Burroughs

B 6700

NETWORK DEFINITION LANGUAGE

INFORMATION MANUAL

THIS MANUAL REPLACES FORM NO. 5000078 DATED NOVEMBER 1970.

'ed in U.S. America

08-71

5000078

LIST OF EFFECTIVE PAGES

NOTE: Insert latest changed page; dispose of superseded pages.

TOTAL NUMBER OF PAGES IN THIS MANUAL IS 146 CONSISTING OF THE FOLLOWING:

Page No.	Issue	Page No.	Issue
Title	Original		
List of Effective Pages ...	Original		
Table of Contents	Original		
Introduction	Original		
1-1 thru 1-5	Original		
1-6 Blank	Original		
2-1 thru 2-94	Original		
3-1 thru 3-17	Original		
3-18 Blank	Original		
A-1 thru A-3	Original		
A-4 Blank	Original		
B-1 thru B-15	Original		
B-16 Blank	Original		

B6700 NETWORK DEFINITION LANGUAGE

TABLE OF CONTENTS

PREFACE . . . . . PAGE 1
INTRODUCTION. . . . . PAGE 3
1. GENERAL INFORMATION. 1- 1
1.1. LANGUAGE CHARACTERISTICS. 1- 1
1.1.1. DESCRIPTION OF THE LANGUAGE 1- 1
1.1.2. TYPES OF DEFINITIONS. 1- 2
1.1.3. STRUCTURE OF THE NETWORK DEFINITION LANGUAGE. . . . 1- 3
1.1.4. STATEMENT REQUIREMENTS AND DEFAULTS 1- 5
2. STRUCTURE OF THE LANGUAGE. 2- 1
2.1. SYNTAX CONVENTIONS. 2- 1
2.2. BASIC ELEMENTS. 2- 3
2.3. SYNTAX AND SEMANTICS OF SECTIONS. 2- 9
2.3.1. CONSTANT SECTION. 2- 10
2.3.2. REQUEST SECTION 2- 12
2.3.3. MODEM SECTION 2- 44
2.3.4. TERMINAL SECTION. 2- 48
2.3.5. STATION SECTION 2- 65
2.3.6. LINE SECTION. 2- 77
2.3.7. DCP SECTION 2- 89
2.3.8. FILE SECTION. 2- 92
3. USING THE NETWORK DEFINITION LANGUAGE PROCESSOR. 3- 1
3.1. PREPARING PROGRAMS. 3- 1
3.1.1. CONTROL CARDS AND FILES 3- 1
3.1.2. USING OPTIONS 3- 3
3.2. RESERVED WORDS. 3- 5
3.3. DIAGNOSTIC MESSAGES 3- 7
APPENDIX. . . . . A- 1
INDEX OF METALINGUISTIC VARIABLES . . . . . B- 1

PREFACE

PURPOSE

THIS DOCUMENT IS INTENDED FOR SOMEONE WHO IS RESPONSIBLE FOR PLANNING AND INITIALIZING THE DATA COMMUNICATION NETWORK ON A B6700 INSTALLATION. THE PERSON SHOULD HAVE INFORMATION FOR THE DEVICES WHICH THE INSTALLATION INTENDS TO USE FOR REMOTE ENTRY TO THEIR SYSTEM, AND THE ORGANIZATION OF ACTIVITIES ON THAT SYSTEM.

PREREQUISITE INFORMATION

BEFORE ATTEMPTING TO CREATE THE DEFINITION OF A NETWORK FOR AN INSTALLATION, THE INDIVIDUAL RESPONSIBLE FOR THIS MUST BE AWARE OF CERTAIN ELEMENTS, SUCH AS THE FOLLOWING:

CHARACTERISTICS OF TERMINAL DEVICES; WHAT HARDWARE WILL BE AVAILABLE FOR DATA COMMUNICATIONS, SUCH AS PERIPHERAL PROCESSORS, MODEMS AND ADAPTERS; WHAT APPLICATIONS THESE DEVICES HANDLE AND HOW TO RUN THEM EFFICIENTLY; WHAT SECURITY MEASURES ARE USED, AND HOW THEY ARE TO BE ENFORCED FOR THE REMOTE USERS.

RELATED PUBLICATIONS

PREREQUISITE INFORMATION ABOUT THE B6700 AND ITS DATACOM CAPABILITY IS INCLUDED WITHIN THE FOLLOWING DOCUMENTS:

FUNCTIONAL DESCRIPTION OF THE B6700 DATA COMMUNICATION SYSTEM (DOCUMENT 5000060).

B6700 DATA COMMUNICATIONS ALGOL LANGUAGE (DOCUMENT 5000052).

OF THESE, DETAILED UNDERSTANDING OF THE FIRST IS CRUCIAL TO RELATING THE NETWORK DEFINITION LANGUAGE TO THE REMAINING SOFTWARE.

DOCUMENT

THE INTRODUCTION TO THIS DOCUMENT PROVIDES A DESCRIPTION OF THE PURPOSE AND STRUCTURE OF THE LANGUAGE. THIS IS FOLLOWED BY A SECTION-BY-SECTION PRESENTATION OF THE SYNTAX AND SEMANTICS, AND, IN SECTION 3, EXPLANATIONS AND EXAMPLES OF THE PROCEDURES USED TO RUN THE SYSTEM TO GENERATE THE TABLES AND DCP OPERATING SYSTEM(S). IN THE APPENDIX, A SET OF USEFUL TABLES ARE INCLUDED.

INTRODUCTION

A DATA COMMUNICATION NETWORK INCLUDES THE REMOTE DEVICES WHICH ARE ATTACHED TO A CENTRAL COMPUTER AND THE VARIOUS MEANS BY WHICH THESE ARE ATTACHED (I.E., LINES, ADAPTERS, MODEMS, DATA COMMUNICATIONS PROCESSORS). REMOTE DEVICES VARY IN HOW THEY COMMUNICATE WITH THE SYSTEM, AS DO THEIR MEANS OF COMMUNICATION. THE USE OF PARTICULAR COMPONENTS OF A NETWORK MAY VARY FROM INSTALLATION TO INSTALLATION. THIS REQUIRES THAT AN INTERFACE FOR THE REMOTE NETWORK AND THE CENTRAL SYSTEM BE FLEXIBLE, I.E., CAPABLE OF HANDLING A VARIETY OF VARYING ASPECTS OF THE NETWORK.

THE NETWORK DEFINITION LANGUAGE ALLOWS AN INSTALLATION TO TAILOR PORTIONS OF THE DATACOM INTERFACE TO THEIR OWN NETWORK AND TO VARY THAT INTERFACE EASILY AND QUICKLY. THE PROCESSES WHICH FORM THE INTERFACE FOR DATA COMMUNICATION ON A B6700 USE TABLES TO DETERMINE SOME CRITICAL FEATURES OF THE NETWORK. THESE "TABLE-DRIVEN" PROCESSES NEED NOT BE MODIFIED TO ADAPT THEM TO CHANGES IN THE NETWORK, BUT RATHER CHANGING THE CONTENT OF THE TABLES ACCOMPLISHES THIS. THE NETWORK DEFINITION LANGUAGE LETS THE INSTALLATION INITIALLY CREATE THESE TABLES, AND WHEN CRITICAL FEATURES VARY, CHANGE THEM EASILY.

IN ADDITION TO PROCESSES RUNNING ON THE B6700 FOR DATA COMMUNICATION, THE SYSTEM HAS A PROGRAMMABLE PERIPHERAL PROCESSOR FOR DATA COMMUNICATION CALLED THE "DCP" WHICH STANDS FOR "DATA COMMUNICATION PROCESSOR". THIS PROCESSOR REQUIRES AN OPERATING SYSTEM TAILORED TO THE INSTALLATIONS NETWORK TO OPERATE EFFICIENTLY.

THE NETWORK DEFINITION LANGUAGE PARTIALLY DETERMINES ACTUAL CODE FOR THE DCP. THIS RELIEVES THE USER OF PROGRAMMING IN SOMETHING LIKE AN ASSEMBLY LANGUAGE, OR OF HAVING TO LIMIT THE DCP TO A SET OF COMMONLY USED PACKAGED FUNCTIONS. AN APPROPRIATE OPERATING SYSTEM FOR THE DCP IS GENERATED VIA THE NETWORK DEFINITION LANGUAGE. IT WILL CONTAIN ONLY THE CODE NECESSARY FOR DEVICES ACTUALLY SPECIFIED (E.G. IF THERE WERE NO TC500 TERMINALS, IT NEED NOT ACCOMMODATE THEM.) AND IT MAY CONTAIN ROUTINES TO PERFORM COMBINATIONS OF ACTIVITIES WHICH WERE UNFORESEEABLE PRIOR TO A NEW TERMINAL ARRIVING.

1. GENERAL INFORMATION

1.1. LANGUAGE CHARACTERISTICS

1.1.1. DESCRIPTION OF THE LANGUAGE

THE NETWORK DEFINITION LANGUAGE IS NOT A PROGRAMMING LANGUAGE, BUT RATHER A DESCRIPTIVE LANGUAGE. RATHER THAN PRODUCING MACHINE-EXECUTABLE CODE DIRECTLY FROM THE STATEMENTS, TABLES ARE PRODUCED WHICH ARE USED BY THE SOFTWARE OF THE DATA COMMUNICATIONS SYSTEM. THE TABLES ARE SUMMARIES OF INFORMATION ABOUT THE VARIOUS ASPECTS OF THE REMOTE NETWORK. THIS ALSO MAKES A DIFFERENCE IN THE TYPES OF STATEMENTS IN THIS LANGUAGE. AS AN EXAMPLE OF DIFFERENCE BETWEEN A PROGRAMMING LANGUAGE AND A DESCRIPTIVE LANGUAGE, CONSIDER HANDLING A TERMINAL "ON-LINE" TO A B6700, WHICH SENDS CHARACTERS IN A CODE WHICH SHOULD BE TRANSLATED TO THE EBCDIC CHARACTER CODE FOR INTERNAL STORAGE (TO BE READ BY, FOR EXAMPLE, AN OBJECT JOB), THE PROGRAMMER WOULD SAY HOW THE CHARACTERS ARE TO BE TRANSLATED, SUCH AS "PROCEDURE TRANSLATE . . ." AND, THEN TELL HOW THIS IS TO BE DONE. ON THE OTHER HAND, IN A DESCRIPTIVE LANGUAGE LIKE THE NETWORK DEFINITION LANGUAGE, THE TERMINAL IS DESCRIBED AND ONE OF ITS CHARACTERISTICS IS ITS CHARACTER CODE, AND ITS DESCRIPTION INCLUDES A STATEMENT, "CODE = . . . " WHERE THE "PROGRAMMER" GIVES THE NAME OF THE TERMINAL CHARACTER CODE, LIKE BCD, BCL OR ONE OF MANY THAT ARE RECOGNIZED BY THE LANGUAGE PROCESSOR.

1.1.2. TYPES OF DEFINITIONS

THE NETWORK DEFINITION LANGUAGE (NDL) IS USED TO DESCRIBE THREE DIFFERENT ASPECTS OF THE REMOTE NETWORK

A. THE LINE DISCIPLINE DEFINITION INCLUDES A SET OF SPECIFIED PROCEDURES CALLED "REQUESTS" WHICH INDICATE HOW TO COMMUNICATE WITH THE REMOTE NETWORK. PROCESSES ARE ENTERED AS INPUT AND/OR PRODUCED AS OUTPUT IN THE SELECTED FORMAT. BOTH INPUT AND/OR OUTPUT PROCESSES RETURN AN ENTRY IN THE RESULT QUEUE TO THE SYSTEM, OR SWITCH TO OTHER PROCESSES.

B. THE PHYSICAL NETWORK DEFINITION PORTION OF THE NETWORK DEFINITION LANGUAGE IS USED TO DEFINE THE PHYSICAL NETWORK IN TERMS OF THE NUMBER OF DATA COMMUNICATIONS PROCESSORS ON THE SYSTEM, THE NUMBER OF LINES AND LINE ATTRIBUTES FOR EACH DCP, THE NUMBER OF STATIONS AND STATION ATTRIBUTES.

C. FILE DEFINITIONS ALLOW SETS OF STATIONS TO BE GROUPED INTO THE LOGICAL ENTITIES FILES. FILE NAMES ARE USED BY OBJECT PROGRAMS TO IDENTIFY THE STATION OR SET OF STATIONS WITH WHICH THEY COMMUNICATE.

1.1.3. STRUCTURE OF THE NETWORK DEFINITION LANGUAGE

A DESCRIPTION IS THE ONLY MEANINGFUL CONSTRUCT IN THE NETWORK DEFINITION LANGUAGE. ANY CONSTRUCT OR SET OF CONSTRUCTS NOT SUBSUMED UNDER A DESCRIPTION IS SYNTACTICALLY DEVIANT (I.E. AN ERROR). A DESCRIPTION IS THE PHYSICAL AND LOGICAL SPECIFICATION OF A DATA COMMUNICATIONS NETWORK.

A NETWORK DESCRIPTION IS AN ORDERLY ARRANGEMENT OF A SERIES OF SECTIONS. THE TABLE BELOW ILLUSTRATES THE ORDER AND FUNCTIONS OF THE SECTIONS. THE CONSTANT SECTION IS OPTIONAL BUT PROVIDES THE USER WITH A WAY TO USE STRINGS OF CHARACTERS AND SPECIAL CHARACTERS (SUCH AS THE DATACOM CONTROL CHARACTERS LIKE STX, AND ETX). FOR WHICH NO EBCDIC GRAPHIC EXISTS. ANOTHER TYPE OF SECTION PROVIDED FOR PROGRAMMING CONVENIENCE IS THE DEFAULT, WHICH ALLOWS THE USER TO DEFINE COMMON SETS OF CHARACTERISTICS A SINGLE TIME PRECEDING THE SECTION IN WHICH THEY ARE TO BE USED.

WITHIN THE SECTIONS (OTHER THAN THE REQUEST SECTION) THE ORDER IN WHICH THE STATEMENTS ARE LISTED IS NOT SIGNIFICANT.

NAME OF SECTION	FUNCTION
CONSTANT	EQUATE NAMES WITH STRINGS FOR CONVENIENCE IN OTHER DESCRIPTIONS
REQUEST	DEFINE LINE DISCIPLINES USED FOR REMOTE DEVICES
MODEM	DESCRIBE MODEMS USED FOR LINES
TERMINAL	DESCRIBE PHYSICAL "HARDWARE" CHARACTERISTICS OF REMOTE DEVICES
STATION	DESCRIBE LOGICAL ASPECTS AND "FILE" ATTRIBUTES FOR SPECIFIED REMOTE DEVICES
LINE	SPECIFY TYPE OF LINE, ITS ADAPTER AND CONNECTED STATION(S)
DCP	SPECIFY MEMORY AND TRANSLATORS AVAILABLE ON DCP

FILE RELATED SETS OF STATIONS INCLUDED AS A FILE

THE LANGUAGE PROCESSOR WILL CHECK FOR CONTRADICTORY STATEMENTS AND/OR MISSING STATEMENTS AT THE END OF EVERY SPECIFICATION. AFTER EACH TERMINAL, STATION, LINE, DCP, AND FILE DEFINITION HAS BEEN SCANNED, A SUMMARY OF THE ERRORS WILL BE PRINTED OUT IF THE SECTION IS ERRONEOUS OR INCOMPLETE.

1.1.4. STATEMENT REQUIREMENTS AND DEFAULTS

BECAUSE OF THE DIFFERENT CHARACTERISTICS OF VARIOUS REMOTE DEVICES, IT IS POSSIBLE FOR THE REQUIRED STATEMENTS IN A DESCRIPTION TO VARY FROM ONE DEFINITION TO ANOTHER. THIS UNUSUAL NATURE OF NDL CAN BE BRIEFLY SUMMARIZED BY MAKING THE FOLLOWING OBSERVATIONS ABOUT THE STATEMENTS MAKING UP A DESCRIPTION:

A. SOME OF THE ATTRIBUTES DEFINED IN THE SYNTAX ARE REQUIRED.

B. SOME OF THE ATTRIBUTES ARE DEPENDENT ON THE PRESENCE OF OTHER ATTRIBUTES.

C. SOME OF THE ATTRIBUTES DEFINED IN THE SYNTAX ARE ENTIRELY OPTIONAL.

THE INTERDEPENDENCY OF THE VARIOUS ATTRIBUTES WILL BE POINTED OUT IN THE SEMANTICS OF EACH SECTION OF THE MANUAL.

DEFAULTS

THE NDL COMPILER AND DATA COMMUNICATIONS SYSTEM AS A WHOLE MAKE NO ASSUMPTIONS FOR ANY PART OF A NETWORK. THE DEFAULT LISTS WHICH WILL BE DEFINED IN THE SYNTAX THROUGHOUT THIS MANUAL PROVIDE THE USER WITH A CONVENIENT METHOD OF GROUPING UNDER ONE OR MORE IDENTIFIERS, SETS OF ATTRIBUTES FOR TERMINALS, STATIONS, LINES, AND FILES.

THE COMPILER WILL NOT CHECK FOR CONTRADICTORY STATEMENTS WITHIN A DEFAULT DEFINITION. THE ATTRIBUTES SPECIFIED WITHIN A DESCRIPTION ALWAYS OVERRIDE THOSE LISTED IN A DEFAULT DEFINITION WHENEVER CONFLICTS ARISE.

2. STRUCTURE OF THE LANGUAGE

2.1. SYNTAX CONVENTIONS

THIS SECTION DISCUSSES THE BACKUS-NAUR FORM (BNF) USED TO DESCRIBE THE SYNTAX OF THE NETWORK DEFINITION LANGUAGE

METALINGUISTIC SYMBOLS

THE FOLLOWING METALINGUISTIC SYMBOLS WILL BE USED IN THIS DOCUMENT:

< > LEFT AND RIGHT BROKEN BRACKETS ARE USED TO CONTAIN ONE OR MORE DIGITS AND/OR LETTERS REPRESENTING A METALINGUISTIC VARIABLE WHOSE DEFINITION IS GIVEN BY A METALINGUISTIC FORMULA.

::= THE SYMBOL "::=" MEANS "IS DEFINED AS". THE METALINGUISTIC VARIABLE TO THE LEFT OF THIS SYMBOL IS DEFINED BY THE METALINGUISTIC FORMULA ON ITS RIGHT.

/ THE SYMBOL "/" MEANS "OR". IT SEPARATES ALTERNATIVE DEFINITIONS OF A METALINGUISTIC VARIABLE.

≤≥ THE "≤" AND "≥" SYMBOLS ARE USED TO ENCLOSE ENGLISH LANGUAGE EXPRESSIONS WHEN SUCH EXPRESSIONS ARE NECESSARY TO DEFINE A METALINGUISTIC VARIABLE.

METALINGUISTIC FORMULAS

METALINGUISTIC SYMBOLS ARE USED IN FORMING A METALINGUISTIC FORMULA. A METALINGUISTIC FORMULA IS A RULE WHICH WILL PRODUCE AN ALLOWABLE SEQUENCE OF CHARACTERS AND/OR SYMBOLS. THESE FORMULAS ARE USED TO DEFINE THE SYNTAX OF THE B6700 NETWORK DEFINITION LANGUAGE. THE SYNTAX, COMBINED WITH THE SEMANTICS, CONTAINED IN THIS MANUAL DEFINES THE B6700 NETWORK DEFINITION LANGUAGE.

ANY MARK OR SYMBOL IN A METALINGUISTIC FORMULA WHICH IS NOT ONE OF THE METALINGUISTIC SYMBOLS, IS EQUIVALENT TO ITSELF. THE JUXTAPOSITION OF THE METALINGUISTIC VARIABLES AND/OR SYMBOLS IN A METALINGUISTIC FORMULA DENOTES THE JUXTAPOSITION OF THOSE ELEMENTS IN THE CONSTRUCT INDICATED.

AN EXAMPLE OF A METALINGUISTIC FORMULA IS:

<IDENTIFIER> ::= <LETTER> / <IDENTIFIER> <LETTER> /
<IDENTIFIER> <DIGIT>

THIS METALINGUISTIC FORMULA IS READ:

AN IDENTIFIER IS DEFINED AS A LETTER, OR AN IDENTIFIER FOLLOWED BY A LETTER, OR AN IDENTIFIER FOLLOWED BY A DIGIT.

THE METALINGUISTIC FORMULA ABOVE DEFINES A RECURSIVE RELATIONSHIP BY WHICH A CONSTRUCT CALLED AN IDENTIFIER MAY BE FORMED. THAT IS, EVALUATION OF THE FORMULA SHOWS THAT AN IDENTIFIER BEGINS WITH A LETTER. THE LETTER MAY STAND ALONE, OR MAY BE FOLLOWED BY ANY SEQUENCE OF LETTERS AND DIGITS.

2.2. BASIC ELEMENTS

CHARACTER SET

THE CHARACTER SET FOR WHICH THE LANGUAGE IS DEFINED IS DRAWN FROM THE EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC) CHARACTER SET.

SYNTAX

```
<LETTER> ::= A / B / C / D / E / F / G / H / I / J / K / L / M /
      N / O / P / Q / R / S / T / U / V / W / X / Y / Z
<DIGIT> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9
<SPECIAL CHARACTER> ::= . / , / ≤ / ≥ / ( / ) / + / - /
      <SLASH> / < / > / = / % / : / ? / " / ;
<SLASH> ::= /
<STRING CHARACTER> ::= <LETTER> / <DIGIT> / <SPECIAL CHARACTER> /
      <SINGLE SPACE>
<SINGLE SPACE> ::= [ONE HORIZONTAL BLANK POSITION ]
<SPACE> ::= <SINGLE SPACE> / <SPACE> <SINGLE SPACE>
<CHARACTER> ::= <STRING CHARACTER> / <STRING BRACKET CHARACTER>
<STRING BRACKET CHARACTER> ::= "
```

SEMANTICS

THE CHARACTER SET FOR NDL IS A 52 CHARACTER SUBSET OF THE EBCDIC CHARACTER SET CONTAINING LETTERS, DIGITS, SPECIAL CHARACTERS, AND THE SPACE.

BASIC SYMBOLS

SYNTAX

```
<BASIC SYMBOL> ::= <LETTER> / <DIGIT> / <LOGICAL VALUE> /
      <DELIMITER> / <EMPTY>
<DELIMITER> ::= <OPERATOR> / <SEPARATOR> / <BRACKET> / CONSTANT /
      REQUEST / MODEM / TERMINAL DEFAULT / TERMINAL /
      STATION DEFAULT / STATION / LINE DEFAULT / LINE / DCP /
      FILE DEFAULT / FILE
<EMPTY> ::= ≤THE NULL STRING OF SYMBOLS≥
<OPERATOR> ::= <ADD OPERATOR> / <RELATIONAL OPERATOR> /
      <SEQUENTIAL OPERATOR> / <ASSIGNMENT OPERATOR>
<ADD OPERATOR> ::= + / -
<RELATIONAL OPERATOR> ::= GT / GE / LS / LE / NE / EQ / < / > / =
<SEQUENTIAL OPERATOR> ::= GO / GO TO / IF / THEN / ELSE
<ASSIGNMENT OPERATOR> ::=  =
<SEPARATOR> ::= , / . / : / <SPACE>
<BRACKET> ::= ( / ) / ≤ / ≥ / " / BEGIN / END
```

SEMANTICS

ONLY UPPER-CASE LETTERS ARE PERMITTED. DELIMITERS SEPARATE THE VARIOUS ENTITIES THAT MAKE UP A NETWORK DEFINITION.

EACH DELIMITER HAS A FIXED MEANING WHICH IS EXPLAINED IN THIS DOCUMENT IN THE SYNTAX AND SEMANTICS OF THE APPROPRIATE CONSTRUCTS.

A <SPACE> MUST SEPARATE ANY TWO OF THE FOLLOWING:

MULTICHARACTER DELIMITER
LOGICAL VALUE
IDENTIFIER
INTEGER
GENERALIZED IDENTIFIER

BASIC COMPONENTS

SYNTAX

<BASIC COMPONENT> ::= <IDENTIFIER> / <GENERALIZED IDENTIFIER> /
<INTEGER> / <STRING> / <LOGICAL VALUE> / <TIME>

SEMANTICS

BASIC COMPONENTS ARE THE MOST PRIMITIVE STRUCTURES OF THE LANGUAGE.

IDENTIFIERS

SYNTAX

<IDENTIFIER> ::= <LETTER> / <IDENTIFIER> <LETTER> /
<IDENTIFIER> <DIGIT>

SEMANTICS

IDENTIFIERS HAVE NO INTRINSIC MEANING. IDENTIFIERS ARE USED TO NAME LABELS, CONSTANTS, REQUESTS, TERMINALS, AND LINES.

THE MAXIMUM LENGTH OF AN IDENTIFIER IS 17 CHARACTERS. SPACES MAY NOT APPEAR AS PART OF AN IDENTIFIER.

GENERALIZED IDENTIFIERS

SYNTAX

<GENERALIZED IDENTIFIER> ::= <IDENTIFIER COMPONENT> /

<GENERALIZED IDENTIFIER> <SLASH> <IDENTIFIER COMPONENT>
<IDENTIFIER COMPONENT> ::= <IDENTIFIER> / <STRING>

SEMANTICS

A <GENERALIZED IDENTIFIER> IS USED TO NAME A STATION OR A FILE. A <GENERALIZED IDENTIFIER> MAY CONTAIN A MAXIMUM OF 14 <IDENTIFIER COMPONENT>S SEPARATED BY SLASHES. A <STRING> USED AS AN <IDENTIFIER COMPONENT> MUST BE LESS THAN OR EQUAL TO 17 CHARACTERS.

INTEGERS

SYNTAX

<INTEGER> ::= <DIGIT> / <INTEGER> <DIGIT>

SEMANTICS

ONLY POSITIVE INTEGERS ARE ALLOWED IN NDL. SPACES MAY NOT APPEAR WITHIN AN INTEGER.

