SPERRY✦UNIVAC

RELEASE DOCUMENTATION

document

SYSTEM RELEASE DESCRIPTION (SRD) SUPPLEMENT

system ___OS/3 System 80 Model 8___

product level ___8.2___

issue number ___N/A___

software category ___1___

date ___February 1984___

Product: SPERRY Operating System/3 (OS/3)

ANSI'74 COBOL NON-ENGLISH TEXT UTILITY

for System 80 Model 8

distribution

lists:

     B00, 28U, CZ, SBZ .

PREFACE

This software document is a supplement to the 8.2
Software Release Description (SRD) of the Sperry Univac
Operating System OS/3.

The OS/3 Release 8.2 ANSI'74 COBOL non-English text utility
is provided for System 80 Model 8 users. This document
contains information not intended for general distribution.

This product should be ordered using the Software System
Request form (UD1-1423).

TABLE OF CONTENTS

1. Introduction

The Non-English Text Utility is used in conjunction with the ANSI '74 COBOL Compiler to produce listings in languages other than English. The utility is used to provide translated diagnostic messages and listing headers to the COBOL compiler. When compiling his program the user may specify the language in which his messages and headers are to be printed.

The utility is comprised of two programs, the Data Generator (DATGEN) and the Module Generator (MODGEN). The Data Generator reads in translated messages as transactions and builds a master file containing the translated messages. The master file for English messages is supplied with the Text Utility modules. The master file is used as input to the Module Generator. This program builds two assembler language source modules which contain the translated information. These modules will be assembled and provided to the compiler.

2. Data Generator

The Data Generator reads the translated messages and provides a file containing the new messages. The program reads the transactions and compares them to the supplied master file containing English messages. If a translation is provided, it is placed in the output file; if not, the English version is used. The Data Generator can also be used to update existing master files and provide listings. If a (non-English) master file is used as input along with the transactions, an updated version of the (non-English) master file will be produced. If no transactions are input, the master file output will be built from the master file input (English is the default) and a listing of the file will be produced. The master files are SAM type with variable length records in 2048 byte blocks.

2.1 Transaction File

The transaction file contains translations of COBOL messages. The transaction can only apply to an existing message, new ones can't be added. There are four types of messages which may be translated; diagnostics, coded symbols, headers and comments.

2.1.1 Message Types

The diagnostics are the error messages generated by the compiler. These messages may consist of a number of words, including inserts. The inserts indicate where in the message additional data (if any) is to be placed. Each word can have up to 103 characters.

Coded symbols are character strings which are inserted in the COBOL diagnostics. One diagnostic message may be applicable to a number of conditions and these conditions may be indicated by separate symbols. A coded symbol may be a COBOL reserved word, in which case it should not be translated. The listing indicates whether a symbol is reserved or not, and in which messages and inserts the symbol may appear. A coded symbol consists of one word of no more than 103 characters.

The headings used in the compiler listings are the header messages. A header consists of a series of words and inserts which identify the various compiler listings and the data printed thereon. The words in some headings must appear in particular positions of the line (e.g. column headings); these are positional headers. The Data Generator will automatically align words in headers of this type. The total length of the header should not exceed 120 characters. The listing of the master file will contain the header type, length and short description of inserts, and the length of all words in positional headers.

The last type of message is comments. These messages are used to provide additional description and commentary on output lines. A comment consists of one word of no more than 120 characters.

2.1.2 Transaction Format

Each line of the transaction file contains the translation of a message. The first word of the transaction is a single letter which identifies the type of the message: A for diagnostics, B for coded symbols, C for headers and D for comments. This is followed by the identifying number for the message. This number accompanies the message on the listing and it has at most 5 digits, leading zeros not required. The word(s) of the translation complete the transaction.

Each word of the transaction must be separated from the other words by at least one space. The punctuation characters (colon, semicolon, comma, and period) need not be preceded by a space but must be followed by one. Inserts are indicated by suffixing a number to the characters "INS". The English message inserts are numbered from left to right but the order of appearance of the inserts may be changed by simply putting the insert indicator in its new position.

