENABLE-SR-SF (ESRSF)	ENABLE-SS (ESS)
EOT-DISCONNECT (ED)	INITIAL-TIMEOUT (IT)
NAME (NAM)	NEW-SIGNAL (NS)
PAD-DISABLE (PAD)	RECEIVE-DELAY (RD)
SELECT-RATE (SR)	SELECT-STANDBY (SS)
STOP-BITS (SB)	SWITCHED (SW)
TEXT-TIMEOUT (TT)	TRANSMIT-DELAY (TD)
TRANSPARENCY (TR)	

Information pertaining to the definition of the requested parameter and the values permitted is displayed on one or more screens as required. When multiple screens are required, the user can transmit NEXT to receive the next screen of information. Transmitting anything other than NEXT causes a return to the original screen.

INITIAL-TIMEOUT (IT)
    Specifies the timeout value in milliseconds for a READ operation to receive the first character. If a character is not received within the specified time, a TIMEOUT error is reported. Any numeric integer value from 0 to 99999 inclusive can be entered. A value of 0 causes the timeout to be disabled.

NAME (NAM)
    Any non-blank field from 1 to 10 characters in length can be entered for the NAME parameter. Upper-case or lower-case letters, decimal digits, and the underscore character are allowed. The name must start with an upper-case or lower-case letter.

NEW-SIGNAL (NS)
    Specifies whether or not the New Signal (RS449 NS or New-Sync) signal to the data set is to be activated. The NS signal directs the data set (such as a Bell 201, Bell 208A, or Burroughs TA714) to prepare to rapidly respond to a new line signal (rapidly resynchronize) between received messages. It is intended for use at a control station of a multipoint polling system with the remote stations operating with controlled carrier. Use of New Signal is recommended when the time between messages is less than 100 milliseconds.

    Valid for all disciplines except TDI. Either NEW-SIGNAL can be set to YES or ACU-TYPE can be set to TA, but not both.

PAD-DISABLE (PAD)
    Applies to only binary synchronous disciplines and specifies whether or not a pad character is required to terminate a READ operation. A pad character is a character with all bits set to 1. The adapter always ends each transmission with a pad character. A receiving station normally expects each transmission that it receives to end with a pad character. A receiving station must normally verify that the first four bits following a message are from a pad character for the message to be considered valid. The values are Y to disable pad character checking or N to require pad character checking.

RECEIVE-DELAY (RD)
    Also known as transmit-to-receive delay or squelch duration. Specifies the time in milliseconds after the initiation of a READ operation and before receiving characters. Used to ignore reflections on two-wire lines. Any numeric integer value from 0 to 99999 inclusive can be entered.

    Most data sets hold (clamp) the received data to the mark level until the received carrier is above a certain threshold level. RECEIVE-DELAY can be set to zero when operating with these data sets. RECEIVE-DELAY must be less than the TRANSMIT-DELAY of the sending station. When using data sets that have a Request-To-Send to Clear-To-Send turnaround time of less than the squelch duration, set RECEIVE-DELAY to zero. A value of zero can be used with the TDI discipline.

SELECT-RATE (SR)
    Specifies the value of the virtual RATE (or FREQUENCY) switch. If the Signalling Rate Selector (SR)/Select Frequency (SF) (CCITT 111/126) signals have been enabled by setting the ENABLE-SR-SF parameter, the value of the SELECT-RATE parameter controls the SR/SF signal. A value of Y causes the signal to be set to TRUE and a value of N causes the signal to be set to FALSE. The network controller input commands SET DATARATE or RESET DATARATE may be used to dynamically change the state of the virtual RATE switch and the SR/SF signal.

    Valid only when the ENABLE-SR-SF parameter is set to YES.

SELECT-STANDBY (SS)

Specifies the value of the virtual STANDBY switch. If the Select Standby (SS) (CCITT 116) signal has been enabled by setting the ENABLE-SS parameter, the value of the SELECT-STANDBY parameter controls the SS signal. A value of Y causes the signal to be set to TRUE and a value of N causes the signal to be set to FALSE. The network controller input commands SET STANDBY or RESET STANDBY may be used to dynamically change the state of the virtual STANDBY switch and the SS signal.

Valid only when the ENABLE-SS parameter is set to YES.

STOP-BITS (SB)

Specifies the number of stop bits to be transmitted at the end of each character. One stop bit is sufficient on received characters. The value can be 1, 1.5, or 2.

Usually two stop bits are used at a baud rate of 110, 1.5 stop bits are used with five bits per character at slow speeds, and one stop bit is used at speeds higher than 110 bits per second.

Valid only for asynchronous protocols and intended primarily for the teletype discipline.

SWITCHED (SW)

Specifies whether the line is switched or dedicated (leased). A value of Y specifies a switched line and a value of N specifies a dedicated line.

Valid for all disciplines except TDI.

TEXT-TIMEOUT (TT)

Specifies the timeout value in milliseconds for a READ operation between received characters after the detection of a SOH or STX character. If the next character is not received within the specified time, a TIMEOUT error is reported. Any numeric integer value from 0 to 99999 inclusive can be entered. A value of 0 causes the timeout to be disabled.

TRANSMIT-DELAY (TD)

Also known as read-to-write delay, Clear-To-Send delay, or turnaround time. Specifies the time in milliseconds after the receipt of a Clear-To-Send signal from the data set (or the end of a READ operation using the TDI discipline) before data is transmitted. TRANSMIT-DELAY is used to prevent transmitting to a terminal that has just transmitted data before the terminal has had sufficient time to switch from transmit to receive mode. Any numeric integer value from 0 to 99999 inclusive can be entered.

If the Request-To-Send to Clear-To-Send delay of the data set is greater than the turnaround time of the receiving terminal, TRANSMIT-DELAY can be set to zero.

TRANSPARENCY (TR)

Specifies whether or not transparency mode is allowed. A value of Y specifies that transparent mode is allowed and a value of N specifies that transparent mode is not allowed.

Valid only for binary synchronous disciplines. The recommended value is YES.