STRINGS

SYNTAX

```
<STRING> ::= <SIMPLE STRING> / <UNITARY STRING> / <STRING> <STRING>
<SIMPLE STRING> ::= <HEXADECIMAL STRING> / <EBCDIC STRING>
<HEXADECIMAL STRING> ::= 4"<HEXADECIMAL CHARACTER CONCATENATION>"
<HEXADECIMAL CHARACTER CONCATENATION> ::= <HEXADECIMAL CHARACTER>/
      <HEXADECIMAL CHARACTER CONCATENATION> <HEXADECIMAL CHARACTER>
<EBCDIC STRING> ::= "<EBCDIC CHARACTER CONCATENATION>"
<EBCDIC CHARACTER CONCATENATION> ::= <EBCDIC CHARACTER> /
      <EBCDIC CHARACTER CONCATENATION> <EBCDIC CHARACTER>
<UNITARY STRING>::= 4"<HEXADECIMAL CHARACTER>
      <HEXADECIMAL CHARACTER>" / "<EBCDIC CHARACTER>"
<HEXADECIMAL CHARACTER> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 /
      9 / A / B / C / D / E / F
<EBCDIC CHARACTER> ::= <STRING CHARACTER>
```

SEMANTICS

ONLY HEXADECIMAL (4-BIT) OR EBCDIC (8-BIT) STRINGS MAY BE DEFINED.

THE MAXIMUM LENGTH OF A STRING IS 64 CHARACTERS. THE LENGTH OF A <STRING> IS THE MAXIMUM NUMBER OF 8-BIT CHARACTERS REQUIRED TO HOLD THE <STRING>. IF CONCATENATION IS USED AND DIFFERENT CHARACTER SIZES ARE USED, EACH CHARACTER IN A <STRING> MUST BE ALIGNED AT A CHARACTER BOUNDARY APPROPRIATE FOR THAT CHARACTER SIZE.

A SIMPLE <HEXADECIMAL STRING> MUST CONSIST OF AN EVEN NUMBER OF HEXADECIMAL CHARACTERS. THUS, ALL STRINGS ARE EBCDIC STRINGS. (THE DATACOM SYSTEM IS INTERNALLY EBCDIC. WHERE APPROPRIATE ALL CHARACTERS ARE TRANSLATED, AS FOR EXAMPLE, WHEN TRANSMITTING TO A NON-EBCDIC DEVICE.)

STRINGS CONTAINING INTERNAL QUOTES MUST BE BROKEN INTO SEPARATE <STRINGS> CONTAINING THREE QUOTES IN SUCCESSION.

LOGICAL VALUES

SYNTAX

<LOGICAL VALUE> ::= TRUE / FALSE

SEMANTICS

A LOGICAL VALUE CONSISTS OF THE TWO POSSIBLE CONDITIONS THAT A BOOLEAN MAY ASSUME

TIME

SYNTAX

<TIME> ::= <INTEGER> <UNIT> / 0
<UNIT> ::= MICRO / MILLI / SEC / MIN

SEMANTICS

INTERVALS OF TIME MAY BE EXPRESSED IN MICROSECONDS, MILLISECONDS, SECONDS AND MINUTES BY "MICRO", "MILLI", "SEC", AND "MIN" RESPECTIVELY.

2.3. SYNTAX AND SEMANTICS OF SECTIONS

SYNTAX

<NETWORK DESCRIPTION> ::= <CONSTANT SECTION> <REQUEST SECTION>
<MODEM SECTION> <TERMINAL SECTION> <STATION SECTION>
<LINE SECTION> <DCP SECTION> <FILE SECTION>

SEMANTICS

THE SECTION-BY-SECTION EXPLICATION FOLLOWING IS ORDERED BY THE ORDER IN WHICH SECTIONS MUST APPEAR.

THE VARIOUS PARTS OF A <NETWORK DESCRIPTION> AS LISTED ABOVE WILL BE DEFINED BOTH SEMANTICALLY AND SYNTACTICALLY IN SUBSEQUENT SECTIONS OF THIS GUIDE.

2.3.1. CONSTANT SECTION

SYNTAX

```
<CONSTANT SECTION> ::= <CONSTANT LIST> / <EMPTY>
<CONSTANT LIST> ::= CONSTANT <CONSTANT DEFINITION LIST>. /
      <CONSTANT LIST> <CONSTANT LIST>
<CONSTANT DEFINITION LIST> ::= <CONSTANT DEFINITION> /
      <CONSTANT DEFINITION LIST> , <CONSTANT DEFINITION>
<CONSTANT DEFINITION> ::= <CONSTANT IDENTIFIER> = <STRING>
<CONSTANT IDENTIFIER> ::= <IDENTIFIER>
```

SEMANTICS

THE <CONSTANT LIST> PROVIDES A MEANS FOR EQUATING IDENTIFIERS WITH <STRING>S. THE <CONSTANT IDENTIFIER> IN A <CONSTANT DEFINITION> IS EQUATED WITH THE <STRING> IN THE <CONSTANT DEFINITION>.

THE OCCURRENCE OF A <CONSTANT IDENTIFIER> OUTSIDE OF A <CONSTANT DEFINITION> IS SYNTACTICALLY AND SEMANTICALLY EQUIVALENT TO THE OCCURRENCE OF THE <STRING> NAMED BY THE <CONSTANT IDENTIFIER>.

EXAMPLES OF CONSTANTS

1. VALID

```
  CONSTANT NUL = 4"00",
           BAD1 = "ERROR",
           RETRANSMITREQUEST = "PLEASE SEND AGAIN".
  CONSTANT QUOTE = """.
```

II. INVALID

```
CONSTANT 12KANGAROO =          % IDENTIFIER MUST BEGIN WITH LETTER.
                    12.        % NOT A STRING.
CONSTANT CLEAR =               % RESERVED WORD MAY NOT
                               % BE USED AS CONSTANT
                 40"000".      % MALFORMED STRING.
```

(PERCENT SIGN (%) ALLOWS COMMENTS TO BE INCLUDED WITH NDL STATEMENTS BECAUSE THE COMPILER WILL IGNORE REMAINING COLUMNS UP TO SEQUENCE STATEMENTS WHEN THE COMPILER ENCOUNTERS A PERCENT SIGN NOT IN A STRING, SO "CONSTANT PERCENT = "%" . " WILL GIVE THE USER A CONSTANT WHOSE VALUE IS A PERCENT SIGN (%).)

2.3.2. REQUEST SECTION

SYNTAX

```
<REQUEST SECTION> ::= <REQUEST LIST>
<REQUEST LIST> ::= <REQUEST DEFINITION> /
      <REQUEST LIST> <REQUEST DEFINITION>
<REQUEST DEFINITION> ::= REQUEST
      <REQUEST IDENTIFIER> : <REQUEST STATEMENT LIST>
<REQUEST IDENTIFIER> ::= <IDENTIFIER>
<REQUEST STATEMENT LIST> ::= <REQUEST STATEMENT>. /
      <REQUEST STATEMENT LIST> <REQUEST STATEMENT>./ <EMPTY>
```

SEMANTICS

THE REQUEST SECTION CONSISTS OF A SIMPLIFIED PROGRAMMATIC LANGUAGE USED TO DEFINE THE LINE DISCIPLINE ROUTINES USED BY THE DCP IN COMMUNICATING WITH THE VARIOUS REMOTE DEVICES.

A REQUEST MUST BE DEFINED FOR EACH CAPABILITY OF A TERMINAL. THAT IS, IF IT IS POSSIBLE FOR A TERMINAL TO SEND INPUT TO THE SYSTEM AND RECEIVE OUTPUT FROM THE SYSTEM THEN BOTH A RECEIVE AND TRANSMIT REQUEST MUST BE DEFINED FOR THAT TERMINAL. THE <REQUEST IDENTIFIER> TO BE USED FOR EACH OF THESE CONDITIONS IS DEFINED THROUGH THE USE OF THE <TERMINAL REQUEST STATEMENT> AS DISCUSSED IN THE TERMINAL DESCRIPTION SECTION OF THIS MANUAL.

THE DCP INITIATES THESE REQUESTS AS DEFINED IN NDL UPON COMMAND FROM THE VARIOUS PARTS OF THE DATA COMMUNICATIONS SYSTEM. FOR INSTANCE, IF IT IS NECESSARY FOR AN OBJECT PROGRAM TO OUTPUT A MESSAGE TO A TC500 TERMINAL, THE SYSTEM FIRST INFORMS THE DCP AS TO WHICH STATION THE MESSAGE IS TO GO TO, THEN THE DCP SELECTS THE TRANSMIT REQUEST AS SPECIFIED IN THE NETWORK DEFINITION FOR THE DESIRED STATION, AND EXECUTES THE SERIES OF COMMANDS AS DESCRIBED

IN THE <REQUEST DEFINITION>. A SIMILAR SERIES OF EVENTS TAKES PLACE WHEN A STATION WISHES TO SEND A MESSAGE TO THE SYSTEM.

REQUEST STATEMENTS

SYNTAX

<REQUEST STATEMENT> ::= <LABELED REQUEST STATEMENT> /
<UNLABELED REQUEST STATEMENT>

LABELED REQUEST STATEMENTS

SYNTAX

<LABELED REQUEST STATEMENT> ::= <LABEL> :
<UNLABELED REQUEST STATEMENT>
<LABEL> ::= <INTEGER>

SEMANTICS

STATEMENTS IN THE REQUEST SECTION ARE EXECUTED SEQUENTIALLY AS LISTED. IN SOME CASES, HOWEVER, IT IS DESIRABLE TO ALTER THE ORDER OF EXECUTION OF STATEMENTS. THE <LABELED REQUEST STATEMENT> IS ONE MEANS OF ACCOMPLISHING THIS ACTION. THE MAXIMUM LENGTH OF A LABEL IS 5 DIGITS. THE <LABEL> MUST BE FOLLOWED BY A COLON TO SEPARATE IT FROM THE SUBSEQUENT <REQUEST STATEMENT>. THE <GO TO STATEMENT> IS USED TO UNCONDITIONALLY TRANSFER CONTROL TO A <LABELED REQUEST STATEMENT>. (<SYNTAX IS GIVEN WITH <UNCONDITIONAL REQUEST STATEMENT>).

UNLABELED REQUEST STATEMENTS

SYNTAX

```
<UNLABELED REQUEST STATEMENT> ::= <COMPOUND REQUEST STATEMENT> /
      <CONDITIONAL REQUEST STATEMENT> /
      <UNCONDITIONAL REQUEST STATEMENT>
```

COMPOUND REQUEST STATEMENTS

SYNTAX

```
<COMPOUND REQUEST STATEMENT> ::= BEGIN <COMPOUND TAIL>
<COMPOUND TAIL> ::= <REQUEST STATEMENT> END /
      <REQUEST STATEMENT>. END / <REQUEST STATEMENT>.
      <COMPOUND TAIL>
```

SEMANTICS

TO EXECUTE ONE OR MORE <REQUEST STATEMENT>S WHEN A CERTAIN CONDITION IS SATISFIED A <COMPOUND REQUEST STATEMENT> MUST BE USED. A <COMPOUND REQUEST STATEMENT> IS A SERIES OF <REQUEST STATEMENT>S PRECEDED BY A BEGIN AND FOLLOWED BY AN END. THE STATEMENTS COMPRISING A <COMPOUND REQUEST STATEMENT> MAY BE <LABELED REQUEST STATEMENT>S, <UNLABELED REQUEST STATEMENT>S, OR OTHER <COMPOUND REQUEST STATEMENT>S. EACH <REQUEST STATEMENT> MUST BE FOLLOWED BY A PERIOD EXCEPT FOR THE LAST STATEMENT PRECEDING THE END DELIMITER, IN WHICH CASE THE PERIOD IS OPTIONAL.

EXAMPLE:

```
IF BUFOVFL THEN
      BEGIN
      STORE "LOST DATA-RETRANSMIT".
      GO TO 1
      END.
```

CONDITIONAL REQUEST STATEMENTS

SYNTAX

<CONDITIONAL REQUEST STATEMENT > ::= IF <LOGICAL EXPRESSION> THEN
<REQUEST STATEMENT> <ELSE PART>
<LOGICAL EXPRESSION> ::= <TOGGLE> / <RELATIONAL EXPRESSION> /
<LOGICAL VALUE>
<ELSE PART> ::= ELSE <REQUEST STATEMENT> / <EMPTY>

SEMANTICS

THE <CONDITIONAL REQUEST STATEMENT> OFFERS A MEANS BY WHICH PROGRAM CONTROL MAY BE "CONDITIONALLY" ALTERED DEPENDING ON PROGRAMMATIC CHECKS SPECIFIED BY THE USER. THE <LOGICAL EXPRESSION> SYNTAX PROVIDES FOR A VARIETY OF CHECKS WHICH MAY BE DONE BY THE USER, EACH OF WHICH IS DISCUSSED BELOW. NOTE: THE "THEN" MUST BE FOLLOWED BY A <REQUEST STATEMENT>. ("IF <TOGGLE> THEN ELSE <REQUEST STATEMENT> " IS NOT ALLOWED).

TOGGLES

SYNTAX

<TOGGLE> ::= <USER TOGGLE> / <ERROR FLAG> / <VARIANT TOGGLE> /
<STATUS TOGGLE>

SEMANTICS

TOGGLE CAN ASSUME THE VALUE OF TRUE OR FALSE ONLY.

USER TOGGLES

SYNTAX

```
<USER TOGGLE> ::= TOG [<TOGGLE NUMBER>] / NAKONSELECT /
      CONTROLFLAG / WRUFLAG / NAKFLAG
<TOGGLE NUMBER> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7
```

SEMANTICS

IN THE HEADER OF EACH MESSAGE EIGHT (8) BITS HAVE BEEN RESERVED FOR <USER TOGGLE>S TOG[0] THROUGH TOG[7]. THE PROGRAMMER MAY USE THESE TOGGLES IN AN NDL REQUEST AS SUPPLEMENTARY FLAGS. NAKONSELECT IS A TOGGLE PRESERVED IN THE ERROR FLAG FIELD OF A STANDARD MESSAGE CONNOTING THAT A "NAK" (NEGATIVE ACKNOWLEDGE) TO A SELECT WAS RECEIVED. THIS FLAG IS SET EITHER BY THE USER EXPLICITLY OR BY A TERMINATE ENABLEINPUT. THE CONTROL FLAG AND WRU FLAG ARE SET AUTOMATICALLY WHEN THE APPROPRIATE <TEXT CONTROL CHARACTER>S ARE IN A <RECEIVE ACTION PART> AND THE CHARACTERS ARE ENCOUNTERED. THE NAKFLAG IS INTENDED TO BE USED TO FLAG THE CONDITION WHEN TRANMISSION OF THE TEXT WAS "NAK"ED; THIS FLAG IS NEVER IMPLICITLY SET BUT CAN BE USED TO NOTE AN ERROR OF SORTS. ALL <TOGGLES> MAY BE SET AND TESTED INDEPENDENTLY.

ERROR FLAGS

SYNTAX

```
<ERROR FLAG> ::= <RECEIVE ERROR FLAG> / <TRANSMIT ERROR FLAG> /
      <ITEM ERROR FLAG> / BREAK[RECEIVE] /
      BREAK[TRANSMIT] / SEQERR
<RECEIVE ERROR FLAG> ::= PARITY / STOPBIT / BUFOVFL /
      TIMEOUT / BREAK / LOSSOFCARRIER
<TRANSMIT ERROR FLAG> ::= BREAK
```

<ITEM ERROR FLAG> ::= BCCERR / ADDERR / TRANERR / FORMATERR /
ENDOFBUFFER / BLKNERR

SEMANTICS

PARITY — AS ITS NAME IMPLIES, PARITY IS SET WHEN A PARITY ERROR IS DETECTED BY THE DCP WHILE RECEIVING A MESSAGE, BUT ONLY IF PARITY HAS BEEN DEFINED BY A <PARITY STATEMENT> IN THE TERMINAL DESCRIPTION FOR THE STATION.

BCCERR — IS SET IF THE BLOCK CHECK CHARACTER (BCC) COMPUTED BY THE DCP IS NOT EQUIVALENT TO THE BCC RECEIVED FROM A STATION. THE PARITY OF THE BCC IS DEFINED IN THE <PARITY STATEMENT> OF THE TERMINAL DESCRIPTION ASSOCIATED WITH A STATION.

ADDERR — IS SET ON INPUT FROM A STATION WHEN AN ERROR IN THE STATION ADDRESS IS DETECTED. THE SIZE AND VALUE OF THE STATION ADDRESS AS DEFINED IN THE <TERMINAL DESCRIPTION> AND <STATION ATTRIBUTE LIST> ALLOW THE DCP TO CHECK THE CORRECT VALUE AGAINST THAT WHICH IS RECEIVED.

TRANERR — IS SET ON INPUT FROM A STATION IF THE TRANSMISSION NUMBER RECEIVED IS NOT ONE GREATER THAN THE NUMBER OF THE PREVIOUSLY RECEIVED MESSAGE FROM THE SAME STATION.

STOPBIT — WILL BE SET, IF THE DCP IS EXPECTING TO RECEIVE A STOPBIT ON AN ASYNCHRONOUS LINE AS THE NEXT INPUT ON THAT LINE AND DUE TO PARITY OR NOISE THE STOPBIT IS NOT RECOGNIZED BY THE DCP.

BUFOVFL	IS SET IF THE DCP WAS UNABLE TO SERVICE THE "CLUSTER ATTENTION NEEDED" (CAN) INTERRUPT BEFORE THE NEXT CHARACTER WAS RECEIVED ON THE LINE, DESTROYING THE PREVIOUS CHARACTER.
ENDOFBUFFER	MAY BE SET BY TWO CONDITIONS. IN THE CASE OF A RECEIVE REQUEST, THERE IS NO TEXT BUFFER RECEIVED BY THE DCP FROM THE SYSTEM. THUS, AN ENDOFBUFFER CONDITION EXISTS WHEN THE NUMBER OF CHARACTERS STORED IN THE BUFFER OBTAINED BY THE DCP BY IMPLICIT OR EXPLICIT STORE COMMANDS (E.G., STORE "GEORGE...") AND RECEIVE TEXT, EXCEEDS THE MAXIMUM ALLOWED AS SPECIFIED BY EITHER A MAXIMUM INPUT OR BUFFER STATEMENT IN THE TERMINAL DESCRIPTION. FOR A TRANSMIT REQUEST, A TEXT BUFFER CONTAINING THE TEXT TO BE TRANSMITTED IS PRESENTED TO THE DCP BY THE SYSTEM. AN ENDOFBUFFER CONDITION OCCURS WHEN THE NUMBER OF IMPLICITY OR EXPLICITY FETCHED CHARACTERS EXCEEDS THE NUMBER ACTUALLY IN THE BUFFER.
FORMATERR	IS SET IF AN NDL REQUEST USES THE RECEIVE <STRING> OPTION TO INPUT DATA TO THE DCP, AND THE CHARACTERS RECEIVED DO NOT CORRESPOND TO THE <STRING> SPECIFIED.
TIMEOUT	IS SET IF A CHARACTER IS NOT RECEIVED FROM A STATION WITHIN THE TIME SPECIFIED IN THE <TIMEOUT STATEMENT> OF THE TERMINAL DESCRIPTION FOR A STATION; I.E. MAXIMUM TIME BETWEEN INPUT CHARACTERS.
BREAK	IS SET IF A BREAK WAS RECEIVED BY THE DCP.

BREAK[RECEIVE]	REFERS TO "BREAK ON INPUT" AND MUST BE USED EXCEPT WHEN IN RECEIVE ACTION LISTS.
BREAK[TRANSMIT]	REFERS TO "BREAK ON OUTPUT" AND MUST BE USED EXCEPT WHEN IN TRANSMIT ACTION LISTS.
SEQERR	IS SET WHEN AN INCREMENT SEQUENCE STATEMENT CAUSES A SEQUENCE NUMBER LARGER THAN ALLOWED.
LOSSOFCARRIER	IS SET UPON "LOSS OF CARRIER".
BLKNERR	IS SET IF THE RECEIVED BLOCK NUMBER DOES MATCH THE ONE SAVED BY THE DCP.

IN ALL CASES, THE ACTION TO BE TAKEN BY THE DCP WHEN A <TOGGLE> HAS BEEN SET MUST BE SPECIFIED BY THE USER. IN ALL CASES THE ASSOCIATED ERROR FLAG WILL BE SET ON THE OCCURRENCE OF THAT ERROR.

VARIANT TOGGLES

SYNTAX

```
<VARIANT TOGGLE> ::= PAGE / SKIPLINES / CARRIAGE /
                     LINEFEED / BLOCK / PAPERMOTION
```

SEMANTICS

PAGE, WHEN SET, INDICATES PAGE FORMAT CONTROL IS TO BE DONE.

SKIPLINES INDICATES THAT LINES ARE TO BE SKIPPED. (THE METHOD OF INDICATING HOW MANY IS TO BE SPECIFIED).

CARRIAGE IS SET TO INDICATE THAT A CARRIAGE RETURN FORMAT

SHOULD BE BE DONE.

LINEFEED INDICATES THAT LINEFEED FORMAT CONTROL BE SENT.

BLOCK WILL BE SET WHEN THE TEXT CONSISTS OF MORE THAN ONE MESSAGE.

PAPERMOTION, WHEN SET, INDICATES THE A NEW LINE SHOULD BE STARTED BEFORE A LINE IS TRANSMITTED.

STATUS TOGGLE

SYNTAX

<STATUS TOGGLE> ::= SEQUENCE

SEMANTICS

TO BE SPECIFIED

RELATIONAL EXPRESSIONS

SYNTAX

<RELATIONAL EXPRESSION> ::= <VALUE> <RELATIONAL OPERATOR> <VALUE>
<VALUE> ::= <BYTE VARIABLE> / <UNITARY STRING> / <INTEGER>
<BYTE VARIABLE> ::= <CHARACTER REGISTER> / <USER TALLY> / <RETRY> /
<BCC>

SEMANTICS

<RELATIONAL EXPRESSION> PROVIDES A MEANS OF COMPARING THE RELATIONSHIPS BETWEEN THE MEMBERS OF A SET OF <VALUE>S. AS THE DCP USES MODULO 256 ARITHMETIC THE INTEGER <VALUE> IS TAKEN MODULO 256.

CHARACTER REGISTER

SYNTAX

<CHARACTER REGISTER> ::= CHARACTER / CHAR

SEMANTICS

"CHAR" AND "CHARACTER" DENOTE THE SAME THING -- THEY ARE SYNONYMS. THE CONTENTS OF THE <CHARACTER REGISTER> MAY ONLY BE TESTED FOR A SINGLE <VALUE> AT A TIME. THE REGISTER IS LOADED BY AN INSTRUCTION SUCH AS THE FOLLOWING:

```
RECEIVE CHAR.
RECEIVE TEXT.
RECEIVE.
```

THE CHARACTER REGISTER CONTAINS THE LAST CHARACTER RECEIVED OR TRANSMITTED IN MOST CASES. THUS, UPON THE SUCCESSFUL COMPLETION OF RECEIVE TEXT, THE CHARACTER REGISTER CONTAINS THE LAST TEXT CHARACTER RECEIVED. SIMILARLY, UPON SUCCESSFUL COMPLETION OF TRANSMIT TEXT, THE CHARACTER REGISTER CONTAINS THE LAST CHARACTER OF TEXT. THESE GENERAL RULES ARE OVERRIDDEN IN TWO CASES. A FETCH STATEMENT CAUSES THE CHARACTER REGISTER TO BE FILLED WITH THE NEXT CHARACTER OF THE TEXT BUFFER NOT ALREADY FETCHED EXPLICITLY BY A FETCH STATEMENT OR IMPLICITLY BY A TRANSMIT TEXT CONSTRUCT. ALSO AN EXPLICIT ASSIGNMENT STATEMENT SUCH AS CHARACTER = "S" WILL CAUSE A CHANGE IN THE CONTENTS OF THE CHARACTER REGISTER (IN ACCORDANCE

WITH THE RULES OF ASSIGNMENT).

USER TALLYS

SYNTAX

<USER TALLY> ::= TALLY [<TALLY NUMBER>]
<TALLY NUMBER> ::= 0 / 1 / 2

SEMANTICS

<USER TALLY> PROVIDES THE USER WITH THREE SEPARATE BYTES IN THE MESSAGE HEADER WHICH MAY BE INDIVIDUALLY UTILIZED IN ANY MANNER DESIRED. THE MAXIMUM DECIMAL VALUE WHICH MAY BE STORED IN THESE <TALLY>S IS 255. IF A NUMBER IS ASSIGNED TO A <TALLY> WHICH EXCEEDS 255, THE NUMBER WILL BE TRUNCATED ON THE LEFT BEFORE STORING. THE BYTES ARE REFERENCED BY A <TALLY NUMBER> CORRESPONDING TO THE THREE SEPARATE BYTES.

RETRY

SYNTAX

<RETRY> ::= RETRY

SEMANTICS

<RETRY> IS INITIALLY SET BY THE USER. THE MAXIMUM VALUE WHICH MAY BE SPECIFIED FOR RETRY IS 255 DECIMAL. EACH REQUEST TO RECEIVE OR TRANSMIT MADE TO THE DCP CONTAINS THE RETRY COUNT AS LAST GIVEN BY THE ORIGINATOR OF THE REQUEST.

BCC

SYNTAX

<BCC> ::= BCC

SEMANTICS

<BCC> PROVIDES THE USER WITH ACCESS TO THE BLOCK CHECK CHARACTER. A RECEIVE OR TRANSMIT STATEMENT ALWAYS UPDATES THE BCC. (CARE MUST BE TAKEN TO RE-INITIALIZE THE BCC WHEN APPROPRIATE). IN ORDER TO ELIMINATE THOSE CHARACTERS NOT TO BE SUMMED IN THE BCC THE <SUM STATEMENT> SHOULD BE USED.