Two special characters are allowed in the transactions; the continuation ("-") and the connector ("@"). If a transaction needs more than one line, a continuation should appear in column 1 of all succeeding lines. This will append all characters of the continued line to the last non-blank character of the preceding line. Note that a space is required in column 2 if the last word on the preceding line is ended on that line (i.e. not continued). The connector allows two or more words to be joined into one. Although this symbol is always treated as part of the word, it will be changed to a space before it is printed. This symbol is useful for putting multiple words in a coded symbol or comment, and for adding spaces between words in headings.

Examples

TRANSACTION	NEW MESSAGE
Type Number Translation	Result
A 0901 THIS IS TRANSLATED.	THIS IS TRANSLATED.
A 803 TRANSLATION EXAMPLE	TRANSLATION EXAMPLE
A 5 CONTAINS INS1 INSERT.	CONTAINS *insert* INSERT.
B 08005 USE@OF@CONNECTOR	USE OF CONNECTOR
C 11001 THE INS2 INSERT IS INS1.	THE *ins-2* INSERT IS *ins-1*.
C 15007 DOUBLE@ SPACE@ WORDS	DOUBLE SPACE WORDS
D 07001 CONTIN -UED	CONTINUED
A 07002 TOO MANY WORDS FOR - ONE LINE.	TOO MANY WORDS FOR ONE LINE.

2.2 Sample Translation

To perform the translation, a listing of the messages to be translated is required. Below is a sample set of messages to be translated. It contains 3 diagnostics, 1 coded symbol, 2 headers and 2 comments.

Num	Message
1	WORD TOO LONG.
5	TRUNCATION OCCURS ON *insert-1*.
95	THE *ins-1* PICTURE *ins-2* IS INCOMPLETE.
03002	FRACTIONAL
12001	PROGRAM-ID: *ins-1* DIAGNOSTIC MESSAGE PAGE *ins-2*
12002	LINE LVL ERROR DIAGNOSTIC MESSAGE
14035	*** ADDRESSES OF PEP TABLES ***
14038	COLLATING SEQ TABLE

To translate these messages into another language (eg. Pig Latin), a transaction file must be built containing the translations. Each transaction begins with a code letter for the type of message; A for diagnostics, B for coded symbols, C for headers and D for comments. This is followed by the message number and the translation. The translation has indicators for inserts and allows changes in the word order if desired. If a translation is too long for one record it may be continued onto successive cards by using the continuation character (-). The connector symbol (ə) is used to introduce spaces into the translation. Note its use in 12001 to center the words ´DIAGNOSTIC LISTING´. Message 12002 is a positional header, ie. one where the position of the words is important (as in column headers). The utility will place each word in its correct position, and if more than one word is required at one position it can be made with the connector symbol. Note that in the two comments, since more than one word is desired in a coded symbol or comment, the connector is again used to connect multiple words. The following is a possible transaction file for Pig Latin.

```
|A   1   ORDWAY ONTAINSCAY  OOTAY ANYWAY ARACTERSCHAY.
|A   05  UNCATIONTRAY OCCURSAY ONAY INS1.
| A  95   INS2 ISAY INCOMPLETEAY INS1 ICTUREPAY.
|B 03002 ACTIONALFRAY
| C 12001 OGRAM-IDPAY: INS1 əəəəIAGNOSTICDAY ISTINGLAY
|- əəəəAGEPAY INS2
| C  12002  LINE LVL ERRORAY IAGNOSTICDAYəESSAGEMAY
|D 14035   ***əəDDRESSESAYəOFAYəEPPAYəABLESTAYəə***
|D 14038   OLLATINGCAYəEQSAYəAəLETAY
```

2.3 Interface

The listing produced by the Data Generator provides information on the messages contained in the master file. For diagnostic messages this is the number, severity and text of the diagnostic. Besides the number and text of the coded symbol, the usage, in message and insert numbers, and an indication for reserved word (R) or not (N) are provided. The information for headers includes number, type (positional is P, otherwise N), length of inserts, short descriptions of inserts, length of each word for positional headers, and the text of the heading. The comment data has the number, maximum length and text for the comment.

