digital

SEE 1-9

libmsh
users manual

digital equipment corporation

DEC-12-AMLBA-A-D

LIBMSH USER'S MANUAL

SOFTWARE SUPPORT CATEGORY

The software described in this document is supported by Digital Equipment Corporation under Category 1 , as defined on page iii of this document.

For additional copies, order No. DEC-12-AMLBA-A-D from Software
Distribution Center, Digital Equipment Corporation, Maynard,
Massachusetts 01754 Price $5.00

First Printing
July, 1972

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page tells you how to keep up-to-date with DEC's software. Completion and return of the "Reader's Comments" page is beneficial to both you and DEC; all comments received are acknowledged and are considered when documenting subsequent manuals.

The material in this document is for information purposes and is subject to change without notice.

SOFTWARE SUPPORT CATEGORIES

Digital Equipment Corporation (DEC) makes available four categories of software. These categories reflect the types of support a customer may expect from DEC for a specified software product. DEC reserves the right to change the category of a software product at any time. The four categories are as follows:

CATEGORY I
Software Products Supported at no Charge

This classification includes current versions of monitors, programming languages, and support programs provided by DEC. DEC will provide installation (when applicable), advisory, and remedial support at no charge. These services are limited to original purchasers of DEC computer systems who have the requisite DEC equipment and software products.

At the option of DEC, a software product may be recategorized from Category I to Category II for a particular customer if the software product has been modified by the customer or a third party.

CATEGORY II
Software Products that Receive Support for a Fee

This category includes prior versions of Category I programs and all other programs available from DEC for which support is given. Programming assistance (additional support), as available, will be provided on these DEC programs and non-DEC programs when used in conjunction with these DEC programs and equipment supplied by DEC .

CATEGORY III
Pre-Release Software

DEC may elect to release certain software products to customers in order to facilitate final testing and/or customer familiarization. In this event, DEC will limit the use of such pre-release software to internal, non-competitive applications. Category III software is only supported by DEC where this support is consistent with evaluation of the software product. While DEC will be grateful for the reporting of any criticism and suggestions pertaining to a pre-release, there exists no commitment to respond to these reports.

CATEGORY IV
Non-Supported Software

This category includes all programs for which no support is given.

CONTENTS

PREFACE

The reader is expected to be familiar with OS/8 which is described in

Introduction to Programming, Chapter 9

OS/8 Software Support Manual (DEC-08-MEXB-D)

1.0 INTRODUCTION

LIBMSH is the MASH LIBRARY program, written in FORTRAN IV (RTPS), used to catalog pertinent information for individual MASH files. It can also delete entries, replace entries, move entries from one catalog to another, re-patch a catalog by eliminating empty entries due to a delete, list the index of a catalog or the information in a given entry, and search for specific entries.

The MASH LIBRARY package is composed of four programs: a transfer program, a print program, an allocate program, and the MASH library program. Each of these four programs is explained in detail in the following sections.

The LINCtape provided with the MASH library package contains the following:

a. An OS/8 tape system plus three OS/8 system programs -- ABSLDR.SV, PIP.SV, BUILD.SV. The OS/8 system is a LINCtape system (tape units are referred to as DTAØ, DTA1,...,DTA7) and can not be run from any disk. To create an OS/8 system on disk, refer to the section on BUILD in Chapter 9 of Introduction to Programming, 1972.

b. The binary programs used by the MASH library.

c. Six programs -- EZGEN.SV, FORLIB.RL, FORIO.SV, FORRTS.FT, LOAD.SV, MRKRTS.SV, which are also a part of the RTPS FORTRAN IV package, are necessary in order to run two of the programs in the MASH library package which require the loader (namely: PRINT and LIBMSH). It is strongly suggested that these programs reside on the device containing the OS/8 system.

NOTE

When using PIP.SV to transfer any of these files from one device to another, always use the /I switch.

It is necessary to configure the Run Time system in order to run the two FORTRAN programs (PRINT and LIBMSH). Refer to Appendix G of the RTPS FORTRAN IV User's Manual (refer to EZGEN and MRKRTS). Once the Run Time System is configured, the loader (LOAD) may be used to create the load modules (see Sections 3.2.1 and 3.3.1 of this manual).

NOTE

If FORRTS.FT and FORLIB.RL are not on the systems device, then careful reading of the LOADER (re: specifying a library file and specifying alternate run time support) should be carried out before attempting to build the load modules.

2.0 HARDWARE CONFIGURATION

All programs in this package run under the control of the OS/8 Operating System and in a minimum hardware configuration which includes 8K core memory, a console Teletype[1], two LINCtape transports, and a Floating Point Processor (FPP). Other desirable hardware includes a line printer and a disk.

3.0 PROGRAMS

This section provides a description of each of the four programs, along with compile/assemble/load instructions, examples, and input/output dialogue for each of the programs. All examples assume an OS/8 disk operating system.

3.1 The Transfer Program

Since all RTPS FORTRAN programs which run under control of OS/8 require OS/8 compatible files, this program is used to transfer AIPOS (MASS/INTENSITY) files with double precision data values to OS/8 files with floating point data values. If the file to be transferred is not a MASS/INTENSITY file, the following message is written on the Teletype, and a different file may be entered.

```
NOT M/I FILE
```

3.1.1 Create a SAVE File:

Save the binary of the transfer program TRANS.BN as follows:[2]

```
.R  ABSLDR)
*TRANS.BN$
.SAVE SYS TRANS.SV 12ØØØ-12577;12ØØØ)
```

[1] Teletype is a registered trademark of the Teletype Corporation

[2] System output is underlined throughout this manual. $ = ALT MODE key.) = RETURN key.

3.1.2 Input/Output:

Use the following input sequence to make a transfer:

```
.R  TRANS)
*odev:oname.ex< )
UNIT: u)
STBLK: s)
LEN: ln)
```

where: odev = output device for the transfer (OS/8 tape).

oname.ex= name and extension of the OS/8 file.

u = LINCtape unit number (0,1,...,7) of AIPOS file.

s = starting block of AIPOS file to be transferred.

l_n = number of blocks in length of this AIPOS file.

The output of TRANS is an OS/8 floating point file containing MASS/INTENSITY pairs.

Example:

Assume a partial listing of the index of an AIPOS data tape on LT4 is:

```
    :
WHEE.ØØ1  452  3
WHEE.ØØ2  455  4
    :
```

Now, transfer each of the above files (assumed to be MASS/INTENSITY files) to an OS/8 tape on unit 2, and keep the same names (refer to NOTE 3 on next page).

```
.R  TRANS↵
* DTA2:WHEE.Ø1<↵
UNIT:  4↵
STBLK: 452↵
LEN:  3↵

* DTA2:WHEE.Ø2<↵
UNIT:  4↵
STBLK:  455↵
LEN:  4↵

* ↑C[1]
.
```

NOTES

1. The output device may be disk as well as LINCtape.
2. The AIPOS files to be transferred must reside on LINCtape.
3. OS/8 allows a six-character file name with a two-character extension; whereas AIPOS allows a six-character file name with a three-character extension.

3.2 The Print Program

This program is used to print the MASS/INTENSITY pairs for a given OS/8 file.

3.2.1 Link and Load:

Link and load the Print program source, PRINT.RL, as follows:

```
.R  LOAD↵
*PRINT.LD,LPT:<PRINT.RL/D↵
```

at this point the Print program is ready to accept the files to be printed.

```
*idev:iname.ex/5↵
```

where: idev = input device of the program to be printed.

iname.ex= name and extension of the program to be printed.

5 = logical device used by the Print program for input. Print equates this number to the file defined by idev:iname.ex.

[1] ↑C = CTRL/C which is typed by holding down the CTRL key while typing the C key.

3.2.2 Input/Output:

Assuming a Load program (PRINT.LD) exists, the only input required is the file to be printed (as created in section 3.2.1),

```
.R  LOAD↵
*PRINT.LD/D↵
*idev:iname.ex/5↵
```

The output is to the line printer, or if no line printer is available, to the Teletype. It consists of two columns of data. The first column contains MASS values times ten (MASS*10), and the second column contains the corresponding intensity values.

Example:

Print the MASS*10/INTENSITY pairs for each file transferred to the OS/8 tape in the TRANS example.

```
.R LOAD↵
*PRINT.LD/D↵
*DTA2:WHEE.Ø1/5↵

.R LOAD↵
*PRINT.LD/D↵
*DTA2:WHEE.Ø2/5↵

.
```

3.3 The Allocate Program

Use the allocate program to open and close a file of user-specified unit, name and length, so that the file can be initialized by the function INIT (refer to section 3.3.2) and used for a new catalogue.

3.3.1 Create a SAVE File:

Save the allocate program binary, ALOCAT.BN, as follows:

```
.R ABSLDR↵
*ALOCAT.BN$
.SAVE SYS ALOCAT.SV 12ØØØ-12177;12ØØØ↵
```

3.3.2 Input/Output:

Use the following input sequence to allocate a file.

```
.R ALOCAT↵
*odev:oname.ex[nb][1]<↵
```

where: odev = mass storage device to contain new catalogue.

oname.ex = name and extension of new catalogue.

[nb] = the length of the new catalogue in blocks.

The output of ALOCAT is a permanent file of user-specified unit, name, and length; which can now be accessed as a catalogue by LIBMSH.

Example:

Create a catalogue 96 blocks long on LINCtape unit 6 (be sure the unit is WRITE ENABLED) and name the catalogue, CATALG.Ø1.

```
.R ALOCAT↵
*DTA6:CATALG.Ø1[96]<↵
*↑C
.
```

NOTE

The largest catalogue that can be allocated is 255 blocks. Since there are four catalogue entries/block, this gives a total of 1Ø2Ø entries/catalogue.

3.4 The MASH Library Program

This program (LIBMSH) makes and manipulates entries in a specific catalogue. In all, the program LIBMSH can perform 8 separate functions. Each of these functions is discussed in the following paragraphs. The signal to enter a function (or to type CTRL/C to abort) is the word

CMD

which is printed on the Teletype.

3.4.1 Link and Load

Don't NEED

To create one program from the binaries of LIBMSH.RL, ENTER.RL, INIT.RL, INTERP.RL, XLIST.RL, DELETE.RL, SQUISH.RL, MOVE.RL, NEXT.RL, GETBLK.RL, PUTBLK.RL, COMPAR.RL, STRNG.RL and CCGET.RL type:

NOTE

The binaries must be on the system device (SYS) for the following LOAD and LINK sequence to work.

[1] The open bracket ([) is produced by holding down the SHIFT key while typing a K (i.e., SHIFT/K); the close bracket is produced by typing a SHIFT/M.

```
.R LOAD↲
*LIBMSH.LD,LPT:< LIBMSH,STRNG,CCGET/O ↲
*ENTER↲
*INTERP↲
*XLIST↲
*DELETE↲
*SQUISH↲
*MOVE/O↲
*COMPAR↲
*INIT↲
*NEXT↲
*GETBLK↲
*PUTBLK↲
*/D↲
```

Can't change order

(If LIBMSH is to be used now, the /D indicates that logical devices are to be specified; (refer to INPUT/OUTPUT Section 3.4.2) otherwise, use CTRL/C to return to the Monitor.)

3.4.2 Input/Output:

Anywhere from one to three files can be used during the course of running LIBMSH.LD. FORTRAN has only one way to refer to input/output devices and files; and this is by a logical device number. The assignment of these numbers takes place at LOAD time. Logical device number 6 must always be specified, numbers 5 and 7 depend on the functions to be specified.

<u>Logical Device Numbers</u> ✓

FORTRAN

DATA 5 - This number is associated with the OS/8 MASH file that is to be used in a SEARCH, or is to be ENTERed into the library catalogue on logical unit 6.

LIB EXITD 6 - This number is associated with the particular catalogue that will receive a file via a SEARCH or an ENTER.

2nd LIB 7 - This number is associated with a second catalogue. The purpose of this catalogue is to receive files via a MOVE. Each MOVE is <u>from</u> logical unit 6 <u>to</u> logical unit 7.

In general then, assuming a LOADed program (LIBMSH.LD) exists; and a MASH file: FILE.Ø1 is on SYS with catalogues: CATALG.Ø1 on DTA1 and CATALG.Ø2 on DTA5, the commands are:

```
.R LOAD↲
*LIBMSH.LD/D↲              BVI.LM/D
*FILE.Ø1/5/C ↲             FILE/5/C
*DTA1:CATALG.Ø1/6/C↲       MSLIDR.01/6
*DTA5:CATALG.Ø2/7↲

CMD
```

It should be noted as shown in the previous example that /C is needed whenever another logical device must be assigned.

The output of this last program is determined by which of the eight functions is specified. In like manner, each function requires its own input, as described in the following paragraphs.

The EIGHT LIBRARY Functions:

INIT Function Call: INn↵ ; where n=6,7

INIT initializes a catalogue for the first time or re-initializes an existing catalogue. A catalogue which was created by the program ALOCAT needs to be initialized before it can be used. Upon loading of LIBMSH, a check is immediately made to see if the catalogue designated by logical unit 6 is initialized. If it is not, the user is forced to initialize it immediately. The same check is made on the catalogue designated by logical unit 7 whenever it is referenced. (Refer to NOTE on next page.)

Associated with each catalogue is a HEADER block which contains 17 floating point words of information. Word 1 (the first word) contains the number, 1934 which signals that the catalogue is initialized. Word 2 contains the number of the next available entry slot in the catalogue. Words 3 and 4 contain a maximum of twelve characters which the user enters as the name of the catalogue. For example, *

```
Program Request:   NAME OF CATALOGUE--12 CHARS MAX ✓
Example Response:  CATALG.Ø1↵
```

Words 5 and 6 contain a maximum of twelve characters which the user enters as the DATE.

```
Program Request:   DAT~~A~~E--12 CHARS MAX ✓
Example Response:  6/21/72↵
```

Words 7-17 contain a maximum of 66 characters which the user enters as descriptive information.

```
Program Request:   USER MESSAGE--66 CHARS MAX
Example Response:  THIS CATALOGUE HOLDS ALL BENZENE COMPOUNDS↵ ✓
```

NOTE

1. If an INIT is automatically forced a message is printed at the Teletype. For example, assuming logical device 6.

   LOGICAL DEV 6 IS BEING INITIALIZED

2. Whenever initialization is requested, the user is given a chance to reconsider his request just prior to the actual initialization of the catalogue.

