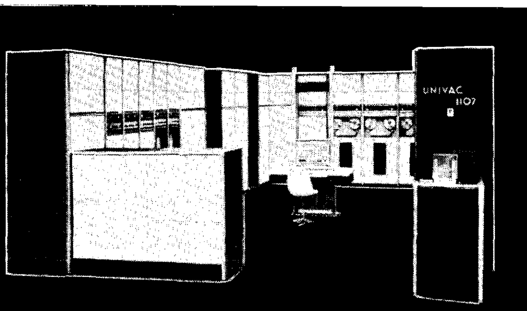


UNIVAC® 1107

TECHNICAL BULLETIN

LIBRARIAN Routine

Programmers Reference

First Edition

May, 1962

TABLE OF CONTENTS

I. INTRODUCTION

The LIBRARIAN is a UNIVAC® 1107 program which performs two major functions: the creation and maintenance of magnetic tape libraries, and the retrieval of programs, subroutines, and data from these libraries.

The LIBRARIAN program is, accordingly, composed of two sections:

(1) The maintenance section performs the building of a library tape and the updating of that tape. The updating features include addition, deletion, replacement, resequencing, and cataloguing of a library tape.

(2) The retrieval section obtains the information block, the cross-reference labels, and the order in which the records are positioned on the library tape.

The LIBRARIAN is closely related to some of the other elements of the software package. The maintenance section of the LIBRARIAN is called in via a Job Request to the Executive System, EXEC, and is executed in the same manner as are other EXEC job programs. The LIBRARIAN retrieval section may be called in by the SLEUTH Assembly System (for incorporation of subroutines at assembly time) or by CLAMP, the Relative Load Routine. The LIBRARIAN retrieval function may also be incorporated within any other assembler, compiler, or load routine. The LIBRARIAN utilizes EXEC I/O Functional Routines for its tape operations. The retrieval section may be used only for the retrieval of information which is stored on a LIBRARIAN produced tape library.

The information is stored on a LIBRARIAN produced tape in the form of records. Each record may be a programming system, a program, a subroutine, or a data table. Each library must consist of only one reel of magnetic tape, but more than one library tape may be produced. Each record on a library tape may occupy several blocks.

II. INPUT MAINTENANCE FUNCTION REQUESTS

A Maintenance Function Request to the LIBRARIAN consists of a Request Parameter, one or more library record labels, tape unit allocations, and special information pertaining to a record. These items are arranged into a sentence and are separated by appropriate Request Delimiters. The choice of Request Delimiters and of the items contained in a Maintenance Function Request is determined by its purpose.

One or more Maintenance Function Requests may be chained together within one request sentence by the use of the comma (,) Request Delimiter. A sentence is ended with the delimiter END.

Table 1 contains a list of the LIBRARIAN Request Parameters and their functions. The use of these parameters is described in the sections which follow. Table 2 lists the Request Delimiters with a brief description of their uses. Each Maintenance Function Request sentence is accepted by the LIBRARIAN as input data when the maintenance section is executed. The sentence may be entered through any of several input media but will usually take the form of punched cards.

A. LIBNEW

1. Function

The LIBNEW Request Parameter is used to produce a new library tape. Following the Request Parameter are the labels of the records being stored on the new library tape.

2. Limitations

The only Request Parameters that may be used with LIBNEW are LIBIDT, LIBDUP, LIBINP, and LIBCAT.

3. Tape Configurations

A minimum of two tapes is necessary when using the LIBNEW Request Parameter:

a. a blank tape for the new library on tape unit A

b. a tape containing the records to be included in the new library on tape unit B

When assigning tape units, the units must be designated in alphabetical order starting with A and ending with I. However, the user need not designate tape units when using the minimum tape configuration for the specific Request Parameter.

4. Example

An example of the use of the LIBNEW Request Parameter is:

LIBNEW SIN, COS, TAN, SQRT END

The commas (,) are used to separate the labels of each record, and the END Request Delimiter is used to signal the end of the request sentence. The END delimiter is the only means of indicating the termination of a sentence. A delimiter is not needed to separate the END delimiter from the remainder of the sentence.

The records will be located on the library tape in the order in which they have been listed in the request; i.e., the sole determining factor of the sequence of routines on the library will be the order given when using the LIBNEW Request Parameter. In the above example, SIN will be the first record on the library, and SQRT will appear as the last record.

B. LIBADD

1. Function

The LIBADD Request Parameter is used to add records to an existing library tape. The labels of the records being added at the time the LIBADD Request Parameter is used will follow LIBADD. In processing the request, the LIBRARIAN builds a new library tape in which the additions are merged with the obsolete library tape.

2. Limitations

When a record to be added to an existing library tape has the same label as a record on the obsolete library, the new record will not be added to the library. If a record to be added references another record, the referenced record must either appear on the existing library or be included in the LIBADD request. When this condition is not complied with,the record designated in the LIBADD request is omitted from the library.