Three files are used for input and output to the Data Generator; the master file output (MASTOUT), the master file input (MASTIN), and the transaction file (TRANS). The use of these files is indicated by the parameters. These parameters are set by the PARAM statement with the form: // PARAM parms

The parameters are:
```
     MASTOUT=YES    (non-English) master output  file produced    (default)
            NO    file not produced

     MASTIN=YES    (non-English) master input file provided
            NO    file not provided   (default)

     TRANS=YES    transaction file provided
           NO    file not provided   (default)
           SYSTEM control stream
           (modname)    module modname on transaction library file
```

The disc files to be used will have linknames of MASTOUT for the master output file, MASTIN for the (non-English) master input file and TRANS for the transaction file.

3. Module Generator

The Module Generator reads the master file and produces assembler source files for the language. There are two source files for every language. The diagnostics and coded symbols are in one module and the headers and comments are in the other. The Module Generator places the assembler source in files the user specifies and provides listings of the source. The input to the Module Generator is a master file produced by the Data Generator. This file contains the description of the messages. The Module Generator formats the information in a form the compiler can process and builds two assembler source programs containing the information.

One source contains the diagnostics and coded symbols and is used when the error messages are printed. The other source contains listing headers and comments and is used whenever a listing is produced. The two sources are placed in separate files and listings of them are produced.

There are two parameters used by the Module Generator. They are specified by using the PARAM statement with the form: // PARAM parms

The parameters are:
    NAME=language-name (8 characters or less)   default: English
    FILE=
          file-name   default: COBLLANG

Two source modules will be placed in the library file named, with module names consisting of the language name suffixed with 'M' for diagnostic messages and 'H' for headers. The master input file must exist and have linkname MAST.

4. Compiler Interface

The ANSI '74 COBOL compiler will use the information contained in the
message modules to present its output in the different languages. The
two source modules provided by the utility will be assembled and made
available to the compiler. To specify the language to be used, the
MESSAGES clause in the ALPHABET section of the CONTROL division is used.
The language specified in the MESSAGES clause should be the same as that
specified to the Module Generator. The correct language modules will be
selected from those available to the compiler and used for the listings.

The two modules provided by the utility should each be assembled. The
two object modules thus obtained will be linked into two separate load
modules which are placed in the system load library. When a module is
needed by the compiler, a LOAD is performed.

5. Summary

The Non-English Text Utility is made up of two programs, the Data Generator and the Module Generator, which are summarized below.

5.1 Data Generator

Inputs
    Master input file: linkname MASTIN
    Transactions: if disc file, linkname TRANS; else on system
    input stream

Outputs
    Master output file: linkname MASTOUT
    Master file listing

Parameters
    [MASTOUT={YES| NO}] [, MAST IN={YES |NO}]
    [, TRANS={YES |NO| SYSTEM| (mcdname)}]