   Program Request: ARE YOU SURE?--Y OR N

   Response: Y ↲ carries out the initialization.

   N ↲ ignores all previously entered information and leaves the catalogue as it was before the function call was requested

ENTER Function Call: EN ↲

The ENTER function encodes a MASH file (floating point MASS/INTENSITY pairs) either automatically or manually; and collects user response information to create an entry which is placed in a catalogue.

At load time the catalogue that is to receive the entry must be designated as logical unit 6, and if a file is to be automatically encoded, it must be the file designated as logical unit 5.

As soon as the function call ENTER is issued, the Teletype responds with:

AUTO(1) OR MANUAL (2) ENCODE--I1[1]

The I1 indicates that the user types his response as a single digit. A response of 1 indicates automatic encoding which will result in the eight highest MASS/INTENSITY pairs (or fewer if the file contains less than eight pairs), being found. A digit response other then 1 indicates that the user wishes to enter eight (or less) characteristic MASS/INTENSITY pairs manually.

If Auto encode (1) is requested, the next question is:

TOLERANCE:DELT(M)--F3.Ø[1]

[1] Refer to Appendix A for a detailed explanation of user responses.

When the eight highest peaks are being found, only those pairs are considered whose masses lie within the specified tolerance of a whole numbered mass. Each chosen mass is integerized and if more than one MASS/INTENSITY pair has the same mass number, the pair with the highest intensity is chosen. For example, suppose the tolerance is Ø.3 and the three M/I pairs are (15Ø.8, 2531), (151.3, 333) and (152.4, 467). Only the first and second pairs lie within the tolerance, and they become (151, 2531), (151, 333). Since the first pair has a larger intensity, it is kept and the second pair eliminated.

When auto encode is complete, each of the eight intensities is changed to a percent relative to the largest intensity.

If MANUAL encode (2) is requested, the question displayed is:

```
          HOW MANY PEAKS (8 IS MAX)--I1
```

A response of Ø or 9 is interpreted as 8. The next question requests the mass and intensity for the number of peaks (NUMPK) in I4 and F6.Ø format.

```
          MASS VS INTENSITY 1 SET PER LINE [NUMPK] TIMES--I4,F6.Ø
```

Once the eight (or less) MASS/INTENSITY pairs have been entered, the characteristic information for this entry is requested.

```
          ENTER NAME, FORMULA, ETC--3Ø CHARS MAX
```

If the catalogue is not empty LIBMSH gives the option of replacing a catalogue entry.

```
          DO YOU WISH TO REPLACE A CATALOGUE ENTRY--Y OR N
```

A response of Y) outputs the following question to the Teletype:

```
          WHICH ENTRY?--I4
```

If a nonexistent entry number is entered the question is repeated.

If an entry is not to be replaced type N) (or any character except Y). The entry is next placed in the catalogue, the message CMD is printed and another function may be issued.

SEARCH   Function Call: SE↵

This function is similar to ENTER. Its purpose is also to encode a MASH file either automatically or manually and eventually to enter it into the catalogue on logical unit 6 if desired. However, its main purpose, after the encoding is finished, is to find all entries in the catalogue which match it exactly or which match it within the specified intensity tolerance and the required number of peaks(#HITS).

SEARCH displays the question:

TOLERANCE:DELT(I),#HITS--,F3.Ø, I2

The intensity-tolerance is entered in F3.Ø format, and the minimum number of peaks (HITS) that must match up is entered in I2 format.

A report of all entries that are found to match up within the user tolerances is sent out to the line printer (or Teletype by default). The report of each entry consists of the entry number, the 3Ø character identification and the #HITS. If the search is unsuccessful, the message NO MATCHES is output.

The encoded entry can now be entered into the catalogue.

PUT ENTRY IN CATALOGUE?-- Y OR N

An answer of Y↵ requests the remaining information needed to complete the entry (see ENTER). Answer of N↵ (or any character except Y), terminates the function.

XLIST   Function Call: XLn↵ where n=6 or 7

Essentially, XLIST lists the index of a catalogue on Logical Unit 6 or 7. The 17 words of header information of the catalogue (described in INIT), along with the entry number, NAME, FORMULA, etc., (described in ENTER), for each entry of the catalogue are listed on the line printer (or Teletype by default). XLIST provides the option of listing the entire index or entries between specified entry-numbers. Response to the following question determines the index printed.

LO & HI LIMIT OF ENTRY NUMBERS--I4, I4

If the high limit is Ø or is greater than the last entry in the catalogue, its value defaults to the last entry number. If the low limit is Ø, its value defaults to 1. If the low limit is greater than the high limit, the low limit takes on the value of the high limit. For example, assuming a catalogue has twelve entries, XLIST interprets the user entries as shown below[1]:

USER ENTRY: LO HI	PROGRAM VALUES: LO	HI
␣␣␣2␣␣␣␣6↵	2	6
␣␣␣1↵	1	12
3.↵	3	12
↵	1	12
␣␣␣␣␣5.↵	1	5
␣␣5.␣␣␣␣5.↵	5	5
␣␣␣8␣␣␣␣4↵	4	4
3.Ø␣␣23↵	3	12

A catalogue entry that maybe empty (refer to DELETE), is indicated on the printout by:

[num]-----UNUSED-----

where num = number of the entry.

If no entries exist in the catalogue, the message

NO ENTRIES IN THIS CATALOGUE

is output to the Teletype and the function is terminated.

CLIST Function Call: CLn↵ where n = 6 or 7

CLIST is similar to the XLIST function, but lists the contents of each entry of a catalogue found on logical unit 6 or 7. As in XLIST, there is an option to list between specified entry numbers and output is to the line printer (or Teletype by default).

LO & HI LIMIT OF ENTRY NUMBERS--I4, I4

As in XLIST, if the high limit is Ø or is greater than the last entry in the catalogue, its value defaults to the last entry number. If the low limit is Ø, its value defaults to 1. If the low limit is higher than the high limit, the low limit takes the value of the high limit.

If the catalogue contains no entries, the message:

NO ENTRIES IN THIS CATALOGUE

is output to the Teletype before the function terminates.

[1] ␣ = space

DELETE   Function Call: DEn↵   where n = 6 or 7

This function deletes the entry specified in response to the question:

ENTRY NUMBER--I4

from a catalogue on logical unit 6 or 7. Only one entry can be deleted at a time.

If an entry number is specified that is greater than the last entry number, the message

NUMBER [num] IS LAST ENTRY

and a line feed is output and the system waits for the correct entry. If a delete is attempted on an empty catalogue, the message

NO ENTRIES IN THIS CATALOGUE

is output.

When the entry is deleted, a line feed to the Teletype signals that another numbered entry can be deleted. A response of Ø↵ or just ↵ terminates the DELETE function.

SQUISH   Function Call: SQn↵   where n = 6 or 7

When entries, other than the last entry of a catalogue, are deleted, holes, or empty entries are left in the catalogue. Executing a SQUISH on logical unit 6 or 7, causes all entries to be re-packed towards the front of the catalogue and eliminates all empty entries. For example

```
Before:   CMD
          XL6↵
          LO & HI LIMIT OF ENTRY NUMBERS--I4,I4
          ↵

          CATALG.1
          5/11/72

          THIS CATALOGUE IS A REAL CAT.          ⎫
          ENTRY    IDENTIFICATION                ⎪
            1     WHEE.Ø1  TOL:.1                ⎪
            2     -----UNUSED-----               ⎬ Old catalogue listed
            3     WHEE.Ø2  TOL:.3                ⎪ on line printer.
            4     -----UNUSED-----               ⎪
            5     WHEE.Ø2  TOL:.1                ⎭
```

```
SQUISH
Command:  CMD
          SQ6↵

After:    CMD
          XL6↵
          LO & HI LIMIT OF ENTRY NUMBERS--I4,I4
          ↵
          CATALG.1
          5/11/72
          THIS CATALOGUE IS A REAL CAT.
          ENTRY    IDENTIFICATION                New catalogue listed
            1     WHEE.Ø1   TOL:.1               on line printer.
            2     WHEE.Ø2   TOL:.3
            3     WHEE.Ø2   TOL:.1
          CMD
          ↑C
          .
```

MOVE Function Call: MO↵

This function moves a contiguous set of entries from the catalogue on logical unit 6 to the catalogue on logical unit 7. The moved files are placed at the first available entry designated by word 2 of the header block. This function allows movement of a set of entries or duplication of an entire catalogue. MOVE outputs the question:

```
LO & HI LIMIT OF ENTRIES TO MOVE--I4,I4
```

The same rules for inputting numbers apply to MOVE as apply to XLIST.

If the catalogue is empty, the message

```
NO ENTRIES IN THIS CATALOGUE
```

is output and the function terminates.

NOTE

1. When the program LIBMSH is running, any illegal function call (such as XL2, DE, DI, LC, etc.) causes the output of the message

   CMD ERR

   and the CMD question is asked again.

2. When printing (XLIST, CLIST, etc.), a string of ASCII characters on the Teletype (not line printer) that were previously entered into the header block during an INIT, or when a file was entered into a catalogue, a filler of blanks is typed out whenever fewer than the maximum number of characters were entered originally. This is due to the limitations of A-format in FORTRAN.

3. Care must be taken when entering numbers as responses to questions or when entering data values. An invalid character such as a letter (if not corrected before typing the RETURN key) causes a Fortran system fatal error and control returns to the OS/8 Monitor immediately.

APPENDIX A

RESPONSE FORMAT

It is sometimes necessary to respond to a Teletype question by typing one or more integers and/or one or more real numbers. This kind of response is indicated at the end of a question by

```
--I4,F6.Ø
```

for example; or

```
--I1
```

In any event, the letter I means that an integer is expected and the number following the I indicates the maximum number of characters which may be typed to enter the integer. In like manner the letter F means an integral or real number may be entered. The number following F and preceding the decimal point is the maximum number of characters (including a decimal point if one is used), that can be entered.

NOTE

1. A comma, as in the first example above means: type a space, comma or any character to separate the numbers being entered.

2. Spaces entered at the Teletype are interpreted as zeros and if input is terminated before the maximum number of characters is entered, the remaining characters are supplied as zeros.

3. A carriage return terminates a line of input. However, it is not counted as a character. For example, I4 allows four characters and a carriage return.

Examples illustrating each type of user input are given below:

FORMAT	USER INPUT	PROGRAM INTERPRETATION
I4	ØØ35↵	35
	␣␣35↵	35
	␣35.↵	35
	35.Ø↵	35
	35.↵	35
	35↵	35ØØ

FORMAT	USER INPUT	PROGRAM INTERPRETATION
I1	6↵	6
	↵	Ø
F3.Ø	.2↵	.2
	Ø.2↵	.2
	␣Ø.2↵	Ø.Ø
I4,I4	␣␣12␣␣␣␣7↵	12 & 7
	␣12.␣7.ØØ↵	12 & 7
	12.Ø␣ØØØ7↵	12 & 7
	12.Ø,ØØØ7↵	12 & 7
	127↵	127Ø & Ø
	12.Ø␣7↵	12 & 7ØØØ
	12.Ø␣7.↵	12 & 7
I4,F6.Ø	␣123␣567.3↵	123 & 567.3
	123.␣␣567.3↵	123 & 567.3
	␣123,␣␣567.3↵	123 & 567.
F3.Ø,I2	1.3␣␣8↵	1.3 & 8
	1.3␣8.↵	1.3 & 8
	1.3␣8↵	1.3 & 8Ø
	␣1.3␣8↵	1. & 8

NOTE

The form in which a number is entered to a program greatly affects the way it is stored. For example, with an I4, F6.Ø format, the value of the numbers entered (24 and 6 for example) can vary from 24 to 2400 and 6 to 6ØØ,ØØØ (depending on the spaces and punctuation used).

APPENDIX B

ASSEMBLY INSTRUCTIONS

Assemble the program sources (if available) as follows:

TRANS.PA

```
.R PAL8↵
*TRANS.BN, LPT:<TRANS.PA↵
```

PRINT.FT

```
.R  F4↵
*PRINT.RL<PRINT.FT/L↵
```

ALOCAT.PA

```
.R PAL8↵
*ALOCAT.BN,LPT:<ALOCAT.PA↵
```

LIBMSH.FT[1]

```
.R F4↵
*LIBMSH.RL,LPT:<LIBMSH.FT/L/N↵
```

ENTER.FT[1]

```
.R F4↵
*ENTER.RL,LPT:<ENTER.FT/L/N↵
```

INIT.FT[1]

```
.R F4↵
*INIT.RL,LPT:<INIT.FT/L/N↵
```

INTERP.FT[1]

```
.R F4↵
*INTERP.RL,LPT:<INTERP.FT/L/N↵
```

XLIST.FT[1]

```
.R F4↵
*XLIST.RL,LPT:<XLIST.FT/L/N↵
```

DELETE.FT[1]

```
.R F4↵
*DELETE.RL,LPT:<DELETE.FT/L/N↵
```

[1] This program must be compiled using the L and N switches. The line numbers of the program were suppressed to enable the program to meet the core requirement without further expansion of the overlaying scheme.

SQUISH.FT[1]

```
.R F4)
*SQUISH.RL,LPT:<SQUISH.FT/L/N)
```

MOVE.FT[1]

```
.R F4)
*MOVE.RL,LPT:<MOVE.FT/L/N)
```

NEXT.FT[1]

```
.R F4)
*NEXT.RL,LPT:<NEXT.FT/L/N)
```

GETBLK.FT[1]

```
.R F4)
*GETBLK.RL,LPT:<GETBLK.FT/L/N)
```

PUTBLK.FT[1]

```
.R F4)
*PUTBLK.RL,LPT:<PUTBLK.FT/L/N)
```

COMPAR.FT[1]

```
.R F4)
*COMPAR.RL,LPT:<COMPAR.FT/L/N)
```

STRNG.PA

```
.R RALF)
*STRNG.RL,LPT:<STRNG.PA/L)
```

CCGET.PA

```
.R RALF)
*CCGET.RL,LPT:<CCGET.PA/L)
```

Refer to the appropriate sections of this manual for information on saving and/or loading these binary programs.