UNCONDITIONAL REQUEST STATEMENTS

SYNTAX

```
<UNCONDITIONAL REQUEST STATEMENT> ::= <ASSIGNMENT STATEMENT> /
      <GO TO STATEMENT> / <INITIATE STATEMENT> /
      <INITIALIZE STATEMENT> / <FINISH STATEMENT> /
      <FETCH STATEMENT> / <STORE STATEMENT> /
      <SUM STATEMENT> / <TRANSMIT STATEMENT> /
      <RECEIVE STATEMENT> / <TERMINATE STATEMENT> /
      <ERROR ACTION STATEMENT> / <BREAK STATEMENT> /
      <DELAY STATEMENT> / <INCREMENT STATEMENT> /
      <SHIFT STATEMENT>
```

ASSIGNMENT STATEMENTS

SYNTAX

```
<ASSIGNMENT STATEMENT> ::= <LOGICAL ASSIGNMENT STATEMENT> /
      <VALUE ASSIGNMENT STATEMENT>
<LOGICAL ASSIGNMENT STATEMENT> ::= <ASSIGNABLE TOGGLE> = <TOGGLE> /
      <ASSIGNABLE TOGGLE> = <LOGICAL VALUE> / SEQUENCE = FALSE
<ASSIGNABLE TOGGLE> ::= <USER TOGGLE> / <ERROR FLAG>
<VALUE ASSIGNMENT STATEMENT> ::= <BYTE VARIABLE> = <VALUE> /
      <BYTE VARIABLE> = <VALUE> <ADD OPERATOR> <VALUE>
```

SEMANTICS

THE <ASSIGNMENT STATEMENT> IS THE MEANS BY WHICH AN NDL REQUEST ASSIGNS AND/OR UPDATES A <BYTE VARIABLE> OR <ASSIGNABLE TOGGLE> <VALUE ASSIGNMENT>S PRESUME "MODULO 256" ARITHMETIC.

GO TO STATEMENTS

SYNTAX

<GO TO STATEMENT> ::= GO TO <LABEL> / GO <LABEL>

SEMANTICS

THE <GO TO STATEMENT> SPECIFIES THAT THE NEXT INSTRUCTION TO BE EXECUTED IS THE ONE ASSOCIATED WITH THE <LABEL>.

INITIATE STATEMENTS

SYNTAX

<INITIATE STATEMENT> ::= INITIATE <INITIATE ACTION> <DELAY PART>
<INITIATE ACTION> ::= RECEIVE / TRANSMIT
<DELAY PART> ::= (<DELAY VALUE>) / <EMPTY>
<DELAY VALUE> ::= <TIME> / NULL

SEMANTICS

THE INITIATE RECEIVE FORM OF THE <INITIATE STATEMENT> COMMANDS THE ADAPTER, I.E., LINE, TO INITIATE RECEIVE DELAYS. THE DELAYED VALUE IS AS DESCRIBED IN THE SEMANTICS FOR THE MODEM NOISEDELAY STATEMENT.

IF THE <DELAY PART> IS EMPTY OR IS ZERO NO DELAY OCCURS. IF THE <DELAY VALUE> IS "NULL", SIMILARLY. OTHERWISE, IT WILL BE THE TIME SPECIFIED IF THE <DELAY VALUE> IS <TIME>.

THE INITIATE TRANSMIT FORM COMMANDS THE LINE TO A TRANSMIT STATE AND DELAYS ANALOGOUSLY TO THAT DESCRIBED ABOVE.

INITIALIZE STATEMENTS

SYNTAX

```
<INITIALIZE STATEMENT> ::= INITIALIZE <INITIALIZED ITEM>
<INITIALIZED ITEM> ::= BCC / TEXT / TRAN / BLKN /
      <TALLY-TOGGLE LIST>
<TALLY-TOGGLE LIST> ::= <USER TALLY> / TOG[<TOGGLE NUMBER>] /
      <TALLY-TOGGLE LIST> , <TALLY-TOGGLE LIST>
```

SEMANTICS

INITIALIZED BCC CAUSES THE BLOCK CHECK CHARACTER FOR HORIZONTAL PARITY TO BE INITIALIZED IN THE DCP. FOR EVEN PARITY THIS CHARACTER IS INITIALIZED TO ALL ZEROES; FOR ODD PARITY, ALL ONES. THE BCC SHOULD ALWAYS BE RE-INITIALIZED FOR ERROR RECOVERY, USUALLY INITIALIZE TEXT SETS THE MESSAGE POINTERS IN THE DCP TO THE FIRST BYTE OF THE MESSAGE. THIS STATEMENT MUST BE EXECUTED BEFORE USING THE STORE OR FETCH COMMANDS. INITIALIZE TRAN OR BLKN RESETS THE VALUE TO ZERO.

THE "INITIALIZE <TALLY-TOGGLE LIST>" FORM CAUSES THE TALLYS AND TOGGLES LISTED TO BE INITIALIZED FROM THE MESSAGE HEADER, IF THERE IS ONE, OR SET TO ZERO OR FALSE, OTHERWISE.

SYNTAX

```
<FINISH STATEMENT> ::= FINISH TRANSMIT <DELAY PART>
```

SEMANTICS

THE <FINISH STATEMENT> INITIATES THE TIMEOUT AND DELAY TIMES ESTABLISHED FOR A STATION. THIS COMMAND PUTS THE LINE BACK IN RECEIVE MODE. THE <FINISH STATEMENT> WOULD BE USED, FOR EXAMPLE,

TO GET A LINE READY TO RECEIVE FOLLOWING A POLL. (IT IS A NECESSARY STATEMENT TO HAVE WHEN FINISHED TRANSMITTING TO INSURE THE LAST CHARACTER IS ACTUALLY TRANSMITTED.)

FETCH STATEMENT

SYNTAX

```
<FETCH STATEMENT> ::= FETCH <FETCH ACTION PART>
<FETCH ACTION PART> ::= [<ENDOFBUFFER ACTION>] / <EMPTY>
<ENDOFBUFFER ACTION> ::= ENDOFBUFFER:<LABEL> / ENDOFBUFFER:NULL /
      <LABEL> / NULL
```

SEMANTICS

FETCH IS VALID ONLY FOR OUTPUT FROM THE DCP TO A STATION. A FETCH LOADS THE <CHARACTER REGISTER> WITH THE BYTE IN THE MESSAGE TEXT POINTED TO BY THE MESSAGE POINTER. WHEN USING THE FETCH INSTRUCTION, PROVISION SHOULD BE MADE FOR TAKING ACTION IF AN ENDOFBUFFER CONDITION IS ENCOUNTERED. (ON AN ENDOFBUFFER CONDITION THE CHARACTER IS NOT TRANSMITTED AFTER FETCHING. AN ENDOFBUFFER CONDITION RESULTING FROM A <FETCH STATEMENT> IS NOT CONSIDERED TO BE AN ERROR; THUS, THE ENDOFBUFFER FLAG IS NOT SET.) THE <FETCH ACTION PART> SPECIFIES THE ENDOFBUFFER ACTION. AN <EMPTY> <FETCH ACTION PART> WILL RESULT ON ABORTING THE REQUEST UPON ENDOFBUFFER, BUT OTHERWISE AN APPROPRIATE BRANCH WILL BE TAKEN ("FETCH[NULL]" IS EQUIVALENT TO "FETCH[1]. 1:").

STORE STATEMENTS

SYNTAX

```
<STORE STATEMENT> ::= STORE <STORE PARAMETER> <STORE ACTION PART> /
      STORE <TALLY-TOGGLE LIST> / STORE SEQUENCE
```

<STORE ACTION PART> ::= [<ENDOFBUFFER ACTION>] / <EMPTY>
<STORE PARAMETER> ::= <CHARACTER REGISTER> / <STRING> / <EMPTY>

SEMANTICS

THE <STORE STATEMENT> IS USED FOR INPUT OR RECEIVE REQUESTS ONLY. THE STORE AND STORE <CHARACTER REGISTER> FORMS OF THIS STATEMENT ARE EQUIVALENT. IT IS POSSIBLE TO GET AN ENDOFBUFFER CONDITION WHILE USING THIS INSTRUCTION, IF THE BUFFER SIZE (FOR BUFFERED DEVICES) OR THE MAXINPUT SIZE (FOR UNBUFFERED DEVICES) IS EXCEEDED. UNLIKE THE CASE OF THE <FETCH STATEMENT>, AN ENDOFBUFFER CONDITION HERE IS CONSIDERED AN ERROR CONDITION. UPON THE CONDITION, A <EMPTY> <STORE ACTIONPART> WILL RESULT IN AN AUTOMATIC "TERMINATE ERROR", A "NULL" WILL RESULT IN A CONTINUANCE OF THE PROGRAM, AND A "LABEL" WILL RESULT IN A BRANCH TO THE SPECIFIED LABEL.

WITH THE "STORE SEQUENCE" FORM OF THE <STORE STATEMENT> IF THE STATION IS IN SEQUENCE MODE, THE CURRENT VALUE OF THE SEQUENCE NUMBER IS STORED IN MESSAGE[5].[26:27] AS A BINARY INTEGER AND MESSAGE[5].[27:1] IS SET TO ONE TO INDICATE ITS PRESENCE. IF THE STATION IS NOT IN SEQUENCE MODE AT THE TIME THE STATEMENT IS EXECUTED THEN THE STATEMENT IS A NO-OP.

SUM STATEMENTS

SYNTAX

<SUM STATEMENT> ::= SUM <SUM PARAMETER>
<SUM PARAMETER> ::= <CHARACTER REGISTER> / <UNITARY STRING> /
<EMPTY>

SEMANTICS

THE <SUM STATEMENT> MAY BE USED TO NEGATE THE HORIZONTAL SUMMATION

FOR THE PREVIOUS CHARACTER TRANSMITTED OR RECEIVED.

FOR EXAMPLE, ASSUME THAT A CHARACTER HAS BEEN RECEIVED AS PART OF A MESSAGE AND HAS BEEN INCLUDED IN THE HORIZONTAL BLOCK CHECK CHARACTER (BCC) BEING ACCUMULATED. THE CHARACTER MAY BE ELIMINATED FROM THE BCC BY THE INSTRUCTION

SUM CHARACTER

WHICH CAUSES THE CHARACTER CONTAINED IN THE <CHARACTER REGISTER> TO BE "EXCLUSIVELY OR"ED WITH THE BCC, THUS DELETING THE CHARACTER FROM THE PARTIAL SUM.

THE SUM AND SUM CHARACTER FORMS OF THIS STATEMENT ARE EQUIVALENT.

EXAMPLE:

```
RECEIVE.
IF CHAR = STX THEN SUM CHAR.
```

IN THE ABOVE EXAMPLE, THE START-OF-TEXT CHARACTER WAS ELIMINATED FROM THE BCC.

TRANSMIT STATEMENTS

SYNTAX

```
<TRANSMIT STATEMENT> ::= TRANSMIT <ITEM> <TRANSMIT ACTION PART>
<ITEM> ::= <CHARACTER REGISTER> / ADDRESS <ADDRESS QUALIFIER> /
      TRAN / BLKN / TEXT / <STRING> / BCC / <EMPTY>
<ADDRESS QUALIFIER> ::= (RECEIVE) / (TRANSMIT) / <EMPTY>
<TRANSMIT ACTION PART> ::= [<TRANSMIT ERROR FLAG>
      <ACTION LABEL>] / <EMPTY>
<ACTION LABEL> ::= :NULL / :<LABEL> / <EMPTY>
```

SEMANTICS

THE <TRANSMIT STATEMENT> IS USED FOR OUTPUT OPERATIONS TO A STATION. IF <ITEM> IS <EMPTY>, THE CURRENT CONTENTS OF <CHARACTER REGISTER> ARE TRANSMITTED. OTHERWISE, <ITEM> SPECIFIES THE VALUE TO BE LOADED INTO THE <CHARACTER REGISTER> BEFORE TRANSMITTING.

THE TRANSMIT AND TRANSMIT CHAR FORMS OF THE STATEMENT ARE EQUIVALENT. THE STATEMENT

```
TRANSMIT TEXT.
```

IS EQUIVALENT TO

```
1:    FETCH [ENDOFBUFFER :2]
      TRANSMIT.
      GO TO 1.
2:
```

THE ADDRESS, TRAN, BLKN, AND BCC FORMS CAUSE THE STATION ADDRESS, TRANSMISSION NUMBER, BLOCK NUMBER, AND BLOCK-CHECK-CHARACTER, RESPECTIVELY, TO BE LOADED FROM THE MESSAGE BEING PROCESSED AND TO BE TRANSMITTED DOWN THE LINE.

TRANSMIT <STRING> WILL LOAD AND TRANSMIT EACH CHARACTER OF THE <STRING> SPECIFIED UNTIL THE <STRING> IS EXHAUSTED. THE <TRANSMIT ACTION PART> SPECIFIES THE ACTION TO BE TAKEN UPON DETECTION OF A BREAK. IF IT IS <EMPTY>, AN AUTOMATIC "TERMINATE ERROR" WILL RESULT FROM A BREAK. OTHERWISE, A BRANCH SPECIFIED BY THE <ACTION LABEL> WILL RESULT.

AN <EMPTY> <ACTION LABEL> SIGNIFIES A BRANCH TO THE NEXT STATEMENT. A ":NULL" <ACTION LABEL> SIGNIFIES THAT EXECUTION SHOULD PROCEED AS IF THE CONDITION ASSOCIATED WITH THE ACTION LABEL DID NOT OCCUR. A ":<LABEL>" AS AN <ACTION LABEL> SPECIFIES THAT A BRANCH TO THE DESIGNATED LABEL SHOULD OCCUR.

RECEIVE STATEMENTS

SYNTAX

```
<RECEIVE STATEMENT> ::= RECEIVE <TIMEOUT PART> <ITEM>
      <RECEIVE ACTION PART>
<TIMEOUT PART> ::= (<TIME>) / (NULL) / <EMPTY>
<RECEIVE ACTION PART> ::= [<RECEIVE ACTION LIST>] / <EMPTY>
<RECEIVE ACTION LIST> ::= <ACTION ITEM> <ACTION LABEL> /
      <ERROR SWITCH DESIGNATOR> /
      <ACTION ITEM> <ACTION LABEL> , <RECEIVE ACTION LIST> /
      <ERROR SWITCH DESIGNATOR> , <RECEIVE ACTION LIST> /
<ACTION ITEM> ::= <TEXT CONTROL CHARACTER> / <UNITARY STRING>
      / <RECEIVE ERROR FLAG> / <ITEM ERROR FLAG>
<TEXT CONTROL CHARACTER> ::= CONTROL / BACKSPACE / LINEDELETE /
      END / WRU
<ERROR SWITCH DESIGNATOR> ::= ERROR [<INTEGER>] / <INTEGER>
```

SEMANTICS

THE <RECEIVE STATEMENT> PROVIDES THE MEANS TO RECEIVE CHARACTERS AND/OR ERROR CONDITIONS. A <RECEIVE STATEMENT> PRESUMES THE ADAPTER FOR THE STATIONS LINE IS IN RECEIVE MODE. IF IT IS NOT, AN ABORT WILL OCCUR, THAT IS, THE DCP WILL SEND AN ABORT ERROR RESULT TO THE MAIN SYSTEM. A <RECEIVE STATEMENT> RECEIVES THE SPECIFIED ITEM AND IF APPROPRIATE STORES THE ITEM. THE <RECEIVE ACTION PART> SPECIFIES ACTIONS TO BE TAKEN UPON EITHER THE RECEIPT OF SPECIFIC CHARACTERS OR OCCURRENCES OF SPECIFIC ERROR CONDITIONS. FOR ALL ITEMS THE <ERROR SWITCH DESIGNATOR> IS APPROPRIATE; THE ERROR SWITCH DESIGNATED SPECIFIES THE ACTION TO TAKE UPON THE VARIOUS ERROR CONDITIONS WHICH ARE <RECEIVE ERROR FLAG>S. THE PRESENCE OF A <RECEIVE ERROR FLAG> CREATES AN ERROR SWITCH UNIQUE TO THE RECEIVE STATEMENT. AN <ERROR SWITCH DESIGNATOR> AND A <RECEIVE ERROR FLAG> MAY NOT BE IN THE SAME <RECEIVE ACTION LIST>. WHICH <ACTION ITEM>S ARE APPROPRIATE FOR RECEIVING VARIOUS ITEMS ARE

DISCUSSED BELOW.

A. ADDRESS <ADDRESS QUALIFIER> : THE PROPER NUMBER OF RECEIVE OR TRANSMIT ADDRESS CHARACTERS (THE NUMBER OF WHICH IS SPECIFIED BY THE DEFINITION OF THE STATION TERMINAL) ARE RECEIVED AND CHECKED FOR AGREEMENT. SHOULD THE ADDRESS CHARACTERS NOT CORRESPOND, AN ADDRESS ERROR CONDITION RESULTS; IF THE <RECEIVE ACTION PART> HAS "ADDERR" ACTION SPECIFIED THE <ACTION LABEL> IS ADHERED TO BUT OTHERWISE AN "AUTOMATIC" "TERMINATE ERROR" IS DONE. (THE ANALOGOUS ACTION OCCURS FOR ALL ERROR CONDITIONS COVERED BY <ITEM ERROR FLAG>S. THAT IS, FOR EVERY ITEM EXCEPT <CHARACTER > AND <EMPTY>, THERE IS A POSSIBLE ERROR CONDITION WHICH WILL RESULT IN EVENTS ANALOGOUS TO THAT DESCRIBED FOR "ADDRESS <ADDRESS QUALIFIER> AND "ADDERR <ACTION LABEL>"

B. TRAN : THE PROPER NUMBER OF TRANSMISSION NUMBER CHARACTERS ARE RECEIVED AND CHECKED FOR AGREEMENT WITH THE (RECEIVE) TRANSMISSION NUMBER KEPT FOR THE STATION WITH WHICH THE DCP IS COMMUNICATING. SHOULD THERE NOT BE AGREEMENT A TRANSMISSION ERROR CONDITION IS OBTAINED. (THE NUMBER OF TRANSMISSION NUMBER CHARACTERS IS SPECIFIED IN THE DEFINITION OF THE STATION TERMINAL.) THE TRANSMISSION NUMBER IS STORED IN THE MESSAGE HEADER.

C. BLKN: A CHARACTER IS RECEIVED AND CHECKED AGAINST THE BLOCK NUMBER. "BLKNERR" IS APPROPRIATE FOR THIS ITEM. THE BLOCK NUMBER IS STORED IN THE MESSAGE HEADER.

D. BCC: A CHARACTER IS RECEIVED AND CHECKED AGAINST THE BLOCK CHECK CHARACTER THAT THE DCP HAS BEEN ACCUMULATING DURING THE PROCESSING OF THE REQUEST. "BCCERR" IS APPROPRIATE FOR THIS ITEM.

E. <STRING>: CHARACTERS ARE RECEIVED (AND STORED IN THE TEXT STRING AND A CHECK IS MADE FOR CORRESPONDENCE, WHICH IF NOT

FOUND RESULTS IN A FORMAT ERROR CONDITION. "FORMATERR" IS THE <ITEM ERROR FLAG> FOR THIS ITEM.

F. TEXT: CHARACTERS ARE RECEIVED (AND STORED IN THE TEXT PART OF THE MESSAGE SPACE OBTAINED) UNTIL EITHER AN <ACTION ITEM>S <ACTION LABEL> RESULTS IN A BRANCH TO A LABEL OR A TERMINATE ERROR, OR A "NON-RECOVERABLE" ERROR SUCH AS A DISCONNECT OCCURS. IF THE OCCURRENCE OF A CHARACTER IN THE <RECEIVE ACTION LIST> RESULTS IN A BRANCH OUTSIDE THE <RECEIVE STATEMENT> THAT CHARACTER IS NOT STORED AWAY AUTOMATICALLY. APPROPRIATE <ACTION ITEM>S ARE <TEXT CONTROL CHARACTER>, <UNITARY STRING>, AND "ENDOFBUFFER" (AN ENDOFBUFFER CONDITION EXISTS WHEN TOO MANY CHARACTERS HAVE BEEN RECEIVED AS SPECIFIED BY THE <BUFFER STATEMENT> OR <MAXINPUT STATEMENT> IN THE DEFINITION OF THE STATIONS TERMINAL. THE FOLLOWING WILL ELUCIDATE THESE POINTS AND THE SPECIAL ACTIONS TAKEN UPON THE RECEIPT OF <TEXT CONTROL CHARACTER> IF THEY ARE SPECIFIED IN THE <RECEIVE ACTION LIST>:

```
EXAMPLE
1:TALLY[2] = TALLY [2] + 1.

RECEIVE TEXT[END , WRU, BACKSPACE: 1].
TERMINATE NORMAL.
```

IN THIS EXAMPLE, AS EACH CHARACTER IS RECEIVED BY THE DCP A CHECK WILL BE MADE TO SEE IF THE CHARACTER IS THE END, WRU, OR BACKSPACE CODE FOR THE STATION. IF THE CHARACTER IS END OR WRU, AN EXIT TO THE STATEMENT IMMEDIATELY FOLLOWING THE <RECEIVE STATEMENT> WILL BE MADE. IF THE BACKSPACE CHARACTER IS RECEIVED, CONTROL IS TRANSFERRED TO THE STATEMENT LABELLED 1. IF NONE OF THESE THREE CHARACTERS ARE FOUND, THE NEXT CHARACTER IN THE MESSAGE IS RECEIVED, AND THE PROCESS CONTINUES.

IF THE WRU, CONTROL, END, BACKSPACE, OR LINEDELETE ARE USED AS <ACTION ITEM>S, THE SYSTEM WILL TAKE AUTOMATIC ACTION

APPROPRIATE FOR THE CHARACTER, AND THEN BRANCH TO THE INDICATED LABEL OR THE NEXT STATEMENT DEPENDING ON THE <ACTION LABEL> SPECIFIED. THE ACTION TAKEN FOR THE WRU AND CONTROL CHARACTERS IS TO SET THE PERTINENT FLAG. THE ACTION FOR THE END CHARACTER IS TO DO NOTHING (EXCEPT, OF COURSE, TO TAKE THE BRANCH AS SPECIFIED BY THE <ACTION LABEL>). NOTE THAT THE BACKSPACE AND LINE DELETE CHARACTERS ARE NEVER STORED IN THE TEXT PART OF THE MESSAGE SPACE EVEN IF THE <ACTION LABEL> IS ":NULL".

G. <CHARACTER REGISTER> OR <EMPTY> : ONE CHARACTER IS RECEIVED. ALL THE <ACTION ITEMS> APPROPRIATE FOR THE TEXT ITEM ARE APPLICABLE HERE EXCEPT "ENDOFBUFFER" WHICH IS INAPPROPRIATE BECAUSE THE RECEIVED CHARACTER IS NOT STORED ANYWHERE.

IF NULL IS USED AS AN <ACTION LABEL> CONTROL WILL RETURN TO THE <RECEIVE STATEMENT> AFTER APPROPRIATE ACTION IS TAKEN BY THE SYSTEM, IF ANY.

TERMINATE STATEMENTS

SYNTAX

```
<TERMINATE STATEMENT> ::= TERMINATE <TERMINATE TYPE>
<TERMINATE TYPE> ::= LOGICALACK <LOGICALACK TYPE> / NORMAL /
      ERROR / BLOCK / ENABLEINPUT
<LOGICALACK TYPE> ::= (RETURN) / <EMPTY>
```

SEMANTICS

THE <TERMINATE STATEMENT> TERMINATES THE REQUEST PROGRAM. THIS USUALLY ENTAILS THE FOLLOWING GENERAL FUNCTIONS:

1. RESULT DESCRIPTION

   THE MAIN SYSTEM IS TO BE INFORMED OF THE SUCCESS, FAILURE OR STATUS OF THE TERMINATING REQUEST. TO DO THIS, A MESSAGE SPACE MAY BE ALLOCATED FROM THE AVAILABLE MESSAGE POOL, IF NECESSARY. THE RESULT FIELDS OF THE MESSAGE ARE COMPLETED APPROPRIATELY AND THE MESSAGE IS MOVED TO THE DCP RESULT QUEUE.

2. LINE AND STATION STATUS

   IS UPDATED TO INHIBIT OR MODIFY INITIATION OF SUBSEQUENT REQUESTS.

3. LINE INITIATION

   NEXT LINE FUNCTION TO BE INITIATED;

   A. IDLE THE LINE

      NO NEW REQUEST FUNCTION IS INITIATED, BUT ALL LINE EVENTS ARE DISCARDED UNTIL AN APPROPRIATE REQUEST IS RECEIVED

FROM THE SYSTEM.

B. INITIATE A CONTENTION LINE WITH ONE STATION.

IF THE LINE IS MARKED AS IN CONTROL MODE OR
THE STATION IS MARKED NOT READY
THEN IDLE THE LINE; ELSE,
IF THE STATION INITIATE QUEUE HOLDS A REQUEST
THEN MARK THE LINE BUSY AND INITIATE THE REQUEST;
ELSE.
IF THE STATION IS INPUT ENABLED
THEN MARK THE LINE BUSY AND
INITIATE THE INPUT REQUEST LOGIC; ELSE,
IDLE THE LINE.

C. INITIATE A POLLED LINE WITH MANY POSSIBLE STATIONS

IF THE LINE STATUS IS MARKED AS IN CONTROL MODE
THEN IDLE THE LINE; ELSE
SELECT THE NEXT STATION IN TURN AND INVESTIGATE ITS
STATUS:
IF THE STATION STATUS IS MARKED NOT READY
THEN SELECT THE NEXT STATION; ELSE
IF THE STATION INITIATE QUEUE CONTAINS A VALID REQUEST
THEN MARK THE LINE BUSY AND INITIATE THE REQUEST; ELSE
IF THE STATION IS INPUT ENABLED
THEN DECREMENT THE STATION PRIORITY TALLY AND IF ZERO
THEN RESET THE PRIORITY TALLY TO THE STATIONS
PRIORITY, MARK THE LINE BUSY AND
INITIATE THE INPUT REQUEST LOGIC; ELSE
SELECT THE NEXT STATION.

EACH TIME THE NEXT STATION IS SELECTED FOR INVESTIGATION,
THE PROCEDURE YIELDS CONTROL OF THE PROCESSOR,
TEMPORARILY, TO PROCESS LINE EVENTS FOR ACTUALLY
ACTIVE LINES. ALSO, WHEN THE STATION LIST

ASSOCIATED WITH THE LINE IS EXHAUSTED, THE FOLLOWING LOGIC IS INVOKED: IF THE LAST PASS OVER THE STATION LIST FOUND NOT ONE READY STATION WHICH EITHER HAD A VALID REQUEST QUEUED OR WAS ENABLED FOR INPUT (REGARDLESS OF ITS PRIORITY TALLY) THEN THE LINE IS IDLED; ELSE IF THE LAST PASS OVER THE STATIONS DID NOT RESULT IN THE INITIATION OF ANY REQUESTED OR ENABLED FUNCTION THEN THE LINE IS IDLED FOR A TIME COMMENSURATE WITH THE TIME REQUIRED TO PROCESS AN INPUT OR OUTPUT REQUEST BEFORE SCANNING THE STATION LIST AGAIN; OTHERWISE THE STATION LIST IS REINVESTIGATED IMMEDIATELY.