Figure 1

DATGEN INPUTS

```
    -----                                                        -----
   -      -                    --------------+              -         -
  +  SPOOLIN  +                /             |            +    SAM     +
  |  -     -  |               |  Card Reader |            |  -      -  |
  |           |               |              |            |           |
  |           |               |              |            |           |
  +  Disc File +              +--------------+            +  Disc File +
   -         -                      :                      -         -
    -----                           :                        -----
      :                             :                          :
      :              +----------+   :                          :
      :              +          +   :                +-----------------+
      :--------------+  GETCS   +-----------:         +    SAMIO    +
                     +          +                     +            +
                     +--------+                       +--------+
                         :
                         :
                         :------(optional via TRANS=  )-----------:
                         :                            (TRANS)
    -----                            :
   -     -                   :------:             /-\
  +  SAM   +                 :              :     |o|
  |  -   - |                 :              :     |p|        -----
  |        |                 +-----------+  :     |t|       -      -
  |        |                 |           |  :     |i|      +  SAM   +
  +  Disc File +             |           |        |o|      |  -   - |
   -         -               |           |        |n|      |        |
    -----                    |  DATGEN   |--------|a|      |        |
      :                      |  INPUTS   |        |l|      +  Disc File +
      :                      |           |        | |       -         -
      :                      |           |        |v|        -----
      :                      +-----------+        |i|           :
  +-----------+                                   |a|           :
  +           +                    :              | |           :
  +  SAMIO    +                    :              |M|      +-----------+
  +          +                     :              |A|      +           +
  +--------+                       :              |S|      +  SAMIO    +
     :    (LFD = MSTABLE)          :              |T|      +          +
     :    (LBL = COBLERRS)         :              |I|      +--------+
     :-----------------------------:              |N|--------:
                                                  |=|  (MASTIN)
                                                  \-/
```