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

CHANGE Configuration Screen

If the CHANGE function is specified, the SYSTEM/QCONFIGURE program responds with the following screen.

```
                   1         2         3         4         5         6         7         8
                   0         0         0         0         0         0         0         0
    .-----------------------------------------------------------------------------.
  1 | CHANGE ( DEFAULT  IN FILE  SYSTEM/QCONFIGS)                  SYSTEM/QCONFIGURE |
  2 |                                                                               |
  3 | ACTION [            ]                                                          |
  4 |        EYE HOpe FUNCtion CONFiguration                                         |
  5 | ------------------------------------------------------------------            |
  6 | NAMe ------------- [DEFAULT  ]       DISCipline ----------- [TD]     ]          |
  7 | SWitches ------------------- [N]     Data-Encoding ------------- [NRZ  ]        |
  8 | Baud-Rate ------------- [7611] ]     Alternate-Baud-Rate ------- [ ]   ]        |
  9 | Transmit-Delay -------- [ ]    ]     Receive-Delay ------------- [ ]   ]        |
 10 | Text-Timeout ---------- [32    ]     Iritial-Timeout ----------- [12]] ]        |
 11 | Character-Parity ------- [EVEN]      Stop-Bits ----------------- [1    ]        |
 12 | Character-Size ----------- [7]       BCS ----------------------- [EVEN]         |
 13 | Continuous-Carrier -------- [N]      PAD-disable --------------- [N]            |
 14 | Break-Enable ------------- [N]       TRansparency -------------- [N]            |
 15 | Eot-Disconnect ----------- [N]       New-Signal ---------------- [N]            |
 16 | Enable-SR-SF ------------- [N]       Select-Rate-switch -------- [N]            |
 17 | Enable-SS ---------------- [N]       Select-Standby-switch ----- [N]            |
 18 | Enable-SI ---------------- [N]       Enable-SB ----------------- [N]            |
 19 | Acu-Type ---------------- [NONE]     Acu-Timeout --------------- [ ]   ]        |
 20 | Dor-Hold --------------- [ ]    ]                                               |
 21 | ENTER "1" OR "YES" TO CHANGE [   ]  HELP [                    ]                 |
 22 |                                                                               |
 23 |                                                                               |
 24 |                                                                               |
    .-----------------------------------------------------------------------------.
```

Figure 7-8. CHANGE Configuration Screen

The meanings of the parameters are the same as previously defined for the ADD function.

If the ALL option was entered in the configuration screen, then each configuration is returned to the screen, one configuration at a time. If the ALL option is not entered, then only the configuration requested is returned. The configurations are not automatically changed. The screen provides for entry of YES or Y at the bottom to verify that the configuration should be changed. The operator must respond with either of these two answers to cause the configuration to be changed. If a response other than Y or YES is entered, the next screen is displayed.

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

NOTE
The NAME field of the CHANGE screen cannot be accessed since a name
change implies a new configuration.

If the screen was correctly entered and the operator indicates that the change is wanted by entering YES or Y in the lower left screen entry field, the following message is displayed on the next screen display.

CONFIGURATION <name> CHANGED

COPY Configuration Screen

If the COPY function is specified, the SYSTEM/QCONFIGURE program presents the configuration specified in the configuration name display screen as follows:

```
        1         2         3         4         5         6         7         8
   i    0         0         0         0         0         0         c         0
 --------------------------------------------------------------------------------
 1  I COPY (SFECIFIED CONFIGURATION ENTRY)                    SYSTEM/QCONFIGURE  I
 2  I                                                                            I
 3  I ACTION (        )                                                          I
 4  I       EYE HOse FUNCtion CONFiguration                                      I
 5  I  ---------------------------------------------------------------------     I
 6  I NAMe -------------- (        )   DISCipline ------------ (TDI      )       I
 7  I SWitchec ----------------- (N)  Data-Encoding ------------- (NPZ )         I
 8  I Baud-Rate ------------ (361) )  Alternate-Baud-Rate ------- ()     )       I
 9  I Transmit-Delay -------- (32   )  Receive-Delay ------------ ()    )        I
10  I Text-Timeout ---------- (32   )  Iritial-Timeout ---------- (121) )        I
11  I Character-Parity ------- (EVEN)  Stop-Bits ----------------- (1  )         I
12  I Character-Size ----------- (7)   UCS ----------------------- (EVEN)        I
13  I Continucus-Carrier -------- (N)  PAD-cisable ---------------- (N)          I
14  I Break-Erable -------------- (N)  TRansparency --------------- (N)          I
15  I Eot-Disconnect ----------- (N)   New-Siyrat ----------------- (N)          I
16  I Enable-SR-SF ------------- (N)   Select-Rate-switch --------- (N)          I
17  I Enable-SS ----------------- (N)  Select-Standby-switch ------ (N)          I
18  I Enable-SI ----------------- (N)  Erable-SH ------------------ (N)          I
19  I Acu-Type -------------- (NONE)   Acu-Timeout---------------- ()   )        I
20  I Dor-Hole ---------------- ()  )                                           I
21  I ENTER "Y" OR "YES" TO ADD (   ) HELP (                        )           I
22  I                                                                            I
23  I                                                                            I
24  I                                                                            I
 --------------------------------------------------------------------------------
```

Figure 7-9. COPY Configuration Screen

The meanings of the parameters are the same as previously defined for the ADD function.

The line configuration is presented with the name field blank, and the user must provide a unique name prior to transmitting the configuration. If the validity check finds any of the parameter fields are in error, the screen is returned with the erroneous fields highlighted.

If the screen was correctly entered and the user indicates that the ADD is wanted by entering YES or Y in the lower-left screen entry field, the following message is displayed on the next screen display.

CONFIGURATION <name> COPIED AS < name>

DELETE Configuration Screen

If the DELETE function is specified, the SYSTEM/QCONFIGURE program responds with the following screen.

```
            1        2        3-       4.-      5        6        7        8
    1       0        0        0        0       0        0        C        0
    .--.---.-----------------------------------------------------------------
 1 | DELETE ( ASYNC12  IN FILE SYSTEM/QCUNFIGS)                SYSTEM/QCUNFIGURE |
 2 |                                                                            |
 3 | ACTION (              )                                                    |
 4 |       (YE HURe FUNCtion CUNFiguration                                      |
 5 | ------------------------------------------------------------------         |
 6 | NAMe ------------- ASYNC12         DISCipline ------------ ASYNC           |
 7 | Switchec ----------------- N       Data-Encoding ------------- NRZ         |
 8 | Baud-Rate ------------- 1200       Alternate-Baud-Rate ------- 0           |
 9 | Transmit-Delay -------- 0          Receive-Delay ------------- 0           |
10 | Text-Timeout ---------- 1200       Initial-Timeout ---------- 1200         |
11 | Character-Parity ------- EVEN      Stop-Bits ----------------- 1           |
12 | Character-Size ------------ 7      BCS ---------------------- EVEN         |
13 | Continuous-Carrier -------- N      PAD-cisable ---------------- N          |
14 | Break-Erable -------------- N      Transparency -------------- N           |
15 | Lot-Disconnect ----------- N       New-Signal ---------------- N           |
16 | Enable-SR-SF ------------- N       Select-Rate-switch -------- N           |
17 | Enable-SS ---------------- N       Select-Standby-switch ----- N           |
18 | Enable-SI ---------------- N       Enable-SB ----------------- N           |
19 | Acu-Type -------------- NONE       Acu-Timeout--------------- 0            |
20 | Dpr-Holc --------------- 0                                                 |
21 | ENTER "Y" OR "YES" TO DELETE (  )  HELP (                         )        |
22 |                                                                            |
23 |                                                                            |
24 |                                                                            |
    ----------------------------------------------------------------------------
```

Figure 7-10. DELETE Configuration Screen

The configuration is not automatically deleted. The screen provides for entry of YES or Y at the bottom to verify that the configuration should be deleted. The operator must respond with either of these two answers to cause the configuration to be deleted. If an invalid response is entered, the next screen is displayed.