[1] See footnote on previous page.

APPENDIX C

INTERNAL DOCUMENTATION

The internal documentation for LIBMSH is provided for Software Support personnel and advanced programmers who wish to better understand the operation of these programs.

The description of each program includes an explanation of the program listing. The array dimensions are shown in parentheses after the array names. For programs assembled with PAL8, explanations are keyed to statement labels. FORTRAN program explanations are keyed to statement numbers plus or minus the number of lines after or before which a routine begins. Comment lines are not counted.

C.1 TRANSFER PROGRAM

TRANS.PA is an assembly language program which assembles with PAL8. Its purpose is to convert an AIPOS file (2 word - double precision - data) into an OS/8 compatible file (3-word - floating point - data).

<u>Label</u> <u>Explanation</u>

BEGIN: The USR is brought into core; the command decoder is called; the user specified output file is entered on his device.

MESS: Three questions and answers follow:

IUNIT holds a number Ø-7 indicative of the LINCtape unit which holds the AIPOS file.

RSB contains the starting block (HEADER BLK) of the AIPOS file to be transferred.

IFL contains (length+1) of AIPOS file.

The Header block is read and checked to see if the AIPOS file is a MASS/Intensity MASH file (WORD2=4Ø5).

WTR: AIPOS data is read into 2ØØØ-2377, a block at a time.

The data is converted to floating point and entered into the output buffer at 6ØØØ-6377. The transfer is complete when two 37777777's in a row are encountered in the AIPOS file.

Label	Explanation
DONE:	Write out the last block of floating point data, close the file, and go to BEGIN.
READ:	Sets up to read a LINCtape block (maximum 16ØØ), does the read and sets IA to starting address -1 of AIPOS buffer.
FIXUP:	This routine is entered only if the last block of an AIPOS file being transferred does not contain the two 37777777's.
RDHEAD:	Referred to earlier when the Header block of AIPOS file was being checked.
TTYO:	This outputs an ASCII message.
TTYI:	This accepts numeric input along with CTRL/C and RETURN key.
WRITE:	The floating point data in the output buffer is written to the OS/8 file.

C.2 PRINT PROGRAM

PRINT.FT outputs the results of the Transfer program. The MASS times 1Ø [MASS*1Ø] values and the corresponding INTENSITY values are listed on the line printer in two columns.

Arrays used by program	Explanation
ARRAY(85)	Contains a block of data read from the device specified.
B(84)	Contains an equal number of MASS(*10)/ Intensity pairs to be output to the line printer.
Logical unit:	
5	Assigned to the file in question at LOAD time.

C.3 ALLOCATE PROGRAM

ALLOCAT.PA is an assembly language program which assembles with PAL8. Its purpose is to open a file of user-specified length, unit and name.

The USR is brought into core; the command decoder called, the output file opened and then closed.

C.4 LIBMSH PROGRAM

This program is made up of fourteen separate modules, twelve written in FORTRAN and two written in relocatable assembly language code.

Logical Units:

5 - assigned to a converted floating point MASH file
6 - assigned to the only required CATALOGUE
7 - assigned to the optional CATALOGUE

Arrays used by program — Explanation

SLOTS(85) — Contains one block of catalogue information which is divided into four - 21 floating-point-word entries with the 85th floating-point-word ignored.

HEADER(85) — Contains the Header block information of a given catalogue.

Floating Point Wd.	Contents	Description
1	INIT FLG	1934 if initialized
2	integer	number of next open entry slot in the file
3-4	name of catalogue	12 characters maximum
5-6	Date, etc.	12 characters maximum
7-17	User message	66 characters maximum
18-85	unused	

ENTRY(21) — Contains all information necessary for one entry into a given catalogue.

Floating Point Word	Contents
1-5	NAME, FORMULA, etc.
6-21	8 M/I pairs found by ENcoding.

COMLST(3) — Contains eight two-letter command codes to be used by LIBMSH.FT when interpreting a function call.

The Fourteen Modules:

Reference to specific lines of FORTRAN code are made relative to statement numbers with comment lines not being counted. For example, 46+2 means the second line of code after the statement number 46.

C.4.1 LIBMSH.FT is a FORTRAN program which is essentially a command decoder.

Program Statement Numbers	Explanation
2Ø-3	KOUNT - initialize to Ø so that NEXT.FT will work correctly the first time it is called.
	IDEV - set to 6 (at other places and in other routines sometimes set to 7). Used in random access I/O to refer to logical unit 6 (or 7).
2Ø-1	A check is made to see if the catalogue on logical unit 6 has been initialized. Next, the command is requested, interpreted and then executed.
1Ø	An illegal command returns here.

C.4.2 ENTER.FT is a FORTRAN program which enters a file into a catalogue on logical unit 6 as well as searches a catalogue for one or more files that compare with the given file.

Arrays used by Program	Explanation
NAME(5)	these five floating point words are equivalent to the first five floating point words of ENTRY(21).
PKLST(16)	these 16 floating point words are equivalent to words 6-21 of ENTRY(21).

Statement Numbers	Explanation
11-2	Read the Header block of logical unit 6. Determine type of encode. Clear PKLST array.
14	Manual encode. User enters a maximum of 8 Mass/Intensity peaks.
20	Automatic encode. After entering the MASS tolerance (DELTM), the eight largest peaks from the MASH file are found.
201	Initialize a set of constants.
220	NEXT.FT is called to obtain the next Mass/Intensity pair with a check made for end of file. The MASS is divided by 10 since the MASS in the file is a multiple of 10. IMASS - real mass is integerized FRACT - decimal part of real mass XFRACT=1-FRACT

Statement Numbers	Explanation
220+6	Find all masses which satisfy the tolerance, DELTM. Enter Mass/Intensity pair into PKLST(16) if the intensity is greater than one which is found in the list, and eliminate smallest intensity pair.
25	Change the eight intensities to a % relative to the highest one and then report the auto encode.
4Ø	Do a Search if FLGLST=1.
4Ø+1	Call COMPAR which is the subroutine which does the search.
67	Entry is now placed in Catalogue upon user request.
72	The entry may replace a given Catalogue entry.
80	Entry is placed in catalogue; updated header block (word 2) of catalogue is written out to the file catalogue.

C.4.3 INTERP.FT is a FORTRAN subroutine that compares a two-letter function command, which is passed to it, with the list of eight valid commands found in COMLST. The subroutine CCGET is called.

INTERP returns one of the first eight digits to the calling program LIBMSH.FT, as a code number signaling which command was requested. An illegal command returns code number 9.

C.4.4 XLIST.FT is a FORTRAN subroutine callable by the XLn command (FLGLST=Ø) or by the CLn command (FLGLST=1).

Statement Numbers	Explanation
3-3	If the catalogue to be listed is on logical device 7, a check is made to see if it has been initialized.
3+1	Check to see if any entries are in catalogue.
10	Request the set of contiguous entry numbers to be used in the listing. The two limit numbers are checked for validity and readjusted if necessary. If the high limit is Ø or is greater than the last entry in the catalogue its value defaults to the last entry number. If the low limit is Ø its value defaults to 1. If the low limit > high limit, the low limit takes on the value of the high limit.

Statement Number	Explanation
2Ø-2	Check FLGLST for choice of type of list.
2Ø	XLn command-listing of Header block information.
3Ø	CLn & XLn command-listing of 30 characters maximum information for each entry.
33	CLn command-listing of Mass/Intensity pairs.

C.4.5 DELETE.FT is a subroutine which deletes one (21 floating point) entry in a catalogue by replacing all 21 words with Ø.Ø.

Statement Number	Explanation
3-3	If the file to be deleted is on logical unit 7, a check ıs maԁe to see if it has been initialized.
3+1	Check to see if any entries are in catalogue.
9+1	Accept entry numbers to be deleted, check to see if NUM is zero which signals return to CMD . Check to see if NUM is too big.
31	If NUM = last entry number, decrement word(2) of the header block. A further check is made to see if the next entry had been previously deleted. If so, decrement word(2) etc. This sequence moves the pointer to the first available entry slot.
30	Replace the entry with Ø's.
15+1	Put entry back into catalogue and restore updated header block.

C.4.6 SQUISH.FT is a subroutine which eliminates all empty entries previously caused by deletions.

Statement Number	Explanation
3+2	Set up some variables IEND = last entry in catalogue. NBLK = number of blocks containing all entries from first to last. IA = current block of catalogue to be read IB = current block of catalogue to be written to during the squish. JEND = counts four entries per block.
10-8	Squeeze the four entries of the current block in core, eliminating zeroed entries.

Statement Number	Explanation
10+1	Write out a block of squeezed entries when the block has four non-zeroed entries.
10+4	LF signals the need (=1) to write out a partially filled block when the SQUISH is done.
25	Update header information and write it out to the catalogue.

C.4.7 MOVE.FT is a subroutine used to move a contiguous set of entries (1,2,3,...,All) from the catalogue on logical unit 6 to the catalogue on logical unit 7.

Arrays used by Programs	Explanation
SL(85)	contains four entries from catalogue on logical unit 7
SLOTS(85)	contains four entries from catalogue on logical unit 6.
HEAD(85)	contains the HEADER information for the catalogue on logical unit 6.
HEADER(85)	contains the header information for the catalogue on logical unit 7.

Statement Number	Explanation
100-1	Request the set of contiguous entry numbers to be used in the move. The two limit numbers LIMLO & LIMHI are checked for validity.
10-2	Set up some variables. KSTOP = number of entries to move. FIRST6 = current block to read. LIMLO = reset to a number from 1 to 4. Keeps track of current entry of current block in core from logical unit 6. FIRST = set to a number from 1 to 4. Keeps track of current entry of current block in core from logical unit 7. FIRST7 = current block to write. KBEGIN = points to first word of current entry in output buffer KSTART = points to first word of current entry in input buffer KEND = points to last word of current entry in input buffer.
26+1	Transfer one entry to output buffer. Update Header Block of output file. Check need to write out buffer if four entries are present. Check if job is done (KSTOP=Ø).

Statement Number	Explanation
55	Set pointer to next input entry.
59	Job is done. Write out last block. Write out updated header block.

C.4.8 COMPAR.FT is a subroutine called by ENTER.FT and is used to search all files on logical unit 6 for one or more matches with a given entry in ENTRY(21).

Statement Number	Explanation
41	User specified tolerances. DELTI - Itensity tolerance NUMPKS - number of peaks which must match in order to report a valid compare.
43	Output of hard copy heading to be used when reporting any matches.
43+2	FLAG = set to 1 if one or more matches were found IEND = number of last entry in catalogue
43+4	Reads one catalogue entry at a time into SLOTS(85).
43+9	Takes one peak at a time from ENTRY(21) and compares with each of the eight peaks of an entry in SLOTS(85).
57-1	Reports a match when found.

C.4.9 INIT.FT is a subroutine which checks the catalogue on logical unit 6 to make sure it has been initialized (FLG=Ø), and then forces an automatic initialization if it has not. This routine is also called to initialize a catalogue (FLG=1) upon user request.

Statement Number	Explanation
3-3	See if an automatic check is to be made on logical unit 6 (FLG=Ø). Otherwise determine which logical unit (IDEV=6 or 7) has been requested for initialization.
3+1	See if catalogue is presently initialized HEADER(1)=1934.
7-1	Indicates that the catalogue is being initialized for the first time.
8	Set word 1 of header block of catalogue to 1934. Set word 2 of header block to a 1.

Statement Number	Explanation
10	Accept user specified information.
5-1	Determine if Initialization is upon request (FLG=1).
5	Place all Header information in array to be written out.
200+1	Write out Header block.
20	If Initialization is upon request, give user a chance to back out before doing the initialization on his catalogue.

C.4.10 NEXT.FT is called by ENTER. This routine fetches the next floating-point number (either a MASS or INTENSITY) value from a given MASS file which is on logical unit 5.

Arrays used by Program	Explanation
ARRAY(85)	Contains one block of floating-point numbers read from a MASH file.

Statement Number	Explanation
10-3	KOUNT - is a variable which indicates which of the 85 floating-point numbers in ARRAY(85) is to be fetched. It is initially set to Ø by LIBMSH, and thereafter set to zero whenever a new block of data is to be read.

C.4.11 GETBLK.FT is a routine which effectively gets a specified entry (NENT) in a catalogue (NDEV), by actually reading into SLOTS(85) the entire block containing the entry. This routine returns to the caller the starting location (LOC) within SLOTS where the requested entry begins.

C.4.12 PUTBLK.FT is a routine which works like GETBLK and puts an entry into a catalogue.

C.4.13 CCGET.PA is an assembly language program which gets the N^{th} character (6-Bit) in a string, and stores it in F as a normalized number between Ø and 63.

C.4.14 STRNG.PA is an assembly language program which enables entry of an ASCII string of characters in groups of six. If fewer than the maximum number of characters is entered, blanks are automatically filled in on the right.

Labels — Explanation

STRNG — The return address is saved; base page and index registers are reassigned; the arguments of the subroutine call are processed.

SETINT — Since the routine works off the FORTRAN interrupt chain, it is necessary to set up the 4-locations for the interrupt. It is necessary to save, and later restore, the first of the 4-locations (#INT). The field and address of the string of characters is set up for LFRT.

A set of locations are cleared.

FLGDN=1 when the last character has been entered and it is time to return to the calling program.

FLG=Even number if the right 6-bit character of a word in the string is requested.