Figure 2

DATGEN OUTPUTS

```
                              +----------------+
                              |                |
    :---------------------|   |    DATGEN      |   /-\
                          |   |    OUTPUTS     |   |o|
    :(PRNTR)              |   |                |   |p|----------:
    :                     |   |                |   |t|          :
    :                     +----------------+   |i|          :
  +----------------+                           |o|   +----------------+
   +            +                             :  |n|    +            +
    +  SAMIO  +                               :  |a|     +  SAMIO  +
     +        +                               :  |l|      +        +
      +------+                                :  | |       +------+
         :                                    :  | |          :
         :                                    :  |v|          :
         :                                :---|i|          :
         :                            (MAS TOUT) |a|
  +----------------+                             | |       -       -
  |                |                             |M|     +   SAM   +
  |   LISTING   |                             |A|     |  -     -  |
  |             ----+                          |S|     |           |
  |         ---/                               |T|     |  -----    |
  +----/                                       |O|     + Disc File +
                                               |U|     -       -
                                               |T|
                                               |=|      -----
                                               \-/
```
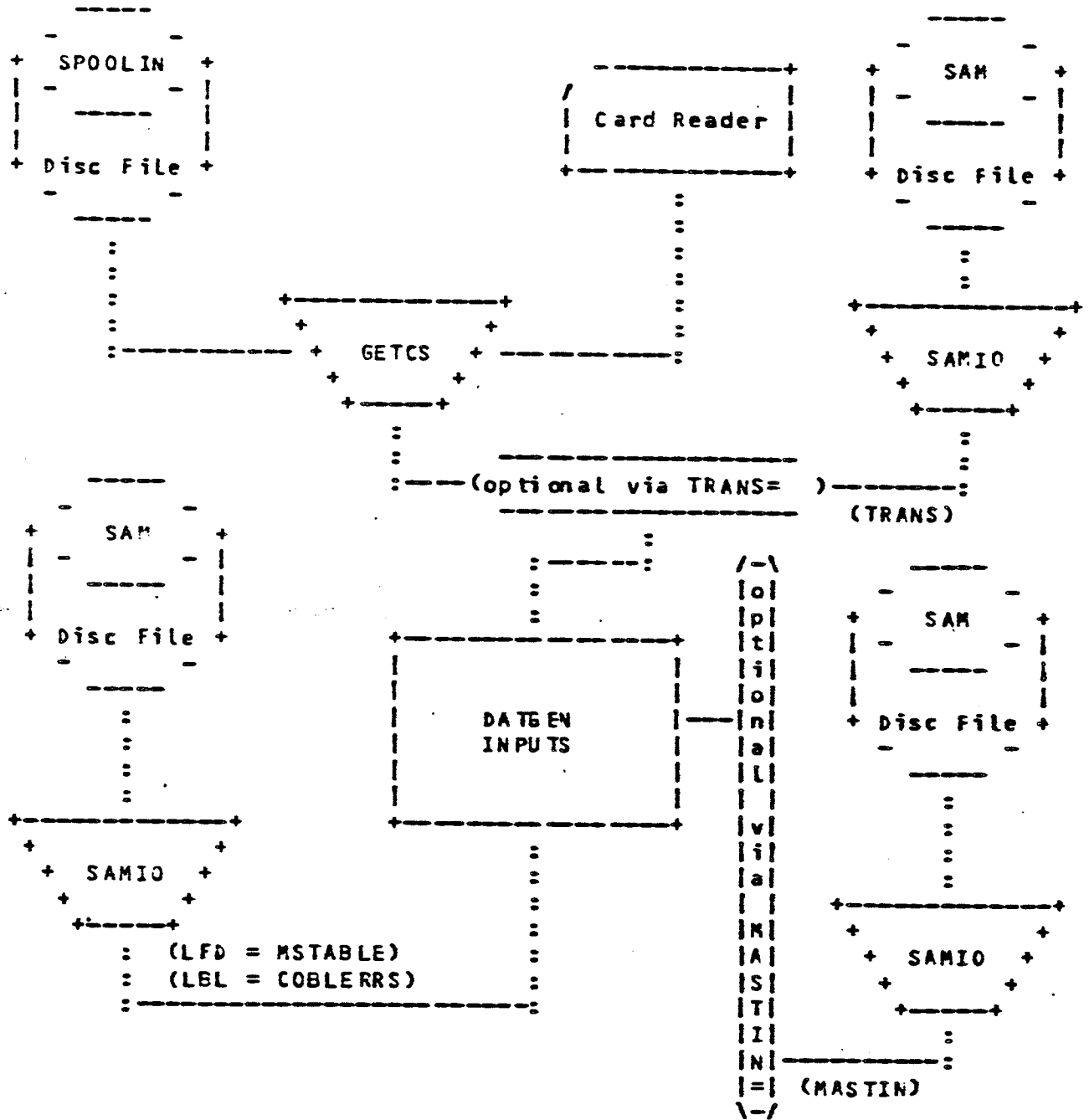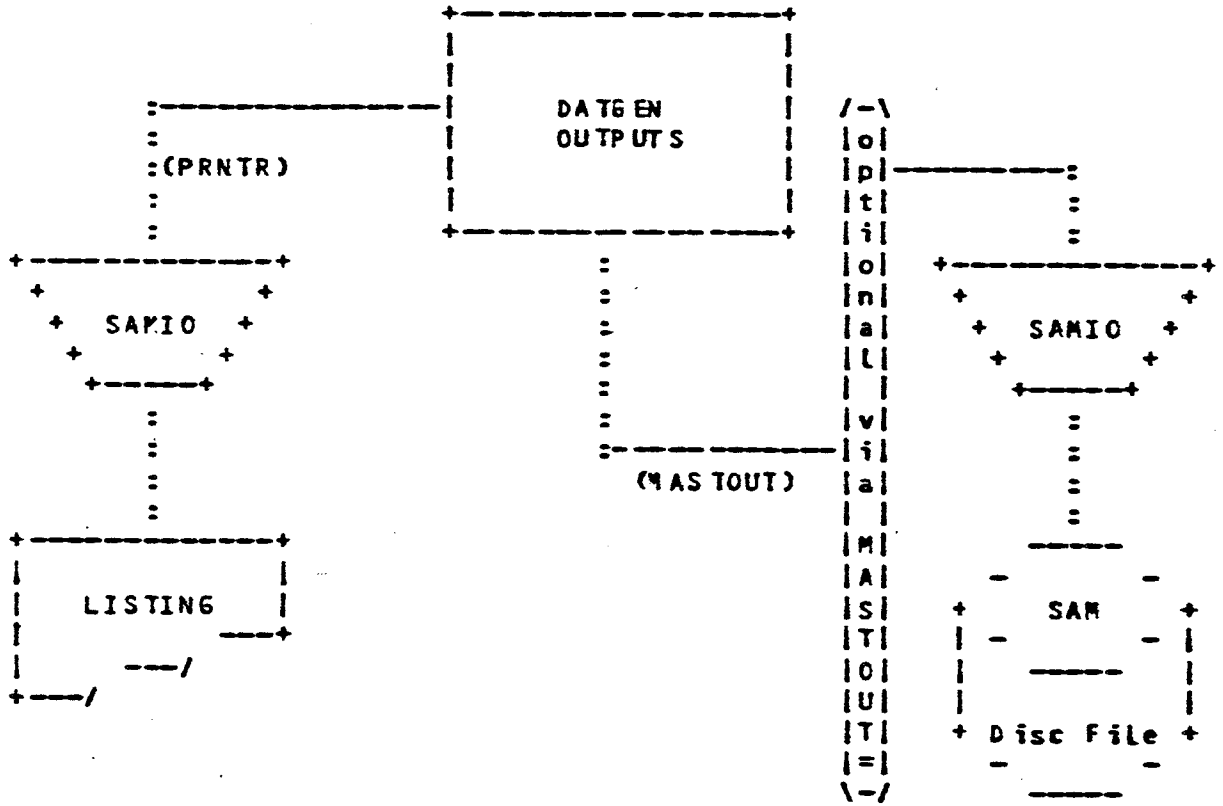
5.2 Module Generator