THE ACTUAL COMBINATION OF THE ABOVE FUNCTIONS AND THEIR SPECIFICS DEPENDS ON THE TYPE OF TERMINATE STATEMENT:

TERMINATE ERROR.

THE TERMINATE ERROR STATEMENT SERVES TO INFORM THE SYSTEM OF A FAILURE TO SATISFACTORILY COMPLETE A REQUESTED OR ENABLED FUNCTION, AND INHIBIT INITIATION OF ANY NEW FUNCTIONS FOR THE STATION UNTIL REQUESTED BY THE SYSTEM.

AN ERROR MESSAGE IS OBTAINED, FILLED AND QUEUED (I.E., FIRST A SMALL MESSAGE IS ALLOCATED, THEN ITS RESULT FLAGS AND RETRY TALLY ARE COPIED EITHER FROM THE CURRENT MESSAGE, OR, THERE BEING NONE CURRENT AS FOR ENABLED INPUT, FROM THE STATION TABLE. TALLYS AND TOGGLES ARE COPIED INTO THE NEWLY ALLOCATED MESSAGE, THE MESSAGE CLASS, DCP, LINE AND STATION NUMBER FIELDS ARE INITIALIZED APPROPRIATELY, AND THE ERROR MESSAGE IS INSERTED INTO THE DCP RESULT QUEUE.) LEAVING THE STATION INITIATE QUEUE INTACT; THE STATION STATUS IS MARKED AS NOT READY; AND THE APPROPRIATE LINE INITIATE ROUTINE IS ENTERED (WHICH WILL IDLE A CONTENTION LINE, BUT PROCEED FOR OTHER READY STATIONS ON A MULTIPOINT POLLED LINE).

TERMINATE NORMAL.

THE TERMINATE NORMAL STATEMENT SIGNALS THE SATISFACTORY COMPLETION OF A REQUESTED OR ENABLED FUNCTION.

IF NO MESSAGE SPACE IS CURRENTLY ALLOCATED FOR THE TERMINATING FUNCTION, THEN ONE IS ALLOCATED (AS IN TERMINATE ERROR) AND FILLED FROM THE STATION TABLE COPY OF RESULT FLAGS AND RETRY TALLY WHICH ARE THEN RESET. IN ANY CASE, TALLYS AND TOGGLES ARE COPIED INTO THE MESSAGE FROM THE STATION TABLE AND THE RESULT INDEX IN THE MESSAGE IS SET TO DESIGNATE NORMAL TERMINATION. THE MESSAGE IS THEN INSERTED INTO THE DCP RESULT QUEUE. FINALLY THE APPROPRIATE LINE INITIATE PROCEDURE IS ENTERED TO BEGIN THE NEXT APPROPRIATE FUNCTION.

TERMINATE LOGICALACK.

THE TERMINATE LOGICALACK STATEMENT SIGNALS THE SUCCESSFUL RECEPTION OF AN INPUT MESSAGE WITH THE POSSIBILITY (DEPENDING UPON A RUN-TIME STATION OPTION) OF DELAYING ACKNOWLEDGMENT OF RECEIPT OF THE MESSAGE UNTIL REQUESTED BY THE SYSTEM TO CONTINUE. THE STATION LOGICALACK OPTION IS TESTED AND IF TRUE

> THEN THE "TO BE ACKNOWLEDGED" RESULT FLAG IS SET, THE MESSAGE IS COMPLETED AND QUEUED AS FOR TERMINATE NORMAL,
>
> THE LINE STATUS IS SET TO ACKNOWLEDGE READY TO INHIBIT INITIATION OF ANY NORMAL REQUESTS EXCEPT THE ACKNOWLEDGE REQUEST, AND
>
> THE LINE IS IDLED; ELSE

LOGICALACK NOT OPTIONED, THE DCP WILL AUTOMATICALLY CONTINUE WITH THE ACKNOWLEDGMENT.

IF <LOGICALACK TYPE> IS "(RETURN)"

> THEN THE MESSAGE IS COMPLETED AND QUEUED AS FOR TERMINATE

NORMAL (ALLOWING OVERLAP OF ACKNOWLEDGMENT WITH SYSTEM PROCESSING OF THE INPUT RECEIVED), A FLAG IS SET TO INHIBIT MESSAGE RESULT ENQUEUEING FOR THE NEXT TERMINATE NORMAL SIGNALLING END OF THE ACKNOWLEDGMENT AND INPUT FUNCTION; ELSE THE MESSAGE IS RETAINED UNTIL THE ACKNOWLEDGMENT TERMINATE NORMAL.

IN ANY CASE, CONTROL IS GIVEN TO THE PROGRAM AT THE NEXT STATEMENT IN SEQUENCE.

TERMINATE BLOCK.

THE TERMINATE BLOCK STATEMENT SERVES A NORMAL TERMINATE FOR A BLOCK (NOT THE LAST) OF INPUT RECEIVED FROM A STATION. THE MESSAGE IS COMPLETED AND QUEUED AS IN TERMINATE NORMAL. CONTROL IS IMMEDIATELY GIVEN BACK TO THE REQUEST PROGRAM AT THE NEXT STATEMENT IN SEQUENCE SO THAT IT MAY BEGIN RECEPTION OF THE NEXT BLOCK OF INPUT FROM THE STATION.

TERMINATE ENABLEINPUT.

THE TERMINATE ENABLEINPUT STATEMENT ALLOWS AN OUTPUT REQUEST THE POSSIBILITY OF SUSPENDING ITSELF IN FAVOR OF THE INPUT LOGIC FOR THE SAME STATION, AS WOULD BE APPROPRIATE TO SOME DEVICES WHICH MAY NAK THE SELECT IF TRANSMIT READY.

IF THE STATION IS INPUT ENABLED

THEN THE CURRENT REQUEST AND ITS MESSAGE ARE HELD IN THE STATION INITIATE QUEUE TO BE REINITIATED FROM THE START LATER, AND THE INPUT LOGIC IS INITIATED; ELSE CONTROL IS IMMEDIATELY RETURNED TO THE WRITE REQUEST PROGRAM AT THE NEXT STATEMENT IN SEQUENCE.

TERMINATE NOINPUT,

THE TERMINATE NOINPUT STATEMENT SIGNALS THE END OF AN UNSUCCESSFUL ATTEMPT TO RECEIVE INPUT FROM A STATION, BUT WITHOUT ERRORS.

IF THE INPUT LOGIC WAS INVOKED BECAUSE THE STATION WAS INPUT ENABLED THEN ANY MESSAGE SPACE THAT MIGHT HAVE BEEN ALLOCATED FOR THE INPUT IS DISCARDED TO ALLOW INITIATION OF REQUESTS THAT MIGHT HAVE OR MAY BE MADE.

IF THE INPUT LOGIC WAS INVOKED BECAUSE OF A READ REQUEST THEN IF THE "TRY ONCE" VARIANT IS SET, TERMINATE NOINPUT PROCEEDS AS TERMINATE NORMAL, FINISHING THE REQUEST; OTHERWISE, THE READ REQUEST IS HELD TO BE REINITIATED LATER. FINALLY, THE APPROPRIATE LINE INITIATE PROCEDURE IS ENTERED.

NOTE: THAT FOR ENABLE INPUT, IF THE "READ ONCE" OPTION IS SET FOR THE STATION, THEN ALL TERMINATION OF ENABLED INPUT EXCEPT TERMINATE NOINPUT WILL CAUSE THE STATION ENABLED OPTION TO BE RESET.

ERROR ACTION STATEMENTS

SYNTAX

```
<ERROR ACTION STATEMENT> ::= ERROR [<ERRORNUMBER>] =
      <ERROR ACTION LIST>
<ERROR ACTION LIST> ::= <ERROR ACTION> /
      <ERROR ACTION LIST>, <ERROR ACTION>
<ERROR ACTION> ::= <RECEIVE ERROR FLAG>
      <LABEL PART>
<LABEL PART> ::= :<LABEL> / : NULL
<ERRORNUMBER> ::= <INTEGER>
```

SEMANTICS

<ERROR ACTION STATEMENT>S CREATE ERROR SWITCHES WHICH ARE BRANCHED

TO UPON THE OCCURRENCE OF ERROR CONDITIONS CORRESPONDING TO <RECEIVE ERROR FLAG>S IN <RECEIVE STATEMENTS> WHOSE <RECEIVE ACTION PART>S SPECIFY AN ERROR SWITCH. (FOR A <RECEIVE STATEMENT> NOT DESIGNATING AN ERROR SWITCH A "TERMINATE ERROR" WILL RESULT UPON THE EXISTENCE OF A RECEIVE ERROR.) FOR EACH ERROR CONDITION MENTIONED IN AN <ERROR ACTION LIST>, EITHER NO ACTION IS TAKEN OR A BRANCH TO A LABEL IS MADE DEPENDING UPON WHETHER THE <LABEL PART> IS ":NULL OR ":<LABEL>". IF AN ERROR FLAG HAS NOT BEEN MENTIONED IN THE <ERROR ACTION LIST>, THE ACTION IS TO "TERMINATE ERROR".

BREAK STATEMENT

SYNTAX

<BREAK STATEMENT> ::= BREAK (<BREAK TIME> , <BREAK DELAY TIME>)
<BREAK TIME> ::= <TIME> / *
<BREAK DELAY TIME> ::= <TIME> / NULL

SEMANTICS

THE <BREAK TIME> SPECIFIES THE TIME TO BREAK. AN ASTERISK INDICATES THAT A STANDARD BREAK OF 2 CHARACTER TIMES SHOULD BE USED. THE <BREAK DELAY TIME> SPECIFIES THE TIME TO DELAY SUBSEQUENT TO THE BREAK BEFORE CONTINUING.

DELAY STATEMENT

SYNTAX

<DELAY STATEMENT> ::= DELAY (<TIME>)

SEMANTICS

THE <DELAY STATEMENT> WILL CAUSE THE SPECIFIED <TIME> TO ELAPSE

BEFORE EXECUTING THE NEXT STATEMENT OF THE REQUEST.

INCREMENT STATEMENT

SYNTAX

```
<INCREMENT STATEMENT> ::= INCREMENT <INCREMENTED ITEM>
<INCREMENTED ITEM> ::= TRAN / BLKN /
      SEQUENCE <SEQUENCE ACTION PART>
<SEQUENCE ACTION PART> ::= <EMPTY> / [<SEQUENCE ACTION>]
<SEQUENCE ACTION> ::= SEQERR : NULL /
      SEQERR : <LABEL> / NULL / <LABEL>
```

SEMANTICS

THE INCREMENT STATEMENT WILL CAUSE 1 TO BE ADDED TO THE TRANSMISSION NUMBER ("TRAN"), TO THE BLOCK NUMBER ("BLKN") OR A SPECIFIED INCREMENT (INITIALLY SET BY AN MCS USING THE DCWRITE CONSTRUCT) TO THE SEQUENCE NUMBER ("SEQUENCE").

THE BRANCH TO THE LABEL OCCURS IF INCREMENTING THE SEQUENCE NUMBER CAUSES AN OVERFLOW.

SHIFT STATEMENT

SYNTAX

```
<SHIFT STATEMENT> ::= SHIFT <SHIFT DIRECTION>
<SHIFT DIRECTION> ::= UP / DOWN
```

SEMANTICS

THE <SHIFT STATEMENT> ALLOWS THE CHANGING OF WHAT IS CONSIDERED TO BE THE CURRENT "CASE" OF CERTAIN CHARACTER SETS.

2.3.3. MODEM SECTION

SYNTAX

```
<MODEM SECTION> ::= <MODEM LIST>
<MODEM LIST> ::= <MODEM DEFINITION> / <MODEM LIST>
      <MODEM DEFINITION>
<MODEM DEFINITION> ::= MODEM <MODEM IDENTIFIER> :
      <MODEM ATTRIBUTE LIST>
<MODEM IDENTIFIER> ::= <IDENTIFIER>
<MODEM ATTRIBUTE LIST> ::= <MODEM STATEMENT>. /
      <MODEM ATTRIBUTE LIST> <MODEM STATEMENT>.
<MODEM STATEMENT> ::= <MODEM ADAPTER STATEMENT> /
      <TRANSMIT DELAY STATEMENT> / <NOISE DELAY STATEMENT>
```

SEMANTICS

THE MODEM SECTION IS USED TO DEFINE THE PERTINENT CHARACTERISTICS OF EACH TYPE OF MODULATOR/DEMODULATOR TO BE USED WITHIN THE DATA COMMUNICATIONS NETWORK.

MODEM ADAPTER STATEMENT

SYNTAX

```
<MODEM ADAPTER STATEMENT> ::= ADAPTER = <ADAPTER TYPE LIST>
<ADAPTER TYPE LIST> ::= <ADAPTER TYPE> /
      <ADAPTER TYPE LIST>, <ADAPTER TYPE>
<ADAPTER TYPE> ::= 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 /
      11 / 12 / 13 / 14 / 15 / 16 / 17 / 18 / 19 / 20 /
      21 / 22 / 23 / 24 / 25 / 26 / 27 / 28 / 29 / 30
```

SEMANTICS

THE <MODEM ADAPTER STATEMENT> IS USED TO LIST ALL OF THE <ADAPTER TYPE>S WITH WHICH THE MODEM BEING DESCRIBED IS COMPATIBLE. THE <INTEGER>S CONTAINED IN THE LIST MUST BE ONE OF THE VALUES LISTED IN THE "TABLE OF DATACOM ADAPTER TYPES" IN THE APPENDIX. TYPE DEPENDS ON THE SPEED OF THE LINE, CHARACTER FORMAT, AND WHETHER THE LINE IS SYNCHRONOUS OR ASYNCHRONOUS.

TRANSMIT DELAY STATEMENT

SYNTAX

```
<TRANSMIT DELAY STATEMENT> ::= TRANSMITDELAY =
      <TIME> <DYNAMIC PART>
<DYNAMIC PART> ::= (DYNAMIC) / <EMPTY>
```

SEMANTICS

THE <TRANSMIT DELAY STATEMENT> DEFINES THE AMOUNT OF TIME REQUIRED FOR THE MODEM TO GO FROM A "REQUEST TO SEND" STATUS TO A "READY TO SEND" STATUS. THE TIME MUST BE DEFINED IN UNITS OF MICROSECONDS (MICRO), MILLISECONDS (MILLI), OR SECONDS (SEC). IF THE DYNAMIC PART IS NOT EMPTY THE TRANSMITDELAY VALUE WILL BE RUN TIME CHANGEABLE, (THE SENSE OF THE DYNAMIC PART IS CONSISTENT, WHERE IT APPEARS, WITHIN A SYNTACTIC CONSTRUCT EXCEPT WHERE NOTED.)

NOISE DELAY STATEMENT

SYNTAX

```
<NOISE DELAY STATEMENT> ::= NOISEDELAY = <TIME> <DYNAMIC PART>
```

SEMANTICS

THE <NOISE DELAY STATEMENT> DEFINES THE AMOUNT OF TIME THAT SHOULD BE DELAYED UPON THE MODEM ENTERING "REQUEST TO SEND" STATUS TO GET AROUND NOISE ON THE LINE TO WHICH THE MODEM IS CONNECTED. UPON DOING AN "INITIATE TRANSMIT" IN A REQUEST COMMUNICATING TO A STATION WITH CONNECTION THROUGH MODEMS, A DELAY WILL BE OBEYED CORRESPONDING TO THE EXCESS OF THE NOISEDELAY OVER THE TRANSMIT

DELAY FOR THE MODEM AT THE ADAPTER. FOR "INITIATE RECEIVE" THE SAME IS DONE EXCEPT THAT THE VALUES OF THE STATIONS MODEM ARE USED. (THIS PICTURE IS SOMEWHAT COMPLICATED BY THE TURNAROUND VALUES FOR THE STATIONS TERMINAL. IF THE INITIATE RECEIVE TURNAROUND DELAY OF THE TERMINAL EXCEEDS BOTH THE NOISE DELAY AND TRANSMIT DELAY VALUES FOR THE MODEM OF THE LINE ADAPTER, THE INITIATE RECEIVE TURNAROUND DELAY OF TERMINAL IS USED FOR THE "INITIATE TRANSMIT" DELAY.)

2.3.4. TERMINAL SECTION

SYNTAX

```
<TERMINAL SECTION> ::= <TERMINAL LIST>
<TERMINAL LIST> ::= <TERMINAL DEFINITION> / <TERMINAL DEFAULT
      DEFINITION>/
      <TERMINAL LIST> <TERMINAL DEFINITION> / <TERMINAL LIST>
      <TERMINAL DEFAULT DEFINITION>
<TERMINAL DEFINITION> ::= TERMINAL <TERMINAL IDENTIFIER> :
      <TERMINAL ATTRIBUTE LIST>
<TERMINAL IDENTIFIER> ::= <IDENTIFIER>
<TERMINAL DEFAULT DEFINITION> ::= TERMINAL DEFAULT
      <TERMINAL DEFAULT IDENTIFIER> :
      <TERMINAL DEFAULT ATTRIBUTE LIST>
<TERMINAL DEFAULT IDENTIFIER> ::= <IDENTIFIER>
<TERMINAL DEFAULT ATTRIBUTE LIST> ::= <TERMINAL DEFAULT ATTRIBUTE
      STATEMENT>./
      <TERMINAL DEFAULT ATTRIBUTE LIST>
      <TERMINAL DEFAULT ATTRIBUTE STATEMENT>.
<TERMINAL DEFAULT ATTRIBUTE STATEMENT> ::=
      <TERMINAL DEFAULT STATEMENT> /
      <TERMINAL ADDRESS STATEMENT> /
      <TERMINAL WIDTH STATEMENT> /
      <TERMINAL PAGE STATEMENT> /
      <TERMINAL TIMEOUT STATEMENT> /
      <ILLEGAL CHARACTER STATEMENT> /
      <FORMAT CHARACTER STATEMENT> /
      <INTER-CHARACTER DELAY STATEMENT> /
      <TERMINAL TURNAROUND STATEMENT> /
      <CODE STATEMENT> /
      <PARITY STATEMENT> /
      <MODE STATEMENT> /
      <BUFFER SIZE STATEMENT> /
```

```
        <MAXINPUT STATEMENT> /
        <BLOCKING STATEMENT> /
        <SCREEN STATEMENT> /
        <TERMINAL DUPLEX STATEMENT> /
        <TERMINAL ADAPTER STATEMENT> /
        <SPECIAL CHARACTER STATEMENT> /
        <TRANSMISSION NUMBER STATEMENT> /
        <WRAPAROUND STATEMENT> /
        <INHIBITSYNC STATEMENT>
<TERMINAL ATTRIBUTE LIST> ::= <TERMINAL ATTRIBUTE STATEMENT>./
        <TERMINAL ATTRIBUTE LIST> <TERMINAL ATTRIBUTE STATEMENT>.
<TERMINAL ATTRIBUTE STATEMENT> ::=
        <TERMINAL DEFAULT ATTRIBUTE STATEMENT> /
        <TERMINAL REQUEST STATEMENT>
```

SEMANTICS

A <TERMINAL DEFINITION> MAY BE THOUGHT OF AS A LIST OF THE PHYSICAL ATTRIBUTES OF THE DEVICES BEING DESCRIBED.

TERMINAL DEFAULT STATEMENT

SYNTAX

```
<TERMINAL DEFAULT STATEMENT> ::= DEFAULT =
      <TERMINAL DEFAULT IDENTIFIER>
```

SEMANTICS

THE <TERMINAL DEFAULT STATEMENT> IS USED TO SPECIFY THE NAME OF A SET OF TERMINAL ATTRIBUTES TO BE USED FOR A TERMINAL WHOSE DESCRIPTION IS INCOMPLETE. FOR TERMINALS WHICH HAVE SEVERAL COMMON CHARACTERISTICS, IT IS ADVANTAGEOUS TO GROUP THESE COMMON CHARACTERISTICS UNDER A <TERMINAL DEFAULT DEFINITION> AND LIST THE

REMAINING ATTRIBUTES UNDER EACH INDIVIDUAL <TERMINAL DEFINITION>. THE COMPILER WILL THEN REFER TO THE DEFAULT LIST TO COMPLETE THE TERMINAL SPECIFICATION.

EXAMPLE

ASSUME TERMINALS A1, A2, AND A3 HAVE THE SAME ATTRIBUTES EXCEPT FOR THE SIZE OF THE LOGICAL LINES TO BE TRANSMITTED ON OUTPUT. A <TERMINAL DEFAULT STATEMENT> COULD BE USED AS ILLUSTRATED BELOW TO SIMPLIFY THE CODING OF THE DESCRIPTIONS OF THESE 3 TERMINALS.

```
TERMINAL DEFAULT A123DFLT:

    CODE            = ASC63.
    MODE            = CONTENTION.
    PARITY          = VERTICAL:EVEN.
    REQUEST         = TWXREAD:RECEIVE, TWXWRITE:TRANSMIT.
    SCREEN          = FALSE.
    BLOCK           = FALSE.
    DUPLEX          = FALSE.
    BUFFER          = NULL.
    ADDRESS         = NULL.
    WIDTH           = 30.
    MAXINPUT        = 72.
    TIMEOUT         = 300 SEC.

    TERMINAL A1:
        DEFAULT     = A123DFLT.
        WIDTH       = 50.

    TERMINAL A2:
        DEFAULT     = A123DFLT.
        WIDTH       = 60.

    TERMINAL A3:
```

```
        DEFAULT    = A123DFLT.
        WIDTH      = 35.

    TERMINAL A4:
        DEFAULT    = A123DFLT.
        DUPLEX     = TRUE.
```

NOTE: IT MUST AGAIN BE POINTED OUT THAT SHOULD A TERMINAL SPECIFICATION, FOR EXAMPLE, CONTAIN AN ATTRIBUTE STATEMENT WHICH ALSO IS SUBSUMED UNDER A TERMINAL DEFAULT SPECIFICATION WHICH IS REFERRED TO BY A DEFAULT STATEMENT, THE ATTRIBUTE STATEMENT UNDER THE TERMINAL SPECIFICATION IS USED. THE ATTRIBUTE STATEMENTS IN A TERMINAL DEFAULT SPECIFICATION ARE DEFAULT STATEMENTS ONLY. IN THE CASE ABOVE, ONLY TERMINAL A4 HAS THE WIDTH SPECIFIED BY A123DFLT.

TERMINAL ADDRESS STATEMENT

SYNTAX

```
<TERMINAL ADDRESS STATEMENT> ::= ADDRESS = <RECEIVE ADDRESS>
      <TRANSMIT ADDRESS PART> <DIFFERENT PART> / ADDRESS = NULL
<RECEIVE ADDRESS> ::= <ADDRESS SIZE>
<TRANSMIT ADDRESS PART> ::=, <TRANSMIT ADDRESS> / <EMPTY>
<TRANSMIT ADDRESS> ::= <ADDRESS SIZE>
<DIFFERENT PART> ::= (DIFFERENT) / <EMPTY>
<ADDRESS SIZE> ::= 1 / 2 / 3
```

SEMANTICS

THE <TERMINAL ADDRESS STATEMENT> SPECIFIES THE NUMBER OF CHARACTERS WHICH ARE USED TO ADDRESS A STATION ASSOCIATED WITH A TERMINAL. AN EMPTY <TRANSMIT ADDRESS PART> IMPLIES THE TRANSMIT ADDRESS IS OF THE SAME LENGTH AS THE RECEIVE ADDRESS. A NON-EMPTY <DIFFERENT

PART> INDICATES THAT THE TRANSMIT ADDRESS MAY BE NON-IDENTICAL TO THE RECEIVE ADDRESS. (NATURALLY, IF THE ADDRESSES ARE OF DIFFERENT LENGTHS, THE ADDRESSES MUST BE DIFFERENT.)

THIS STATEMENT MUST BE USED IF THE MODE IS HYBRID OR POLL. IF THE MODE IS CONTENTION THE ADDRESS MUST BE "NULL". THE INTEGER MUST BE LESS THAN OR EQUAL TO 3.

TERMINAL WIDTH STATEMENT

SYNTAX

<TERMINAL WIDTH STATEMENT> ::= WIDTH = <INTEGER>

SEMANTICS

THE <TERMINAL WIDTH STATEMENT> DEFINES THE LENGTH OF A PHYSICAL LINE OF OUTPUT. THIS LENGTH MUST BE LESS THAN OR EQUAL TO THE LENGTH DEFINED IN THE <BUFFER SIZE STATEMENT> OR <MAXINPUT STATEMENT>. A LINE OF OUTPUT MAY BE LESS THAN THE NUMBER OF CHARACTERS SPECIFIED BY THE <TERMINAL WIDTH STATEMENT>. IF WIDTH IS NOT DEFINED, THE BUFFER SIZE OR MAXINPUT SPECIFIED WILL BE USED.

EXAMPLE

```
TERMINAL A:
          .
          .
BUFFER      = 150.
WIDTH       = 50.
          .
          .
```

IF 130 CHARACTERS WERE CONTAINED IN A SINGLE OUTPUT LINE TO THE ABOVE TERMINAL, THE OUTPUT WOULD APPEAR AS 3 LINES (PROVIDED THE WRAPAROUND ATTRIBUTE FOR THE STATION TO

WHICH THE OUTPUT IS DIRECTED IS SET). THE FIRST 2 LINES EACH CONTAINING 50 CHARACTERS, THE LAST LINE CONTAINING 30 CHARACTERS.

TERMINAL PAGE STATEMENT

SYNTAX

<TERMINAL PAGE STATEMENT> ::= PAGE = <INTEGER>

SEMANTICS

THE <TERMINAL PAGE STATEMENT> SPECIFIES THE NUMBER OF PHYSICAL LINES EQUAL TO A PHYSICAL PAGE. A ZERO VALUE INDICATES THAT PAGINATION CAN BE ARBITRARY.