REQUEST PARAMETER	FUNCTION
LIBNEW	Produces a new library tape
LIBADD	Adds new records to an existing library tape
LIBDEL	Deletes records from an existing library tape
LIBREP	Replaces records on an existing library tape
LIBSEQ	Resequences records on an existing library tape
LIBIDT	Assigns a title to a new or existing library tape
LIBDUP	Duplicates tapes
LIBCAT	Catalogues and lists the record labels of an existing library tape
LIBINP	Defines input tape format

Table 1

REQUEST DELIMITER	PURPOSE
( ) parentheses	Used to enclose record labels for cross-reference purposes
, comma	Used to separate record labels within the same request, or to separate chained requests within a sentence
BEFORE	Used to define the location and/or relocation of records on a library tape
* asterisk	Used to identify a tape unit tag
WITH	Used to designate the record replacing another record
" " quotation marks	Used to designate additional information
END	Used to terminate a request sentence

Table 2

3. Tape Configurations

A minimum of three tapes is necessary when using the LIBADD Request Parameter:

a. a blank tape for the updated library located on tape unit A

b. a tape containing the old library located on tape unit B

c. a tape containing the records to be included in the updated library on tape unit C.

4. Example

An example of the use of the LIBADD Request Parameter is:

```
LIBADD SIN, COS, SQRT BEFORE TAN END
```

The records being added to a library tape are inserted before any existing record of the library by the use of the BEFORE delimiter. When the BEFORE delimiter is not used, the additions are stored following the last record on the library. In either case, the records are stored in the order that the record labels appear in the request. In the above example, the SIN, COS, and SQRT records are inserted before the TAN record and will be ordered in the same sequence as they appear in the request.

The BEFORE Delimiter is used to define the location and/or relocation of records being added or resequenced within a library. Any record label appearing prior to the BEFORE delimiter is inserted immediately preceding the defined library position and, as mentioned, in the sequence defined within the request sentence. There is no restriction on the number of BEFORE delimiters appearing in a request sentence; however, the location of the record defined by BEFORE (that is, the record following the BEFORE delimiter) is restricted to a record already existing on the library. The BEFORE Request Delimiter is valid with the LIBADD, LIBREP, and LIBSEQ Request Parameters only.

C. LIBDEL

1. Function

This Request Parameter is used to delete records from a library tape.

2. Limitations

When a record to be deleted is referenced by an existing library record, the dependent recoras must either be replaced or deleted.

3. Tape Configurations

A minimum of two tape units is needed when using the LIBDEL Request Parameter:

a. a blank tape for the updated library on tape unit A

b. the obsolete library tape on tape unit B

4. Example

An example of the use of the LIBDEL Request Parameter is:

```
LIBDEL SIN, COS, TAN END
```

In the above example the records labelled SIN, COS, and TAN will be deleted from the library tape located on tape unit B.

D. LIBREP

1. Function

The LIBREP Request Parameter is used to replace existing records in a library tape. When using this Request Parameter, the LIBRARIAN removes a record and replaces it with a new one. Therefore, in effect, the function performed when using this Request Parameter, is the same as when using both LIBDEL and LIBADD. The primary use, however, of the LIBREP Request Parameter is to replace a routine with a new routine which has the same label as the old routine.

2. Limitations

Great care must be taken to ensure that routines operating in conjunction with the routine which has been replaced can still operate with the replacing routine. Whenever the LIBREP Request Parameter is used, all the information pertaining to the replacing routine must be included in the request sentence. This means that cross references must be designated in the request sentence, for transfer of this information is not allowed.

3. Tape Configurations

A minimum of three tape units is needed when using the LIBREP Request Parameter:

a. a blank tape for the updated library on tape unit A

b. a tape containing the old library on tape unit B

c. a tape containing the records being added to the updated library on tape unit C.

4. Examples

When the replacing record has the same label as the record to be replaced the request would be:

```
LIBREP TAN END
```

However, if the records being replaced are not labelled with the same label as the replacing records, the request would be:

```
LIBREP EXP WITH SIN (COS, SQRT) END
```

The use of the WITH Request Delimiter designates that the record labelled EXP is being replaced by the SIN record. The record label following WITH must follow the same limitations and requirements as when using the LIBADD Request Parameter.

The use of parentheses indicates that the SIN routine references the SQRT and the COS routines; in other words, the records whose labels are enclosed within the parentheses are referenced by the record whose label immediately precedes the parentheses. The comma (,) is used to separate the record labels enclosed within the parentheses.

There may or may not be a relationship between the COS and SQRT routines. The only record labels necessary are the immediate references in the source record which is the SIN record in the above example. Using the same example above, the COS and SQRT records may reference other records; but this dependence need not be included as they are not source records.

E. LIBSEQ

1. Function

The LIBSEQ Request Parameter causes a resequencing of the records on the library tape. A new library tape is created with the records arranged in the order specified in the request. Only those records which have been designated will be rearranged, and the relative tape sequence of the unspecified records will be left unchanged. The positions of the records being sequenced are defined by the BEFORE Request Delimiter.