    Inputs
        Master input file: linkname MAST

    Outputs
        Two modules in one library file
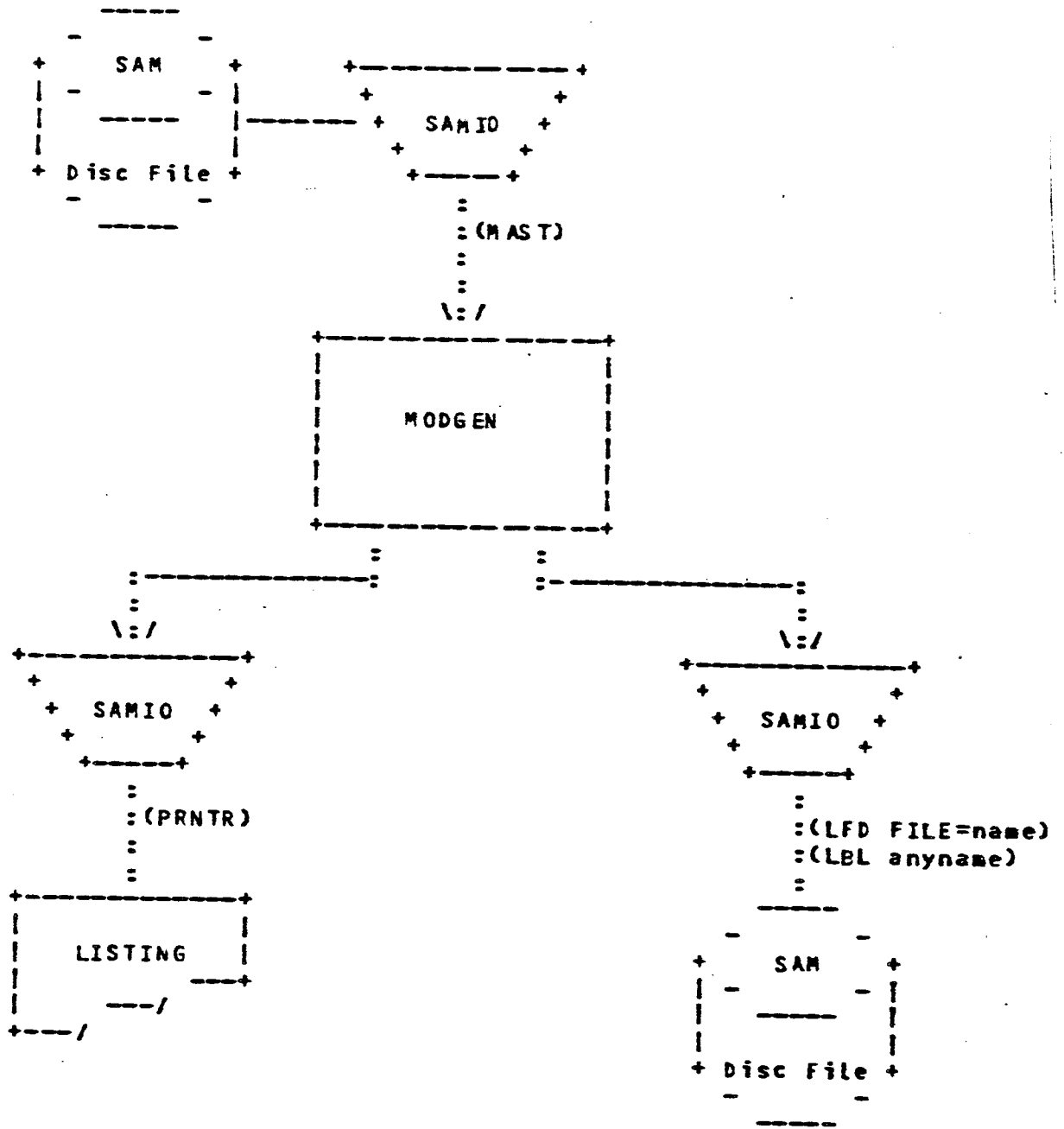
        Listings of the two modules