TERMINAL TIMEOUT STATEMENT

SYNTAX

<TERMINAL TIMEOUT STATEMENT> ::= TIMEOUT = <TIME> <DYNAMIC PART>

SEMANTICS

THE <TERMINAL TIMEOUT STATEMENT> DEFINES THE INTERVAL OF TIME TO WAIT BETWEEN RECEIPT OF ONE CHARACTER AND THE START OF THE NEXT CHARACTER FROM A TERMINAL BEFORE ASSUMING THAT THE STATION ASSOCIATED WITH THE TERMINAL HAS "TIMED OUT".

ILLEGAL CHARACTER STATEMENT

SYNTAX

```
<ILLEGAL CHARACTER STATEMENT> ::= ILLEGALCHR =
      <ILLEGAL CHARACTER LIST>
<ILLEGAL CHARACTER LIST> ::= <UNITARY STRING> /
      <ILLEGAL CHARACTER LIST>, <UNITARY STRING>
```

SEMANTICS

THE <ILLEGAL CHARACTER STATEMENT> LISTS ONE OR MORE CHARACTERS WHOSE PRESENCE IN A MESSAGE TEXT TO BE TRANSMITTED IS ILLEGAL.

FORMAT CHARACTER STATEMENT

SYNTAX

<FORMAT CHARACTER STATEMENT> ::= <FORMAT CHARACTER> =
<UNITARY STRING>
<FORMAT CHARACTER> ::= CARRIAGE / LINEFEED / HOME / CLEAR

SEMANTICS

THE <FORMAT CHARACTER STATEMENT> PROVIDES A MEANS OF DEFINING THE CARRIAGE RETURN, LINEFEED, HOME, AND CLEAR CHARACTERS. THE HOME AND CLEAR CHARACTERS ARE SIGNIFICANT ONLY FOR TERMINALS WHOSE <SCREEN STATEMENT> IS SET TRUE. IF THE DEVICE HAS DISPLAY CAPABILITIES (I.E. SCREEN = TRUE) AND THE CLEAR FORMAT CHARACTER IS NOT SPECIFIED, IT WILL BE ASSUMED THAT THE HOME CHARACTER AUTOMATICALLY CLEARS THE SCREEN. (FOR THE PRESENT, THE SPECIFICATION OF THESE CHARACTERS IS COMMENTARY ONLY).

INTER-CHARACTER DELAY STATEMENT

SYNTAX

<INTER-CHARACTER DELAY STATEMENT> ::= ICTDELAY = <TIME>

SEMANTICS

THE <INTER-CHARACTER DELAY STATEMENT> DEFINES THE INTER-CHARACTER TRANSMISSION DELAY TIME FOR A TERMINAL. THE DCP WILL DELAY THE <TIME> SPECIFIED BETWEEN EACH CHARACTER TRANSMITTED AS PART OF A MESSAGE. (NOTE THAT THIS DELAY MAY NOT BE DYNAMIC.)

TERMINAL TURNAROUND STATEMENT

SYNTAX

<TERMINAL TURNAROUND STATEMENT> ::= TURNAROUND = <IRT>
<IRT> ::= <TIME>

SEMANTICS

THE <TERMINAL TURNAROUND STATEMENT> SPECIFIES THE TIME BETWEEN THE DCP "TURNING THE LINE AROUND" AND THE TERMINAL BEING ABLE TO RECEIVE CHARACTERS. <IRT> IS THE TIME FROM THE DCF INITIATING TRANSMIT TO THE TIME THE TERMINAL COULD RECEIVE CHARACTERS.

CODE STATEMENT

SYNTAX

<CODE STATEMENT> ::= CODE = <CODE TYPE>
<CODE TYPE> ::= EBCDIC / BCL / BCD / BAUDOT / ASC63 / ASC67

SEMANTICS

THE <CODE STATEMENT> SPECIFIES THE CHARACTER SET WHICH WILL BE USED TO COMMUNICATE WITH A SPECIFIC TERMINAL. TRANSLATION CANNOT BE AVOIDED UPON INPUT TO THE DCP FROM A TERMINAL. INPUT CHARACTERS ARE ALWAYS TRANSLATED TO EBCDIC.

PARITY STATEMENT

SYNTAX

```
<PARITY STATEMENT> ::= PARITY = <PARITY LIST> / PARITY = NULL
<PARITY LIST> ::= <PARITY TYPE> / <PARITY TYPE>, <PARITY TYPE>
<PARITY TYPE> ::= VERTICAL : <PARITY VALUE> /
      HORIZONTAL <HORIZONTAL PARITY TYPE> : <PARITY VALUE>
<PARITY VALUE> ::= ODD / EVEN
<HORIZONTAL PARITY TYPE> ::= (<HORIZONTAL PARITY VARIANT>) /
      <EMPTY>
<HORIZONTAL PARITY VARIANT> ::= 0 / 1
```

SEMANTICS

THE <PARITY STATEMENT> IS USED TO DEFINE THE TYPE OF PARITY CHECKING OR GENERATION TO BE PERFORMED BY THE DCP FOR A TERMINAL. HORIZONTAL OR VERTICAL CHECKING MAY BE DONE FOR EVEN OR ODD PARITY. THE VERTICAL OPTION REFERS TO THE PARITY OF THE CHARACTER ITSELF, WHILE THE HORIZONTAL OPTION IS USED FOR PARITY OF THE BLOCK CHECK CHARACTER (BCC). IF THE NULL OPTION IS USED, PARITY IS NOT CHECKED REGARDLESS OF ITS PRESENCE. ONLY ONE OF EACH OF THE TWO <PARITY TYPE>S IS ALLOWED. SHOULD A REQUEST MENTIONED IN A TERMINAL REQUEST STATEMENT MENTION THE "BCC" , IT IS MANDATORY THAT HORIZONTAL PARITY BE SPECIFIED AS NON-NULL. A <HORIZONTAL PARITY VARIANT> OF 0 MEANS THAT THE VERTICAL PARITY BIT OF THE BLOCK CHECK CHARACTER IS A TRUE PARITY BIT WHEREAS A VALUE OF 1 INDICATES THAT THE VERTICAL PARITY IS THE SUM OF THE VERTICAL PARITY BITS OF THE SUMMED CHARACTERS.

AN <EMPTY> <HORIZONTAL PARITY TYPE> IS SEMANTICALLY EQUIVALENT TO SPECIFYING A NON-EMPTY ONE WITH A VARIANT OF 0.

MODE STATEMENT

SYNTAX

```
<MODE STATEMENT> ::= MODE = <MODE TYPE>
<MODE TYPE> ::= POLL / CONTENTION / HYBRID
```

SEMANTICS

THE <MODE STATEMENT> DESIGNATES THE MODE OF OPERATION OF THE TERMINAL, THAT IS, WHETHER THIS DEVICE IS OF POLL, CONTENTION, OR HYBRID TYPE.

BUFFER SIZE STATEMENT

SYNTAX

<BUFFER SIZE STATEMENT> ::= BUFFER = <INTEGER> / BUFFER = NULL

SEMANTICS

THE <BUFFER SIZE STATEMENT> DEFINES THE MAXIMUM SIZE OF A PHYSICAL LINE AND MUST BE INCLUDED IN THE TERMINAL DESCRIPTION OF A BUFFERED DEVICE. IF THE TERMINAL DEVICE HAS NO BUFFER, THE NULL OPTION SHOULD BE USED.

MAXINPUT STATEMENT

SYNTAX

<MAXINPUT STATEMENT> ::= MAXINPUT = <INTEGER>

SEMANTICS

THE <MAXINPUT STATEMENT> APPLIES ONLY TO INPUTS FROM THE TERMINAL AND DEFINES THE MAXIMUM SIZE OF A PHYSICAL LINE FOR UNBUFFERED DEVICES. A <MAXINPUT STATEMENT> AND A <BUFFER SIZE STATEMENT> MAY NOT APPEAR IN THE SAME TERMINAL DESCRIPTION UNLESS BUFFER IS DEFINED AS NULL.

BLOCKING STATEMENT

SYNTAX

<BLOCKING STATEMENT> ::= BLOCKED = <LOGICAL VALUE>

SEMANTICS

THE <BLOCKING STATEMENT> IF SET TRUE, INFORMS THE DCP THAT THE TERMINAL IS CAPABLE OF SENDING AND/OR RECEIVING DATA IN BLOCKED FORMAT. THE REQUESTS DEFINED FOR THE TERMINAL SHOULD BE CODED IN A MANNER THAT ALLOWS THE BLOCKED FORMAT. THE DATA COMMUNICATIONS SYSTEM WILL, UNDER APPROPRIATE CIRCUMSTANCES, ATTEMPT BLOCKED OUTPUT. (THIS IS EXEMPLARY OF THE CONVENTIONS THAT MUST BE OBEYED IN THE WRITING OF REQUESTS.)

SCREEN STATEMENT

SYNTAX

<SCREEN STATEMENT> ::= SCREEN = <LOGICAL VALUE>

SEMANTICS

THE <SCREEN STATEMENT> IS SET TRUE ONLY IF A TERMINAL HAS DISPLAY CAPABILITIES. (AT PRESENT THIS ATTRIBUTE IS NOT UTILIZED EXCEPT IN ALLOWING THE SPECIFICATION OF THE HOME AND/OR CLEAR CHARACTERS.)

TERMINAL DUPLEX STATEMENT

SYNTAX

<TERMINAL DUPLEX STATEMENT> ::= DUPLEX = <LOGICAL VALUE>

SEMANTICS

THE <TERMINAL DUPLEX STATEMENT> SPECIFIES WHETHER OR NOT THE TERMINAL MUST BE ATTACHED TO LINES OF DUPLEX TYPE. THIS STATEMENT MUST BE INCLUDED IN THE TERMINAL DESCRIPTION.

TERMINAL ADAPTER STATEMENT

SYNTAX

<TERMINAL ADAPTER STATEMENT> ::= ADAPTER = <ADAPTER TYPE LIST>

SEE MODEM ADAPTER STATEMENT AS DESCRIBED UNDER THE MODEM SECTION. THIS STATEMENT SPECIFIES THE ADAPTER TYPES WITH WHICH THE TERMINAL MAY BE INTERFACED WITH THE DCP.

TERMINAL REQUEST STATEMENT

SYNTAX

```
<TERMINAL REQUEST STATEMENT > ::= REQUEST <APPLICATION QUALIFIER> =
      <REQUEST IDENTIFICATION LIST>
<APPLICATION QUALIFIER> ::= [<APPLICATION NUMBER>] / <EMPTY>
<APPLICATION NUMBER> ::= 1 / 2 / 3 / 4 / 5 / 6
<REQUEST IDENTIFICATION LIST> ::= <REQUEST IDENTIFICATION> /
      <REQUEST IDENTIFICATION>, <REQUEST IDENTIFICATION>
<REQUEST IDENTIFICATION> ::= <REQUEST IDENTIFIER> :
```

<REQUEST ACTION>
<REQUEST ACTION> ::= RECEIVE / TRANSMIT

SEMANTICS

THE <TERMINAL REQUEST STATEMENT> ASSOCIATES THE INPUT-OUTPUT CAPABILITIES OF A TERMINAL WITH THE <REQUEST IDENTIFIER>S WHICH WILL HANDLE INPUT FROM (RECEIVE) AND OUTPUT TO (TRANSMIT) THE TERMINAL.
THE <REQUEST ACTION>S DEFINED FOR THE TERMINAL WILL BE CHECKED FOR CONSISTENCY WITH THE <MYUSE STATEMENT> IN ANY STATION DEFINITION REFERENCING THE TERMINAL. IF, FOR EXAMPLE, A STATION SPECIFIES BOTH INPUT AND OUTPUT CAPABILITIES IN THE <MYUSE STATEMENT>, BUT THE ASSOCIATED TERMINAL HAS ONLY A RECEIVE REQUEST DEFINED, AN ERROR WILL OCCUR. ONLY ONE OF EACH OF THE TWO <REQUEST ACTION>S IS PERMITTED.
THE <EMPTY> APPLICATION QUALIFIER IS EQUIVALENT TO THE NON-EMPTY FORM HAVING AN APPLICATION NUMBER EQUAL TO ONE. THE APPLICATION QUALIFIER ALLOWS THE SPECIFICATION OF SIX OR LESS LINE DISCIPLINE ROUTINES FOR THE SAME DEVICE WHICH PRESUMABLY DIFFER FOR "APPLICATION" DEPENDENT REASONS. THE "STANDARD" APPLICATION- THAT USED FOR INITIALIZATION- IS THE ONE HAVING AN APPLICATION NUMBER EQUALING ONE; FOR THIS REASON, THIS "APPLICATION REQUEST SET" MUST BE SPECIFIED.

SPECIAL CHARACTER STATEMENT

SYNTAX

```
<SPECIAL CHARACTER STATEMENT> ::= <SPECIAL CHARACTER NAME> =
      <UNITARY STRING> <DYNAMIC PART>
<SPECIAL CHARACTER NAME> ::= END / BACKSPACE / LINEDELETE / WRU
```

SEMANTICS

THE <SPECIAL CHARACTER STATEMENT> DEFINES THE END, BACKSPACE, LINEDELETE, AND WRU CHARACTER FOR A TERMINAL. THE <SPECIAL CHARACTER STATEMENT> FOR A PARTICULAR SPECIAL CHARACTER MUST BE GIVEN IF A REQUEST FOR THE TERMINAL MENTIONS THE SPECIAL CHARACTER. (BACKSPACE CHARACTER MEANS BOTH BACKSPACE AND DELETE PRECEDING CHARACTER NOT MERELY BACKSPACE AS CARRIAGE OR CARRIER MOVEMENT.)

TRANSMISSION NUMBER STATEMENT

SYNTAX

```
<TRANSMISSION NUMBER STATEMENT> ::= TRANSMISSION =
      <TRANSMISSION NUMBER>
<TRANSMISSION NUMBER> ::= 0 / 1 / 2 / 3 / NULL
```

SEMANTICS

THE <TRANSMISSION NUMBER STATEMENT> SPECIFIES THE LENGTH OF THE TRANSMISSION NUMBER. THE NUMBER ZERO AND NULL ARE EQUIVALENT AND SPECIFY NO TRANSMISSION NUMBER WILL BE USED. A NON-NULL TRANSMISSION NUMBER MUST BE SPECIFIED IF THE ITEM "TRAN" IS MENTIONED IN A REQUEST OF THE TERMINAL.

WRAPAROUND STATEMENT

SYNTAX

```
<WRAPAROUND STATEMENT> ::= WRAPAROUND = <LOGICAL VALUE>
```

SEMANTICS

THE <WRAPAROUND STATEMENT> SPECIFIES WHETHER AUTOMATIC WRAPAROUND OF MESSAGES EXCEEDING THE LINE WIDTH OCCURS -- THAT IS, WHETHER THE DEVICE, IN RECEIVING A MESSAGE EXCEEDING THE LINE WIDTH, SEGMENTS AUTOMATICALLY THE RECEIVED MESSAGE, PLACING THE EXCESS ON SUCCESSIVE LINES OR NOT.

INHIBIT SYNC STATEMENT

SYNTAX

<INHIBITSYNC STATEMENT> ::= INHIBITSYNC = <LOGICAL VALUE>

SEMANTICS

THE INHIBITSYNC STATEMENT INDICATES THAT SYNCRONIZATION CHARACTERS SHOULD NOT BE TRANSMITTED AND WILL NOT BE RECEIVED FROM THE TERMINAL.

2.3.5. STATION SECTION

SYNTAX

```
<STATION SECTION> ::= <STATION LIST>
<STATION LIST> ::= <STATION DEFINITION> /
      <STATION DEFAULT DEFINITION> / <STATION LIST>
      <STATION DEFINITION> / <STATION LIST>
      <STATION DEFAULT DEFINITION>
<STATION DEFINITION> ::= STATION
      <STATION IDENTIFIER> : <STATION ATTRIBUTE LIST>
<STATION IDENTIFIER> ::= <GENERALIZED IDENTIFIER>
<STATION DEFAULT DEFINITION> ::= STATION DEFAULT
      <STATION DEFAULT IDENTIFIER> :
      <STATION DEFAULT ATTRIBUTE LIST>
<STATION DEFAULT IDENTIFIER>::= <GENERALIZED IDENTIFIER>
<STATION DEFAULT ATTRIBUTE LIST>::=<STATION DEFAULT ATTRIBUTE
      STATEMENT>./
      <STATION DEFAULT ATTRIBUTE LIST> <STATION DEFAULT ATTRIBUTE
      STATEMENT>.
<STATION DEFAULT ATTRIBUTE STATEMENT> ::=
      <STATION DEFAULT STATEMENT> / <STATION ADDRESS STATEMENT> /
      <STATION WIDTH STATEMENT> / <STATION PAGE STATEMENT> /
      <STATION TIMELIMIT STATEMENT> /
      <CONTROL CHARACTER STATEMENT> / <WRAPAROUND STATEMENT> /
      <MYUSE STATEMENT> / <SPO STATEMENT> /
      <LOGICALACK STATEMENT> / <MCS STATEMENT> /
      <STATION PHONE STATEMENT> / <ENABLEINPUT STATEMENT> /
      <FREQUENCY STATEMENT> / <STATION ADAPTER STATEMENT> /
      <STATION MODEM STATEMENT> / <STATION TERMINAL STATEMENT> /
      <RETRY STATEMENT>
<STATION ATTRIBUTE LIST> ::= <STATION ATTRIBUTE STATEMENT>. /
      <STATION ATTRIBUTE LIST> <STATION ATTRIBUTE STATEMENT>.
<STATION ATTRIBUTE STATEMENT> ::=
```

<STATION DEFAULT ATTRIBUTE STATEMENT> /
<ALTERNATE STATEMENT>

SEMANTICS

A <STATION DEFINITION> MAY BE THOUGHT OF AS A LIST OF THE LOGICAL ATTRIBUTES OF THE STATION BEING DESCRIBED.

STATION DEFAULT STATEMENT

SYNTAX

<STATION DEFAULT STATEMENT> ::= DEFAULT =
<STATION DEFAULT IDENTIFIER>

SEMANTICS

THE <STATION DEFAULT STATEMENT> IS USED IN THE SAME MANNER THAT A <TERMINAL DEFAULT STATEMENT> IS USED. THE <TERMINAL DEFAULT STATEMENT> IS DISCUSSED UNDER THE DESCRIPTION OF TERMINAL DEFAULT SEMANTICS.

STATION ADDRESS STATEMENT

SYNTAX

<STATION ADDRESS STATEMENT> ::= ADDRESS = <ADDRESS LIST PART>
<ADDRESS LIST PART> ::= <ADDRESS PART> / <ADDRESS LIST PART>,
<ADDRESS PART>
<ADDRESS PART> ::= <ADDRESS LIST> / <ADDRESS PAIR LIST>
<ADDRESS LIST> ::= <ADDRESS> / <ADDRESS LIST>, <ADDRESS>
<ADDRESS> ::= <STRING>
<ADDRESS PAIR LIST> ::= <ADDRESS PAIR> / <ADDRESS PAIR LIST>,

<ADDRESS PAIR>
<ADDRESS PAIR> ::= (<ADDRESS>,<ADDRESS>)

SEMANTICS

THE <STATION ADDRESS STATEMENT> IS USED TO SPECIFY THE ADDRESS OF A STATION FOR OPERATIONS SUCH AS POLLING AND SELECTING.
IF NO <STATION ADDRESS STATEMENT> IS DEFINED FOR A STATION, THE MODE OF THE ASSOCIATED TERMINAL MUST BE CONTENTION.
THE LENGTH OF THE ADDRESS SPECIFIED MUST BE EQUAL TO THE LENGTH AS DEFINED IN THE <TERMINAL ADDRESS STATEMENT> OF THE ASSOCIATED TERMINAL.
THE MULTIPLE ENTRY FORM OF THE <ADDRESS LIST> IS USED IN CONJUNCTION WITH DIALIN LINES. DIALIN LINES ARE DISCUSSED IN THE SECTION ON LINE SEMANTICS. THE ADDRESS PAIR FORM OF THE ADDRESS LIST IS APPROPRIATE ONLY IF THE STATIONS TERMINALS ADDRESS STATEMENT ALLOWED THAT THE RECEIVE ADDRESS MAY BE DIFFERENT FROM THE TRANSMIT ADDRESS. (THE FIRST MEMBER OF AN ADDRESS PAIR IS THE RECEIVE ADDRESS, THE SECOND THE TRANSMIT ADDRESS.)

STATION WIDTH STATEMENT

SYNTAX

<STATION WIDTH STATEMENT> ::= WIDTH = <INTEGER>

SEMANTICS

THE <STATION WIDTH STATEMENT> SPECIFIES THE MAXIMUM NUMBER OF CHARACTERS IN A LOGICAL LINE OF OUTPUT. THE <INTEGER> DEFINED MUST BE LESS THAN OR EQUAL TO THE NUMBER OF CHARACTERS SPECIFIED FOR THE <TERMINAL WIDTH STATEMENT>, UNDER THE DEFINITION OF THE STATIONS TERMINAL. IF NO <STATION WIDTH STATEMENT> IS INCLUDED IN THE STATION DESCRIPTION, THEN INFORMATION SPECIFIED FOR THE TERMINAL

ASSOCIATED WITH THE STATION WILL BE USED AS A DEFAULT WIDTH SIZE.

STATION PAGE STATEMENT

SYNTAX

<STATION PAGE STATEMENT> ::= PAGE = <INTEGER>

SEMANTICS

THE <STATION PAGE STATEMENT> DEFINES THE NUMBER OF LOGICAL LINES PER LOGICAL PAGE. THE <INTEGER> SPECIFIED MUST BE LESS THAN OR EQUAL TO THE NUMBER OF LINES SPECIFIED FOR THE <TERMINAL PAGE STATEMENT> FOR THE STATIONS TERMINAL UNLESS THE NUMBER IS ZERO INDICATING PAGING CAN BE ARBITRARY. IF A <STATION PAGE STATEMENT> IS NOT INCLUDED IN THE STATION DESCRIPTION, THE STATIONS TERMINAL SPECIFICATIONS FOR PAGINATION ARE ASSUMED.

STATION TIME LIMIT STATEMENT

SYNTAX

<STATION TIMELIMIT STATEMENT> ::= TIMELIMIT = <TIME>

SEMANTICS

THE <STATION TIMELIMIT STATEMENT> DEFINES THE INTERVAL OF TIME TO WAIT AFTER ACTIVITY CEASES FROM THAT STATION BEFORE ASSUMING THAT IT IS NO LONGER THERE. (ACTION UPON THE OCCURRENCE OF THIS CONDITION IS TO BE SPECIFIED.)

CONTROL CHARACTER STATEMENT

SYNTAX

<CONTROL CHARACTER STATEMENT> ::= CONTROL = <UNITARY STRING>

SEMANTICS

THE <CONTROL CHARACTER STATEMENT> DEFINES A CHARACTER WHOSE APPEARANCE IN A RECEIVE TEXT FROM THE STATION SPECIFIES THAT THE CHARACTERS FOLLOWING IT ARE TO BE SENT DIRECTLY TO THE STATIONS CONTROLLING MCS SHOULD THE CHARACTER BE THE FIRST NON-BLANK CHARACTER WITHIN THE TEXT, AND THE REQUEST DETECTS IT.

WRAPAROUND STATEMENT

SYNTAX

<WRAPAROUND STATEMENT> ::= WRAPAROUND = <LOGICAL VALUE>

SEMANTICS

THE <WRAPAROUND STATEMENT> IF SET TRUE PREVENTS THE TRUNCATION OF MESSAGES WHOSE LENGTH IS GREATER THAN THAT SPECIFIED BY THE <WIDTH STATEMENT>. IF THE WIDTH WAS DEFINED TO BE 35 CHARACTERS AND 40 CHARACTERS ARE TRANSMITTED AS OUTPUT, THE LAST 5 CHARACTERS WILL BE PRINTED ON THE NEXT LINE IF WRAPAROUND IS SET TRUE.

MYUSE STATEMENT

SYNTAX

<MYUSE STATEMENT> ::= MYUSE = <IO LIST>
<IO LIST> ::= INPUT / OUTPUT / INPUT, OUTPUT / OUTPUT, INPUT

SEMANTICS

THE <MYUSE STATEMENT> SPECIFIES IF A STATION IS TO BE USED FOR INPUT ONLY, OUTPUT ONLY, OR BOTH INPUT AND OUTPUT.

A <REQUEST STATEMENT> MUST BE DEFINED WITHIN THE TERMINAL DESCRIPTION ASSOCIATED WITH THIS STATION FOR HANDLING THE INPUT AND/OR OUTPUT CAPABILITIES AS SPECIFIED IN THE <MYUSE STATEMENT>. THUS, IF THE STATION IS TO RECEIVE INPUT FROM AND TRANSMIT OUTPUT TO THE SYSTEM, THE TERMINAL NAMED IN THE <TERMINAL STATEMENT> FOR THE STATION MUST CONTAIN A TRANSMIT AND RECEIVE REQUEST.

SPO STATEMENT

SYNTAX

<SPO STATEMENT> ::= SPO = <LOGICAL VALUE>

SEMANTICS

THE <SPO STATEMENT> SPECIFIES WHETHER OR NOT THE STATION CAN BE TREATED AS A SPO. THAT IS, ALL SYSTEM MESSAGES MAY BE DIRECTED TO THE STATION, AND THE STATION MAY INPUT MESSAGES TO THE SYSTEM AS IF IT WERE THE CENTRAL SPO. (THIS IS A TENTATIVE POSITION.)

LOGICALACK STATEMENT

SYNTAX

<LOGICALACK STATEMENT> ::= LOGICALACK = <LOGICAL VALUE>

SEMANTICS

THE CONSEQUENCES OF THIS STATEMENT ARE EXPLAINED WITH THE RELATED REQUEST STATEMENTS, "TERMINATE LOGICALACK" AND "TERMINATE LOGICALACK(RETURN)"

MCS STATEMENT

SYNTAX

<MCS STATEMENT> ::= MCS = <MCS IDENTIFIER>
<MCS IDENTIFIER> ::= <GENERALIZED IDENTIFIER>

SEMANTICS

THE <MCS STATEMENT> GIVES THE NAME OF THE MCS WHICH IS TO CONTROL AND HANDLE MESSAGES TO AND FROM THE STATION AND MAINTAIN POSSIBLE SECURITY AND HANDLE ERROR RECOVERY FOR THE STATION.