If the user indicates that the change is wanted by entering YES or Y in the lower-left screen entry field, the following message is displayed on the next screen display.

    CONFIGURATION <name> DELETED

DISPLAY Configuration Screen

The DISPLAY function displays one or all line configurations. The SYSTEM/QCONFIGURE program responds with the following screen.

```
             1        2        3        4        5        6        7        8
    1        0        0        0        0        0        0        0        0
   --------------------------------------------------------------------------
  1 I DISPLAY ( ASYNC12  IN FILE SYSTEM/QCCNFIGS)              SYSTEM/QCONFIGURE I
  2 I                                                                          I
  3 I ACTION (            )                                                    I
  4 I       EYE HOme FUNCtion CUNFiguration                                    I
  5 I ------------------------------------------------------------------       I
  6 I NAMe -------------- ASYNC12         DISCipline ----------- ASYNC         I
  7 I SWitches --------------------- N    Data-Encoding ------------- NRZ      I
  8 I Baud-Rate ------------- 1200        Alternate-Baud-Rate ------- 0        I
  9 I Transmit-Delay -------- 0           Receive-Delay ------------- 0        I
 10 I Text-Timeout --------- 1200         Iritial-Timeout ----------- 1200     I
 11 I Character-Parity ------- EVEN       Stop-Bits ----------------- 1        I
 12 I Character-Size ----------- 7        DCS ----------------------- EVEN     I
 13 I Continuous-Carrier -------- N       P/O-disable ---------------- N       I
 14 I Break-Erable -------------- N       Transparency --------------- N       I
 15 I Eot-Disconnect ----------- N        New-Signal ----------------- N       I
 16 I Enable-SR-SF ------------- N        Select-Rate-switch --------- N       I
 17 I Enable-SS ---------------- N        Select-Standby-switch ------ N       I
 18 I Enable-SI ---------------- N        Erable-SB ------------------ N       I
 19 I Acu-Type --------------- NONE       Acu-Timeout---------------- 0        I
 20 I Dpr-Hole ------------- 0                                                 I
 21 I TRANSMIT ANYTHING TO RETURN ( )  HELF (                       )          I
 22 I                                                                          I
 23 I                                                                          I
 24 I                                                                          I
   --------------------------------------------------------------------------
```

Figure 7-11. DISPLAY Configuration Screen

If ALL was entered in the CONFIGURATION NAME screen, then all line configurations are displayed. A prompt is required to proceed to the next configuration.

If the name of a configuration is entered, then the name is checked for validity, and if present, is displayed. When the configurations have been displayed, the program returns to the function select menu.

PRINT Configuration

The PRINT function generates a printed copy of the line configurations.

If ALL was entered in the configuration name screen, 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 display screen.

If the name of a configuration is entered, then the name is checked for validity, and if present, is printed on the line printer.

When all requested configurations are printed, the program returns to the function select menu.

APPENDIX A
GLOSSARY OF TERMS

American Standard Code for Information Interchange (ASCII)   This code, established by the American Standards Association as an American standard, defines codes for a set of characters to be employed in the interchange of information between business equipment and over telephone and telegraph circuits. The code consists of 128 control and graphic characters. Each character consists of seven bits of information.

ASCII   A mnemonic abbreviation for American Standard Code for Information Interchange.

attach initiator   The file that writes an ATTACH message.

application program   A program that processes data and is usually unique to one type of application.

Automatic Calling Unit (ACU)   A device that can be furnished by a communications carrier that allows a business machine to automatically establish a dialed link over the communications network.

autopoll   A technique used by the central computer to interrogate terminals to determine if they have anything to transmit without interrupting the network controller for the initiation of each station poll.

asynchronous transmission   A means of transmission which requires that there always be an integral number of unit intervals within a character between any two significant instants in time.

baud   Baud is a unit measure of data flow, commonly bits per second. Bits per second (BPS) is used to express the speed at which bits are transmitted over a data communications line or internally within a computer system.

binary synchronous (BISYNC)   A method of data transmission which allows sending or receiving of data streams. The data streams can contain bit patterns which would normally be detected as control character sequences. Therefore, data streams containing object program code can be transmitted. The transmission is always synchronous.

bit   A bit is the smallest unit of information in a binary system.

block   A group of characters or bits sent as an integral unit.

block check character (BCC)   The block check character is a redundant character added to the end of a transmission block for the purpose of error detection.

block number   This option is used when data must be subdivided into separate units for transmission. The block number consists of a 2-character number identifying the sequential block number in a blocked message. The first character is always a DLE character.

BOJ   A mnemonic abbreviation for beginning of job.

break   The facility to enable a receiving device to interrupt the transmitting device and, by doing so, to be able to take control of the circuit.

broadcast   The simultaneous transmission of a message to several stations of a network.

buffer   A buffer is a temporary storage area that is used during the transfer of data from one device to another. It is also used to compensate for the variations in data flow rates or timing of events within a system.

character A set of elements arranged in orderly groups to represent digits, symbols, or letters. Represented in two language forms: 1) for use by computers, business machines, communications facilities, etc., usually in groups of binary bits; and 2) for use by people in conveying an understandable form of decimal digits, alphabetic characters, punctuation, or other special symbols. Characters are represented by using groups of bits. These groups are commonly five, six, seven, or eight bits in size.

character synchronization A process in which the character frequencies of the transmitting and receiving portions of a transmission circuit are maintained in a phase relationship in order that the receiver can derive the transmitted characters from the signals received.

continuous operation A type of message transmission in which the master station need not stop for a reply after transmitting each acknowledgement unit, but can continue transmission with the next acknowledgement unit.

control character, data communications A functional character intended to control or facilitate transmission of information over communication networks. The major communication control characters provided are ACK, BEL, CAN, CR, DC1, DC2, DC3, DC4, DEL, DLE, ENQ, EOT, ESC, ETX, FIB, FF, FSL, GS, GSL, HT, LF, NAK, NUL, POL, RS, SI, SO, SOH, STX, SYN, US, and VT. Refer to appendix B, Data Communications Control Characters, for more information on the function of these control characters.

control station A permanently designated station on a data link, unaffected by link control procedures, having the overall responsibility for the orderly operation of the link. A control station generally has additional control capabilities. One example is the capability to poll other stations, which is not allowed at non-control stations. The control station is also designated to initiate error recovery procedures in the event of certain abnormal conditions on the link.

conversational mode An operational mode in which message information is used in lieu of, or in addition to, control characters as replies for message information.

controlling remote file (CRF) The file with HEADERS to which a station was most recently assigned, either with an ATTACH or with an OPEN. If the primary file has HEADERS, it is the CRF. Otherwise, the secondary file is the CRF.