    Parameters
        [FILE=filename] [,NAME=language-name]
        default FILE is COBLLANG
        default NAME is ENGLISH

MODGEN INPUTS AND OUTPUTS

Figure 3

```
     -----                          +------------+
   -       -              +---------+            +
  + SAM     +             |         |            |
  | -     - |------------ +   SAMIO +            +
  |         |             |         |            |
  |  ----   |             +    +----+            +
  + Disc File +                :
   -       -                   :(MAST)
     -----                     :
                               :
                              \:/
                  +--------------------+
                  |                    |
                  |                    |
                  |                    |
                  |      MODGEN        |
                  |                    |
                  |                    |
                  |                    |
                  +--------------------+
             :                              :
          +-----------------+            +------------------+
          :                 :            :                  :
         \:/                             \:/
    +---------------+              +---------------+
   +                 +            +                 +
  +    SAMIO          +          +    SAMIO          +
   +                 +            +                 +
    +    +-----+    +              +    +-----+    +
         :                              :
         :(PRNTR)                       :(LFD FILE=name)
         :                              :(LBL anyname)
         :                              :
    +---------------+                  -----
    |               |                -       -
    |   LISTING     |               + SAM     +
    |          -----|               | -     - |
    |          ---/ |               |  ----   |
    +--/                            |         |
                                    + Disc File +
                                     -       -
                                       -----
```

6. Sample Job Stream

The following show different uses of the utility with the corresponding
job stream.

```
// JOB UPDATE,,C000                                            1
// DVC 20 // LFD PRNTR                                         2
// DVC 50 // VOL vol // LBL UPDATES // LFD TRANS               3
// DVC 51 // VOL vol1 // LBL ERRORS // LFD MASTIN              4
// DVC 50 // VOL vol // LBL ERROUT // LFD MASTOUT              5
// DVC RES // LBL COBLERRS // LFD MSTABLE                      6
//TRNERR WORK1                                                 7
// EXEC DATGEN                                                 8
// PARAM MASTIN=YES,TRANS= (mod name)                          9
// DVC 50 // VOL vol // LBL ERROUT // LFD MAST                 10
// DVC 51 // VOL vol1 // LBL ERRASM // LFD CBFRENCH            11
// EXEC MODGEN                                                 12
// PARAM NAME=FRENCH,FILE=CB FRENCH                            13
/&
```

```
// FIN
```