STATION PHONE STATEMENT

SYNTAX

<STATION PHONE STATEMENT> ::= PHONE = <INTEGER>

SEMANTICS

THE <PHONE STATEMENT> LISTS THE TELEPHONE NUMBER ASSOCIATED WITH THE STATION BEING DESCRIBED. IMBEDDED HYPHENS (AS NORMALLY USED IN PHONE NUMBERS) ARE NOT ALLOWED AND SHOULD BE DELETED. THE PRESENCE OF THE PHONE STATEMENT IMPLIES THAT THE SYSTEM CAN DIAL THE STATION.

ENABLEINPUT STATEMENT

SYNTAX

<ENABLEINPUT STATEMENT> ::= ENABLEINPUT = <LOGICAL VALUE>

SEMANTICS

THE <ENABLEINPUT STATEMENT> IF SET TRUE IMPLIES THAT, IF THE STATION IS ON AN ATTACHED LINE, INPUT FROM THE STATION WILL BE ACCEPTED AND/OR SOLICITED (IF THE STATIONS TERMINAL IS A POLLED DEVICE) UPON COMPLETION OF THE INITIALIZATION OF THE DATA COMMUNICATIONS SYSTEM WITHOUT HAVING A SPECIFIC REQUEST TO DO SO FROM A MCS.

FREQUENCY STATEMENT

SYNTAX

<FREQUENCY STATEMENT> ::= FREQUENCY = <TIME>

SEMANTICS

THE <FREQUENCY STATEMENT> SPECIFIES THE APPROXIMATE INTERVALS IN SECONDS AT WHICH THE STATION BEING DESCRIBED IS TO BE POLLED. THE MAXIMUM NUMBER WHICH MAY BE SPECIFIED IS 255 SECONDS. THE <TIME> WILL BE ROUNDED TO THE NEAREST SECOND.

STATION ADAPTER STATEMENT

SYNTAX

<STATION ADAPTER STATEMENT> ::= ADAPTER = <ADAPTER TYPE>

SEMANTICS

THE <STATION ADAPTER STATEMENT> SPECIFIES THE <ADAPTER TYPE> TO BE USED WITH THE STATION BEING DESCRIBED. THE <INTEGER> MUST BE CHOSEN FROM THE LIST OF VALUES DEFINED IN THE "TABLE OF DATACOM ADAPTER TYPES" IN AN APPENDIX TO THIS DOCUMENT. THE <ADAPTER TYPE> MUST BE COMPATIBLE WITH THE LIST OF THE TYPES NAMED IN THE TERMINAL DESCRIPTION ASSOCIATED WITH THE STATION.

STATION MODEM STATEMENT

SYNTAX

<STATION MODEM STATEMENT> ::= MODEM = <MODEM IDENTIFIER>

SEMANTICS

THE <MODEM IDENTIFIER> MUST REFERENCE A DESCRIPTION PREVIOUSLY DEFINED IN THE MODEM SECTION. THE LIST OF ADAPTER TYPES FOR THE <MODEM IDENTIFIER> WILL BE CHECKED AGAINST THE STATION AND TERMINAL <ADAPTER TYPE LIST>S FOR COMPATIBILITY. IF INCONSISTENCIES ARISE, AN ERROR WILL BE GENERATED BY THE COMPILER. IF THE STATION BEING DESCRIBED, OR THE LINE ASSOCIATED WITH THE STATION HAS NO MODEM STATEMENT, NDL WILL ASSUME A DIRECT CONNECTION BETWEEN THE ADAPTER AND THE STATION. THE STATEMENT IS NECESSARY IF THE STATIONS TERMINAL IS COMPATIBLE WITH SEVERAL TYPES OF ADAPTERS IN ORDER TO RESOLVE AMBIGUITY. THE <MODEM IDENTIFIER> MUST REFERENCE A DEFINITION PREVIOUSLY GENERATED BY THE NDL PROCESSOR.

STATION TERMINAL STATEMENT

SYNTAX

<STATION TERMINAL STATEMENT> ::= TERMINAL = <TERMINAL IDENTIFIER>

SEMANTICS

THE <STATION TERMINAL STATEMENT> SPECIFIES THE NAME OF THE TERMINAL DESCRIPTION WHICH DEFINES THE PHYSICAL CHARACTERISTICS OF THE STATION BEING DESCRIBED. MORE THAN ONE STATION DESCRIPTION MAY REFERENCE THE SAME TERMINAL DESCRIPTION.

ALTERNATE STATEMENT

SYNTAX

<ALTERNATE STATEMENT> ::= ALTERNATE = <ALTERNATE LIST>
<ALTERNATE LIST> ::= <STATION IDENTIFIER> / <ALTERNATE LIST>,
<STATION IDENTIFIER>

SEMANTICS

THE <ALTERNATE STATEMENT> IS COMMENTARY AS FAR AS THE B6700 DATACOM SYSTEM IS CONCERNED.

RETRY STATEMENT

SYNTAX

<RETRY STATEMENT> ::= RETRY = <INTEGER>

SEMANTICS

THE <RETRY STATEMENT> GIVES INITIAL AND/OR DEFAULT RETRY COUNT FOR THE STATION. THE VALUE CAN BE CHANGED BY THE STATIONS CONTROLLING MCS ONCE IT IS ATTACHED. (FOR DETAILS SEE DCALGOL INFORMATION MANUAL - DOCUMENT NUMBER 5000052.)

2.3.6. LINE SECTION

SYNTAX

```
<LINE SECTION> ::= <LINE LIST>
<LINE LIST> ::= <LINE DEFINITION> / <LINE LIST> <LINE DEFINITION> /
      <LINE DEFAULT DEFINITION> / <LINE LIST>
      <LINE DEFAULT DEFINITION>
<LINE DEFAULT DEFINITION> ::= LINE DEFAULT
      <LINE DEFAULT IDENTIFIER> : <LINE DEFAULT ATTRIBUTE LIST>
<LINE DEFAULT IDENTIFIER> ::= <IDENTIFIER>
<LINE DEFINITION> ::= LINE <LINE IDENTIFIER> :
      <LINE ATTRIBUTE LIST>
<LINE IDENTIFIER> ::= <IDENTIFIER>
<LINE DEFAULT ATTRIBUTE LIST> ::=
      <LINE DEFAULT ATTRIBUTE STATEMENT>. /
      <LINE DEFAULT ATTRIBUTE LIST>
      <LINE DEFAULT ATTRIBUTE STATEMENT>.
<LINE DEFAULT ATTRIBUTE STATEMENT> ::= <LINE DEFAULT STATEMENT> /
      <LINE ADDRESS STATEMENT> /
      <LINE ADAPTER CLASS STATEMENT> / <LINE PHONE STATEMENT> /
      <LINE MODEM STATEMENT> / <MAXSTATIONS STATEMENT>
<LINE ATTRIBUTE LIST> ::= <LINE ATTRIBUTE STATEMENT>. /
      <LINE ATTRIBUTE LIST> <LINE ATTRIBUTE STATEMENT>.
<LINE ATTRIBUTE STATEMENT> ::= <LINE DEFAULT ATTRIBUTE STATEMENT> /
      <LINE TYPE STATEMENT> / <LINE STATION STATEMENT> /
      <ANSWER STATEMENT> / <ENDOFNUMBER STATEMENT>
```

SEMANTICS

THE LINE SECTION OF A NETWORK DEFINITION DESCRIBES THE INITIAL STATE OF ALL LINES IN THE NETWORK AND THEIR LOGICAL RELATIONSHIP TO THE STATIONS IN THE NETWORK.

LINE DEFAULT STATEMENT

SYNTAX

<LINE DEFAULT STATEMENT> ::= DEFAULT = <LINE DEFAULT IDENTIFIER>

SEMANTICS

THE <LINE DEFAULT STATEMENT> IS USED IN THE SAME MANNER THAT A <TERMINAL DEFAULT STATEMENT> IS USED. THE <TERMINAL DEFAULT STATEMENT> IS DISCUSSED UNDER THE DESCRIPTION OF TERMINAL DEFAULT SEMANTICS.

LINE ADDRESS STATEMENT

SYNTAX

```
<LINE ADDRESS STATEMENT> ::= ADDRESS =
      <DCP NUMBER> : <CLUSTER ADDRESS> : <ADAPTER ADDRESS>
<DCP NUMBER> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7
<CLUSTER ADDRESS> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 /
      10 / 11 / 12 / 13 / 14 / 15
<ADAPTER ADDRESS> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 /
      10 / 11 / 12 / 13 / 14 / 15
```

SEMANTICS

THE LINE ADDRESS STATEMENT IS USED TO SPECIFY THE DCP NUMBER, THE CLUSTER NUMBER AND THE ADAPTER NUMBER WITHIN THE CLUSTER TO WHICH THE LINE IS ATTACHED.

EXAMPLE:

```
ADDRESS = 2 : 0 : 15.
```

THIS STATEMENT WOULD APPEAR IN THE LINE DESCRIPTION OF THE LINE WHICH IS CONNECTED TO CLUSTER NUMBER ZERO AT ADAPTER NUMBER 15 OF DCP NUMBER TWO. THAT WOULD BE THE LOWEST-NUMBERED CLUSTER AND THE HIGHEST-NUMBERED ADAPTER WITHIN THAT CLUSTER.

LINE TYPE STATEMENT

SYNTAX

```
<LINE TYPE STATEMENT> ::= TYPE = <TYPE LIST>
<TYPE LIST> ::= <TYPE> / <TYPE LIST>, <TYPE>
<TYPE> ::= DIALIN / DIALOUT: <LINE IDENTIFIER> / REMOTE / DUPLEX /
      DUPLEX (RECEIVE): <LINE IDENTIFIER> /
      DUPLEX (TRANSMIT) : <LINE IDENTIFIER> /
      TOUCHTONE : <LINE IDENTIFIER> /
      PASSIVE
```

SEMANTICS

THE TYPE STATEMENT SPECIFIES THE FOLLOWING INFORMATION:

1. LINES WHICH CAN BE DIALED (BY THE SYSTEM)

2. LINES WHICH DIAL IN (TO THE SYSTEM)

3. SPARE ADAPTERS (PASSIVE LINES)

4. TOUCHTONE-VOICE RESPONSE PAIRS

5. PAIRS OF LINES USED TO SIMULATE DUPLEX MODE USING TWO HALF-DUPLEX LINES

6. PSEUDO-LINES EQUIVALENT TO DIALABLE SETS OF STATIONS.

THE "TABLE OF ALLOWABLE COMBINATIONS OF LINE TYPES" CONTAINED IN THE APPENDIX SUMMARIZES THE ALLOWABLE COMBINATIONS OF TYPES, WHICH MAY BE SPECIFIED FOR ONE LINE.

NOTE: LINES RELATED THROUGH A TYPE STATEMENT MUST BE IN

THE SAME DCP.

DIALOUT

A LINE WHICH IS ATTACHED TO AN AUTOMATIC CALLING UNIT (ACU) AND IS CAPABLE OF DIALING OUT TO A DATASET IS CALLED DIALOUT. THE <LINE IDENTIFIER> DESIGNATES THE ACU ASSOCIATED WITH THE DIALOUT LINE. THE DIALOUT LINE AND ITS ASSOCIATED ACU MAY BE USED BY THE SYSTEM TO CALL UP ANY COMPATIBLE REMOTE LINE OR STATION WHICH CAN BE DIALED. THE <LINE IDENTIFIER> NAMED IN THE DIALOUT LINES DESCRIPTION WILL BE CHECKED TO ASSURE THAT THE LINE HAS A <LINE ADAPTER CLASS STATEMENT> SPECIFYING THE ACU ADAPTER.

DIALIN

A LINE WHICH MAY BE DIALED FROM A REMOTE STATION IS A DIALED LINE. THE STATION GIVEN IS A "PSUEDO" STATION, WHICH IS USED ONLY TO DEFINE A LIST OF ADDRESSES WHICH MAY DIAL-UP THAT LINE AND ARE TO BE RECOGNIZED AS VALID STATIONS ON IT. IF A <LINE PHONE STATEMENT> IS GIVEN FOR A DIALED LINE, IT IS CONSIDERED DOCUMENTARY ONLY.

PASSIVE

AN ADAPTER TO WHICH NO LINE IS CONNECTED -- A SPARE ADAPTER -- IS CALLED A PASSIVE LINE. FOR THE DESCRIPTION OF A PASSIVE LINE ONLY THE <LINE ADDRESS STATEMENT>, <LINE ADAPTER CLASS STATEMENT>, AND THE <LINE TYPE STATEMENT> MAY BE SPECIFIED.

EXAMPLE:

```
LINE L1:
     ADDRESS = 4:2:1.
     ADAPTER = 1.
     TYPE    = PASSIVE.
```

THESE THREE STATEMENTS MUST APPEAR FOR EACH DESCRIPTION OF A

PASSIVE LINE.

REMOTE

A REMOTE LINE IS EQUIVALENT TO A SET OF STATIONS WHICH CAN BE DIALED OUT BY THE SYSTEM SIMULTANEOUSLY WITH ONE PHONE NUMBER. THE REMOTE LINE SHOULD NOT BE ASSOCIATED WITH ANY SPECIFIC CLUSTER-ADAPTER ADDRESS. THE REMOTE LINE WILL BE DIALED BY USING A DIALOUT LINE WHOSE ADAPTER TYPE IS COMPATIBLE WITH THAT OF THE REMOTE. THE <LINE PHONE STATEMENT> SPECIFIES THE NUMBER OF THE DATASET ASSOCIATED WITH THE REMOTE LINE.

EXAMPLE:

```
LINE XXX:
     TYPE    = REMOTE.
     PHONE   = 2143750123.
     STATION = STAT1, THEM, US.

LINE YYY:
     ADDRESS = 2:4:1.
     TYPE    = DIALOUT: LACU1.
```

TO REACH "THEM" (STATION IDENTIFIER IN THE STATION ATTACH STATEMENT IN THE LINE DESCRIPTION OF "XXX"), THE ACU WOULD DIAL THE PHONE NUMBER GIVEN IN THE <LINE PHONE STATEMENT> AND WOULD FIND A LINE SUCH AS "YYY", WHICH CAN BE DIALED OUT, AS SPECIFIED IN THE <LINE TYPE STATEMENT>

DUPLEX

A PAIR OF LINES, ONE TO TRANSMIT AND ONE TO RECEIVE, WHICH SIMULTANEOUSLY SEND AND RECEIVE TO/FROM A PARTICULAR STATION ARE CALLED A "DUPLEX LINE PAIR". THE LINE SPECIFIED AS FOR TRANSMISSION OF DATA WILL ALWAYS BE USED FOR TRANSMITTING DATA. NEVER FOR RECEIVING. FOR A TYPE DUPLEX THE RECEIVING LINE IS THE MASTER. THE MASTER LINE MUST ALWAYS HAVE THE LIST OF STATIONS ATTACHED TO THE LINE PAIR.

NOTE: THE RECEIVING AND/OR TRANSMITTING LINE IS ALWAYS DESIGNATED WITH RESPECT TO THE SYSTEM.

THE RECEIVING LINE IS DESIGNED AS TYPE DUPLEX (RECEIVE). THE <LINE IDENTIFIER> FOLLOWING THE <LINE TYPE STATEMENT> SPECIFIES THE OTHER LINE OF THE DUPLEX PAIR, WHICH MUST BE A DUPLEX (TRANSMIT) LINE. BOTH LINES OF A DUPLEX PAIR MUST BE ON THE SAME DCP.

```
LINE L1:

    ADAPTER = 1.
    ADDRESS = 1:1:7.
    TYPE    = DUPLEX(TRANSMIT):L2.

LINE L2:

    ADAPTER = 1.
    ADDRESS = 1:2:8.
    TYPE    = DUPLEX(RECEIVE):L1.
    STATION = NUMBER1.
```

THESE TWO LINE DESCRIPTIONS WOULD DESCRIBE THE SITUATION SHOWN BELOW:

LINE "L1" TRANSMITS INFORMATION TO STATION "NUMBER1" AND ALL INFORMATION FROM THE STATION TO THE SYSTEM GOES ON LINE "L2".

STATIONS WHICH ARE ASSOCIATED WITH DUPLEX LINES MUST HAVE THE <TERMINAL DUPLEX STATEMENT> SET TRUE IN THEIR <TERMINAL DESCRIPTION>.

LINES WHICH ARE DEFINED TO THE NETWORK AS REMOTE MAY ALSO BE DUPLEX, BUT CAN NOT BE SPECIFIED AS DUPLEX (RECEIVE) OR DUPLEX (TRANSMIT), CONVERSELY DUPLEX BY ITSELF IS ALLOWED ONLY FOR USE WITH REMOTE.

LINE ADAPTER CLASS STATEMENT

SYNTAX

<LINE ADAPTER CLASS STATEMENT> ::= ADAPTER = <LINE ADAPTER CLASS>
<CONNECTION TYPE>
<LINE ADAPTER CLASS> ::= 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8
<CONNECTION TYPE> ::= (DIRECT) / (MODEM) / <EMPTY>

SEMANTICS

THE <LINE ADAPTER CLASS STATEMENT> DEFINES THE UNIQUE ADAPTER CLASS ASSOCIATED WITH THE LINE. THE <INTEGER> LISTED MUST BE A NUMBER FROM ONE TO EIGHT INCLUSIVE, WHICH IS CHOSEN FROM THE "TABLE OF DATACOM ADAPTER CLASSES" DEPENDING ON THE CHARACTERISTICS OF THE ADAPTER ASSOCIATED WITH THE LINE. THE <LINE ADAPTER CLASS STATEMENT> WILL BE CHECKED FOR VALIDITY AGAINST THE <LINE MODEM STATEMENT>. THE <CONNECTION TYPE> SPECIFIES WHICH OF THE VARIANTS CLASSES IS ON THE LINE (SEE THE TABLE OF ADAPTER CLASSES FOR THE POSSIBLE VARIANTS.) A <CONNECTION TYPE> OF <EMPTY> SPECIFIES THAT THE PRESENCE OR ABSENCE OF A <MODEM STATEMENT> DETERMINES THE ADAPTER CLASS VARIANT.

LINE PHONE STATEMENT

SYNTAX

<LINE PHONE STATEMENT> ::= PHONE = <INTEGER>

SEMANTICS

THE <LINE PHONE STATEMENT> LISTS THE TELEPHONE NUMBER OF THE LINE

BEING DESCRIBED (A DIAL UP LINE). IMBEDDED HYPHENS (AS NORMALLY APPEAR IN PHONE NUMBERS) ARE NOT ALLOWED AND SHOULD BE OMITTED.

LINE MODEM STATEMENT

SYNTAX

<LINE MODEM STATEMENT> ::= MODEM = <MODEM IDENTIFIER>

SEMANTICS

THE <MODEM IDENTIFIER> MUST REFERENCE A DESCRIPTION PREVIOUSLY DEFINED IN THE MODEM SECTION. THE <ADAPTER TYPE LIST> IN THE MODEM DESCRIPTION WILL BE CHECKED FOR COMPATIBILITY WITH THE ADAPTER CLASS SPECIFIED FOR THE LINE. IF INCONSISTENCIES EXIST, THE COMPILER WILL GENERATE AN ERROR.

LINE STATION STATEMENT

SYNTAX:

<LINE STATION STATEMENT> ::= STATION = <LIST OF STATIONS>
<LIST OF STATIONS> ::= <STATION IDENTIFIER> / <STATION LIST>,
<STATION IDENTIFIER>

SEMANTICS

THE <LINE STATION STATEMENT> IDENTIFIES THE STATIONS ON A LINE. IN THE CASE OF DIAL-IN LINES, THE <STATION LIST> MUST LIST ONE (1) PSUEDO STATION THAT IS REPRESENTATIVE OF ALL THE STATIONS THAT CAN DIAL IN ON THE LINE. LINES DESIGNATED AS PASSIVE OR DUPLEX (TRANSMIT), AND LINES WHICH HAVE AN ACU ADAPTER CLASS MUST NOT HAVE A <LINE STATION STATEMENT>. THE STATIONS ON A DUPLEX PAIR MUST BE MENTIONED ONLY ON LINE THAT IS DUPLEX (RECEIVE).

ANSWER STATEMENT

SYNTAX

<ANSWER STATEMENT> ::= ANSWER = <LOGICAL VALUE>

SEMANTICS

IF THE LINE TYPE IS DIALIN, AN <ANSWER STATEMENT> IS REQUIRED TO INDICATE WHETHER OR NOT THE DCP IS TO ANSWER THE RING AUTOMATICALLY. AN MCS MAY ALTER THIS VALUE AFTER INITIALIZATION.

ENDOFNUMBER STATEMENT

SYNTAX

<ENDOFNUMBER STATEMENT> ::= ENDOFNUMBER = <LOGICAL VALUE>

SEMANTICS

LINES WHICH HAVE AN ACU ADAPTER CLASS (ADAPTER = 8) MUST ALSO HAVE SPECIFIED WHETHER THE ATTACHED ACU HAS THE "END OF NUMBER" OPTION.

MAXSTATIONS STATEMENT

SYNTAX

<MAXSTATIONS STATEMENT> ::= MAXSTATIONS = <INTEGER>

SEMANTICS

THE <MAXSTATIONS STATEMENT> SPECIFIES THE NUMBER OF STATIONS THAT MAY BE ATTACHED TO A LINE. THE <MAXSTATIONS STATEMENT> MAY BE SPECIFIED FOR ANY LINE HAVING OR POTENTIALLY HAVING STATIONS. IF THE STATEMENT IS NOT SPECIFIED FOR A LINE HAVING STATIONS IT IS ASSUMED THAT THE MAXIMUM NUMBER OF STATIONS EQUALS THE NUMBER EXPLICITLY SPECIFIED AS BEING ON THE LINE. THE <INTEGER> SPECIFIED MUST NOT EQUAL ZERO, EXCEED 25, OR BE LESS THAN THE ACTUAL NUMBER OF STATIONS SPECIFIED FOR THE LINE.

2.3.7. DCP SECTION

SYNTAX

```
<DCP SECTION> ::= <DCP LIST>
<DCP LIST> ::= <DCP DEFINITION> / <DCP LIST> <DCP DEFINITION>
<DCP DEFINITION> ::= DCP <DCP IDENTIFIER> :<DCP ATTRIBUTE LIST>
<DCP IDENTIFIER> ::= 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7
<DCP ATTRIBUTE LIST> ::= <DCP ATTRIBUTE STATEMENT>. /
      <DCP ATTRIBUTE LIST> <DCP ATTRIBUTE STATEMENT>.
<DCP ATTRIBUTE STATEMENT> ::= <MEMORY STATEMENT> /
      <EXCHANGE STATEMENT> / <DCP TERMINAL STATEMENT>
```

SEMANTICS

A DCP SECTION DESCRIBES A DATA COMMUNICATIONS PROCESSOR. THE IDENTIFIER FOR A DCP IS A NUMBER FROM ZERO THROUGH SEVEN. THIS VALUE CORRESPONDS WITH THE SCAN BUS ADDRESS OF THE DCP.

MEMORY STATEMENT

SYNTAX

```
<MEMORY STATEMENT>::= MEMORY = <MEMORY SIZE>
<MEMORY SIZE> ::= <INTEGER> / NULL
```

SEMANTICS

THE <MEMORY STATEMENT> SPECIFIES THE AMOUNT OF LOCAL MEMORY.

(NOTE: MEMORY = 0 IMPLIES NO LOCAL MEMORY.
MEMORY = N WHERE N IS NON-ZERO IMPLIES INFINITE LOCAL MEMORY.

I.E. ENOUGH TO CONTAIN ALL THE DCP OPERATING CODE.)

EXCHANGE STATEMENT

SYNTAX

<EXCHANGE STATEMENT> ::= EXCHANGE = <DCP IDENTIFIER>

SEMANTICS

AN <EXCHANGE STATEMENT> SPECIFIES THAT CLUSTER EXCHANGING IS ALLOWED BETWEEN TWO DCPS. IT IS PRESUMED THAT ALL CLUSTERS ON THE DCPS INVOLVED ARE EXCHANGED AND THAT THEY ARE DESIGNATED IDENTICALLY ON EACH DCP. (IT IS REQUIRED THAT THE LINES SPECIFIED FOR EACH DCP BE GIVEN ADDRESSES SUCH THAT THE INITIAL CONDITIONS ARE SUCH THAT THE DCPS DO NOT HAVE LINES ON THE SAME CLUSTER.)

A DCP NOT HAVING ANY LINES SPECIFIED FOR IT MUST HAVE AN <EXCHANGE STATEMENT>.

DCP TERMINAL STATEMENT

SYNTAX

<DCP TERMINAL STATEMENT> ::= TERMINAL = <TERMINAL LIST>
<TERMINAL LIST> ::= <TERMINAL IDENTIFIER> / <TERMINAL LIST>,
<TERMINAL IDENTIFIER>

SEMANTICS

THE <DCP TERMINAL STATEMENT> ALLOWS THE SPECIFICATION OF THE TERMINALS THAT THE DCP IS TO HANDLE - THAT IS, SYSTEM/DCPPROGEN WILL GENERATE CODE ONLY FOR THOSE TERMINALS MENTIONED.

SHOULD THIS STATEMENT NOT BE GIVEN, THE CODE WILL BE GENERATED TO HANDLE ALL OF THE TERMINALS SPECIFED IN THE <NETWORK DESCRIPTION>.