CRF A mnemonic abbreviation for controlling remote file.

data communications The transfer of encoded information by means of electrical transmission systems.

data communications system One or more data links, each of which can be operating in the same or a different mode.

data link This is an ensemble of terminal installations and the interconnected network, together with controlling procedures operating in a particular mode, that permits information to be exchanged between locations.

data set A circuit termination device used to provide an interface between a data communications circuit and a data terminal. A modulation and/or demodulation function is typically performed by a data set.

dial-up The use of a dial or pushbutton telephone to establish a station-to-station telephone contact.

EBCDIC A mnemonic abbreviation for Extended Binary Coded Decimal Interchange Code.

EOJ   A mnemonic abbreviation for end of job.

Extended Binary Coded Decimal Interchange Code (EBCDIC) This code, which defines a set of 256 characters, is used to exchange information between computer systems and store information on computer readable media. Each character consists of eight bits of information.

family   A multistation file.

family population   The number of active stations in the family of the file. Family size is the number of stations assigned to the file.

fast select (FSL)   The fast select character is used to indicate "This is a fast select" in a selection sequence transmitted by the central computer. Fast select is followed immediately by a transmission block, without requiring acknowledgement of the selection.

filler   A character used as a time-fill or space-fill when a block of a specified size is required and when the heading and/or text characters are of sufficient length for the requirement.

format   The predetermined arrangement of characters, field, lines, page numbers, punctuation marks, and so forth used to transfer data from one location to another. This refers to input, output, and files.

group poll   Group poll is used to poll a group of stations using a group poll address. Only the station that has a message to send responds to the poll. For example, for TD830 terminals, the group poll address is the transmit address. Using group poll, a poll string would be issued using the group poll address as all stations in the group have the same transmit or group poll address. The station answering the poll masks out the other stations in the group; therefore, only the answering station receives the ACK of the message just transmitted in response to the group poll. The output Request procedure then uses the input address in a select string to send a message to a particular station in the group, instead of using the transmit (group poll) address in the select string.

handshaking   This is the exchange of control signals between data sets when the connection is established.

header   A sequence of characters that can precede the text of a message to provide the information necessary to route the message to any number of ultimate destinations. A message header can also contain communications relating information other than routine instructions.

headers   An option on a remote file which allows system control functions and provides a 50-byte header on all data messages moving through that remote file.

identification   A process using a sequence of characters to identify a station during the establishment of a connection and subsequently, if needed.

identifier   A sequence of one or more characters transmitted by a station to identify itself.

information block   A sequence of characters of fixed or variable length which is a subdivision of an information message formed for the purpose of meeting transmission requirements.

information message   A sequence of characters conveying the text. It can also convey supplementary information forming a heading.

I/O   A mnemonic abbreviation for input/output.

line adapter   A line adapter consists of a logic card used to interface a data set or communications line into a multiline or single-line I/O control.

line switching   A switching technique that temporarily connects two lines so that two stations can directly exchange information.

link, data communications   The logical association of two or more stations interconnected by the same communication channel. A data link includes the communication control capability of the connected stations.

logical station number (LSN)   The number by which the network controller uniquely identifies a station for normal transactions. LSNs begin with 001 and proceed sequentially through all the stations declared in the Station Section of the network controller symbolic code.

LSN   A mnemonic abbreviation for logical station number.

master station   A station temporarily designated can be changed by the link control procedures to have control of the data link at a given instant. Master status is normally conferred upon a station so that it can transmit a message.

MCP   A mnemonic abbreviation for Master Control Program II.

MCPII   A mnemonic abbreviation for Master Control Program II.

MCS   A mnemonic abbreviation for Message Control System.

message   A sequence of characters arranged in a form suitable for the purpose of conveying information from an originator to one or more destinations or addresses. It contains the information to be conveyed (called the text) and can, in addition, contain communication information to aid in the routing or handling of the message (called the header).

Message Control System (MCS)   Any program which opens a remote file with the headers option and thereby controls the stations in that remote file.

message format   Rules for the placement of such portions of a message as message header, address, text, and end of message.

message numbering   The identification of each message within a communications system by the assignment of a sequential number.

message switching   The technique of receiving a message, storing it until the proper outgoing circuit and station are available, and then retransmitting the message toward its destination.

MLC   A mnemonic abbreviation for multiline control.

modem   Same as data set.

modulation   The process by which a characteristic of one wave is varied according to some characteristic of another wave. Modulation is used in data sets to make business machine signals compatible with communication facilities.

multiline control (MLC)   The multiline control is an I/O control that provides the functional control between the computer system and the line adapters. This is an integral part of a multiline data communications subsystem.

multipoint network   1. A data communication line connecting three or more stations. 2. A data communication link with the control capability necessary to interconnect three or more stations.

NC A mnemonic abbreviation for Network Controller.

NDL A mnemonic abbreviation for Network Definition Language.

network The ensemble of equipment through which connections are made between terminal installations. This equipment operates in real time and does not introduce, store, or forward delays. A switched telephone network is the network of telephone lines normally used for dialed telephone calls. A private network is a network of communication channels reserved for the use of one customer.

network controller The program generated through compilation of a network controller symbolic code. The network controller handles the line discipline for the data communications devices of a system and for the interface queue between an MCS and the operating system.

Network Definition Language (NDL) A descriptive, free-form language for defining and implementing a data communications network. The NDL compiler analyzes the input statements and generates a custom network controller.