FLG=Odd number if the left 6-bit character is requested.

```
FLGCHR = negative number signals job is done
       = Ø   signals Teletype input
       = 1   signals time to issue carriage
             return
       = 2   signals time to issue line feed
```

STRING — Checks interrupt flags for a previous echoed character or a current input character. If an echoed character caused interrupt, a check is made (FLGCHR) whether to output line feed, carriage return, or service Teletype.

CRTN — This routine is called only when the user exceeded the maximum number of input string characters before issuing a carriage return.

LPTTYI — This routine services the keyboard.

BLANKS — Come here if user issued carriage return before the maximum number of characters had been entered. Blanks are filled in the string to the right.

LFRT — Depending on the contents of FLG (ODD or EVEN) the left or right 6-bit character of a word in the string is stored with the keyboard character.

```
          C      DEC-12-AMLBA-A-LD
          C      COPYRIGHT 1972
          C      DIGITAL EQUIPMENT CORPORATION
          C      MAYNARD, MA 01754
          C      -----LIBMSH.FT-----
          C      THIS IS THE MAIN PROGRAM FOR THE
          C      MASH LIBRARY PROGRAM.
          C   CALLING SEQ: R LOAD
          C                   LIBMSH/D
          C                   SPECTR/5/C      SPECT TO INTERACT WITH CATELOG
          C                   CATALOG1/6/C    GIVEN CATALOGUE NAME
          C                   CATALOG2/7      OPTIONAL-USE WHEN DOING A MOVE
          C      COMAND DECODER
0002             COMMON SLOTS(85),HEADER(85),ENTRY(21),COMLST(3),KOUNT
0003             COMMON/A/IDEV
0004             DATA COMLST/'XLCLDE','SQINEN','SEMO  '/
0005             DEFINE FILE 6(1,85,U,LV1)
0006             KOUNT=0
0007             IDEV=6
0010             CALL INIT(0)
0011      20     WRITE(4,100)
0012      100    FORMAT(//1X,'CMD'/)
0013             READ(4,110) COMD
0014      110    FORMAT(A3)
0015             CALL INTERP(8,COMD,ICODE)
0016             GO TO(9,9,9,9,9,6,7,8,10),ICODE
0017      1      CALL XLIST(0)
0020             GO TO 20
          C-----THIS IS THE 'CLIST' COMD
0021      2      CALL XLIST(1)
0022             GO TO 20
0023      3      CALL DELETE
0024             GO TO 20
0025      4      CALL SQUISH
0026             GO TO 20
0027      5      CALL INIT(1)
0030             GO TO 20
0031      6      CALL ENTER(0)
0032             GO TO 20
          C-----THIS IS THE 'SEARCH COMD'
0033      7      CALL ENTER(1)
0034             GO TO 20
0035      8      DEFINE FILE 7(1,85,U,LV2)
0036             IDEV=7
0037             CALL INIT(0)
0040             CALL MOVE
0041             GO TO 20
0042      9      CALL CCGET(COMD,3,IDEV)
0043             IDEV=IDEV-48
0044             IF(IDEV.LT.6.OR.IDEV.GT.7)GO TO 10
0045             GO TO(1,2,3,4,5),ICODE
          C-----THIS IS THE ERROR RTN-----
0046      10     WRITE(4,22)
0047      22     FORMAT(' CMD ERR'/)
0050             GO TO 20
```

```
ØØ51              END
```

```
                /       DEC-12-AMLBA-A-UD
                /       COPYRIGHT 1972
                /       DIGITAL EQUIPMENT CORPORATION
                /       MAYNARD, MA 01754

                /       -----STRNG.FA-----
                /
                /       THIS IS A DSB ROUTINE THAT IS CALLABLE
                /       FROM FORTRAN. IT ENABLES ONE TO ENTER
                /       AN ASCII STRING OF CHARACTERS IN GROUPS
                /       OF SIX (6).
                /
                /       CALLING SEQ.  CALL STRNG(MAX,ARRAY) WHERE

                /       MAX=6*DIMENSION OF ARRAY
                /       ARRAY=ADDRESS OF WHERE TO PACK THE CHARS
                /
                /
                /

                        COMMZ GLOB
00000 1030              JA      STRNG
00001 0040
00002 2324              TEXT    +STRNG+
00003 2216
00004 0700
00005 0000      BSTRNG, 0;0;0
00006 0000
00007 0000
00010 0000      X,      0;0;0
00011 0000
00012 0000
00013 0000      Y,      0
00014 0000      AFLD,   0
00015 0000      ARPTR,  0
00016 0000      XSTRNG, 0;0;0
00017 0000
00020 0000
                        ORG BSTRNG+30
00035 0040      GOBAK,  FNOP;JA .
00036 1030
00037 0036
                        ENTRY STRNG
                        BASE 0
00040 0006      STRNG,  STARTD
00041 0210              FLDA 30
00042 6400              FSTA    GOBAK
00043 0035
00044 0200              FLDA    0
00045 1100              SETX    XSTRNG
00046 0016
00047 1110              SETB    BSTRNG
00050 0005
                        BASE BSTRNG
00051 0101              LDX     1,1
00052 0001
00053 6200              FSTA    BSTRNG
```

```
ØØØ54 Ø61Ø              FLDA%   BSTRNG,1        /ADR OF MAX
ØØØ55 62Ø1              FSTA    X
ØØØ56 Ø71Ø              FLDA%   BSTRNG,1+       /ADR OF ARRAY
ØØØ57 62Ø2              FSTA    Y
ØØØ6Ø ØØØ5              STARTF
ØØØ61 Ø6Ø1              FLDA%   X               /GET MAXNUM
ØØØ62 ØØ2Ø              ATX Ø
ØØØ63 4ØØØ              TRAP4   SETINT
ØØØ64 ØØ7Ø
ØØØ65 ØØØ5              STARTF
ØØØ66 1Ø3Ø              JA      GOBAK
ØØØ67 ØØ35
ØØØ7Ø ØØØØ    SETINT,   Ø
ØØØ71 62Ø1              CDF Ø
ØØØ72 1744              TAD%    INTADR+1        /SAVE C(#INT)
ØØØ73 3352              DCA     INTØ
ØØØ74 1351              TAD     ASKP            /STORE 'SKP'
ØØØ75 3744              DCA%    INTADR+1        /AT #INT
ØØØ76 2344              ISZ     INTADR+1
ØØØ77 1346              TAD     JOB+1           /ADR OF 'STRING'
ØØ1ØØ 3744              DCA%    INTADR+1
ØØ1Ø1 2344              ISZ     INTACR+1
ØØ1Ø2 1345              TAD     JOB             /FLD OF 'STRING'
ØØ1Ø3 71Ø6              CLL RTL
ØØ1Ø4 7ØØ4              RAL
ØØ1Ø5 135Ø              TAD     AFIEL
ØØ1Ø6 3744              DCA%    INTADR+1
ØØ1Ø7 1216              TAD     XSTRNG          /GET MAXNUM
ØØ11Ø 6211              CDF 1Ø
ØØ111 3756              DCA%    MAXN+1          /MAXCNT
ØØ112 1756              TAD%    MAXN+1
ØØ113 7Ø41              CIA
ØØ114 376Ø              DCA%    MAXC+1
ØØ115 1214              TAD     AFLD            /FLD OF ARRAY
ØØ116 Ø371              AND     P7
ØØ117 71Ø6              CLL RTL
ØØ12Ø 7ØØ4              RAL
ØØ121 1347              TAD     ACDF
ØØ122 3766              DCA%    CDFAR+1
ØØ123 1215              TAD     ARPTR
ØØ124 3754              DCA%    ARPT+1
ØØ125 3764              DCA%    FLGD+1
ØØ126 3762              DCA%    FL+1
ØØ127 377Ø              DCA%    FLGCH+1
ØØ13Ø 1764              TAD%    FLGD+1
ØØ131 765Ø              SNA CLA
ØØ132 533Ø              JMP     .-2
ØØ133 62Ø1              CDF Ø
ØØ134 1372              TAD     M2
ØØ135 1344              TAD     INTADR+1        /RTN PTR TO #INT
ØØ136 3344              DCA     INTADR+1
ØØ137 1352              TAD     INTØ            /RESTORE C(#INT)
ØØ14Ø 3744              DCA%    INTADR+1
ØØ141 62Ø3              CDF CIE
ØØ142 5Ø7Ø              JMP%    SETINT
```

```
                          EXTERN #INT
ØØ143 ØØØØ  INTADR, ADDR #INT
ØØ144 ØØØØ
ØØ145 ØØØØ  JOB,    ADDR STRING
ØØ146 Ø2ØØ
ØØ147 62Ø1  ACDF,   62Ø1
ØØ15Ø 62Ø3  AFIEL.  62Ø3
ØØ151 741Ø  ASKP,   741Ø
ØØ152 ØØØØ  INTØ,   Ø
ØØ153 ØØØØ  ARPT,   ADDR APTR
ØØ154 Ø3Ø7
ØØ155 ØØØØ  MAXN,   ADDR MAXNUM
ØØ156 Ø3Ø2
ØØ157 ØØØØ  MAXC,   ADDR MAXCTR
ØØ16Ø Ø3Ø3
ØØ161 ØØØØ  FL,     ADDR FLG
ØØ162 Ø3Ø4
ØØ163 ØØØØ  FLGD,   ADDR FLGON
ØØ164 Ø3Ø5
ØØ165 ØØØØ  CDFAR,  ADDR CDFARY
ØØ166 Ø322
ØØ167 ØØØØ  FLGCH,  ADDR FLGCHR
ØØ17Ø Ø3Ø1
ØØ171 ØØØ7  P7,     7
ØØ172 7776  M2,     -2
                    ORG 2ØØ


ØØ2ØØ ØØØØ  STRING, Ø
ØØ2Ø1 6Ø41          TSF                     /CHK 'OUTPUT' FLG
ØØ2Ø2 5225          JMP     LPTTYI          /NOT SET
ØØ2Ø3 6Ø42          TCF                     /SET, CLR IT
ØØ2Ø4 13Ø1          TAD     FLGCHR          /FLG WAS DUE TO
ØØ2Ø5 745Ø          SNA                     /PREVIOUS ECHO CHAR?
ØØ2Ø6 5225          JMP     LPTTYI
ØØ2Ø7 751Ø          SPA
ØØ21Ø 5351          JMP     DONE
ØØ211 1314          TAD     M1
ØØ212 775Ø          SPA SNA CLA
ØØ213 5221          JMP     CRTN            /ISSUE A CR
ØØ214 131Ø          TAD     P212            /ISSUE LF
ØØ215 6Ø46          TLS
ØØ216 724Ø          CLA CMA
ØØ217 33Ø1          DCA     FLGCHR
ØØ22Ø 5353          JMP     RETURN
            /WILL COME HERE ONLY WHEN USER ISSUED NO CR

ØØ221 1312  CRTN,   TAD     P215
ØØ222 6Ø46          TLS
ØØ223 33Ø1          DCA     FLGCHR
ØØ224 5353          JMP     RETURN
ØØ225 6Ø31  LPTTYI, KSF
ØØ226 5353          JMP     RETURN
ØØ227 6Ø36          KRB
ØØ23Ø 33Ø6          DCA     TMP
```

```
00231 1306           TAD     TMP
00232 1315           TAD     M377                /TEST FOR RUBOUT
00233 7450           SNA
00234 5256           JMP     RUBOUT
00235 1313           TAD     CR
00236 7650           SNA CLA                     /TEST FR CR
00237 5246           JMP     BLANKS
00240 1306           TAD     TMP
00241 6046           TLS                         /ECHO CHAR
00242 0317           AND     P77
00243 3306           DCA     TMP
00244 4321           JMS     LFRT
00245 5353           JMP     RETURN
00246 1311 BLANKS,   TAD     P40
00247 3306           DCA     TMP
00250 1312           TAD     P215
00251 6046           TLS
00252 7200           CLA
00253 2301           ISZ     FLGCHR
00254 4321           JMS     LFRT
00255 5254           JMP     .-1

00256 1302 RUBOUT,   TAD     MAXNUM
00257 1303           TAD     MAXCTR
00260 7750           SPA SNA CLA                 /OK TO RUBOUT?
00261 5353           JMP     RETURN              /NO, AT TOP OF LIST
00262 1316           TAD     P334                /YES, ECHO BACKSLASH
00263 6046           TLS
00264 7240           CLA CMA
00265 1303           TAD     MAXCTR              /BACKUP CTR
00266 3303           DCA     MAXCTR
00267 1304           TAD     FLG
00270 7110           CLL RAR
00271 7630           SZL CLA                     /LEFT 6BIT CAR?
00272 5276           JMP     .+4                 /YES
00273 7040           CMA                         /NO, BACKUP PTR
00274 1307           TAD     APTR
00275 3307           DCA     APTR
00276 2304           ISZ     FLG
00277 7000           NOP
00300 5353           JMP     RETURN

00301 0000 FLGCHR,   0
00302 0000 MAXNUM,   0
00303 0000 MAXCTR,   0
00304 0000 FLG,      0
00305 0000 FLGDN,    0
00306 0000 TMP,      0
00307 0000 APTR,     0
00310 0212 P212,     212
00311 0040 P40,      40
00312 0215 P215,     215
00313 0162 CR,       377-215
00314 7777 M1,       -1
00315 7401 M377,     -377
00316 0334 P334,     334
```

```
00317 0077  P77,      77
00320 7700  P7700,    7700


00321 0000  LFRT,     0
00322 0000  CDFARY,   0
00323 2304            ISZ     FLG
00324 7000            NOP
00325 1304            TAD     FLG
00326 7110            CLL RAR
00327 7620            SNL CLA             /LF CHAR?
00330 5337            JMP     R1                    /NO
00331 1306  LF,       TAD     TMP
00332 7106            CLL RTL
00333 7006            RTL
00334 7006            RTL
00335 3707            DCA%    APTR
00336 5344            JMP     RTN
00337 1707  R1,       TAD%    APTR
00340 0320            AND     P7700
00341 1306            TAD     TMP
00342 3707            DCA%    APTR
00343 2307            ISZ     APTR
00344 6211  RTN,      CDF 10
00345 2303            ISZ     MAXCTR
00346 5721            JMP%    LFRT
00347 2301            ISZ     FLGCHR
00350 5353            JMP     RETURN