2. Limitations

The LIBSEQ Request Parameter may not be used in a request sentence that contains LIBNEW. The reason for this is that only those records which exist on a library tape may be resequenced.

3. Tape Configurations

A minimum of two tapes is needed when using the LIBSEQ Request Parameter:

a. a blank tape for the updated library on tape unit A.

b. a tape containing the old library on tape unit B.

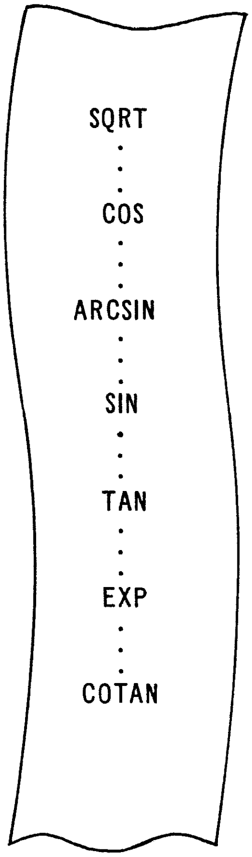
4. Example

Assume that a library tape is arranged in the sequence indicated in the illustration below. Then the use of the LIBSEQ maintenance request

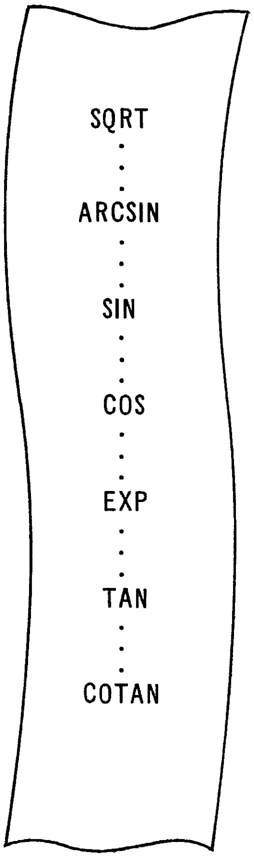
LIBSEQ SIN, COS, EXP BEFORE TAN END

would result in the resequenced library as shown.

The order of this new library is obtained as follows. The LIBRARIAN checks the first record on the tape and determines whether this record label is named in the request sentence. If it is not, it is copied onto the blank tape. If it is named in the request sentence, it is put aside and the procedure is repeated with the next record. When all of the records named in the LIBSEQ request have been encountered, they are copied onto the blank tape in the sequence designated in the request. The remainder of the tape is then copied onto the blank tape.



EXISTING TAPE



RESEQUENCED TAPE

F. LIBIDT

1. Function

The LIBIDT Request Parameter is used to assign a title to a library tape. This title appears in the library identification block, and will be discussed in greater detail in Section IV. The library title contains one to six alphanumeric characters.

2. Limitations

The LIBIDT Request Parameter must always be present when using a LIBNEW Request Parameter. However, when using LIBADD, LIBREP, or LIBSEQ, the LIBIDT Request Parameter need not be present in the request; if the LIBIDT Request Parameter is not used, the title of the old library will be duplicated on the updated library tape.

3. Tape Configurations

The minimum tape configuration required when using the LIBIDT Request Parameter is dependent upon the Request Parameter being used in conjunction with LIBIDT. For example, the minimum tape configuration requirement when using LIBADD is three tape units. Therefore, when LIBIDT is used in conjunction with LIBADD, three tape units are required.

4. Example

An example of the use of the LIBIDT Request Parameter is:

```
LIBIDT TRANS1, LIBADD SQRT (SIN, COS) “221012313415" BEFORE EXP END
```

In the above example, the library title is renamed TRANS 1; and EXP, SIN, and COS are assumed to be existing records in the obsolete library. The record labelled SQRT will be inserted before the record labelled EXP, and SQRT cross-references the SIN and COS records. The records labelled SIN and COS will not be relocated.

The use of the quotation marks (" ") within a request is limited to the function of designating additional information about a record being included within the library. The additional information, which can be any information the user wishes to include for any specific record is contained in information blocks. An example of additional information is the date a routine was written and the programmer responsible. Additional information must be written within the request in octal characters. If alphabetic information is desired, the coding must conform to FIELDATA code. A further discussion of information blocks and of this Request Delimiter follows in Section IV.

G. LIBDUP

1. Function

The LIBDUP Request Parameter is used for duplicating tapes, and is not restricted to reproducing LIBRARIAN produced tapes. An exact copy of the specified tape is produced and compared block for block to ensure accurate reproduction.

Up to eight copies of the original tape may be produced; the number to be duplicated is determined by the Job Request to the Executive System.

2. Limitations

The LIBDUP Request Parameter may be used with any other Request Parameter, but care must be exercised to define the proper tape unit for duplication. This will be explained further in the following paragraphs. The tape being duplicated must always appear on tape unit A.