ODT A mnemonic abbreviation for Operator Display Terminal.

off line For data communications, off line refers to a station which is not under the control of the network controller, that is a station which is either turned off or not being polled.

on line For data communications, on line refers to a station which is under the control of the network controller, that is a which station is both turned on and being polled.

parity bit A bit associated with a character or block for the purpose of checking for the presence of error within that character or block. This bit is chosen to make the modulo 2 sum of the bits, including the parity bit, in the character or block a zero (even) or a one (odd) as the system requires.

parity check A check that tests whether the sum of the binary 1's and 0's in an array of binary digits is odd or even.

point-to-point connection A configuration in which a connection is established between two, and only two, terminal installations. The connection can include switching facilities.

polling A technique for assignment of master status to a particular station on a data communications link. Polling is centrally controlled in order to maintain a strict discipline over the operation of a number of stations.

polling supervisory sequence A supervisory sequence that performs a polling function.

primary file The file to which a station was most recently attached or was included in an OPEN message. Normal input goes to the primary file.

queue A linear list for which all insertions are usually made at one end of the list and all deletions and other accesses are made at the other end.

relative station number (RSN) The number by which a user program with a remote file using a remote key uniquely identifies a station. An RSN equal to 1 implies the first station in the FAMILY statement for that file. RSNs proceed sequentially through the stations delineated by the FAMILY statement of a file, except that a controlling remote file can modify the LSN list and thereby modify the RSNs of the remote file.

remote file A file declared in a program which, in conjunction with the network controller, provides input, output, or I/O with a set of data communication devices.

remote job entry (RJE)   A computer communications system in which a central computer executes programs that were sent to it by other computers that appear as terminals.

RJE   A mnemonic abbreviation for remote job entry.

RSN   A mnemonic abbreviation for relative station number.

secondary file   The file to which a station was just previously attached or which was included in an OPEN request. The secondary file must have headers, and must have approved the ATTACH or OPEN message of the primary file. Input, whose first character matches the designated signal character, goes to the secondary file.

shutdown time   The time at which the network controller performs several functions necessary to the operation of the data communications network. The major functions are the disabling of any station input in progress and the issuing of a termination notification to all executing application programs or Message Control Systems. These functions are performed before the network controller terminates itself.

signal character   A signal character is used by the network controller to identify messages intended for the MCS from a station. All messages whose first character is the designated signal character go to the secondary file. All other messages go to the primary file.

signal-conversion equipment   That part of the terminal installation belonging to the data channel, comprising at least one modulator or one demodulator. According to the signals to be transmitted, this equipment provides modulation and/or demodulation of the signals received.

simple headers   Simple headers is an option on a remote file which provides many of the capabilities of a headers file without the responsibilities of being an MCS. The user can do REMOTE_FILE_INFO, STATUS, DETACH, GOODRESULTS, and set tallies and toggles in the header. The user does not receive OPEN, CLOSE, ATTACH, and LOGICALACK messages.

single-line control (SLC)   The single-line control is an I/O control that provides the functional control between a computer system and a line adapter. The SLC is an integral part of a single-line data communications subsystem.

SLC   A mnemonic abbreviation for single-line control.

start/stop transmission   Start/stop transmission is an asynchronous transmission in which a group of code elements corresponding to a character signal is preceded by a start signal. The start signal prepares the receiving mechanism for the reception and registration of a character and is followed by a stop signal. The stop signal brings the receiving mechanism to rest in preparation for the reception of the next character.

station   An input/output device designed to both receive and send source data in an environment associated with the job to be performed. Also, it is capable of sending entries to, and obtaining output from, the system of which it is a part.

switching   Operations involved in interconnecting circuits in order to establish a temporary communication channel between two or more stations.

synchronous transmission   A transmission process in which there is always an integral number of unit intervals between any two significant instants.

System Development Language (SDL)   A compiler-level language used primarily for writing B 1000 system software.

transparent mode   An operational mode in which all coded combinations of 8-bit characters are allowed as message characters within the message text.

turnaround time   This is the time required by data sets when switching between transmit and receive modes.

User Programming Language (UPL)   A compiler-level language used primarily for writing Message Control Systems (MCS) for B 1000 Data Communications Systems.

APPENDIX B
DATA COMMUNICATIONS CONTROL CHARACTERS

The following are the data communications control characters and their functions.

ACK

Acknowledgement. The ACK character indicates an affirmative response to a selection and/or a message directed to or from the station.

ACK0, ACK1

Acknowledgement. The ACK0 and ACK1 control sequences are used in binary synchronous protocols and indicate, if received in the proper sequence, that the previous message was accepted without error, and that the station is ready to accept the next block of the transmission. The ACK0 control sequence is used as a response to a selection for binary synchronous multipoint protocols or a line bid for point-to-point protocols. The alternate use of the ACK0 and ACK1 control sequences is that of affirmative replies and provides a sequential checking mechanism for a series of replies. This makes it possible to maintain a continuous check to ensure that each reply corresponds to the immediately preceding message block. The affirmative response to a poll is the transmission of a message.

The ACK0 control sequence is comprised of a DLE character followed by a EBCDIC hexadecimal 70 (DLE[70]). The ACK1 control sequence is comprised of a DLE character followed by a EBCDIC hexadecimal 61 (DLE/).

AD1, AD2

Address. A special 2-character address that is assigned to each terminal in the network for identification purposes. AD1 and AD2 cannot be special data communications control characters.

The address is used to identify a station in polling or selection, and is used in the message header to identify the terminal from which a message is transmitted. The address is also used to identify from which station an ACK or NAK control character originated. On receipt of a message, the receiving station can use the AD1-AD2 characters to verify that the message originated at the polled terminal.

For group polling, the AD1 and AD2 characters are assigned to a group of terminals that are in one concatenated string. Group polling allows any or all of the terminals in a concatenated string to sequentially answer a single polling sequence without the necessity for polling each terminal address. The AD1-AD2 characters indicate the station that acknowledges receipt of the message.

BCC

Block Check Character. A longitudinal parity check character. For example, bit 1 of the BCC character is a parity summation of the first bit contained in all characters following the SOH or STX character and including the ETX character. The BCC represents an even parity count and is used in all received or transmitted messages containing text data.

The use of a BCC character is optional for some types of line adapters.

BEL

Bell Character. The receipt of a BEL character causes the audible alarm on screen terminals.

BSL

Broadcast Select. The BSL character causes all terminals to accept the message. In the broadcast sequence, the AD1-AD2 characters identify the station that acknowledged receipt of the message. A broadcast select message, which is followed immediately by a transmission block, does not require acknowledgement of the selection.

BS

Backspace. When contained within the text portion of a message, the BS character causes the terminal cursor to be moved one space to the left. When the cursor is located in the first position of a line, the BS character causes the cursor to move to the last cursor position of the preceding line.

CAN

Cancel or Blink Video. On a non-screen terminal such as the AP 300 series, the CAN character causes the terminal to be put into an idle status. On a screen terminal, the CAN character causes the data following to blink.

CON

Contention. The CON character is used to place terminals in a standby or idle mode.

CR

Carriage Return. On a non-screen device, the CR character causes the cursor to move to column 1 of the current line. On a screen device, the CR character typically causes the cursor to move to column 1 of the next line. This is generally a terminal option.