00351 2305  DONE,     ISZ     FLGDN
00352 7200            CLA
00353 6203  RETURN,   CIF CDF
00354 5600            JMP%    STRING
```

```
RALF V  51  MAY 11, 72 PAGE 2

NO ERRORS
54 SYMBOLS, NO ABS REFS

 #        C 00000   #INT    X 00000   #MAIN  S 00000   ACOF       00147
 AFIEL      00150   AFLD      00014   APTR     00307   ARPT       00153
 ARPTR      00015   ASKP      00151   BLANKS   00246   BSTRNG     00005
 CDFAR      00165   CDFARY    00322   CR       00313   CRTN       00221
 DONE       00351   FL        00161   FLG      00304   FLGCH      00167
 FLGCHR     00301   FLGD      00163   FLGDN    00305   GLOB     Z 00355
 GOBAR      00035   INTADR    00143   INT0     00152   JOB        00145
 LF         00331   LFRT      00321   LPTTYI   00225   MAXC       00157
 MAXCTR     00303   MAXN      00155   MAXNUM   00302   M1         00314
 M2         00172   M377      00315   P212     00310   P215       00312
 P334       00316   P40       00311   P7       00171   P77        00317
 P7700      00320   RETURN    00353   RT       00337   RTN        00344
 RUBOUT     00256   SETINT    00070   STRING   00200   STRNG      00040
 TMP        00306   X         00010   XSTRNG   00016   Y          00013
```

```
                  /       DEC-12-AMLBA-A-UO
                  /       COPYRIGHT 1972
                  /       DIGITAL EQUIPMENT CORPORATION
                  /       MAYNARD, MA Ø1754

                  /       -----CCGET.PA-----
                  /       CHARACTER HANDLING ROUTINES FOR FORTRAN
                  /
                  /       CALLING SEQUENCE:
                  /       CALL CCGET(STR,N,F)
                  /               GETS THE NTH CHARACTER (6-BIT) IN STR, AND STORES
                  /               IT IN F AS A NORMALIZED NUMBER BETWEEN Ø AND 63.
                  /
                  /
                  /               COLLATING SEQUENCE IS STRIPPED ASCII.
                  /
                          SECT    CH
                          ENTRY   CCGET
ØØØØØ ØØ31Ø              TEXT    +CH+
ØØØØ1 11ØØ      RET,     SETX    XØ
ØØØØ2 ØØ43
ØØØØ3 111Ø               SETB    BASE
ØØØØ4 ØØØ5
ØØØØ5 ØØ4Ø      BASE,    FNOP;JA .
ØØØØ6 1Ø3Ø
ØØØØ7 ØØØØ
ØØØ1Ø ØØØØ      N,       F       Ø
ØØØ11 ØØØØ
ØØØ12 ØØØØ
ØØØ13 ØØØØ      F,       F       Ø
ØØØ14 ØØØØ
ØØØ15 ØØØØ
ØØØ16 ØØØØ      KSETX,   Ø;1Ø77;7777     /BECOMES SETX TO STR(N/2)
ØØØ17 1Ø77
ØØØ2Ø 7777
ØØØ21 ØØØØ      KSGN15, Ø;1Ø;Ø           /STRIP THE SIGN FROM 15-BIT ADDRESS
ØØØ22 ØØ1Ø
ØØØ23 ØØØØ
ØØØ24 ØØØØ      KSGN12, Ø;1;Ø            /12 BIT ADDR
ØØØ25 ØØØ1
ØØØ26 ØØØØ
ØØØ27 ØØØØ      KSGN6,   Ø;Ø;1ØØ         /LIKEWISE FOR CHAR
ØØØ3Ø ØØØØ
ØØØ31 Ø1ØØ
```

```
                        ORG     BASE+30 /RETURN, TRACEBACK INFO
00035 0040              FNOP;JA RET
00036 1030
00037 0001
00040 0040              FNOP
00041 1030    GOBACK,   JA      .       /RETURN TO CALLER'S BASE
00042 0041
              /
              /         THERE ARE TWO SETS OF INDEX REGISTERS, WHICH MUST JOIN
              /         AT REMAIN; THAT IS, REMAIN MUST BE IN BOTH SETS.
              /         IN ADDITION, XTMP1 AND XTMP2 MUST BE THE HIGH
              /         AND LOW FRACTION WORDS, RESPECTIVELY, OF ONE
              /         BASE PAGE LOCATION.
              /
00043 0000    X0,       0       /UNUSED
00044 0001    X1,       1       /FOR GETTING 1ST PARAM
00045 0002    X2,       2       /FOR GETTING 2ND
00046 0003    X3,       3       /FOR GETTING 3RD
00047 0014    K14,      14      /SHIFT COUNTER
00050 7767    KM11,     -11     /ME, TOO
00051 0011    K11,      11      /ALSO A SHIFTER
              XX,               /BEGIN OF SECOND SET
              /         REMAIN MUST BE IN BOTH SETS OF INDEX REGISTERS
00052 0000    REMAIN,   0       /NON-ZERO IF N ODD
00053 0006    K6,       6
00054 7772    KM6,      -6
00055 7764    KM14,     -14
00056 0022    K22,      22
00057 0000              0       /OBTAIN ALIGNMENT
00060 0000    XTMP1,    0       /MUST BE FIRST WORD OF FRACTION
00061 0000    XTMP2,    0       /MUST BE SECOND
```

```
                  /       COMMON CODE FOR CHAR ROUTINES
                  /
                          BASE    0
00062 1030  PROLOG,       JA      .         /SUBR ENTRY
00063 0062
00064 0006                STARTD
00065 0210                FLDA    30
00066 6400                FSTA    GOBACK
00067 0041
00070 0200                FLDA    0
00071 1100                SETX    X0
00072 0043
                          INDEX   X0
00073 1110                SETB    BASE
00074 0005
                          BASE    BASE
00075 6200                FSTA    BASE      /CALL POINTER
00076 0620                FLDA%   BASE,X2 /ADDR OF SUBSCRIPT
00077 6201                FSTA    N
00100 0630                FLDA%   BASE,X3 /SCALAR RESULT
00101 6202                FSTA    F
00102 0005                STARTF
00103 0601                FLDA%   N         /GET N VALUE
00104 0010                ALN     X0        /STRIP ANY FRACTION
00105 0014                ALN     K14       /BINARY POINT FOLLOWS BIT 12
00106 0006                STARTD
00107 0027                ATX     REMAIN    /SAVE ODD-EVEN BIT
00110 0014                ALN     K14       /SHIFT RIGHT 12
00111 1010                JGE     .+3       /N MUST BE POSITIVE
00112 0114
00113 1205                FADD    KSGN12    /SO CLEAR SIGN
00114 6400                FSTA#   T1        /SAVE TO TEST
00115 0145
00116 0037                XTA     REMAIN    /CHECK ODD EVEN
00117 1000                JEQ     T2        /JUMP IF EVEN
00120 0123
00121 0400                FLDA#   ONE       /INCREMENT ADDRESSS
00122 0147
00123 1400  T2,           FADD#   T1        /CONTINUE FOR BOTH
00124 0145
00125 1610                FADD%   BASE,X1 /ADD ADDRESS OF STRING
00126 0015                ALN     KM11      /SHIFT LEFT 9
00127 0016                ALN     K11       /AND RIGHT AGAIN
00130 1010                JGE     .+3       /IF NECESSARY
00131 0133
00132 1204                FADD    KSGN15    /REMOVE SIGN BITS
00133 1203                FADD    KSETX     /CREATE SETX INSTR
00134 6400                FSTA#   .+2       /AND EXECUTE IT
00135 0136
00136 1100                SETX    .
00137 0136
00140 0030                XTA     X0        /GET TWO CHARS
00141 1100                SETX    XX        /SETUP FOR SECOND SET OF REGISTERS
00142 0052
                          INDEX   XX
00143 1030                JA      PROLOG  /RETURN TO CALLER
```

```
ØØ144 ØØ62
ØØ145 ØØØØ  T1,     Ø;Ø
ØØ146 ØØØØ
ØØ147 ØØØØ  ONE,    Ø;1
ØØ15Ø ØØØ1
```

```
                /       NOW THE ACTUAL ROUTINES
                /
00151 1120  CCGET,  JSA     PROLOG  /GET EVERYTHING SET UP
00152 0062
00153 2000          JXN     GC1,REMAIN      /JMP IF ODD-NUMBERED CHAR WANTED
00154 0156
00155 0012          ALN     KM6     /IF EVEN, SHIFT IT INTO PLACE
00156 0013  GC1,    ALN     KM14    /THROW AWAY HIGH BITS
00157 0014          ALN     K22     /SHIFT EVERYTHING BACK
00160 1010          JGE     .+3     /IF NEG,
00161 0163
00162 1200          FADD    KSGN6   /DISCARD SIGN BITS
00163 0026          ATX     XTMP1   /HOLD IT A MO
00164 0005          STARTF          /TILL WE GET THE MODE
00165 0036          XTA     XTMP1
00166 6002          FSTA%   F       /PUT IT AWAY
00167 1030          JA      GOBACK  /ALL DONE
00170 0041
```

```
RALF V  51  MAY 11, 72 PAGE 4-1

NO ERRORS
31 SYMBOLS, NO ABS REFS

 #        C 00000   *MAIN  S 00002   BASE      00005   CCGET     00151
 CH       S 00171   F        00013   GC1       00156   GOBACK    00041
 KM11       00050   KM14     00055   KM6       00054   KSETX     00016
 KSGN12     00024   KSGN15   00021   KSGN6     00027   K11       00051
 K14        00047   K22      00056   K6        00053   N         00010
 ONE        00147   PROLOG   00062   REMAIN    00052   RET       00001
 T1         00145   T2       00123   XTMP1     00060   XTMP2     00061
 XX         00052   X0       00043   X1        00044   X2        00045
 X3         00046
```

```
C     DEC-12-AMLBA-A-UO
C     COPYRIGHT 1972
C     DIGITAL EQUIPMENT CORPORATION
C     MAYNARD, MA 01754
C     -----ENTER.FT-----
C     THIS SUBROUTINE IS USED TO ENTER A FILE IN
C     A CATALOGUE ON LOGICAL DEV 6 AS WELL AS SEARCH FOR ONE OR
C     MORE FILES THAT COMPARE WITH A GIVEN FILE
C     AND THEN THE OPTION IS GIVEN TO ENTER
C     THE GIVEN FILE IN THE CATALOGUE.
      SUBROUTINE ENTER(FLGLST)
      COMMON SLOTS(85),HEADER(85),ENTRY(21),COMLST(3),KOUNT
      DIMENSION NAME(5),PKLST(16)
      EQUIVALENCE (ENTRY(1),NAME(1)),(ENTRY(6),PKLST(1))
      READ(6'1)HEADER
      WRITE(4,11)
11    FORMAT(' AUTO(1) OR MANUAL(2) ENCODE--I1'/)
      READ(4,12) ENCODE
12    FORMAT(I1)
      DO 13 I=1,16
      PKLST(I)=0.
13    CONTINUE
      IF(ENCODE.EQ.1.)GO TO 20
C     MANUAL ENCODE
      WRITE(4,14)
14    FORMAT(' HOW MANY PEAKS(8 IS MAX)--I1'/)
      READ(4,12) NUMPKS
      IF(NUMPKS.GT.8.OR.NUMPKS.EQ.0) NUMPKS=8
      WRITE(4,15) NUMPKS
15    FORMAT(' MASS VS INTENSITY 1 SET PER LINE 'I1,' TIMES
     1--I4,F6.0'/)
      DO 17 I=1,NUMPKS
      J=I*2-1
      READ(4,16)PKLST(J),PKLST(J+1)
16    FORMAT(I4,1X,F6.0)
17    CONTINUE
      GO TO 40
C     AUTO ENCODE
20    WRITE(4,200)
200   FORMAT(' TOLERANCE:DELT(M)--F3.0'/)
      READ(4,201) DELTM
201   FORMAT(F3.0)
      MASS=0
      TENSTY=0.
      EEND=0.
      Y=8388607.
21    IF(EEND.EQ.1.)GO TO 25
      I=1
220   CALL NEXT(XMASS)
      IF(XMASS.EQ.Y)GO TO 250
      CALL NEXT(XINTEN)
      XMASS=XMASS/10.
      IMASS=XMASS
      FRACT=XMASS-IMASS
      IF(FRACT.LE.DELTM)GO TO 300
```

```
                XFRACT=1-FRACT
                IF(XFRACT.GT.DELTM)GO TO 220
                IMASS=IMASS+1
        300     IF(MASS.EQ.0)MASS=IMASS
                IF(IMASS.GT.MASS)GO TO 22
                IF(XINTEN.GT.TENSTY)TENSTY=XINTEN
                GO TO 220
        22      J=I*2-1
                IF(TENSTY.LE.PKLST(J+1))GO TO 24
                IF(I.EQ.8)GO TO 23
                M=15
                DO 23 L=I,7
                PKLST(M)=PKLST(M-2)
                PKLST(M+1)=PKLST(M-1)
                M=M-2
        23      CONTINUE
                PKLST(J)=MASS
                PKLST(J+1)=TENSTY
        240     MASS=IMASS
                TENSTY=XINTEN
                GO TO 21
        24      I=I+1
                IF(I-9)22,240,240
        250     EEND=1.
                GO TO 22
        C       CHANGE INTENSITIES TO %TAGES
        C       REPORT ON AUTO ENCODE
        25      DO 260 IK=2,8
                JK=2*IK
                PKLST(JK)=PKLST(JK)/PKLST(2)*100.
        260     CONTINUE
                PKLST(2)=100.
                WRITE(3,26)
        26      FORMAT(' AUTO ENCODE'/'   MASS   INTENSITY')
                DO 28 K=1,8
                J=K*2-1
                IF(PKLST(J).EQ.0.)GO TO 28
                WRITE(3,27) PKLST(J),PKLST(J+1)
        27      FORMAT(3X,I4,4X,F7.2)
        28      CONTINUE
                KOUNT=0
                REWIND 5
        40      IF(FLGLST.EQ.0)GO TO 70
        C       SEARCH MODE
                CALL COMPAR
                WRITE(4,67)
        67      FORMAT(' PUT ENTRY IN CATALOGUE?--Y OR N'/)
                READ(4,68)RESPON
        68      FORMAT(A1)
                CALL CCGET(RESPON,1,CHAR1)
                IF(CHAR1.NE.25)GO TO 90
        C       PUT ENTRY IN CATALOGUE ON REQUEST BY 'SEARCH'
        C       AND AUTOATICALLY ENTER IF 'ENTER'.
        70      WRITE(4,71)
        71      FORMAT(' ENTER NAME,FORMULA,ETC--30 CHARS MAX'/)