2.3.8. FILE SECTION

SYNTAX

```
<FILE SECTION> ::= <FILE DEFAULT LIST> <FILE LIST>
<FILE DEFAULT LIST> ::= <FILE DEFAULT DEFINITION> /
      <FILE DEFAULT LIST> <FILE DEFAULT DEFINITION> / <EMPTY>
<FILE DEFAULT DEFINITION> ::= FILE DEFAULT
      <FILE DEFAULT IDENTIFIER> : <FILE ATTRIBUTE LIST>
<FILE DEFAULT IDENTIFIER> ::= <GENERALIZED IDENTIFIER>
<FILE LIST> ::= <FILE DEFINITION> / <FILE LIST>, <FILE DEFINITION>
<FILE DEFINITION> ::= FILE <FILE IDENTIFIER> :
      <FILE ATTRIBUTE LIST>
<FILE IDENTIFIER> ::= <GENERALIZED IDENTIFIER>
<FILE ATTRIBUTE LIST> ::= <FILE ATTRIBUTE STATEMENT>. /
      <FILE ATTRIBUTE LIST> <FILE ATTRIBUTE STATEMENT>.
<FILE ATTRIBUTE STATEMENT> ::= <FILE DEFAULT STATEMENT> /
      <FAMILY STATEMENT> / <CONFLICTING ATTRIBUTE>
```

SEMANTICS

> NOTE: ALL CURRENT SEMANTICS FOR <FILE ATTRIBUTE STATEMENT>S ARE TENTATIVE AT THIS POINT.

THE FILE SECTION OF A NETWORK DEFINITION IS TO ASSOCIATE STATIONS WHICH ARE PART OF THE NETWORK WITH THE DATACOM FILE OF AN OBJECT JOB OR JOBS. THE STATIONS REFERENCED IN THE FILE SECTION MUST HAVE BEEN DEFINED IN THE STATION DESCRIPTION SECTION OF THE NETWORK DEFINITION.

FILE DEFAULT STATEMENT

SYNTAX

<FILE DEFAULT STATEMENT> ::= DEFAULT = <FILE DEFAULT IDENTIFIER>

SEMANTICS

THE <FILE DEFAULT STATEMENT> IS USED IN THE SAME MANNER THAT A <TERMINAL DEFAULT STATEMENT> IS USED. THE <TERMINAL DEFAULT STATEMENT> IS DISCUSSED UNDER THE DESCRIPTION OF TERMINAL DEFAULT SEMANTICS.

FAMILY STATEMENT

SYNTAX

```
<FAMILY STATEMENT> ::= FAMILY = <FAMILY LIST>
<FAMILY LIST> ::= <FAMILY DESIGNATOR> /
      <FAMILY LIST>, <FAMILY DESIGNATOR>
<FAMILY DESIGNATOR> ::= <FILE IDENTIFIER> / <STATION IDENTIFIER>
      <STATION ATTRIBUTE PART>
<STATION ATTRIBUTE PART> ::= (<ATTRIBUTE LIST>) / <EMPTY>
<ATTRIBUTE LIST> ::= [ TO BE SPECIFIED ]
```

SEMANTICS

THE <FAMILY STATEMENT> LISTS THE NAMES OF STATIONS OR OTHER FILES WHICH ARE TO BE CONSIDERED MEMBERS OF THE FILE BEING DESCRIBED. IF THE <FILE IDENTIFIER> OPTION IS USED AS PART OF THE <FAMILY DESIGNATOR>, THE <FILE IDENTIFIER> MUST BE PART OF A <FILE DEFINITION> WITHIN THE <FILE LIST>.

CONFLICTING ATTRIBUTES

SYNTAX

```
<CONFLICTING ATTRIBUTE> ::= <BACKSPACE STATEMENT> /
      <WRAPAROUND STATEMENT> / <LOGICALACK STATEMENT> /
      <MCS STATEMENT> / <USER STATEMENT> /
      <SECURITY STATEMENT> / <ENABLEINPUT STATEMENT>
```

SEMANTICS

TO BE SPECIFIED

3. USING THE NETWORK DEFINITION LANGUAGE PROCESSOR

3.1. PREPARING PROGRAMS

3.1.1. CONTROL CARDS AND FILES

NDL SOURCE LANGUAGE STATEMENTS NEED THE FOLLOWING CONTROL CARDS FOR COMPILATION ON THE B6700 BY SYSTEM/NDL.

```
        <I> RUN SYSTEM/NDL
        <I> BCL NDLSOURCE
            OR, FOR EBCDIC,
            <I> DATA NDLSOURCE
        [SOURCE LANGUAGE STATEMENTS]
        <I> END
```

<I> INDICATES AN INVALID PUNCH
IN COLUMN 1 .

INPUT FILES

EXTERNAL NAME	INTERNAL NAME	KIND
NDLSOURCE	CARD	
SOURCE/NDL	TAPE	DISK(1)

OUTPUT FILES

EXTERNAL NAME	INTERNAL NAME	KIND
LINE	LINE	PRINTER(7)
SYSTEM/NIF	NIF	DISK(1)
DC/CODE	DCPCODE	DISK(1)
SYSTEM/REQUESTIMAGE	REQIM	DISK(1)

DC/CODE, SYSTEM/REQUESTIMAGE AND SYSTEM/NIF ARE CREATED ONLY WHEN THE EXECUTION OF SYSTEM/NDL IS ERROR FREE.

3.1.2. USING OPTIONS

VARIOUS OPTIONS ARE AVAILABLE DURING COMPILATION AND ARE ACTIVATED BY DOLLAR SIGN ($) CONTROL CARDS.

SYNTAX FOR CONTROL STATEMENTS

```
<CONTROL STATEMENT>::=  $ <OPTION LIST>
<OPTION LIST>      ::=  <OPTION> / <OPTION LIST> <OPTION>
<OPTION>           ::=  <OPTION ACTION> <OPTION ITEM>
<OPTION ACTION>    ::=  SET / RESET / POP / <EMPTY>
<OPTION ITEM>      ::= LIST / CODE / VOID / VOIDT / SINGLE /
                       MERGE / NEW / SEQ <BASE> + <INCREMENT> /
                       SYNTAX / SEQ
<BASE>             ::= <INTEGER>
<INCREMENT>        ::= <INTEGER>
```

SEMANTICS

THE OPTION ACTIONS TURN ON AND OFF THE SPECIFIED ITEMS. SET CAUSES THE CURRENT SETTING OF AN OPTION TO BE PLACED IN A STACK FOR STORING OPTIONS, AND THE OPTION TO BE TURNED ON
RESET CAUSES THE CURRENT SETTING OF THE OPTION ITEM TO BE PLACED IN THE STACK, IN THE SAME MANNER AS SET DOES, BUT THEN TURNS OFF THE OPTION.
POP CAUSES THE CURRENT SETTING OF AN OPTION TO BE DISCARDED AND TO BE REPLACED BY THE PRIOR SETTING FROM THE STACK.

OPTIONS ARE IDENTICAL TO THOSE FOR THE ALGOL COMPILER EXCEPT WHERE INDICATED BELOW.

LIST -- CAUSES A LISTING TO BE GENERATED CONTAINING THE SOURCE LANGUAGE INPUT AND POSSIBLE ERROR MESSAGES, AND THE DCP PROGRAM GENERATOR LISTS THE DCP OPERATING SYSTEM(S).

CODE -- LISTS THE MACROS GENERATED BY SYSTEM/NDL FOR REQUESTS AND CAUSES THE OPERATING SYSTEM FOR THE DCP TO BE LISTED BY DCPPROGEN.

VOID -- CAUSES THE SOURCE-LANGUAGE INPUT PROCESS TO OPERATE NORMALLY, WITH SOURCE-LANGUAGE PRIMARY AND SECONDARY INPUT NOT COMPILED.

VOIDT -- LIKE VOID, HOWEVER THE CARD IMAGES ON THE SECONDARY INPUT ARE NOT COMPILED. ONLY PRIMARY COMPILED.

SINGLE -- CAUSES THE LISTING TO BE SINGLE SPACED.

MERGE -- CAUSES THE PRIMARY INPUT TO BE MERGED WITH SECONDARY INPUT. IF MATCHING SEQUENCE NUMBERS OCCUR, THE PRIMARY INPUT WILL REPLACE THE SECONDARY. WHEN THE MERGE OPTION IS OFF THE SECONDARY INPUT IS NOT USED.

NEW -- CREATE A SOURCE/NDL SYMBOLIC FILE.

SEQ -- CAUSES NEW SEQUENCE NUMBERS TO BE CREATED WITH THE FIRST INTEGER GIVEN AS THE INITIAL NEW SEQUENCE NUMBER AND SECOND AS AN INCREMENT.

SYNTAX -- IF SET BEFORE SOURCE IMAGES ARE PROCESSED, THE FILES "SYSTEM/NIF", "SYSTEM/REQUESTIMAGE", AND "DC/CODE" WILL NOT BE CREATED.

3.2. RESERVED WORDS

THE FOLLOWING IS A COMPLETE LIST OF RESERVED WORDS USED IN THE NETWORK DEFINITION LANGUAGE. THESE WORDS HAVE SPECIAL MEANING TO THE COMPILER AND CAN NOT BE USED AS IDENTIFIERS OR IN ANY OTHER MANNER EXCEPT FOR THEIR DEFINED MEANING.

ABORT
ACU
ADAPTER
ADAPTOR
ADDERR
ADDRESS
ALTERNATE
AUTOANSWER
ASC63
ASC67
ASC68
BACKSPACE
BAUDOT
BCC
BCCERR
BCD
BCL
BEGIN
BKSP
BLOCK
BLOCKED
BUFFER
BUFOVFL
CARRIAGE
CHAR
CHARACTER
CLEAR
CODE
COMMENT
CONNECTION
CONSTANT
CONTENTION
CR
DCP
DEFAULT
DELAY
DIALIN
DIALOUT
DISCONNECT
DUPLEX
EBCDIC
ELSE
ENABLEINPUT
END
EQ
ERROR
EVEN
EXCHANGE
FALSE
FETCH
FILE
FINISH
FOR
FORMAT
FORMATERR
FREQUENCY
GE
GO
GT
HOME

HORIZONTAL
HYBRID
ICTDELAY
IF
ILLEGALCHR
INHIBITSYNC
INITIALIZE
INITIATE
INPUT
LE
LD
LF
LS
LINE
LINEDELETE
LINEFEED
LOGICALACK
LOGIN
MAXINPUT
MAXSTATIONS
MCS
MEMORY
MICRO
MILLI
MIN
MODE
MODEM
MYUSE
NE
NOISEDELAY
NORMAL
NULL
ODD

ON
OUTPUT
PAGE
PARITY
PASSIVE
PHONE
POLL
RECEIVE
REMOTE
REQUEST
RETRY
SCREEN
SEC

SECURITY
SPECIAL
SPO
STATION
STOPBIT
STORE
SUM
TALLY
TASK
TERMINAL
TERMINATE
TEXT
THEN

TIMEOUT
TO
TOG
TRAN
TRANERR
TRANSMISSION
TRANSMIT
TRUE
TURNAROUND
TYPE
USER
VERTICAL
WIDTH

WRAPAROUND
WRU

3.3. DIAGNOSTIC MESSAGES

THE MESSAGES BELOW WILL BE GENERATED BY THE NETWORK DEFINITION LANGUAGE PROCESSOR WHEN IT ENCOUNTERS INCORRECT STATEMENTS, INCOMPATIBILITIES OR INCOMPLETENESS IN DESCRIPTIONS. (A PERCENT SIGN IN THE MESSAGES BELOW MERELY INDICATES CONTINUATION ON THE NEXT CARD FOR THE PROGRAM WHICH PACKS THESE ERROR MESSAGES FOR THE LANGUAGE PROCESSOR AND DOES NOT APPEAR IN THE LISTING OF THE MESSAGE.) ONLY THE TEXT OF MESSAGES IS LISTED .

NUMBER	TEXT OF ERROR MESSAGE.
300	IDENTIFIER TOO LONG
301	STRING TOO LONG
302	ILLEGAL CHARACTER
303	INTEGER TOO LARGE (EXCEEDS 11 DIGITS)
310	CONSTRUCT NOT IMPLEMENTED
400	ILLEGAL DESCRIPTION DIVISION
401	MISPLACED DEFAULT PART
402	MISSING COLON
403	EXTRA DESCRIPTION PART
404	MISPLACED DIVISION
405	ILLEGAL CONSTRUCT
406	ILLEGAL STRING CODE
407	ILLEGAL STRING CHARACTER
408	CONSTANT GREATER THAN 64 CHARACTERS
450	REQUEST STATEMENT EXPECTED
451	THEN EXPECTED
452	DUPLICATE LABEL
453	ERROR ACTION EXPECTED
454	TERMINATE TYPE EXPECTED
455	INITIATE TYPE EXPECTED
456	INITIALIZE TYPE EXPECTED
457	CHARACTER OR UNITARY STRING EXPECTED
458	CHARACTER OR STRING EXPECTED
459	RECEIVE ACTION ITEM EXPECTED

460 ITEM EXPECTED
461 VALUE EXPECTED
462 TALLY NUMBER EXPECTED
463 TOGGLE NUMBER EXPECTED
464 RECEIVE OR TRANSMIT EXPECTED
465 APPLICATION NUMBER EXPECTED
466 ALL ERROR SWITCH ASSIGNMENTS MUST PRECEDE ALL OTHER REQUEST STATEMENTS
467 ERROR SWITCH HAS NOT BEEN SPECIFIED
468 DUPLICATE ERROR SWITCH DESIGNATOR
469 TRANSMIT ACTION ITEM EXPECTED
500 ILLEGAL CONSTANT IDENTIFIER
501 MISSING PERIOD
502 MISSING EQUAL SIGN
503 CONSTANT NOT A STRING
504 TRUE OR FALSE EXPECTED
505 INTEGER EXPECTED
506 STRING EXPECTED
507 UNITARY STRING EXPECTED
508 UNITARY STRING OR INTEGER EXPECTED
509 STRING OR INTEGER EXPECTED
510 LEFT PAREN EXPECTED
511 RIGHT PAREN EXPECTED
512 COLON EXPECTED
513 TIME UNIT EXPECTED
514 LEFT BRACKET EXPECTED
515 RIGHT BRACKET EXPECTED
516 RELATIONAL OPERATOR EXPECTED
517 COMMA EXPECTED
518 QUOTE EXPECTED
519 CHARACTER NOT ALIGNED
520 MISSING CONDITION
521 LIST EXPECTED
522 MISSING REQUEST IDENTIFIER
523 RECEIVE REQUEST NOT SPECIFIED
524 TRANSMIT REQUEST NOT SPECIFIED

525 UNDEFINED REQUEST IDENTIFIER
526 DUPLICATE REQUEST CONDITION
527 DCP ATTRIBUTE STATEMENT EXPECTED
528 MCS IDENTIFIER EXPECTED
529 TASK IDENTIFIER EXPECTED
530 DYNAMIC EXPECTED
531 LINE TYPE STATEMENT MAY NOT BE PART OF LINE DEFAULT DEFINITIONS
532 LINE STATION STATEMENT MAY NOT BE PART OF LINE DEFAULT DEFINITION
533 NO. LINES / PAGE TOO LARGE
534 STANDARD CODE EXPECTED
535 ILLEGAL ADAPTER CLASS
536 ILLEGAL ADAPTER TYPE
537 LINE WIDTH TOO LARGE
538 ADDRESS LIST EXCEEDS 50 MEMBERS
539 INCONSISTENT ADDRESS LENGTH
540 LABEL EXPECTED
541 ERROR TOGGLE LIST EXPECTED
542 ERROR TOGGLE EXPECTED
543 ITEM EXPECTED
544 -OUTPUT- EXPECTED
545 -INPUT- EXPECTED
546 -INPUT- OR -OUTPUT- EXPECTED
547 MAX. ADDRESS < MIN. ADDRESS
548 TERMINAL IDENTIFIER EXPECTED
549 ADDRESS EXCEEDS 3 CHARACTERS
550 BUFFER SIZE TOO LARGE
551 -ODD- OR -EVEN- EXPECTED
552 -HORIZONTAL- OR -VERTICAL- EXPECTED
553 TERMINAL MODE EXPECTED
554 CONNECTION TYPE EXPECTED
555 CHARACTER SIZE TOO LARGE
556 ILLEGAL CORE SIZE
557 TERMINAL IDENTIFIER EXPECTED
558 STATION IDENTIFIER EXPECTED

559 LINE IDENTIFIER EXPECTED
560 DCP IDENTIFIER EXPECTED
561 STATION DESIGNATOR EXPECTED
562 A DUPLEX(RECEIVE) LINE MUST POINT TO A DUPLEX(TRANSMIT) LINE
563 THE DIALOUT TYPE MUST BE ASSOCIATED WITH THE DUPLEX(RECEIVE) LINE OF A DUPLEX PAIR OF LINES
564 DUPLICATE TYPES
565 THE DIALIN TYPE MUST BE ASSOCIATED WITH THE DUPLEX(RECEIVE) LINE OF A DUPLEX PAIR OF LINES
566 RETRY VALUE MUST NOT EXCEED 255
567 DUPLEX(RECEIVE) AND DUPLEX(TRANSMIT) ARE INCOMPATIBLE TYPES
568 LINE DOES NOT POINT BACK
569 A DIALOUT LINE MUST POINT TO A LINE WITH AN ACU ADAPTER CLASS
570 A DUPLEX(TRANSMIT) LINE MUST POINT TO A DUPLEX(RECEIVE) LINE
571 LINE TYPE EXPECTED
572 THE PASSIVE TYPE AND ALL OTHER TYPES ARE INCOMPATIBLE
573 THE REMOTE TYPE IS ONLY COMPATIBLE WITH THE TYPE DUPLEX
574 TRANSMISSION NUMBER LENGTH EXPECTED
575 DUPLICATE STATION IN LIST
576 STATION CANNOT BE ALTERNATE OF ITSELF
577 TYPE DUPLEX IS COMPATIBLE ONLY WITH TYPE REMOTE
578 STATION ALREADY ATTACHED TO A LINE
579 DUPLICATE ADDRESS
580 ILLEGAL PHONE NUMBER
581 ILLEGAL DCP ADDRESS
582 ILLEGAL ADAPTER ADDRESS
583 ILLEGAL CLUSTER ADDRESS

584 DUPLICATE STATION ADDRESSES ON LINE

585 MODE OF STATIONS TERMINAL INCONSISTENT WITH OTHER STATIONS ON LINE

586 DUPLEX ATTRIBUTE OF STATIONS TERMINAL

587 ADDRESS LENGTH OF STATIONS TERMINAL INCONSISTENT WITH OTHER STATIONS ON LINE

588 REQUEST FOR STATIONS TERMINAL INCONSISTENT WITH OTHER STATIONS ON LINE

589 STATIONS ADAPTER TYPE NOT CONSISTENT WITH OTHER STATIONS ON LINE

590 CODE OF STATIONS TERMINAL INCONSISTENT WITH OTHER STATIONS ON LINE

591 A STATION WITH MULTIPLE ADDRESSES MUST BE THE ONLY STATION ON A LINE

592 A LINES TYPE STATEMENT MAY NOT REFER TO ITS SELF

593 TERMINAL DEFAULT IDENTIFIER EXPECTED

594 STATION DEFAULT IDENTIFIER EXPECTED

595 LINE DEFAULT IDENTIFIER EXPECTED

596 FILE DEFAULT IDENTIFIER EXPECTED

597 ADDRESS SIZE EXCEEDS 3 CHARACTERS

598 DIFFERENT EXPECTED

599 EXISTENCE OF MODEM ON STATION INCONSISTENT WITH OTHER STATIONS ON LINE

600

601 DUPLICATE STATEMENT

602 MAXINPUT NOT SPECIFIED FOR UNBUFFERED DEVICE

603 DUPLEX ATTRIBUTE NOT SPECIFIED

604 REQUESTS FOR TERMINAL USE BCC BUT TERMINAL DOES NOT HAVE HORIZONTAL PARITY

605 REQUESTS FOR TERMINAL SPECIFY END CHARACTER ACTION BUT TERMINAL DOES NOT SPECIFY END CHARACTER

606 REQUESTS FOR TERMINAL SPECIFY BACKSPACE CHARACTER ACTION BUT TERMINAL DOES NOT

	SPECIFY BACKSPACE CHARACTER
607	REQUESTS FOR TERMINAL SPECIFY LINEDELETE CHARACTER ACTION BUT TERMINAL DOES NOT SPECIFY LINEDELETE CHARACTER
608	ADDRESS NOT SIGNIFICANT FOR STATIONS HAVING CONTENTION MODE TERMINALS
609	REQUESTS FOR TERMINAL SPECIFY WRU CHARACTER ACTION BUT TERMINAL DOES NOT SPECIFY WRU CHARACTER
610	ALTERNATE STATEMENT CANNOT BE PART OF A STATION DEFAULT DEFINITION
611	DUPLICATE APPLICATION REQUEST SET
612	APPLICATION REQUEST SET ONE NOT SPECIFIED
613	APPLICATION REQUEST SETS NOT CONSISTENTLY SPECIFIED FOR RECEIVE AND/OR TRANSMIT
614	DUPLICATE ERROR SWITCH NUMBER
615	DUPLICATE RECEIVE ERROR TOGGLE
616	
617	TIMEOUT VALUE OR NULL EXPECTED
618	ADDRESS STATEMENTS MAY NOT BE INCLUDED IN STATION DEFAULT DEFINITIONS
619	A RESERVED WORD MAY NOT BE USED AS AN IDENTIFIER
620	ALREADY USED IDENTIFIER
621	REQUEST STATEMENTS MAY NOT BE INCLUDED IN TERMINAL DEFAULT DEFINITIONS
622	ILLEGAL TERMINAL ATTRIBUTE
623	CODE NOT SPECIFIED
624	MODE NOT SPECIFIED
625	CHARACTER SIZE NOT SPECIFIED
626	IN - OUT CAPABILITY NOT SPECIFIED
627	REQUESTS NOT SPECIFIED
628	BLOCKING NOT SPECIFIED
629	SCREEN NOT SPECIFIED

630 LINE WIDTH EXCEEDS BUFFER SIZE
631 PAGE SIZE OF SCREEN DEVICE NOT SPECIFIED
632 HOME CHARACTER OF SCREEN DEVICE NOT SPECIFIED *
633 CARRIAGE RETURN CHARACTER NOT SPECIFIED
634 MAXINPUT ILLEGAL FOR BUFFERED DEVICE
635 LINEFEED CHARACTER NOT SPECIFIED
636 TIMEOUT NOT SPECIFIED
637 ILLEGAL STATION ATTRIBUTE
638 TERMINAL NOT SPECIFIED
639 MCS NOT SPECIFIED
640 SECURITY NOT SPECIFIED
641 WRU NOT SPECIFIED
642 MYUSE ATTRIBUTE OF STATION NOT CONSISTENT WITH I/O CAPABILITY OF ITS TERMINAL *
643 STATIONS BLOCKING ATTRIBUTE NOT COMPATIBLE WITH ITS TERMINAL *
644 ADDRESS NOT SPECIFIED (STATION HAS A NON-CONTENTION MODE TERMINAL) *
645 SPECIAL AND WRU CHARACTERS EQUAL
646 REQUESTS FOR STATIONS TERMINAL SPECIFY CONTROL CHARACTER ACTION BUT STATION DOES NOT SPECIFY CONTROL CHARACTER * *
647 SPECIAL AND END CHARACTERS EQUAL
648 SPECIAL AND BACKSPACE CHARACTERS EQUAL
649 SPECIAL AND LINEDELETE CHARACTERS EQUAL
650 WRU AND END CHARACTERS EQUAL
651 WRU AND BACKSPACE CHARACTERS EQUAL
652 WRU AND LINEDELETE CHARACTERS EQUAL
653 END AND BACKSPACE CHARACTERS EQUAL
654 END AND LINEDELETE CHARACTERS EQUAL
655 BACKSPACE AND LINEDELETE CHARACTERS EQUAL
656 STATION PAGE SIZE EXCEEDS TERMINAL PAGE SIZE *
657 LOGIN NOT SPECIFIED
658 STATION ADAPTER TYPE NOT COMPATIBLE *

WITH ITS TERMINALS ADAPTER TYPES

659 TERMINAL AND STATION ADAPTER TYPES NOT COMPATIBLE WITH MODEM

660 MODEM IDENTIFIER EXPECTED

661 TASK MUST BE SPECIFIED WHEN LOGIN IS SET FALSE

662 ILLEGAL LINE ATTRIBUTE

663 ADDRESS NOT SPECIFIED

664 ADAPTER NOT SPECIFIED

665 INPUT/OUTPUT NOT SPECIFIED

666 CONNECTION TYPE NOT SPECIFIED

667 LINE HAS NO STATIONS ATTACHED

668 DIALOUT, DUPLEX(TRANSMIT), AND PASSIVE LINES MAY NOT HAVE ATTACHED STATIONS

669 DIALIN LINES MUST HAVE ONLY ONE ATTACHED STATION

670 LINE TYPE STATEMENTS MAY NOT REFER TO LINES ON OTHER DCPS

671 ONLY DIALIN TYPE LINES MAY HAVE A MULTIPLE ADDRESS STATION ATTACHED

672 ADAPTER CLASS OF LINE AND ADAPTER TYPE OF STATIONS INCOMPATIBLE

673 LENGTH OF STATIONS ADDRESS INCONSISTENT WITH LENGTH RESTRICTIONS SPECIFIED IN STATIONS TERMINAL DEFINITION

674 PARITY NOT SPECIFIED

675 TERMINAL CANNOT HAVE AN ACU ADAPTER TYPE SPECIFIED IN ITS ADAPTER STATEMENT

676 THE CHARACTER SIZE OF THE TERMINALS ADAPTER TYPES ARE TOO SMALL TO CONTAIN THE CHARACTER SET SPECIFIED IN THE TERMINALS CODE STATEMENT