DC1

Device Control 1. The DC1 character is a character that performs a specific function for different terminals. On a screen terminal, the DC1 character causes the terminal to remain in receive mode and optionally erase all data from the cursor position to the end of the line.

DC2

Device Control 2. The DC2 character is a character that performs a specific function for different terminals. The DC2 character has no specific function for the B 1000 system software.

DC3

Device Control 3. The DC3 character is a character that performs a specific function for different terminals. On a screen terminal, the DC3 character causes the terminal to perform a reverse line feed.

DC4

Device Control 4. The DC4 character is a character that performs a specific function for different terminals. On a screen terminal, the DC4 character causes the cursor to move to the home position.

DLE
> Data Link Escape. The DLE character is used to change the meaning of a limited number of contiguous characters, which then become supplementary data transmission control functions.
>
> The data link escape character combinations are used primarily for binary synchronous operation. The data link escape character is always the first character of the 2-character combinations. The DLE combination characters are listed as follows:
>
> ACK0 (DLE 4"70")
> ACK1 (DLE 4"61")
> RVI (DLE 4"7C")
> WACK (DLE 4"6B")
> DLE STX
> DLE DLE
> DLE EOT
> DLE ENQ
> DLE SYN
> DLE ETB
> DLE ETX
> DLE ITB

DEOT
> The mnemonic for the DLE EOT communication control sequence. The DLE EOT control sequence is used to indicate a mandatory disconnect of a switched line.

EM
> The EM character is used in screen terminals to provide a secure video function.

ENQ
> Enquiry. The ENQ character is used as a request for a response of station status or for a retransmission of control characters. The ENQ character is also used as the final character of a poll or in a selection when response from the other station is required.

EOT
> End of Transmission. The EOT character indicates either the end of a communications sequence or the first character in a poll or select message. An EOT character can be transmitted by the master station in a point-to-point protocol to abort a communications sequence. If a station receives a poll message and it is not ready to transmit, it responds with an EOT character.

ESC
> Escape. The ESC character indicates a software controllable escape sequence.

ETB
> End of Text Block. The ETB character is used, in the Bisync protocols, to indicate the end of a block of text. The ETB character is generally used when messages are of sufficient length to warrant their being broken into smaller transmission blocks. The ETB character indicates the end of a block of data, either in a heading or in the message text. The heading or text is resumed after transmission of a block number and either a SOH or STX character.

ETX

End of Text. The ETX character is used to terminate the text portion of received or transmitted text. The ETX character is immediatedly followed by a BCC character (optional).

FF

Forms Feed. On a non-screen terminal, the FF character causes the paper to advance to the top of the next page. On a screen terminal, the FF character causes the screen to be cleared and the cursor to go to the home position.

FS

Form Separator. On a screen terminal, the FS character is used as the opening delimiter for a protected data field that can be transmitted. The RS character is used as the closing delimiter for the field.

FSL

Fast Select. On a screen terminal, the fast select character causes a message to be displayed without inquiring into the ready status of the terminal.

GS

Group Separator. On a screen terminal, the GS character is used to right justify protected data in forms mode. The RS character is used as the closing character.

GSL

Group Select. The GSL character is any selected character that is assigned to a group of terminals that causes all the terminals in the group to accept the message. However, only the terminal that is being addressed responds to the GSL message.

HT

Horizontal Tab. The HT character causes the cursor to advance to the next tab stop.

ITB

Intermediate Text Block. The ITB character is used for binary synchronous operations to divide a message (heading or text) for error-checking purposes, without causing a reversal of transmission direction. The block check character immediately follows ITB and resets the block check count. After the first intermediate block, successive intermediate blocks need not be preceded by STX or SOH. For transparent data, which can contain control character sequences, each successive intermediate block must begin with DLE STX. If two consecutive intermediate blocks that are heading blocks are followed by an intermediate block that is a text block, then the text block must begin with an STX.

Normal line turnaround occurs after the last intermediate block, which is terminated by ETB or ETX (DLE ETB or DLE ETX for transparency). When óne of these ending characters is received, the receiving station responds to the entire transmission. If a block check error is detected for any of the intermediate blocks, a negative reply is sent, which requires retransmission of all intermediate blocks.

All binary synchronous stations must have the ability to receive the ITB character and its attendant BCC. The ability to transmit the ITB character is a station option.

LF

Line Feed. The LF character causes the cursor to move down to the next line.

NAK

Negative Acknowledgement. The NAK character is used as a negative response to a normal selection, indicating not ready to receive, or as a negative response to a transmission, indicating character parity failure for any character of a block or, in a message, a failure of the BCC, or block number sequence check failure. The NAK character can optionally be preceded by station identification, AD1-AD2, or by other information.

NUL

Null or time fill. The NUL character is used as a time fill.

POL

Poll. The POL character is part of a polling message that causes the addressed terminal to answer. If the addressed terminal is in transmit mode, it responds with a text message or a control message. If the addressed terminal is not in transmit mode, it responds with an EOT character. The POL character precedes an ENQ in poll sequence.

RS

Right Separator. The RS character acts as a right delimiter to a FS or GS character in forms mode. On screen devices, the RS character also terminates the highlighting function.

RVI

Reverse Interrupt. The RVI control sequence is transmitted from a slave station in a binary synchronous point-to-point protocol and causes the master station to cease transmission.

The RVI control sequence is comprised of a DLE character followed by the at sign (@) character.

SEL

Select. The SEL character is used when polling terminals and causes the addressed terminal to respond with either an ACK character if the terminal is in receive mode or a NAK character if the terminal is not in receive mode. The SEL character precedes an ENQ character in the select sequence.

SI

Shift In or underline. The SI character is used to indicate that an alternate character set is to be used. On a screen device, the SI character indicates that the text following the SI character is to be underlined.

SO

Shift Out or reverse video. The SO character is used to indicate that the primary character set is to be used. On a screen device, the SO character indicates that the text following the SO character is to be displayed in reverse video.

SOH

Start of Header. The SOH character is required when a message header is to be sent with the transmission. When used, the SOH character is the first of a sequence of characters used for the message header. The message header also can contain the station identification, AD1-AD2, and can, depending on the specific application, contain other information pertinent to the transmission number. A header is ended by the STX character.

STX
> Start of Text. The STX character is used to indicate the start of the text data of the message.

SUB
> Bright Video. On a screen terminal, the SUB character indicates that the data following the SUB character is to be displayed in bright video.