3. Tape Configurations

A minimum of two tapes is needed when using the LIBDUP Request Parameter:

a. the tape being duplicated on tape unit A

b. a blank tape for reproducing on tape unit B.

Note that the tape assignments do not follow the normal rule.

4. Examples

a. An example of the use of the LIBDUP Request Parameter when duplicating a non-LIBRARIAN produced tape is:

```
LIBDUP END
```

b. An example of the use of LIBDUP when the tape being duplicated is a LIBRARIAN produced tape, and the request sentence contains more than one Request Parameter is:

```
LIBIDT TRIG3, LIBADD EXP, SIN, COS, LIBDUP *D END
```

In the above example, the library will have the title .TRIG3; and the records labelled EXP, SIN, and COS will be added to the library. The tape unit requirements for the above example are the same as when using the LIBADD Request Parameter (i.e., a blank tape on tape unit A, the obsoletê library on tape unit B, and the records being included on tape unit C.) However, the request sentence contains the request for duplication which requires a blank tape on tape unit D.

When writing a LIBRARIAN request, care must be taken to ensure that the LIBRARIAN is given information about tape unit allocations. The tape tags appearing in the parameter request must be identical to the coded facility tags of the LIBRARIAN, of which there are nine, A through I. When a LIBRARIAN request is submitted to the Executive System, the number of tape assignments on the facility card is correlated in consecutive order with the LIBRARIAN tags. Thus, if three tags are required, A, B, and C will be assigned. In the same manner, if five tags are required, A, B, C, D, and E are assigned. However, the tape unit tags do not have to be specified when only the minimum tape configurations are needed.

The asterisk (*) Request Delimiter is used to identify a tape unit tag. It should be noted that *A must never be used except with the LIBCAT Request Parameter. Immediately following the asterisk is the tape unit tag followed by either the records which are located on the particular tape, or the blank tapes for duplication. Thus, in the above example, the new library labelled TRIG3 will be duplicated on tape unit D after it has been updated. Thus, the new library appears on tape unit A and a copy on tape unit D.

c. An example of the use of the LIBDUP Request Parameter when several copies of the original library with updated library titles is desired would appear as:

```
LIBIDT IOLIB, LIBDUP *B IOLIB1, *C, *D IOLIB2 END
```

In the above illustration, the library titled IOLIB is to be duplicated. The copies on tape units B and D will contain the updated title (IOLIB1 and IOLIB2) specified in the request. These two copies will be compared block by block; however, because the library identification block is not the same, this block will not be compared. Wherever a title is missing in an asterisked portion of the request, an exact copy of the original tape is made and compared block by block, as with tape unit C in the above example.

H. LIBCAT

1. Function

The LIBCAT Request Parameter causes the LIBRARIAN to catalogue one or more specified libraries. A catalogue is the listing of record labels according to their location on the specified library tape. Also included in the catalogue will be cross-references and the information blocks of each record.

2. Limitations

Cataloguing is an automatic procedure when building, updating, or resequencing a library tape. However, any library tape used in updating may be catalogued in addition to the new library by specifying the library titles.

3. Tape Configurations

A minimum of one tape unit is needed when using the LIBCAT Request Parameter. The tape unit assignments must be specified when using the LIBCAT Request Parameter; the tape units are specified by the asterisk (*) Request Delimiter and the tape tags to be used are A through I. However, the tape unit tag A will be used only when LIBCAT is not used in conjunction with another Request Parameter.

4. Examples

a. An example of the use of the LIBCAT Request Parameter is:

```
LIBCAT *A TRANS END
```

In the above example, the LIBRARIAN lists the record labels with their cross-references and information blocks according to the sequence within the library tape titled TRANS. The library will appear on tape unit A.

b. An example of the use of the LIBCAT Request Parameter in a request containing other Request Parameters is:

```
LIBADD SIN, COS, TAN BEFORE SQRT, LIBCAT *C TRANS END
```

In the above example, the SIN, COS, and TAN records are being added to the library tape, and will be inserted immediately preceding the SQRT routine. Following the rules of tape configurations, the new library will be on tape unit A, the old library on tape unit B, and the tape containing the records to be added on tape unit C. The tape, labelled TRANS, on tape unit C will be catalogued.

I. LIBINP

1. Function

The LIBINP Request Parameter is used to define the input tape format. The records to be included in a library must be on magnetic tape when updating or building a new library; and the input tapes may be other libraries, prestored symbolic tapes, assembler or compiler produced tapes, and any combinations of input tapes. The asterisk (*) Request Delimiter is used with LIBINP to denote the logical tape unit of the input tape. Various formats are available:

a. ROC input tapes
b. other library tapes
c. off-line card-to-tape conversion
d. on-line card-to-tape conversion

These formats will be described in greater detail in the following paragraphs.

2. Tape Configurations

The tape configurations are defined by the user with the use of the asterisk (*) Request Delimiter. The tape configurations are dependent upon the Request Parameters used within the request sentence.