677 THE TERMINAL HAS VERTICAL PARITY SPECIFIED AND COMPATIBLE ADAPTER TYPES NOT HAVING A PARITY BIT ALSO SPECIFIED

678 INHIBITSYNC ATTRIBUTE NOT SPECIFIED

679 ENABLEINPUT ATTRIBUTE NOT SPECIFIED

680 LINES HAVING AN ACU ADAPTER CLASS MAY BE OF TYPE PASSIVE ONLY

681 LINES HAVE AN ACU ADAPTER CLASS MAY NOT HAVE ATTACHED STATIONS

682 LINES ADAPTER CLASS NECESSITATES A MODEM BE SPECIFIED FOR THE LINE

683 STATIONS TERMINAL IS A POLLED DEVICE AND FREQUENCY IS NOT SPECIFIED

684 LOGICALACK NOT SPECIFIED

685 LINES ADAPTER CLASS REQUIRES THAT ITS STATIONS HAVE MODEMS

686 EXISTENCE OF MODEMS ON LINE AND ITS STATIONS INCONSISTENT

687 STATIONS ADAPTER TYPE REQUIRES MODEM BE SPECIFIED FOR STATION

688 NON-ZERO TIMEOUT VALUES EXPECTED

689 ACTION ITEM INAPPROPRIATE FOR ITEM

690 ILLEGAL MODEM ATTRIBUTE

691 ADAPTER STATEMENT NOT SPECIFIED

692 TURNAROUND NOT SPECIFIED

693 NOISEDELAY NOT SPECIFIED

694 INTER-CHARACTER TRANSMIT DELAY NOT SPECIFIED

695 PAGE NOT SPECIFIED

696 NO LINES SPECIFIED FOR DCP

697 MEMORY NOT SPECIFIED

701 ERROR SWITCH DESIGNATORS APPROPRIATE TO RECEIVE ACTION LISTS ONLY

702 ACTION LABEL EXPECTED

703 REQUEST FOR TERMINAL REQUIRES A RECEIVE ADDRESS

704 REQUEST FOR TERMINAL REQUIRES A TRANSMIT ADDRESS

705 REQUEST FOR TERMINAL REQUIRES A TRANS- MISSION NUMBER

706 REQUEST FOR TERMINAL REQUIRES THAT THE TERMINAL BE CAPABLE OF BLOCKED INPUT AND/OR OUTPUT

707 ADAPTER CLASS EXPECTED

708 MODEM OR DIRECT EXPECTED

709 ADAPTER CLASS REQUIRES MODEM

710 INCREMENT ITEM EXPECTED

711 RETURN EXPECTED

712 FILE ATTRIBUTE STATEMENT EXPECTED

713 FAMILY NOT SPECIFIED

714 FAMILY STATEMENT NOT FILE DEFAULT ATTRIBUTE STATEMENT

715 FILE OR STATION IDENTIFIER EXPECTED

716 TIMELIMIT NOT SPECIFIED

717 TRANSMITDELAY NOT SPECIFIED

718 NO STATION DEFINITIONS GIVEN

719 NO TERMINAL DEFINITIONS GIVEN

720 NO LINE DEFINITIONS GIVEN

721 NO DCP DEFINITIONS GIVEN

722 INCOMPLETE DESCRIPTION

728 LINE AND MODEM ADAPTER TYPES ARE INCONSISTENT

729 STATION AND MODEM ADAPTER TYPES ARE INCONSISTENT

730 ANSWER STATEMENT REQUIRED FOR LINES OF TYPE DIALIN

731 ANSWER STATEMENT NOT APPROPRIATE FOR NON-DIALIN LINES

732 END ENDOFNUMBER STATEMENT REQUIRED FOR LINES HAVING AN ADAPTER CLASS OF 8

733 ENDOFNUMBER STATEMENTS ARE APPROPRIATE ONLY WITH LINES HAVING AN ADAPTER CLASS OF 8

734 ANSWER STATEMENT IS NOT A LINE DEFAULT ATTRIBUTE STATEMENT

735 ENDOFNUMBER STATEMENT IS NOT A LINE DEFAULT ATTRIBUTE STATEMENT

737 HORIZONTAL PARITY VARIANT EXPECTED

738 TALLY OR TOGGLE EXPECTED

739 DCP CANNOT REFERENCE SELF IN EXCHANGE STATEMENT

740 NUMBER OF STATIONS ON LINE CANNOT EXCEED 25

741	MAXIMUM NUMBER OF STATIONS IS LESS THAN ACTUAL NUMBER OF STATIONS
742	A DCP MAY BE EXCHANGED WITH ONLY ONE OTHER DCP
743	TWO DCPS HAVE INCONSISTENT EXCHANGE STATEMENTS
744	DCP AND DCP WITH WHICH EXCHANGED HAVE NO LINES ON THEM
745	DCP AND DCP WITH WHICH EXCHANGED HAVE THE SAME CLUSTERS DEFINED ON THEM
746	THE SET OF TERMINALS SPECIFIED IN TERMINAL STATEMENT DOES NOT INCLUDE ALL THE TERMINALS ON STATIONS ON DCP.
747	TERMINAL WIDTH EQUALS ZERO
748	DCP MENTIONED IN EXCHANGE STATEMENT NOT DEFINED

APPENDIX

TABLE OF DATACOM ADAPTER CLASSES

CLASS	STYLE NO.	MAX BAUD RATE	SERIAL	PARA-LLEL	SYNC	ASYNC	CON-NECTION	ACU	TOUCH TONE	AUD IO RES-PONSE
1	B6650-1	600	X			X	D-M			
2	B6650-2	1800	X			X	D-M			
3	B6650-3	2400	X		X	X	D-M			
4	B6650-4	4800	X		X	X	D-M			
5	B6650-5	9600	X		X	X	D-M			
6	B6650-6	40		X			M		X	
7	B6650-7	16	X				M			X
8	B6650-8	40		X			M	X		

D = DIRECT CONNECT
M = MODEM

TABLE OF DATACOM ADAPTER TYPES

TYPE	SPEED	SIZE		START	STOP	PARITY	ASYNC	ADAPTER CLASS
	(BPS)	CHAR	UNIT	BIT	BIT		SYNC	12345678
1	45.5	5.0	7.5	1	1.5	-	ASYNC	XXXXX
2	56.9	5.0	7.5	1	1.5	-	ASYNC	XXXXX
3	75.0	5.0	7.5	1	1.5	-	ASYNC	XXXXX
4	110.0	7.0	11.0	1	2	1	ASYNC	XXXXX
5	134.5	6.0	9.0	1	1	1	ASYNC	XXXXX
6	150.0	7.0	10.0	1	1	1	ASYNC	XXXXX
7	3000.0	7.0	10.0	1	1	1	ASTNC	XXXXX
8	600.0	7.0	10.0	1	1	1	ASYNC	XXXXX
9	1200.0	7.0	10.0	1	1	1	ASYNC	XXXX
10	1200.0	4.0	6.0	1	1	-	ASYNC	XXXX
11	1800.0	7.0	10.0	1	1	1	ASYNC	XXXX
12	2400.0	7.0	10.0	1	1	1	ASYNC	XXX
13	3600.0	7.0	10.0	1	1	1	ASYNC	XX
14	3600.0	7.0	10.0	1	1	1	ASYNC	XX
15	9600.0	7.0	10.0	1	1	1	ASYNC	X
16	2000.0	6.0	7.0	-	-	1	SYNC	XXX
17	2000.0	7.0	8.0	-	-	1	SYNC	XXX
18	2000.0	8.0	9.0	-	-	1	SYNC	XXX
19	2400.0	6.0	7.0	-	-	1	SYNC	XXX
20	2400.0	7.0	8.0	-	-	1	SYNC	XXX
21	2400.0	8.0	9.0	-	-	1	SYNC	XXX
22	4800.0	6.0	7.0	-	-	1	SYNC	XX
23	4800.0	7.0	8.0	-	-	1	SYNC	XX
24	4800.0	8.0	9.0	-	-	1	SYNC	XX
25	9600.0	6.0	7.0	-	-	1	SYNC	X
26	9600.0	7.0	8.0	-	-	1	SYNC	X
27	9600.0	8.0	9.0	-	-	1	SYNC	X
28	40.0	4.0	4.0	-	-	-	SYNC	X
29	16.0	7.0	8.0	-	-	1	SYNC	X
30	40.0	4.0	4.0	-	-	-	SYNC	X

TABLE OF ALLOWABLE COMBINATIONS OF LINE TYPES

	DUPLEX	DUPLEX TRANSMIT	DUPLEX RECEIVE	REMOTE	PASSIVE	DIALOUT	DIALIN	TOUCHTONE
TOUCHTONE			X			X	X	
DIALIN			X			X		
DIALOUT			X					
PASSIVE								
REMOTE	X							
DUPLEX RECEIVE								
DUPLEX TRANSMIT								
DUPLEX								

TABLE OF ALLOWABLE COMBINATIONS
OF LINE TYPE
X = ALLOWED COMBINATION

INDEX OF METALINGUISTIC VARIABLES

ITEM	PAGE
<ACTION ITEM>	2-32;2-32,
<ACTION LABEL>	2-30;2-30,2-32,
<ADAPTER ADDRESS>	2-78;2-78,
<ADAPTER TYPE>	2-44;2-44,2-74,
<ADAPTER TYPE LIST>	2-44;2-44,2-60,
<ADD OPERATOR>	2-4;2-4,2-24,
<ADDRESS>	2-66;2-66,2-67,
<ADDRESS LIST>	2-66;2-66,
<ADDRESS LIST PART>	2-66;2-66,
<ADDRESS PAIR>	2-67;2-66,2-67,
<ADDRESS PAIR LIST>	2-66;2-66,
<ADDRESS PART>	2-66;2-66,
<ADDRESS QUALIFIER>	2-30;2-30,
<ADDRESS SIZE>	2-51;2-51,
<ALTERNATE LIST>	2-76;2-76,
<ALTERNATE STATEMENT>	2-76;2-66,
<ANSWER STATEMENT>	2-87;2-77,
<APPLICATION NUMBER>	2-60;2-60,
<APPLICATION QUALIFIER>	2-60;2-60,
<ASSIGNABLE TOGGLE>	2-24;2-24,

ITEM	PAGE
<ASSIGNMENT OPERATOR>	.2-4;2-4,
<ASSIGNMENT STATEMENT>	.2-24;2-24,
<ATTRIBUTE LIST>	.2-93;2-93,
<BACKSPACE STATEMENT>	2-94,
<BASIC COMPONENT>	.2-5;
<BASIC SYMBOL>	.2-4;
<BCC>	.2-23;2-20,
<BLOCKING STATEMENT>	.2-59;2-49,
<BRACKET>	.2-4;2-4,
<BREAK DELAY TIME>	.2-42;2-42,
<BREAK STATEMENT>	.2-42;2-24,
<BREAK TIME>	.2-42;2-42,
<BUFFER SIZE STATEMENT>	.2-58;2-48,
<BYTE VARIABLE>	2-20;2-20,2-24,
<CHARACTER>	.2-3;
<CHARACTER REGISTER>	.2-21;2-20,2-28,2-30,
<CLUSTER ADDRESS>	.2-78;2-78,
<CODE STATEMENT>	.2-56;2-48,
<CODE TYPE>	.2-56;2-56,
<COMPOUND REQUEST STATEMENT>	.2-14;2-14,

ITEM	PAGE
<COMPOUND TAIL>	.2-14;2-14,
<CONDITIONAL REQUEST STATEMENT>	.2-15;2-14,
<CONFLICTING ATTRIBUTE>	.2-94;2-92,
<CONNECTION TYPE>	.2-85;2-85,
<CONSTANT DEFINITION>	.2-10;2-10,
<CONSTANT DEFINITION LIST>	.2-10;2-10,
<CONSTANT IDENTIFIER>	.2-10;2-10,
<CONSTANT LIST>	.2-10;2-10,
<CONSTANT SECTION>	2-10;2-9,
<CONTROL CHARACTER STATEMENT>	.2-69;2-65,
<DCP ATTRIBUTE LIST>	.2-89;2-89,
DCP ATTRIBUTE STATEMENT>	.2-89;2-89,
<DCP DEFINITION>	.2-89;2-89,
<DCP IDENTIFIER>	2-89;2-89,2-90,
<DCP LIST>	.2-89;2-89,
<DCP NUMBER>	.2-78;2-78,
<DCP SECTION>	2-89;2-9,
<DCP TERMINAL STATEMENT>	.2-90;2-89,
<DELAY PART>	2-25;2-25,2-26,
<DELAY STATEMENT>	.2-42;2-24,

ITEM	PAGE
<DELAY VALUE>	.2-25;2-25,
<DELIMITER>	.2-4;2-4,
<DIFFERENT PART>	.2-51;2-51,
<DIGIT>	.2-3;2-3,2-4,2-5,2-6,
<DYNAMIC PART>	.2-46;2-46,2-53,2-62,
<EBCDIC CHARACTER>	.2-7;2-7,
<EBCDIC CHARACTER CONCATENATION>	.2-7;2-7,
<EBCDIC STRING>	.2-7;2-7,
<ELSE PART>	.2-15;2-15,
<ENABLEINPUT STATEMENT>	2-72;2-65,2-94,
<ENDOFBUFFER ACTION>	2-27;2-27,2-28,
<ENDOFNUMBER STATEMENT>	.2-88;2-77,
<ERROR ACTION>	.2-41;2-41,
<ERROR ACTION LIST>	.2-41;2-41,
<ERROR ACTION STATEMENT>	.2-41;2-24,
<ERROR FLAG>	2-16;2-15,2-24,
<ERROR SWITCH DESIGNATOR>	.2-32;2-32,
<ERRORNUMBER>	.2-41;2-41,
<EXCHANGE STATEMENT>	.2-90;2-89,

ITEM	PAGE
<FAMILY DESIGNATOR>	.2-93;2-93,
<FAMILY LIST>	.2-93;2-93,
<FAMILY STATEMENT>	.2-93;2-92,
<FETCH ACTION PART>	.2-27;2-27,
<FETCH STATEMENT>	.2-27;2-24,
<FILE ATTRIBUTE LIST>	.2-92;2-92,
<FILE ATTRIBUTE STATEMENT>	.2-92;2-92,
<FILE DEFAULT DEFINITION>	.2-92;2-92,
<FILE DEFAULT IDENTIFIER>	2-92;2-92,2-93,
<FILE DEFAULT LIST>	.2-92;2-92,
<FILE DEFAULT STATEMENT>	.2-93;2-92,
FILE DEFINITION>	.2-92;2-92,
<FILE IDENTIFIER>	2-92;2-92,2-93,
<FILE LIST>	.2-92;2-92,
<FILE SECTION>	2-92;2-9,
<FINISH STATEMENT>	.2-26;2-24,
<FORMAT CHARACTER>	.2-55;2-55,
<FORMAT CHARACTER STATEMENT>	.2-55;2-48,
<FREQUENCY STATEMENT>	.2-74;2-65,
<GENERALIZED IDENTIFIER>	2-5;2-5,2-6,2-65,2-72,2-92,

ITEM	PAGE
<GO TO STATEMENT>	.2-25;2-24,
<HEXADECIMAL CHARACTER>	.2-7;2-7,
<HEXADECIMAL CHARACTER CONCATENATION>	.2-7;2-7,
<HEXADECIMAL STRING>	.2-7;2-7,
<HORIZONTAL PARITY TYPE>	.2-57;2-57,
<HORIZONTAL PARITY VARIANT>	.2-57;2-57,
<IDENTIFIER>	.2-5;2-5,2-6,2-10,2-12,2-44,2-48, 2-77,
<IDENTIFIER COMPONENT>	.2-6;2-5,2-6,
<ILLEGAL CHARACTER LIST>	.2-54;2-54,
<ILLEGAL CHARACTER STATEMENT>	.2-54;2-48,
<INCREMENT STATEMENT>	.2-43;2-24,
<INCREMENTED ITEM>	.2-43;2-43,
<INHIBITSYNC STATEMENT>	.2-64;2-49,
<INITIALIZE STATEMENT>	.2-26;2-24,
<INITIALIZED ITEM>	.2-26;2-26,
<INITIATE ACTION>	.2-25;2-25,
<INITIATE STATEMENT>	.2-25;2-24,
<INTEGER>	2-6;2-5,2-6,2-8,2-13,2-20,2-32, 2-41,2-52,2-53,2-58,2-67,2-68, 2-72,2-76,2-85,2-88,2-89,
<INTER-CHARACTER DELAY STATEMENT>	.2-55;2-48,

ITEM	PAGE
<IO LIST>	.2-70;2-70,
<IRT>	.2-56;2-56,
<ITEM>	2-30;2-30,2-32,
<ITEM ERROR FLAG>	2-17;2-16,2-32,
<LABEL>	.2-13;2-13,2-25,2-27,2-30,2-41, 2-43,
<LABEL PART>	.2-41;2-41,
<LABELED REQUEST STATEMENT>	.2-13;2-13,
<LETTER>	.2-3;2-3,2-4,2-5,
<LINE ADAPTER CLASS>	.2-85;2-85,
<LINE ADAPTER CLASS STATEMENT>	.2-85;2-77,
<LINE ADDRESS STATEMENT>	.2-78;2-77,
<LINE ATTRIBUTE LIST>	.2-77;2-77,
<LINE ATTRIBUTE STATEMENT>	.2-77;2-77,
<LINE DEFAULT ATTRIBUTE LIST>	.2-77;2-77,
<LINE DEFAULT ATTRIBUTE STATEMENT>	.2-77;2-77,
<LINE DEFAULT DEFINITION>	.2-77;2-77,
<LINE DEFAULT IDENTIFIER>	2-77;2-77,2-78,
<LINE DEFAULT STATEMENT>	.2-78;2-77,
<LINE DEFINITION>	.2-77;2-77,
<LINE IDENTIFIER>	2-77;2-77,2-80,

ITEM	PAGE
<LINE LIST>	.2-77;2-77,
<LINE MODEM STATEMENT>	.2-86;2-77,
<LINE PHONE STATEMENT>	.2-85;2-77,
<LINE SECTION>	2-77;2-9,
<LINE STATION STATEMENT>	.2-87;2-77,
<LINE TYPE STATEMENT>	.2-80;2-77,
<LIST OF STATIONS>	.2-87;2-87,
<LOGICAL ASSIGNMENT STATEMENT>	.2-24;2-24,
<LOGICAL EXPRESSION>	.2-15;2-15,
<LOGICAL VALUE>	.2-8;2-4,2-5,2-15,2-24,2-59,2-60, 2-63,2-64,2-69,2-70,2-71,2-72, 2-87,2-88,
<LOGICALACK STATEMENT>	2-71;2-65,2-94,
<LOGICALACK TYPE>	.2-36;2-36,
<MAXINPUT STATEMENT>	.2-58;2-49,
<MAXSTATIONS STATEMENT>	.2-88;2-77,
<MCS IDENTIFIER>	.2-72;2-72,
<MCS STATEMENT>	2-72;2-65,2-94,
<MEMORY SIZE>	.2-89;2-89,
<MEMORY STATEMENT>	.2-89;2-89,
<MODE STATEMENT>	.2-57;2-48,
<MODE TYPE>	.2-57;2-57,

ITEM	PAGE
<MODEM ADAPTER STATEMENT>	.2-44;2-44,
<MODEM ATTRIBUTE LIST>	.2-44;2-44,
<MODEM DEFINITION>	.2-44;2-44,
<MODEM IDENTIFIER>	.2-44;2-44,2-75,2-86,
<MODEM LIST>	.2-44;2-44,
<MODEM SECTION>	2-44;2-9,
<MODEM STATEMENT>	.2-44;2-44,
<MYUSE STATEMENT>	.2-70;2-65,
<NETWORK DESCRIPTION>	.2-9;
<NOISE DELAY STATEMENT>	.2-46;2-44,
<OPERATOR>	.2-4;2-4,
<PARITY LIST>	.2-57;2-57,
<PARITY STATEMENT>	.2-57;2-48,
<PARITY TYPE>	.2-57;2-57,
<PARITY VALUE>	.2-57;2-57,
<RECEIVE ACTION LIST>	.2-32;2-32,
<RECEIVE ACTION PART>	.2-32;2-32,
<RECEIVE ADDRESS>	.2-51;2-51,
<RECEIVE ERROR FLAG>	.2-16;2-16,2-32,2-41,
<RECEIVE STATEMENT>	.2-32;2-24,

ITEM	PAGE
<RELATIONAL EXPRESSION>	.2-20;2-15,
<RELATIONAL OPERATOR>	2-4;2-4,2-20,
<REQUEST ACTION>	.2-61;2-61,
<REQUEST DEFINITION>	.2-12;2-12,
<REQUEST IDENTIFICATION>	.2-60;2-60,
<REQUEST IDENTIFICATION LIST>	.2-60;2-60,
<REQUEST IDENTIFIER>	2-12;2-12,2-60,
<REQUEST LIST>	.2-12;2-12,
<REQUEST SECTION>	2-12;2-9,
<REQUEST STATEMENT>	.2-13;2-12,2-14,2-15,
<REQUEST STATEMENT LIST>	.2-12;2-12,
<RETRY>	.2-22;2-20,
<RETRY STATEMENT>	.2-76;2-65,
<SCREEN STATEMENT>	.2-59;2-49,
<SECURITY STATEMENT>	2-94,
<SEPARATOR>	.2-4;2-4,
<SEQUENCE ACTION>	.2-43;2-43,
<SEQUENCE ACTION PART>	.2-43;2-43,
<SEQUENTIAL OPERATOR>	.2-4;2-4,
<SHIFT DIRECTION>	.2-43;2-43,

ITEM	PAGE
<SHIFT STATEMENT>	.2-43;2-24,
<SIMPLE STRING>	.2-7;2-7,
<SINGLE SPACE>	.2-3;2-3,
<SLASH>	.2-3;2-3,2-6,
<SPACE>	.2-3;2-3,2-4,
<SPECIAL CHARACTER>	.2-3;2-3,
<SPECIAL CHARACTER NAME>	.2-62;2-62,
<SPECIAL CHARACTER STATEMENT>	.2-62;2-49,
<SPO STATEMENT>	.2-70;2-65,
<STATION ADAPTER STATEMENT>	.2-74;2-65,
<STATION ADDRESS STATEMENT>	.2-66;2-65,
<STATION ATTRIBUTE LIST>	.2-65;2-65,
<STATION ATTRIBUTE PART>	.2-93;2-93,
<STATION ATTRIBUTE STATEMENT>	.2-65;2-65,
<STATION DEFAULT ATTRIBUTE LIST>	.2-65;2-65,
<STATION DEFAULT ATTRIBUTE STATEMENT>	2-65;2-65,2-66,
<STATION DEFAULT DEFINITION>	.2-65;2-65,
<STATION DEFAULT IDENTIFIER>	2-65;2-65,2-66,
<STATION DEFAULT STATEMENT>	.2-66;2-65,
<STATION DEFINITION>	.2-65;2-65,

ITEM PAGE

<STATION IDENTIFIER> 2-65;2-65,2-76,2-87,2-93,
<STATION LIST> 2-65;2-65,2-87,
<STATION MODEM STATEMENT>. 2-75;2-65,
<STATION PAGE STATEMENT> 2-68;2-65,
<STATION PHONE STATEMENT>. 2-72;2-65,
<STATION SECTION>. 2-65;2-9,
<STATION TERMINAL STATEMENT> 2-75;2-65,
<STATION TIMELIMIT STATEMENT>. 2-68;2-65,
<STATION WIDTH STATEMENT>. 2-67;2-65,
<STATUS TOGGLE>. 2-20;2-15,
<STORE ACTION PART>. 2-28;2-27,
<STORE PARAMETER>. 2-28;2-27,
<STORE STATEMENT>. 2-27;2-24,
<STRING> 2-7;2-5,2-6,2-7,2-10,2-28,2-30,
2-66,
<STRING BRACKET CHARACTER> 2-3;2-3,
<STRING CHARACTER> 2-3;2-3,2-7,
<SUM PARAMETER>. 2-28;2-28,
<SUM STATEMENT>. 2-28;2-24,
<TALLY NUMBER> 2-22;2-22,
<TALLY-TOGGLE LIST>. 2-26;2-26,2-27,

ITEM	PAGE
<TERMINAL ADAPTER STATEMENT>	.2-60;2-49,
<TERMINAL ADDRESS STATEMENT>	.2-51;2-48,
<TERMINAL ATTRIBUTE LIST>	2-49;2-48,2-49,
<TERMINAL ATTRIBUTE STATEMENT>	.2-49;2-49,
<TERMINAL DEFAULT ATTRIBUTE LIST>	.2-48;2-48,
<TERMINAL DEFAULT ATTRIBUTE STATEMENT>	2-48;2-48,2-49,
<TERMINAL DEFAULT DEFINITION>	.2-48;2-48,
<TERMINAL DEFAULT IDENTIFIER>	2-48;2-48,2-49,
<TERMINAL DEFAULT STATEMENT>	.2-49;2-48,
<TERMINAL DEFINITION>	.2-48;2-48,
<TERMINAL DUPLEX STATEMENT>	.2-60;2-49,
<TERMINAL IDENTIFIER>	.2-48;2-48,2-75,2-90,
<TERMINAL LIST>	.2-48;2-90;2-48,2-90,
<TERMINAL PAGE STATEMENT>	.2-53;2-48,
<TERMINAL REQUEST STATEMENT>	.2-60;2-49,
<TERMINAL SECTION>	2-48;2-9,
<TERMINAL TIMEOUT STATEMENT>	.2-53;2-48,
<TERMINAL TURNAROUND STATEMENT>	.2-56;2-48,
<TERMINAL WIDTH STATEMENT>	.2-52;2-48,
<TERMINATE STATEMENT>	.2-36;2-24,

ITEM	PAGE
<TERMINATE TYPE>	.2-36;2-36,
<TEXT CONTROL CHARACTER>	.2-32;2-32,
<TIME>	.2-8;2-5,2-25,2-32,2-42,2-46, 2-53,2-55,2-56,2-68,2-74,
<TIMEOUT PART>	.2-32;2-32,
<TOGGLE>	2-15;2-15,2-24,
<TOGGLE NUMBER>	2-16;2-16,2-26,
<TRANSMISSION NUMBER>	.2-63;2-63,
<TRANSMISSION NUMBER STATEMENT>	.2-63;2-49,
<TRANSMIT ACTION PART>	.2-30;2-30,
<TRANSMIT ADDRESS>	.2-51;2-51,
<TRANSMIT ADDRESS PART>	.2-51;2-51,
<TRANSMIT DELAY STATEMENT>	.2-46;2-44,
<TRANSMIT ERROR FLAG>	2-16;2-16,2-30,
<TRANSMIT STATEMENT>	.2-30;2-24,
<TYPE>	.2-80;2-80,
<TYPE LIST>	.2-80;2-80,
<UNCONDITIONAL REQUEST STATEMENT>	.2-24;2-14,
<UNIT>	.2-8;2-8,
<UNITARY STRING>	.2-7;2-7,2-20,2-28,2-32,2-54, 2-55,2-62,2-69,