SYN
> Synchronous Idle. The SYN character is used only for synchronous transmission in the absence of any other character to provide a signal for establishing and retaining character synchronization. On initiation of synchronous transmission, a number of SYN characters are transmitted before transmitting any other character to permit the receiving station to acquire character synchronization.
>
> The SYN character is also used as a time fill when no other characters are available for transmission at any point in a character sequence except between the ETB and ETX characters and the next BCC character. The SYN character is purged at the receiving station and is not included in the BCC character.

US
> Forms delimiter. On a screen terminal, the US character produces a left delimiter that precedes unprotected data.

VT
> Vertical Tab or Variable Tab. On a non-screen terminal such as the AP 300, the VT character causes the paper to be advanced a specified number of lines. On a screen terminal, the VT character is used to set variable tab stops.

WACK
> Wait before Acknowledgement. The WACK control sequence is used in the binary synchronous protocol to indicate that the receiving station has accepted the message, but is temporarily not ready to receive from the transmitting station. The WACK control sequence can be transmitted as a response to text or heading block, selection sequence, line bid, or as an identification line bid sequence.
>
> The WACK control sequence is comprised of the DLE character followed by a comma (,) character.

APPENDIX C
CHARACTER SETS

Table C-1 is a listing of the 8-bit EBCDIC internal codes with the corresponding USASCII-7 codes or graphics.

Hexadecimal representation means that standard convention is followed for the 8-bit internal codes, which are presented as pairs of hexadecimal numbers that are translated as shown in the following examples.

Examples:

		8 4 2 1	8 4 2 1
@39@	=	0 0 1 1	1 0 0 1
@BE@	=	1 0 1 1	1 1 1 0
@0F@	=	0 0 0 0	1 1 1 1

Table C-1. B 1000 Codes in EBCDIC Sequence

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
0	00	NUL	NUL
1	01	SOH	SOH
2	02	STX	STX
3	03	ETX	ETX
4	04		EOT
5	05	HT	ENQ
6	06		ACK
7	07	DEL	BEL
8	08		BS
9	09		HT
10	0A		LF
11	0B	VT	VT
12	0C	FF	FF
13	0D	CR	CR
14	0E	SO	SO
15	0F	SI	SI
16	10	DLE	DLE
17	11	DC1	DC1
18	12	DC2	DC2
19	13	DC3	DC3
20	14		DC4
21	15	NL	NAK
22	16	BS	SYN
23	17		ETB
24	18	CAN	CAN
25	19	EM	EM
26	1A		SUB

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
27	1B		ESC
28	1C	FS	FS
29	1D	GS	GS
30	1E	RS	RS
31	1F	US	US
32	20		[space]
33	21		!
34	22		"
35	23		#
36	24		$
37	25	LF	%
38	26	ETB	&
39	27	ESC	'
40	28		(
41	29		)
42	2A		*
43	2B		+
44	2C		,
45	2D	ENQ	-
46	2E	ACK	.
47	2F	BEL	/
48	30		0
49	31		1
50	32	SYN	2
51	33		3
52	34		4
53	35		5
54	36		6
55	37	EOT	7
56	38		8
57	39		9
58	3A		:
59	3B		;
60	3C	DC4	<
61	3D	NAK	=
62	3E		>
63	3F	SUB	?
64	40	[space]	@
65	41		A
66	42		B
67	43		C
68	44		D
69	45		E
70	46		F
71	47		G
72	48		H
73	49		I
74	4A	[	J
75	4B	.	K
76	4C	<	L

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
77	4D	(	M
78	4E	+	N
79	4F	!	O
80	50	&	P
81	51		Q
82	52		R
83	53		S
84	54		T
85	55		U
86	56		V
87	57		W
88	58		X
89	59		Y
90	5A	]	Z
91	5B	$	[
92	5C	*	\
93	5D	)	]
94	5E	;	^
95	5F	^	—
96	60	-	`
97	61	/	a
98	62		b
99	63		c
100	64		d
101	65		e
102	66		f
103	67		g
104	68		h
105	69		i
106	6A	¦	j
107	6B	,	k
108	6C	%	l
109	6D	__	m
110	6E	>	n
111	6F	?	o
112	70	!	p
113	71		q
114	72		r
115	73		s
116	74		t
117	75		u
118	76		v
119	77		w
120	78		x
121	79	`	y
122	7A	:	z
123	7B	#	{
124	7C	@	¦
125	7D	'	}

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
126	7E	=	~
127	7F	"	DEL
128	80		
129	81	a	
130	82	b	
131	83	c	
132	84	d	
133	85	e	
134	86	f	
135	87	g	
136	88	h	
137	89	i	
138	8A		
139	8B		
140	8C		
141	8D		
142	8E		
143	8F		
144	90		
145	91	j	
146	92	k	
147	93	l	
148	94	m	
149	95	n	
150	96	o	
151	97	p	
152	98	q	
153	99	r	
154	9A		
155	9B		
156	9C		
157	9D		
158	9E		
159	9F		
160	A0		
161	A1	~	
162	A2	s	
163	A3	t	
164	A4	u	
165	A5	v	
166	A6	w	
167	A7	x	
168	A8	y	
169	A9	z	
170	AA		
171	AB		
172	AC		
173	AD		

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
174	AE		
175	AF		
176	B0		
177	B1		
178	B2		
179	B3		
180	B4		
181	B5		
182	B6		
183	B7		
184	B8		
185	B9		
186	BA		
187	BB		
188	BC		
189	BD		
190	BE		
191	BF		
192	C0	{	
193	C1	A	
194	C2	B	
195	C3	C	
196	C4	D	
197	C5	E	
198	C6	F	
199	C7	G	
200	C8	H	
201	C9	I	
202	CA		
203	CB		
204	CC		
205	CD		
206	CE		
207	CF		
208	D0	}	
209	D1	J	
210	D2	K	
211	D3	L	
212	D4	M	
213	D5	N	
214	D6	O	
215	D7	P	
216	D8	Q	
217	D9	R	
218	DA		
219	DB		
220	DC		
221	DD		

Decimal	Hex	EBCDIC Graphic	ASCII7 Graphic
222	DE		
223	DF		
224	E0	\	
225	E1		
226	E2	S	
227	E3	T	
228	E4	U	
229	E5	V	
230	E6	W	
231	E7	X	
232	E8	Y	
233	E9	Z	
234	EA		
235	EB		
236	EC		
237	ED		
238	EE		
239	EF		
240	F0	0	
241	F1	1	
242	F2	2	
243	F3	3	
244	F4	4	
245	F5	5	
246	F6	6	
247	F7	7	
248	F8	8	
249	F9	9	
250	FA		
251	FB		
252	FC		
253	FD		
254	FE		
255	FF		

Table C-2 is a chart of the 8-bit EBCDIC internal codes with their corresponding graphic characters.