```

```
              CALL STRNG(30,NAME)
              IF(HEADER(2).NE.1)GO TO 720
              NENTRY=1
              GO TO 800
      720     WRITE(4,72)
      72      FORMAT(' DO YOU WISH TO REPLACE A CATALOGUE ENTRY==Y OR N'/)
              READ(4,68) RESPON
              CALL CCGET(RESPON,1,CHAR1)
              IF(CHAR1.NE.25)GO TO 79
      730     WRITE(4,73)
      73      FORMAT(' WHICH ENTRY?==I4'/)
              READ(4,74) NENTRY
      74      FORMAT(I4)
              LENTRY=HEADER(2)=1
              IF(NENTRY.LE.LENTRY)GO TO 80
              WRITE(4,101) LENTRY
      101     FORMAT(' WHAT? ',I4,' IS LAST ENTRY'/)
              GO TO 730
      79      NENTRY=HEADER(2)
      800     HEADER(2)=HEADER(2)+1
      80      CALL GETBLK(NENTRY,LOCATE,6)
      C       ADD IN OR REPLACE ENTRY BLK
              DO 81 I=1,21
              SLOTS(LOCATE)=ENTRY(I)
              LOCATE=LOCATE+1
      81      CONTINUE
              CALL PUTBLK(NENTRY,6)
              WRITE(6'1)HEADER
      90      RETURN
              END
```

```
        C      DEC-12-AMLBA-A-UD
        C      COPYRIGHT 1972
        C      DIGITAL EQUIPMENT CORPORATION
        C      MAYNARD, MA 01754
        C      -----INTERP.FT-----
        C      THIS SUBROUTINE GETS ANY ONE OF 6 ASCII CHARS
        C      FROM A GIVEN WORD.
0002           SUBROUTINE INTERP(NCMDS,CMD,ICDE)
0003           COMMON SLOTS(85),HEADER(85),ENTRY(21),COMLST(3)
0004           CALL CCGET(CMD,1,CHAR1)
0005           CHAR1=CHAR1*100
0006           CALL CCGET(CMD,2,CHAR2)
0007           CHAR1=CHAR1+CHAR2
0010           DO 10 I=1,NCMDS
0011           CALL CCGET(COMLST,2*I-1,CHAR2)
0012           CHAR2=CHAR2*100
0013           CALL CCGET(COMLST,2*I,CHAR3)
0014           CHAR2=CHAR2+CHAR3
0015           IF(CHAR1.EQ.CHAR2)GO TO 20
0016    10     CONTINUE
        C      -----ILLEGAL COMMAND RETURN
0017           ICDE=9
0020           RETURN
0021    20     ICDE=I
0022           RETURN
0023           END
```

```
        C       DEC-12-AMLBA-A-DO
        C       COPYRIGHT 1972
        C       DIGITAL EQUIPMENT CORPORATION
        C       MAYNARD, MA Ø1754
        C       -----XLIST.FT-----
        C       THIS SUBROUTINE DOES TWO DIFFERENT TYPES OF LISTS
        C       (1) XLIST-LIST THE HEADER INFO & EACH ENTRY NAME
        C       (2) CLIST-LIST EACH ENTRY(NAME AND PEAKS)
ØØØ2            SUBROUTINE XLIST(FLGLST)
ØØØ3            COMMON SLOTS(85),HEADER(85)
ØØØ4            COMMON/A/IDEV
ØØØ5            IF(IDEV.EQ.6)GO TO 3
ØØØ6            DEFINE FILE 7(1,85,U,LV2)
ØØØ7            CALL INIT(Ø)
ØØ1Ø    3       READ(IDEV'1)HEADER
ØØ11            IF(HEADER(2).EQ.1)GO TO 4Ø
ØØ12            WRITE(4,1Ø)
ØØ13    1Ø      FORMAT(' LO & HI LIMIT OF ENTRY NUMBERS--I4,I4'/)
ØØ14            READ(4,11) LIMLO,LIMHI
ØØ15    11      FORMAT(I4,1X,I4)
ØØ16            LENTRY=HEADER(2)-1
ØØ17            IF(LIMHI.EQ.Ø.OR.LIMHI.GT.LENTRY)LIMHI=LENTRY
ØØ2Ø            IF(LIMLO.EQ.Ø)LIMLO=1
ØØ21            IF(LIMLO.GT.LIMHI)LIMLO=LIMHI
ØØ22            IF(FLGLST.EQ.1)GO TO 3Ø
        C       LIST HEADER INFO
ØØ23            WRITE(3,2Ø)(HEADER(I),I=3,6)
ØØ24    2Ø      FORMAT(2(1X,2A6,/))
ØØ25            WRITE(3,21)(HEADER(I),I=7,17)
ØØ26    21      FORMAT(1X,11A6,/)
        C       LIST ENTRY NUM,USER MESSAGE
        C       ALSO LIST PEAKS IF FLGLST=1
ØØ27    3Ø      WRITE(3,31)
ØØ3Ø    31      FORMAT(' ENTRY      IDENTIFICATION')
ØØ31            DO 39 I=LIMLO,LIMHI
ØØ32            CALL GETBLK(I,LOC,IDEV)
ØØ33            IF(SLOTS(LOC).EQ.Ø)GO TO 36
ØØ34            WRITE(3,32)I,(SLOTS(J),J=LOC,LOC+4)
ØØ35    32      FORMAT(/1X,I4,4X,5A6)
ØØ36            IF(FLGLST.EQ.Ø) GO TO 39
ØØ37            WRITE(3,33)
ØØ4Ø    33      FORMAT('   MASS   INTENSITY')
ØØ41            DO 35 K=1,8
ØØ42            J=LOC+3+K*2
ØØ43            IF(SLOTS(J).EQ.Ø)GO TO 35
ØØ44            WRITE(3,34)SLOTS(J),SLOTS(J+1)
ØØ45    34      FORMAT(3X,I4,4X,F7.2)
ØØ46    35      CONTINUE
ØØ47            GO TO 39
ØØ5Ø    36      WRITE(3,37) I
ØØ51    37      FORMAT(/1X,I4,4X,'-----UNUSED-----')
ØØ52    39      CONTINUE
ØØ53            RETURN
ØØ54    4Ø      WRITE(4,41)
ØØ55    41      FORMAT(' NO ENTRIES IN THIS CATALOGUE'/)
```

```
ØØ56           RETURN
ØØ57           END
```

```
RTPS  FORTRAN   1.Ø6                    MAY 11  1972                 PAGE  ONE

              C       DEC-12-AMLBA-A-UD
              C       COPYRIGHT 1972
              C       DIGITAL EQUIPMENT CORPORATION
              C       MAYNARD, MA Ø1754
              C       -----DELETE.FT-----
              C       THIS SUBROUTINE DELETES ONE 21 FL PT WORD
              C       ENTRY IN A CATALOGUE BY REPLACING ALL 21
              C       WORDS WITH Ø.Ø
              C       A 'CR' OR 'Ø' CAUSES A RTN TO 'CMD'
ØØØ2                  SUBROUTINE DELETE
ØØØ3                  COMMON SLOTS(85),HEADER(85)
ØØØ4                  COMMON/A/IDEV
ØØØ5                  IF(IDEV.EQ.6)GO TO 3
ØØØ6                  DEFINE FILE 7(1,85,U,LV2)
ØØØ7                  CALL INIT(Ø)
ØØ1Ø          3       READ(IDEV'1)HEADER
ØØ11                  IF(HEADER(2).EQ.1)GO TO 4Ø
ØØ12                  WRITE(4,1Ø)
ØØ13          1Ø      FORMAT(' ENTRY NUMBER--I4'/)
ØØ14          9       IF(HEADER(2).EQ.1)GO TO 4Ø
ØØ15                  READ(4,11) NUM
ØØ16          11      FORMAT(I4)
ØØ17                  IF(NUM.EQ.Ø)GO TO 32
ØØ2Ø                  LENTRY=HEADER(2)-1
ØØ21                  IF(NUM.GT.LENTRY) GO TO 2Ø
ØØ22                  IF(NUM.LT.LENTRY) GO TO 3Ø
ØØ23          31      HEADER(2)=HEADER(2)-1
ØØ24                  NUM=NUM-1
ØØ25                  CALL GETBLK(NUM,LOCA,IDEV)
ØØ26                  IF(SLOTS(LOCA).EQ.Ø)GO TO 31
ØØ27                  GO TO 7
ØØ3Ø          3Ø      CALL GETBLK(NUM,LOCA,IDEV)
ØØ31                  IEND=LOCA+2Ø
ØØ32                  DO 15 I=LOCA,IEND
ØØ33                  SLOTS(I)=Ø.Ø
ØØ34          15      CONTINUE
ØØ35                  CALL PUTBLK(NUM,IDEV)
ØØ36                  GO TO 7
ØØ37          32      WRITE(IDEV'1)HEADER
ØØ4Ø                  RETURN
ØØ41          2Ø      WRITE(4,21) LENTRY
ØØ42          21      FORMAT(' NUMBER ',I4,' IS LAST ENTRY'/)
ØØ43          7       WRITE(4,8)
ØØ44          8       FORMAT(1H )
ØØ45                  GO TO 9
ØØ46          4Ø      WRITE(4,41)
ØØ47          41      FORMAT(' NO ENTRIES IN THIS CATALOGUE'/)
ØØ5Ø                  WRITE(IDEV'1)HEADER
ØØ51                  RETURN
ØØ52                  END
```

```
          C       DEC-12-AMLBA-A-UD
          C       COPYRIGHT 1972
          C       DIGITAL EQUIPMENT CORPORATION
          C       MAYNARD, MA 01752
          C       -----SQUISH.FT-----
          C       THIS SUBROUTINE PACKS ALL ENTRIES OF A
          C       GIVEN CATALOGUE ELIMINATING ALL EMPTY
          C       ENTRIES THAT WERE CAUSED BY DELETIONS.
0002              SUBROUTINE SQUISH
0003              COMMON SLOTS(85),HEADER(85)
0004              COMMON/A/IDEV
0005              DIMENSION TEMP(85)
0006              IF(IDEV.EQ.6)GO TO 3
0007              DEFINE FILE 7(1,85,U,LV2)
0010              CALL INIT(0)
0011      3       READ(IDEV'1)HEADER
0012              IEND=HEADER(2)-1
0013              NBLKS=(IEND-1)/4+1
0014              IA=2
0015              IB=2
0016              NENTT=0
0017              LOCB=1
0020              LF=0
0021              DO 30 K=1,NBLKS
0022              LOCA=1
0023              READ(IDEV'IA)SLOTS
0024              IA=IA+1
0025              JEND=4
0026              IF(K.EQ.NBLKS)JEND=IEND
0027              IEND=IEND-4
0030              DO 20 J=1,JEND
0031              IF(SLOTS(LOCA).EQ.0)GO TO 12
0032              LF=1
0033              NENTT=NENTT+1
0034              LEND=LOCA+20
0035              DO 10 I=LOCA,LEND
0036              TEMP(LOCB)=SLOTS(I)
0037              LOCB=LOCB+1
0040      10      CONTINUE
0041              IF(LOCB.LT.85)GO TO 12
0042              WRITE(IDEV'IB) TEMP
0043              IB=IB+1
0044              LF=0
0045              LOCB=1
0046      12      LOCA=LOCA+21
0047      20      CONTINUE
0050      30      CONTINUE
0051              IF(LF.EQ.0)GO TO 25
0052              WRITE(IDEV'IB) TEMP
0053      25      HEADER(2)=NENTT+1
0054              WRITE(IDEV'1) HEADER
0055              RETURN
0056              END
```

```
      C       DEC-12-AMLBA-A-UD
      C       COPYRIGHT 1972
      C       DIGITAL EQUIPMENT CORPORATION
      C       MAYNARD, MA 01754
      C       =====MOVE.FT=====
      C       THIS SUBROUTINE MOVES A CONTIGUOUS SET OF
      C       ENTRIES (K1,2,3,...,ALL) FROM THE CATALOGUE
      C       ON DEV6 TO THE CATALOGUE ON DEV7.  BOTH DEVICES
      C       ARE CHECKED FOR INIT.
              SUBROUTINE MOVE
              COMMON SLOTS(85),HEADER(85)
              DIMENSION SL(85),HEAD(85)
              READ(6'1) HEAD
              IF(HEAD(2).EQ.1)GO TO 40
              WRITE(4,100)
      100     FORMAT(' LO & HI LIMIT OF ENTRIES TO MOVE--I4,I4'/)
              READ(4,110) LIMLO,LIMHI
      110     FORMAT(I4,1X,I4)
              LENTRY=HEAD(2)-1
              IF(LIMHI.EQ.0.OR.LIMHI.GT.LENTRY)LIMHI=LENTRY
              IF(LIMLO.EQ.0)LIMLO=1
              IF(LIMLO.GT.LIMHI)LIMLO=LIMHI
      C       DO MOVE
              KSTOP=LIMHI-LIMLO+1
              FIRST6=(LIMLO+3)/4+1
      10      IF(LIMLO.LE.4)GO TO 11
              LIMLO=LIMLO-4
              GO TO 10
      11      FIRST=HEADER(2)
              FIRST7=(FIRST+3)/4+1
      12      IF(FIRST.LE.4)GO TO 13
              FIRST=FIRST-4
              GO TO 12
      13      KBEGIN=(FIRST-1)*21+1
              KSTART=(LIMLO-1)*21+1
              READ(7'FIRST7)SL
      25      READ(6'FIRST6) SLOTS
      26      KEND=KSTART+20
              DO 50 I=KSTART,KEND
              SL(KBEGIN)=SLOTS(I)
              KBEGIN=KBEGIN+1
      50      CONTINUE
              HEADER(2)=HEADER(2)+1
              KSTOP=KSTOP-1
              IF(KSTOP.EQ.0)GO TO 59
              IF(KBEGIN.LT.85)GO TO 55
              KBEGIN=1
              WRITE(7'FIRST7) SL
              FIRST7=FIRST7+1
      55      KSTART=KSTART+21
              IF(KSTART.LT.85)GO TO 26
              KSTART=1
              FIRST6=FIRST6+1
              GO TO 25
      59      WRITE(7'FIRST7) SL
```

```
          WRITE(7'1)HEADER
          RETURN
   40     WRITE(4,41)
   41     FORMAT(' NO ENTRIES IN THIS CATALOGUE'/)
          RETURN
          END
```

```
C       DEC-12-AMLBA-A-UD
C       COPYRIGHT 1972
C       DIGITAL EQUIPMENT CORPORATION
C       MAYNARD, MA 01754
C       -----COMPAR.FT-----
C       THIS ROUTINE IS USED TO SEARCH FOR A
C       MATCH BETWEEN A GIVEN ENTRY AND
C       ALL FILES IN THE GIVEN CATALOGUE
        SUBROUTINE COMPAR
        COMMON SLOTS(85),HEADER(85),ENTRY(21)
        WRITE(4,41)
41      FORMAT(' TOLERANCE: DELT(I),#HITS--F3.0,I2'/)
        READ(4,42) DELTI,NUMPKS
42      FORMAT(F3.0,1X,I2)
C       DO THE SEARCH
        WRITE(3,43)
43      FORMAT(/' REPORT OF SEARCH'/' ENTRY   IDENTI
       1FICATION                 #HITS')
        FLAG=0.
        IEND=HEADER(2)-1
        DO 65 I=1,IEND
        IBLK=(I+3)/4+1
        LOCATE=((I-4*(IBLK-2))-1)*21+1
        READ(6'IBLK)SLOTS
        ICTR=0
        DO 55 L=1,8
        LL=4+L*2
        IF(ENTRY(LL).EQ.0)GO TO 55
        DO 50 K=1,8
        KK=LOCATE+3+K*2
        IF(SLOTS(KK).NE.ENTRY(LL)) GO TO 50
        DIFF=SLOTS(KK+1)-ENTRY(LL+1)
        IF(DIFF.GE.0) GO TO 45
        DIFF=-DIFF