3. Examples

Examples of the use of the LIBINP Request Parameter for defining any of the four formats are:

a. ROC Input Tapes

The following example illustrates the use of the LIBINP Request Parameter defining ROC input:

```
LIBINP *B ROC ...
```

ROC defines the input tape format and *B represents the logical tape assignment of the defined input tape.

b. Other Libraries

The request sentence when using other library tapes as input to build a new library would be:

```
LIBINP *B TRIG1 ...
```

TRIG1 is the title of the input library tape and *B represents the logical tape assignment to the LIBRARIAN.

When transferring a record from a library source in updating or building a new library, only the record label need be cited in the Parameter Requests of LIBNEW, LIBADD, or LIBREP. When a record has cross-references and/or an information block on the input library tape, the LIBRARIAN includes this information automatically. An example is:

```
LIBIDT SERIES, LIBINP *C TRIG1, LIBADD *C TAN END
```

For the above example, assume that the TAN record cross-references SIN and COS records and also has an information block.

The TAN record appears on the library tape labelled TRIG1, which is the input tape and is located on tape unit C. The library to which the TAN record is being added is located on tape unit B (this does not need to be specified for it follows the rules of tape configurations specified throughout the above discussions), and the updated library tape SERIES is located on tape unit A. The LIBRARIAN inspects the TAN entry of the TRIG1 directory for the information block and cross-references. The SIN and COS records are added to the SERIES library, if and only if, these records do not appear in the directory of the library on tape unit B or on the tape located on tape unit B. The COS and SIN records are inspected in the TRIG1 directory for listings of their cross-references and information block, and in turn their cross-references, if any, are inspected, etc. These record labels are added to the SERIES directory and their information blocks are copied onto the SERIES library with the actual records.

c. Off-Line Card-to-Tape Conversion

An example of the use of the LIBINP Request Parameter to define a pre-stored card symbolic input tape produced by the off-line card-to-tape converter is:

LIBINP *C CARD x . . .

In the above example, the input tape is on tape unit C and CARD indicates the format of the input. The x represents the number of cards to be combined into one block on the library tape, where $1 \leq x \leq 18$. When one card per block is desired, the x portion may be omitted.

When using the off-line equipment to produce a prestored symbolic tape, two additional cards will be necessary:

(1) Label Card

This card contains the label of the record to be included in the library, and must precede the cards of the record. The label card must contain the sentinel /././. in columns 1-6; starting in column 13 is the record label which may consist of a maximum of 12 characters. If the record label does not consist of 12 characters, it is left justified and space-filled.

(2) End of Record Card

This card concludes each symbolic card record deck, and has the sentinel *ENDRC in columns 1-6.

d. On-Line Card-to-Tape Conversion

An example of the use of the LIBINP Request Parameter when using prestored card symbolic input tapes produced by the on-line card-to-tape conversion routine (supplied in the software package LION) is:

```
LIBINP *B NCARD ...
```

The logical tape assignment of the input tape is defined by *B, and NCARD defines the tape format. When using this tape format, the block lengths of the input tape are duplicated on the library.

III. LIBRARY RETRIEVAL

The retrieval section of the LIBRARIAN is used to obtain information about records. The retrieval section is made an integral part of a program with which it is associated. This program, which may be an assembler, compiler, or load routine, references the retrieval section whenever necessary. For example, in the case of the SLEUTH Assembly System, the retrieval section of the LIBRARIAN is called upon by SLEUTH for the incorporation of subroutines at assembly time.[1] If subroutines are to be added at load time, CLAMP, the Relative Load Routine, will call upon the retrieval section.

The information which follows is presented in order to facilitate the use of the retrieval section in conjunction with those assemblers, compilers, or load routines produced by the user.

The retrieval section consists of routines, each of which processes a specific request. The communication between these routines and the user is effected by parameter words which will be described in greater detail in "Retrieval Requests" below. The Retrieval routines are entered by a Return Jump. Because the Retrieval routines are assembled with the program using them, all tags and labels used in the Retrieval routines begin with the letters "LR" to reduce the possibility of duplication.

The retrieval section only accepts LIBRARIAN produced tapes in the format described in Section IV.

A. Retrieval Requests

The Retrieval routines will list all cross-references for any number of records; but only one information block will be retrieved at any one time. There are four Retrieval Requests available to the user:

1. Initialize Request
2. Information Block Request
3. Cross-References Request
4. Order Label List Request

[1] If the SLEUTH source program contains a macro-call, the Assembly System first determines if this is a call for (1) a program-defined macro, (2) a system macro, or (3) a library record. If the first two checks fail, then control is given to the retrieval section of the LIBRARIAN (an image of which is contained within the Assembly System). The retrieval section then proceeds to incorporate the desired record into the object program. See manual on SLEUTH Assembly System.

The user is required to provide a list of record labels for the last two of these requests. This labels list contains the two word record labels, space filled through 12 characters if necessary. The labels are listed in consecutive 2 word locations. In the case of the Cross-References Request, the labels list must have sufficient space at the end to contain any cross-references that are to be added.