1.   Specify JOB card. Note that 48K must be available.
2.   Specify printer device.
3.   Specify library file that contains transactions.
4.   Specify master file to be updated.
5.   Specify output master file.
6.   Specify message master. This must always be specified.
7.   Specify work file. This must be done if any transactions are done.
8.   Execute the Data Generator.
9.   Specify master file used and transaction module.
10. Assign LFD name for master file.
11. Specify output file for language modules.
12. Execute Module Generator.
13. Specify language as French and output file LFD. Modules FRENCH and
FRENCH will be built.

```
// JOB LISTENG,,8000                                           1
// DVC 20 // LFD PRNTR                                         2
// DVC RES // LBL COBLERRS // LFD MSTABLE                      3
// EXEC DATGEN                                                 4
// PARAM MASTOUT=NO                                            5
/&
// FIN
```

1.   Specify JOB card. Note that 32K must be available.
2.   Specify printer device.
3.   Specify message master.
4.   Execute the Data Generator.
5.   Set to get output listing only.

7. Error Messages

If an error is encountered during processing of the translations, a message is printed. The following table lists the 11 errors and 2 warnings which may be printed. Severity 2 indicates the transaction will be deleted, severity 1 indicates the transaction will be applied with errors and a simple warning is severity 0.

Sev	Message	Cause
2	ILLEGAL ID CATEGORY	An ID other than A, B, C, or D was encountered.
2	ILLEGAL MESSAGE NUMBER	A non-numeric message number or a number with too many characters was encountered
2	NON-EXISTANT MESSAGE IS TRANSLATED	The message number does not correspond to any known message.
2	MESSAGE CONTAINS WRONG NUMBER OF INSERTS	The number of inserts in the translation does not correspond to the actual number of inserts in the message
1	DIAGNOSTIC WORD OVER 103 CHARACTERS	The word is too long and is truncated.
1	CODED SYMBOL OVER 103 CHARACTERS	The coded symbol is too long and is truncated.
1	POSITIONAL HEADER WORD TOO LONG	A word in a positional header is too long and is truncated.
1	COMMENT CONTAINS TOO MANY CHARACTERS	The comment exceeds the maximum size allowed.
1	CHARACTER STRING EXCEEDS 120 CHARACTERS	The character string is too long.
1	MESSAGE TEXT NOT FOUND	No translation is provided for the message.
1	CODED SYMBOL OR COMMENT HAS >1 WORD	Coded symbols and comments can only have one word. See connector symbol for multiple words.

Sev	Message	Cause
0	RESERVED WORD SYMBOL WAS TRANSLATED	A coded symbol comprised of reserved word(s) was translated.
0	WARNING: nnn MESSAGES NOT TRANSLATED	When a translatIion of the master file is being done (not an update), the number of non-translated messages is printed.

8. English Listing

The default language used is English. The Data Generator (DATGEN) should be used to obtain formatted printed listings of the master file for the English Language.

9. MASTER FILE (ENGLISH)

The master file used by the Data Generator is supplied with the Text Utility programs. This file must be used by the Data Generator program.

The master file is supplied with the Text Utility possibly on tape (as a SAM file). The tape file must be converted to a SAM DISK file to be used by the Data Generator program. The following show a sample use of OS/3 DATA UTILITIES to put the tape file onto a DISK in a form usable by the Data Generator.

```
// JCB LOADIT,,8000
// DVC 20 // LFD PRNTR
// DVC 91 // VOL S04095 // LFD INPUT1
// DVC RES // EXT SQ,C,0,CYL,5
// LBL COBLERRS // LFD OUTPUT1,,INIT
// EXEC DATA
/$
UTD A=(2044,2048),FV,LD,IR,OR
/*
/&
// FIN
```