45      IF(DIFF.GT.DELTI)GO TO 50
        ICTR=ICTR+1
50      CONTINUE
55      CONTINUE
        IF(ICTR.LT.NUMPKS)GO TO 65
C       REPORT THE MATCH
        WRITE(3,57)I,(SLOTS(J),J=LOCATE,LOCATE+4),ICTR
57      FORMAT(1X,I4,2X,5A6,I3)
        FLAG=1.
65      CONTINUE
        IF(FLAG.EQ.1)GO TO 67
        WRITE(3,66)
66      FORMAT(' NO MATCHES'/)
67      RETURN
        END
```

```
C       DEC-12-AMLBA-A-UO
C       COPYRIGHT 1972
C       DIGITAL EQUIPMENT CORPORATION
C       MAYNARD, MA 01754
C       -----INIT.FT-----
C       THIS SUBROUTINE CHECKS A CATALOGUE TO SEE
C       IF IT HAS BEEN INITED.  IF IT HASN'T THE
C       INIT IS DONE AND IF IT HAS, THE INIT IS DONE
C       ONLY IF REQUESTED.
        SUBROUTINE INIT(FLG)
        COMMON SLOTS(85),HEADER(85),ENTRY(21),COMLST(3)
        COMMON/A/IDEV
        IF(FLG.EQ.0)GO TO 3
        IF(IDEV.EQ.6)GO TO 3
        DEFINE FILE 7(1,85,U,LV2)
3       READ(IDEV'1)HEADER
        IF(HEADER(1).EQ.1934.)GO TO 4
        WRITE(4,7)IDEV
7       FORMAT(' LOGICAL DEV',I2,' IS BEING INITIALIZED'/)
        GO TO 8
4       IF(FLG.EQ.0)GO TO 30
8       ENTRY(1)=1934.
        ENTRY(2)=1
        WRITE(4,10)
10      FORMAT(' NAME OF CATALOGUE--12 CHARS MAX'/)
        CALL STRNG(12,ENTRY(3))
        WRITE(4,11)
11      FORMAT(' DATE--12 CHARS MAX'/)
        CALL STRNG(12,ENTRY(5))
        WRITE(4,12)
12      FORMAT(' USER MESSAGE--66 CHARS MAX'/)
        CALL STRNG(66,ENTRY(7))
        IF(FLG.EQ.1)GO TO 20
5       DO 200 I=1,17
        HEADER(I)=ENTRY(I)
200     CONTINUE
        WRITE(IDEV'1)HEADER
        RETURN
20      WRITE(4,21)
21      FORMAT(' ARE YOU SURE?--Y OR N'/)
        READ(4,22) RESPON
22      FORMAT(A1)
        CALL CCGET(RESPON,1,CHAR1)
        IF(CHAR1.EQ.25)GO TO 5
30      RETURN
        END
```

```
RTPS  FORTRAN   1.06                  MAY 11  1972          PAGE  ONE

C       DEC-12-AMLBA-A-UD
C       COPYRIGHT 1972
C       DIGITAL EQUIPMENT CORPORATION
C       MAYNARD, MA 01754
C       -----NEXT.FT-----
C       THIS SUBROUTINE GETS THE NEXT VALUE OF
C       A GIVEN HASH FILE.
        SUBROUTINE NEXT(VALUE)
        COMMON SLOTS(85),HEADER(85),ENTRY(21),COMLST(3),KOUNT
        DIMENSION ARRAY(85)
        IF(KOUNT.GT.0.AND.KOUNT.LT.85)GO TO 10
        KOUNT=0
        READ(5) ARRAY
10      KOUNT=KOUNT+1
        VALUE=ARRAY(KOUNT)
        RETURN
        END
```

```
   RTPS  FORTRAN   1.Ø6                    MAY 11  1972              PAGE  ONE

               C       DEC-12-AMLBA-A-UD
               C       COPYRIGHT 1972
               C       DIGITAL EQUIPMENT CORPORATION
               C       MAYNARD, MA Ø1754
               C       -----PUTBLK.FT-----
               C       THIS SUBROUTINE PUTS ONE BLK OF 4 ENTRY SLOTS
               C       INTO THE CATALOGUE
     ØØØ2              SUBROUTINE PUTBLK(NENT,NDEV)
     ØØØ3              COMMON SLOTS(85)
     ØØØ4              IBLK=(NENT+3)/4+1
     ØØØ5              WRITE(NDEV'IBLK)SLOTS
     ØØØ6              RETURN
     ØØØ7              END
```

```
        C      DEC-12-AMLBA-A-UD
        C      COPYRIGHT 1972
        C      DIGITAL EQUIPMENT CORPORTION
        C      MAYNARD, MA 01754
        C      -----GETBLK.FT-----
        C      THIS SUBROUTINE GETS ONE BLK OF 4 ENTRY SLOTS
0002           SUBROUTINE GETBLK(NENT,LOC,NDEV)
0003           COMMON SLOTS(85)
0004           IBLK=(NENT+3)/4+1
0005           LOC=((NENT-4*(IBLK-2))-1)*21+1
0006           READ(NDEV'IBLK)SLOTS
0007           RETURN
0010           END
```

```
                    /       DEC-12-AMLBA-A-UC
                    /       COPYRIGHT 1972
                    /       DIGITAL EQUIPMENT CORPORATION
                    /       MAYNARD, MA 01754

                    /       -----ALOCAT.PA-----
                    /       THIS PROGRAM CPENS A FILE OF USER-SPECIFIED
                    /       LENGTH, UNIT, & NAME  AND THEN CLOSES IT.
                    /       THIS WILL BE THE NEW CATEGORY FILE TO BE
                    /       INITED AND THEN ACCESSED VIA DIRECT ACCESS I/O.

                    /       CALLING SEQ:  ODEV:ONAME.EX[NB]<
                    /       WHERE NB = NUMBER BLKS OF OUTPUT FILE
                    /

          0012              NDX2=12
          0001              FIELD 1
          2000              *2000
    12000 6201              CDF 0
    12001 1377              TAD     (7003           /JOB STATUS WD
    12002 3776'             DCA     7746            /NO SAVING & NO RESTART
    12003 6211              CDF 10
    12004 4775              JMS I   (7700           /GET USR
    12005 0010              10
    12006 4774              JMS I   (200            /GET CD
    12007 0005              5
    12010 2001              2001
    12011 1773'             TAD     7600
    12012 4774              JMS I   (200            /GET OUTPUT HANDLER
    12013 0001              1
    12014 3001              3001
    12015 5241              JMP     ERR
    12016 1773'             TAD     7600
    12017 4774              JMS I   (200            /ENTER OUTPUT FILE
    12020 0003              3
    12021 7601              7601
    12022 0000              0
    12023 5242              JMP     ERR+1
    12024 1773'             TAD     7600
    12025 0372              AND     (7760
    12026 7112              CLL RTR
    12027 7012              RTR
    12030 3235              DCA     .+5             /LENGTH OF FILE
    12031 1773'             TAD     7600
    12032 4774              JMS I   (200            /CLOSE OUTPUT FILE
    12033 0004              4
    12034 7601              7601
    12035 0000              0
    12036 5243              JMP     ERR+2
    12037 6203      EXIT,   CDF CIF
    12040 5771              JMP I   (7605           /GO TC MONITOR
    12041 7001      ERR,    IAC                     /BAD OUTPUT DEV
    12042 7001              IAC                     /OUTPUT FILE ENTRY ERRR
    12043 7001              IAC                     /FILE CLOSE OR WRITE ERRCR
    12044 3247              DCA     .+3
```

```
12Ø45  4774        JMS I   [2ØØ              /TELL USER
12Ø46  ØØØ7        7
12Ø47  ØØØØ        Ø
12Ø5Ø  5257        JMP     EXIT              /AND GO TO MONITOR
12171  76Ø5
12172  776Ø
12173  76ØØ
12174  Ø2ØØ
12175  77ØØ
12176  7746
12177  7ØØ3
                   $
```

```
ERR     2Ø41
EXIT    2Ø37
NDX2    ØØ12
```

```
                        /       DEC-12-AMLBA-A-UD
                        /       COPYRIGHT 1972
                        /       DIGITAL EQUIPMENT CORPORATION
                        /       MAYNARD, MA 01754

                        /       -----TRANS.FA-----
                                /TRANS WRITTEN TO TRANSCRIBE AIPOS FILES
                                /INTO FPP FORTRN 3WORD/85 PER BLOCK
                                /DATA FILES UNDER OS8
                                /
                                /CALLING SEQUENCE
                                /ODEV:ONAME.EXT<
                                /
                                /Q&A INPUT
                                /       UNIT:
                                /       STBLK:
                                /       LEN:
                                /
                                /


        0010                    IA=10
        0011                    CA=11
        0012                    NDX2=12
        0001                    FIELD 1
        2000                    *2000
12000   1377    BEGIN,          TAD     (3001
12001   3220                    DCA     ODA
12002   1376                    TAD     (7601
12003   3225                    DCA     OSB
12004   6201                    CDF 0
12005   1375                    TAD     (7003   /JOB STATUS WORD
12006   3774'                   DCA     7746    /NO SAVING & NO RESTART
12007   6211                    CDF 10
12010   4773                    JMS I   (7700   /GET USR
12011   0010                    10
12012   4772                    JMS I   (200    /GET CD, SPECIAL MODE
12013   0005                    5
12014   5200                    5200
12015   1771'                   TAD     7600
12016   4772                    JMS I   (200    /GET OUTPUT HANDLER
12017   0001                    1
12020   3001    ODA,            3001
12021   5770'                   JMP     ERR
12022   1771'                   TAD     7600
12023   4772                    JMS I   (200    /ENTER OUTPUT FILE
12024   0003                    3
12025   7601    OSB,            7601
12026   0000    OFL,            0
12027   5767'                   JMP     ERR+2
```

```
/       DEC-12-AMLBA-A-LD          PAL8-V7  5/11/72   PAGE 1-1

                         EJECT
                /
                /Q&A STUFF
                /
12030   1366    MESS,    TAD     (MESS1-1
12031   3012             DCA     NDX2
12032   4765             JMS I   (TTYO
12033   4764             JMS I   (TTYI   /GET UNIT NUMBER
12034   1763'            TAD     NUM
12035   1362             TAD     (-7
12036   7740             SMA SZA CLA            /NUM<=7?
12037   5230             JMP     MESS    /NO
12040   1763'            TAD     NUM     /YES
12041   3335             DCA     1UNIT
12042   1361             TAD     (MESS2-1
12043   3012             DCA     NDX2
12044   4765             JMS I   (TTYO
12045   4764             JMS I   (TTYI   /GET STBLK INPUT
12046   7200             CLA
12047   1763'            TAD     NUM
12050   3760'            DCA     RSB     /STBLK OF 'AIPOS DATA'
12051   1357             TAD     (MESS3-1
12052   3012             DCA     NDX2
12053   4765             JMS I   (TTYO
12054   4764             JMS I   (TTYI   /GET LEN INPUT
12055   7201             CLA IAC
12056   1763'            TAD     NUM
12057   7041             CIA
12060   3333             DCA     IFL     /-(LEN+1)OF 'AIPOS FILE'
```

```
                              EJECT
                        /
                        /NOW DO YOUR STUFF
                        /
12061  1356                   TAD     (-2
12062  3334                   DCA     FLGINF
12063  1225                   TAD     OSB
12064  3755'                  DCA     WSB
12065  1220                   TAD     ODA
12066  3754'                  DCA     ODAX
12067  4753'                  JMS     RDHEAD  /READ IN HEADER BLK OF AIPOS FILE
12070  6211                   CDF 10
12071  4752'                  JMS     READ    /GET FIRST BLOCK
12072  1351    WTR,           TAD     (5777
12073  3011                   DCA     OA      /OUTPUT BUFFER-1
12074  1350                   TAD     (-125
12075  3332                   DCA     OC      /AND CTR
12076  6201    LDR,           CDF 0
12077  1347                   TAD     (27
12100  3411                   DCA I   OA
12101  1410                   TAD I   IA
12102  3336                   DCA     TMP
12103  1336                   TAD     TMP
12104  3411                   DCA I   OA
12105  1410                   TAD I   IA
12106  3411                   DCA I   OA
12107  1336                   TAD     TMP     /CHECK FOR
12110  1346                   TAD     (-3777  /37777777
12111  7650                   SNA CLA         /NO
12112  2334                   ISZ     FLGINF  /YES, 2 SETS YET?
12113  7410                   SKP             /NO, 1 SET SO FAR
12114  5745'                  JMP     DONE    /YES,ALL DONE
12115  6211                   CDF     10
12116  2332                   ISZ     OC
12117  7410                   SKP