1. Initialize Request

   The Initialize Request reads the library tape directory into core and must precede any of the other Retrieval Requests when any of the following have occurred:

   a. The Library Retrieval routine has been loaded and must be readied for use.

   b. The physical unit or channel of the library tape is changed.

   c. The library tape is changed.

   d. The user has read record(s) from the library tape.

   The Initialize Request is initiated by return jumping to the entry point LRINTL.

2. Information Block Request

   When the Information Block Request is used, the information block for the record whose label is in two consecutive A registers is read at LRINBK. This request is initiated by putting the record label (space filled through 12 characters if necessary) in two pre-determined A registers and return jumping to the entry point LRINFO.

3. Cross-References Request

   When the Cross-Reference Request is used, all the cross-reference labels are added to the labels list. The user originally references the Cross-Reference Request with a list of labels of one or more records for which the user wants the cross-references. The Retrieval routine adds the cross-references to the labels list. Only the labels of those records not in the list are added; duplicates are eliminated.

The Cross-References Request is initiated by storing the address of the first label in the labels list in the right half of a pre-determined A register, the number of record labels in the list in the left half of the same A register, and the size of the labels list storage area into another predetermined A register. A return jump to the entry point LRXREF is then executed.

4. Order Label List Request

When the Order Label List Request is used the record labels on the labels list will be relisted in the order of appearance on the library tape. This request is initiated by storing the address of the first label in the labels list in a predetermined A register, the number of labels in the labels list into another predetermined A register and return jumping to the entry point LRLIST.

B. Retrieval Error

Any errors found during the execution of any of the Retrieval routines will be indicated by an error status word. If the error status word is zero when returning from the Retrieval Request, the request was completed without error. However, if an error does occur, the type of error will be defined by the error status word.

C. Record Retrieval

The user must call upon the Executive System directly for reading records. In order to read library records, the user must be familiar with the library tape format described in Section IV. When the user does read records from a library tape, the next reference to the Retrieval routine must be an Initialize Request.

A suggested method of reading records from a library tape follows. The Retrieval routine will never position the library tape beyond any of the records on the tape. Therefore, once the labels of the records to be read have been ordered by an Order Label List Request, the tape may be positioned in front of the first record to be read by initializing a Search-Read Tape Forward Request by the Executive System using the first 6 characters of the record label as the search sentinel. This will position the tape just beyond the record identification block. A check should be made on the second 6 characters of the label to ensure accuracy. The record may be read block by block by the Executive Systems however, the user must check for the end-of-library record block. Or a Read Tape Forward With Sentinel Check Request may be initiated through the Executive System with the first word of the end-of-library record block (*TRMRC) as the sentinel. Since the next record desired is beyond the record just read, this process may be repeated until all desired records are read.

IV. LIBRARY TAPE FORMAT

The Library tape format is comprised of three sections:

A. Library Identification section
B. Directory Information section
C. Library Records section

A. Library Identification

The Library Identification consists of one block containing four words:

	35 18	17 0
Word 1	L I B	A R Y
Word 2	Library Title (6 characters)	
Word 3	XREF Length	DIRECT Length
Word 4	checksum	

The first word contains the standard Library Identification, LIBARY, in FIELDATA format.

2. Word 2.

This word contains the library source title defined by LIBIDT in FIELDATA format.

3. Word 3.

The third word contains the lengths of the XREF and DIRECT table for the library.

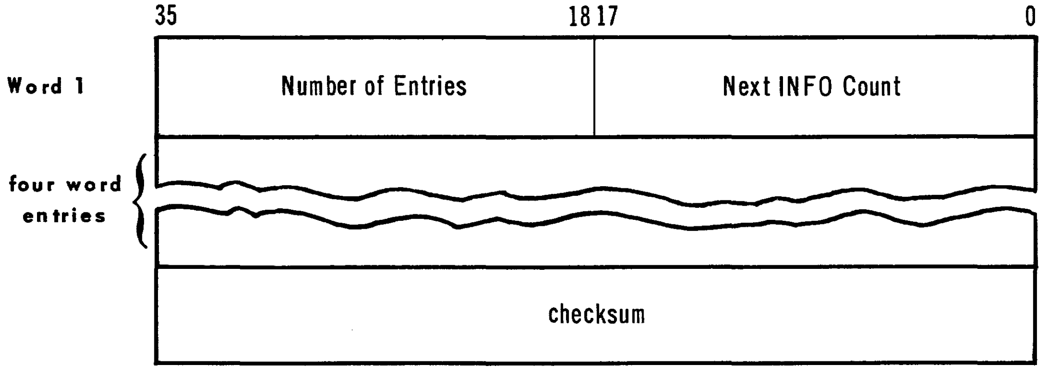
4. Word 4.

The fourth word contains the block check sum.