Table C-2. EBCDIC Chart.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE			SP	&	–						{	}	\	0
1	SOH	DC1					/		a	j	~		A	J		1
2	STX	DC2		SYN					b	k	s		B	K	S	2
3	ETX	DC3							c	l	t		C	L	T	3
4									d	m	u		D	M	U	4
5	HT	NL	LF						e	n	v		E	N	V	5
6		BS	ETB						f	o	w		F	O	W	6
7	DEL		ESC	EOT					g	p	x		G	P	X	7
8		CAN							h	q	y		H	Q	Y	8
9		EM						`	i	r	z		I	R	Z	9
A					[	]	¦	:								
B	VT				.	$	,	#								
C	FF	FS		DC4	<	*	%	@								
D	CR	GS	ENQ	NAK	(	)	_	'								
E	SO	RS	ACK		+	;	>	=								
F	SI	US	BEL	SUB	!	^	?	"								

Table C-3 is a chart of the 7-bit ASCII internal codes with their corresponding graphic characters.

Table C-3. ASCII Chart.

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

APPENDIX D
SYNTAX AND NOTATION CONVENTIONS

Railroad syntax diagrams show how syntactically valid statements can be constructed. Traversing a rail-road syntax diagram from left to right, down on the left vertical line, up on the right vertical line, or in the direction of the arrowheads, and adhering to the limits illustrated by bridges produces a syntactically valid statement. Continuation from one line of the diagram to another 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:
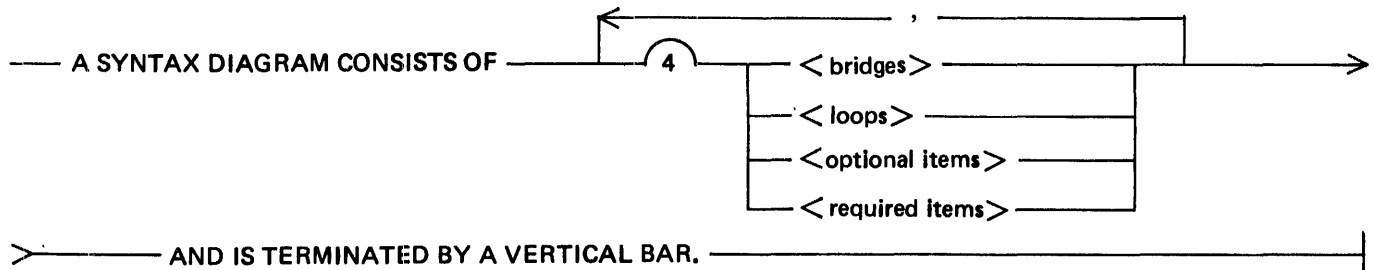
The following syntactically valid statements can be constructed from the above diagram:

A SYNTAX DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTICAL BAR.

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

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

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

REQUIRED ITEMS

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

Example:

—— REQUIRED ITEM ———————————————————————————————————————|

G50052
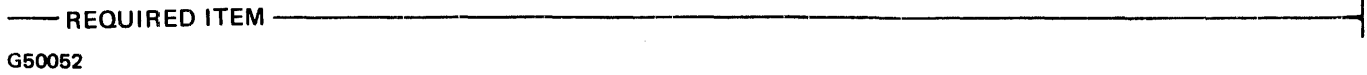
OPTIONAL ITEMS

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:

—— REQUIRED ITEM ——┬——————————————————————————————————|
                    ├—— <optional item-1> ——┤
                    └—— <optional item-2> ——┘

G50053
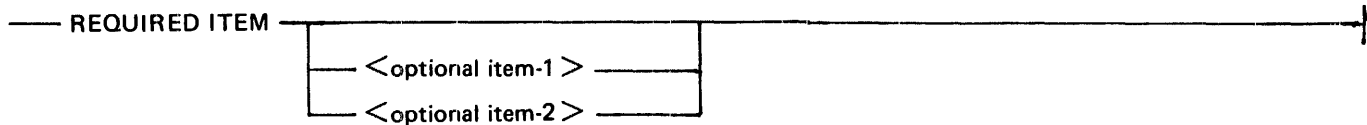
The following valid statements can be constructed from the above diagram:

    REQUIRED ITEM

    REQUIRED ITEM <optional item-1>

    REQUIRED ITEM <optional item-2>

LOOPS

A loop is a recurrent path through a syntax diagram and has the following general format.
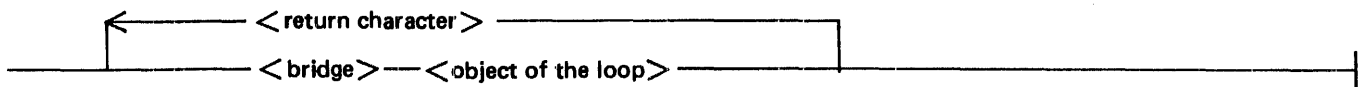
```
         ┌←————— <return character> ——————┐
—————————┤      <bridge>——<object of the loop>——————┤———————————|
```

Example:

```
       ┌←—————————— , ——┐
———————┤  2  ┬—— <optional item-1> ——┤——————————————————|
          └—— <optional item-2> ——┘
```
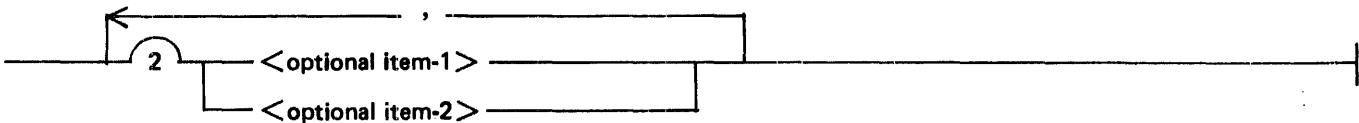
The following statements can be constructed from the syntax diagram in the example.

<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 arrowheads, 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 syntax diagram.

There are two forms of bridges.

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

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

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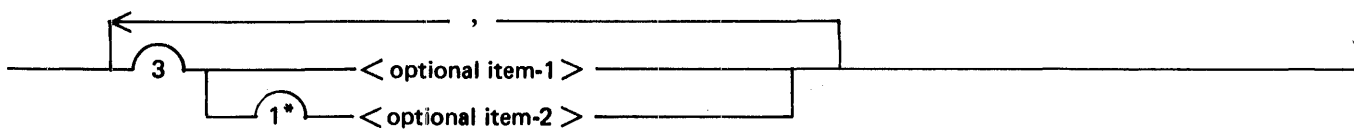
The path inside the loop can be traversed a maximum of three times; however, the path for <optional item-2> must be traversed at least one time.

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

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

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

<optional item-2>

INDEX