12120  5325                   JMP     ODN     /OUTPUT BUFFER IS LOADED
12121  2331                   ISZ     IC
12122  5276                   JMP     LDR
12123  4752'                  JMS     READ    /GET NEXT INPUT BLK
12124  5276                   JMP     LDR
12125  4744'   ODN,           JMS     WRITE
12126  2331                   ISZ     IC
12127  5272                   JMP     WTR     /RESET OUTPUT JUNK
12130  5271                   JMP     WTR-1   /INPUT ALSO
12131  0000    IC,            0
12132  0000    OC,            0
12133  0000    IFL,           0
12134  0000    FLGINF,        0
12135  0000    IUNIT,         0
12136  0000    TMP,           0
```

```
                                EJECT
12144  2457
12145  22ØØ
12146  4ØØ1
12147  ØØ27
1215Ø  7653
12151  5777
12152  2222
12153  2274
12154  247Ø
12155  2464
12156  7776
12157  2522
1216Ø  2246
12161  2511
12162  7771
12163  2455
12164  2412
12165  24ØØ
12166  25Ø1
12167  2473
1217Ø  2471
12171  76ØØ
12172  Ø2ØØ
12173  77ØØ
12174  7746
12175  7ØØ3
12176  76Ø1
12177  3ØØ1
       22ØØ             PAGE
122ØØ  6211   DONE,     CDF 1Ø
122Ø1  4777'            JMS     WRITE    /LAST WRITE
122Ø2  1776'            TAD     OSB      /CLOSE OUTPUT FILE
122Ø3  7Ø41             CIA
122Ø4  1775'            TAD     WSB
122Ø5  3212             DCA     .+5
122Ø6  1774'            TAD     76ØØ
122Ø7  4773             JMS I   (2ØØ
1221Ø  ØØØ4             4
12211  76Ø1             76Ø1
12212  ØØØØ             Ø
12213  5772'            JMP     ERR+4    /FILE CLOSE EROR
12214  1371             TAD     (MESS4-1
12215  3Ø12             DCA     NDX2
12216  477Ø             JMS I   (TTYC
12217  5767'            JMP     BEGIN
1222Ø  62Ø3   EXIT,     CDF CIF Ø
12221  5766             JMP I   (76Ø5    /GO TO MONITOR
                               EJECT
         6141                           LINK=6141
         ØØØ2                           PDP=2
         1Ø2Ø                           LDAIØ=1Ø2Ø
         ØØ23                           TMA=23
         ØØØ1                           AXO=1
12222    ØØØØ    READ,         Ø
12223    2765'                 ISZ      IFL
12224    741Ø                  SKP
12225    5257                  JMP      FIXUP
12226    1764'                 TAD      IUNIT
12227    711Ø                  CLL RAR                 /GET EXTENDED UNIT
1223Ø    1363                  TAD      (2Ø            /FLD=Ø,ENABLE EXT ADR
12231    3243                  DCA      XOB            /XOB BITS SET
12232    72Ø6                  CLA RTL                 /GET INSTRUCT UNIT &
12233    7ØØ6                  RTL                     /PUT IN TO BIT 8
12234    1362                  TAD      (7ØØ           /OCTAL FOR'RDC'
12235    3245                  DCA      RDC
12236    6141                  LINK
12237    1Ø2Ø                  LDAIØ
1224Ø    2ØØØ                  2ØØØ
12241    ØØ23                  TMA
12242    1Ø2Ø                  LDAIØ
12243    ØØØØ    XOB,          Ø
12244    ØØØ1                  AXO
12245    ØØØØ    RDC,          Ø
12246    ØØØØ    RSB,          Ø                       /BLK NUMBER
12247    ØØØ2                  PDP
1225Ø    2246                  ISZ      RSB
12251    72ØØ    X,            CLA
12252    1361                  TAD      (1777
12253    3Ø1Ø                  DCA      IA
12254    1374                  TAD      (-2ØØ
12255    3766'                 DCA      IC
12256    5622                  JMP I    READ

12257    62Ø1    FIXUP,        CDF Ø
1226Ø    1361                  TAD      (1777
12261    3Ø1Ø                  DCA      IA
12262    1357                  TAD      (3777
12263    341Ø                  DCA I    IA
12264    1356                  TAD      (7777
12265    341Ø                  DCA I    IA
12266    1357                  TAD      (3777
12267    341Ø                  DCA I    IA
1227Ø    1356                  TAD      (7777
12271    341Ø                  DCA I    IA
12272    6211                  CDF 1Ø
12273    5251                  JMP      X
```

```
                            EJECT
12274  0000  RDHEAD, 0
12275  4222              JMS       READ
12276  6201              CDF 0
12277  2010              ISZ       IA
12300  2010              ISZ       IA
12301  1410              TAD I     IA
12302  1355              TAD       (-0405   /MASS/INTEN CODE NUMBER
12303  7650              SNA CLA
12304  5674              JMP I     RDHEAD
12305  6211              CDF 10
12306  1354              TAD       (MESS5-1
12307  3012              DCA       NDX2
12310  4770              JMS I     (TTYO
12311  5767'             JMP       BEGIN
12354  2535
12355  7373
12356  7777
12357  3777
12360  2131
12361  1777
12362  0700
12363  0020
12364  2135
12365  2133
12366  7605
12367  2000
12370  2400
12371  2531
12372  2475
12373  0200
12374  7600
12375  2404
12376  2025
12377  2457
       2400              PAGE
```

```
                           EJECT

12400  0000  TTYO,    0
12401  7200           CLA
12402  6046           TLS
12403  6041           TSF
12404  5203           JMP     .-1
12405  1412           TAD I   NDX2
12406  6046           TLS
12407  7640           SZA CLA          /E.O.M.?
12410  5203           JMP     TTYO+3   /NO
12411  5600           JMP I   TTYO     /YES

12412  0000  TTYI,    0
12413  1377           TAD     (-5
12414  3254           DCA     DIGCTR
12415  3255           DCA     NUM
12416  6031           KSF
12417  5216           JMP     .-1
12420  6036           KRB
12421  1376           TAD     (-203
12422  7450           SNA              /AC?
12423  5775'          JMP     EXIT     /YES
12424  1374           TAD     (203
12425  6046           TLS              /ECHO CHAR
12426  1373           TAD     (-215
12427  7450           SNA              /CR?
12430  5612           JMP I   TTYI     /YES
12431  1372           TAD     (215-267
12432  7540           SMA SZA
12433  5247           JMP     ERROR
12434  1371           TAD     (7
12435  7510           SPA
12436  5247           JMP     ERROR
12437  3256           DCA     DIGIT
12440  1255           TAD     NUM
12441  7106           CLL RTL
12442  7004           RAL
12443  1256           TAD     DIGIT
12444  3255           DCA     NUM
12445  2254           ISZ     DIGCTR
12446  5216           JMP     TTYI+4
12447  7200  ERROR,   CLA
12450  1212           TAD TTYI
12451  1370           TAD     (-4      /REPEAT CURRENT MESSAGE
12452  3212           DCA     TTYI
12453  5612           JMP I   TTYI
12454  0000  DIGCTR,  0
12455  0000  NUM,     0
12456  0000  DIGIT,   0
```

```
                         EJECT

12457   ØØØØ   WRITE,    Ø
1246Ø   62Ø2             CIF Ø
12461   467Ø             JMS I   ODAX    /WRITE OUT
12462   42ØØ             42ØØ
12463   6ØØØ             6ØØØ
12464   ØØØØ   WSB,      Ø
12465   5275             JMP     ERR+4
12466   22Ø4             ISZ     WSB
12467   5657             JMP I   WRITE
1247Ø   ØØØØ   ODAX,     Ø

12471   7ØØ1   ERR,      IAC             /BAD OUTPUT DEVICE
12472   7ØØ1             IAC             /BAD INPUT DEVICE
12473   7ØØ1             IAC             /OUTPUT FILE ENTRY ERROR
12474   7ØØ1             IAC             /FILE CLOSE O WRITE ERROR
12475   33ØØ             DCA     .+3
12476   4767             JMS I   (2ØØ    /TELL USER
12477   ØØØ7             7
125ØØ   ØØØØ             Ø
125Ø1   5775'            JMP     EXIT    /AND GO TO MONITOR
```

```
                          EJECT

12502  0215  MESS1,  215
12503  0212          212
12504  0325          325              /UNIT:
12505  0316          316
12506  0311          311
12507  0324          324
12510  0272          272
12511  0000          0                /EOM
12512  0215  MESS2,  215
12513  0212          212
12514  0323          323              /STBLK:
12515  0324          324
12516  0302          302
12517  0314          314
12520  0313          313
12521  0272          272
12522  0000          0                /EOM
12523  0215  MESS3,  215
12524  0212          212
12525  0314          314              /LEN:
12526  0305          305
12527  0316          316
12530  0272          272
12531  0000          0                /EOM
12532  0212  MESS4,  212
12533  0212          212
12534  0212          212
12535  0000          0                /EOM
12536  0215  MESS5,  215
12537  0212          212
12540  0316          316              /NOT M/I FILE
12541  0317          317
12542  0324          324
12543  0240          240
12544  0315          315
12545  0257          257
12546  0311          311
12547  0240          240
12550  0306          306
12551  0311          311
12552  0314          314
12553  0305          305
12554  0215          215
12555  0212          212
12556  0000          0
12557  0200
12570  7774
12571  0007
12572  7726
12573  7563
12574  0203
12575  2220
12576  7575
```

```
12577  7773
                    $
```

```
AXO     ØØØ1
BEGIN   2ØØØ
DIGCTR  2454
DIGIT   2456
DONE    22ØØ
ERR     2471
ERROR   2447
EXIT    222Ø
FIXUP   2257
FLGINP  2134
IA      ØØ1Ø
IC      2131
IFL     2133
IUNIT   2135
LDAIØ   1Ø2Ø
LDR     2Ø7Ø
LINK    6141
MESS    2Ø3Ø
MESS1   25Ø2
MESS2   2512
MESS3   2523
MESS4   2532
MESS5   2536
NDX2    ØØ12
NUM     2455
OA      ØØ11
OC      2132
ODA     2Ø2Ø
ODAX    247Ø
ODN     2125
OFL     2Ø26
OSB     2Ø25
PDP     ØØØ2
RDC     2245
RDHEAD  2274
READ    2222
RSB     2246
TMA     ØØ23
TMP     2136
TTYI    2412
TTYO    24ØØ
WRITE   2457
WSB     2464
WTR     2Ø72
X       2251
XOB     2243
```

```
C       DEC-12-AMLBA-A-UO
C       COPYRIGHT 1972
C       DIGITAL EQUIPMENT CORPORATION
C       MAYNARD, MA Ø1754
C       -----PRINT.FT-----
C       THIS PRINTS THE CONVERTED AIPOS FILES NOW
C       IN FPP MODE ONTO THE LINEPRINTER
C       DEV SPECS: IDEV:NAME.EX/5
        DIMENSION ARRAY(85),B(84)
        WRITE(3,5Ø)
5Ø      FORMAT(1H1,4X,'MASS*1Ø',6X,'INTENSITY'//)
1ØØ     FORMAT(2X,F1Ø.2,3X,F1Ø.2)
        Y=83886Ø7.
        X=1
        K=1
        J=1
14      READ(5) ARRAY
15      B(K)=ARRAY(J)
        IF(ARRAY(J).EQ.Y.AND.X.EQ.2) GO TO 16
        IF(ARRAY(J).EQ.Y.AND.X.EQ.1) X=X+1
        J=J+1
        K=K+1
        IF(K.EQ.85) GO TO 11
12      IF(J.LT.86) GO TO 15
        J=1
        GO TO 14
11      WRITE(3,1ØØ) B
        K=1
        GO TO 15
16      WRITE(3,1ØØ) B
        END
```

INDEX

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12
Digital Software News for the PDP-11
Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library, Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's Software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are provided in the software kit should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software and manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. U.S.A. customers may order directly from the Program Library in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

LIBMSH USER'S MANUAL
DEC-12-AMLBA-A-D

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability.

Did you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Please state your position. ____________________ Date: __________

Name: ____________________ Organization: ____________________

Street: ____________________ Department: ____________________

City: ____________ State: ____________ Zip or Country __________

- - - - - - - - - - - - - - - - - Fold Here -

- - - - - - - - - - - - - - Do Not Tear - Fold Here and Staple - - - - - - - - - - - - - -

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754