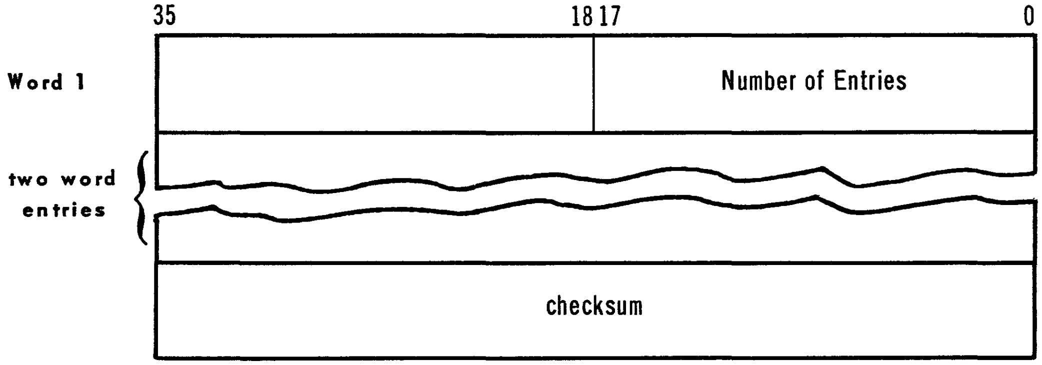
B. Library Directory

The Library Directory contains the DIRECT and XREF tables and the record information blocks. The Direct table consists of one block and follows the identification block. The Direct table format is described in Section V. The XREF table also consists of one block and directly follows the DIRECT block. The XREF table will be described in greater detail in Section V.

The DIRECT block would appear as:



And the .XREF block would be in the following format:



Included in the Directory portion of the tape will be information blocks, which contain any additional information pertaining to the functional use of any record contained in the library. The information block for each record is entered on the library at the time the record is added to the library tape. When an information block is to be included, it should follow the right parenthesis (or close parenthesis) or the record label. The information is enclosed within quotation marks, which defines the beginning and end of the information, and is to be written in the request in FIELDATA octal characters. The octal characters are converted to binary words and then written on tape. The octal words are separated in the request by commas. When a full computer word (12 octal characters) is not utilized, the characters will be right justified after conversion to binary code.

When locating a record, which has an information block, and utilizing the BEFORE Parameter the format is:

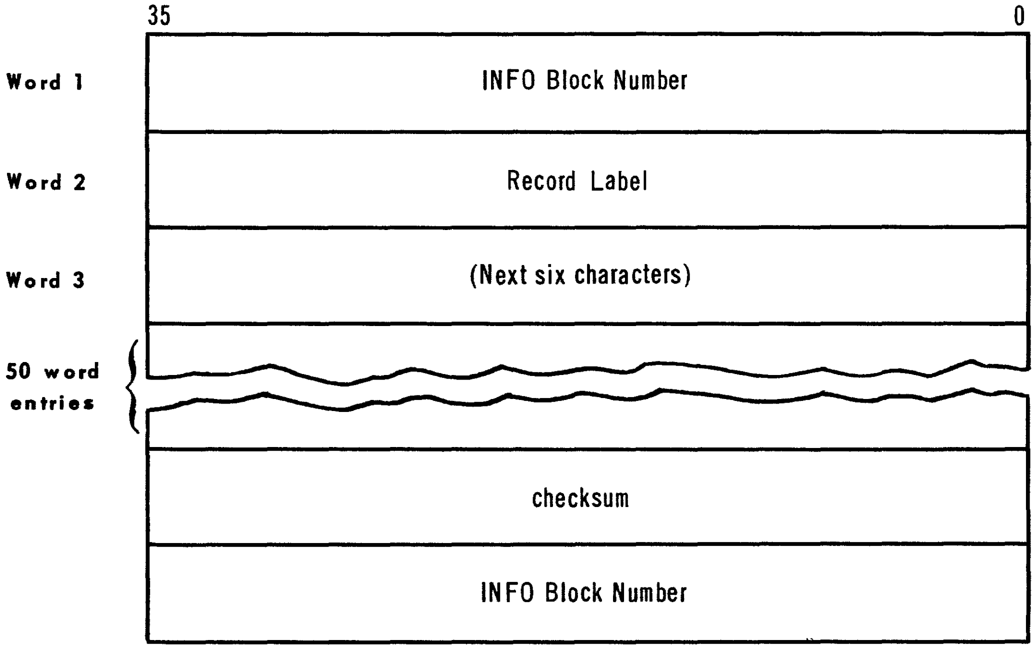
TAN (COS,SIN) "15122323" BEFORE ARCSIN

TAN "12220623" BEFORE ARCSIN

Because of the varying lengths of the additional information per record, the information blocks will be located on the library tape in individual blocks. All of the information blocks will be written on the library tape after the XREF block and before the location of the library records.

Each information block will be assigned a number in the library directory. This number will be the first and last word of each block and facilitates the search-read operation. The number will determine whether the search-read function is to be performed in the backward or forward mode depending on the magnitude of the last information block number which was read.

The record label associated with each information block is contained in words two and three of each block. The word preceding the last word of each information block contains the block check sum. An illustration of an information block is:



C. Library Records

The Library Records Section contains the stored records. The identification block of each record contains 12 space filled characters of the record label in the first two words, and the first six characters of the record label in the last word. This format facilitates the forward or backward search-read mode of operation. The third and fourth words contain the sentinel $/$/$/. The fifth word contains the checksum, while the sixth word contains the first six characters of the record label. An illustration of the block is:

	35					0
Word 1	Record Label					
Word 2	(next six characters)					
Word 3	$	/	$	/	$	/
Word 4	$	/	$	/	$	/
Word 5	checksum					
Word 6	Record Label (first six characters)					

Following the identification block will be several blocks necessary to contain the record.

The end-of-record block is in the following format:

	35					0
Word 1	*	E	N	D	R	C
Word 2	Record Label					
Word 3	(next six characters)					
Word 4	checksum					

V. DIRECTORY TABLES

The directory tables consists of the library directory (DIRECT), the cross-reference listing for each record (XREF), and the information blocks for each record (INFO). The latter two tables, containing variable length entries, supplements the library directory. The library directory is composed of fixed length entries only. The directory tables are generated by the LIBRARIAN as a result of the parameters.

A. Directory Table Format (DIRECT)

The first word of the DIRECT table contains the number of entries in the library and the last assigned information block count number. The information block count is needed for future assignment of block count numbers when updating the library.

Each record of the library will have a four word entry in the DIRECT table. The first two words contain the record label, whose maximum size will be 12 characters. When the record label contains less than 12 characters, the remaining positions of the two words will be spaced filled with a FIELDATA code of 05. For example, the six character label of ARCSIN would appear in the DIRECT table in FIELDATA code as:

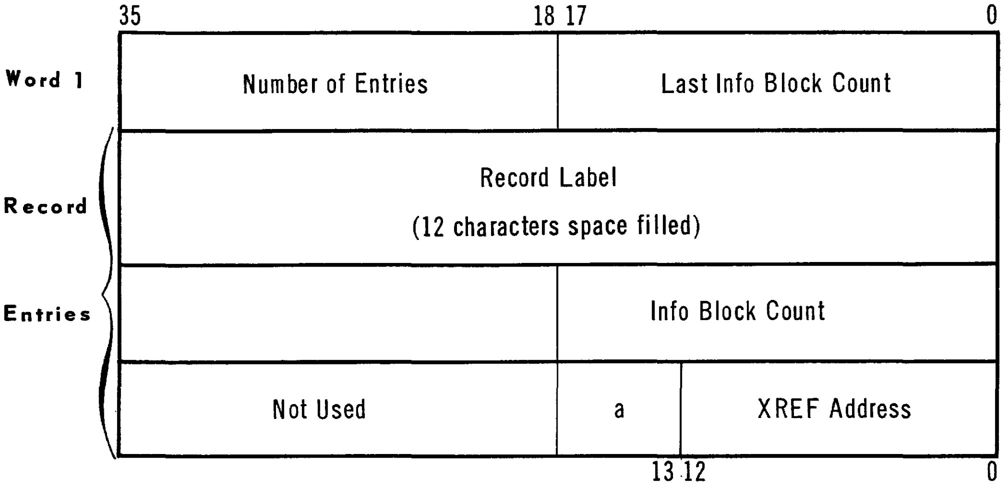
RECORD LABEL	DIRECT ENTRY
ARCSIN	***Word 1*** ***062710301623*** ***Word 2*** ***050505050505***

or, the three character label of SIN would appear in the DIRECT in FIELDATA code as:

RECORD LABEL	DIRECT ENTRY
SIN	***Word 1*** ***301623050505*** ***Word 2*** ***050505050505***

The third word contains the information block count of this library entry. The last word of each entry contains a value representing an index into the XREF table, and the number of core locations needed for storing the cross-reference labels.

Bits 35-18 of the third word of each DIRECT table entry are used during record retrieval and have been described in an earlier section. The DIRECT is illustrated in the following:



	35–18	17–13	12–0
Word 1	Number of Entries	Last Info Block Count	
Record Entries	Record Label (12 characters space filled)		
		Info Block Count	
	Not Used	a	XREF Address

a = Number of core locations needed for XREF storage

B. Cross-Reference Table (XREF)

The first word of the XREF table contains the number of two word entries in XREF. The record labels of the cross-references are space filled and appear in XREF in the identical format as the record labels in the DIRECT table.

C. Information Block Storage Area (INFO)

Each entry of INFO, related to a DIRECT entry, is stored on the library tape in individual blocks, and the minimum length of the storage area is 50 words. If a greater area is required, the area may be enlarged by the EXEC TAL Parameter Card (see manual on the Executive System.)

INDEX

INDEX

UNIVAC

DIVISION OF SPERRY RAND CORPORATION