User's Guide to LIBEDIT

(2/71) V0 LIBEDIT

CONTENTS

I. Introduction

LIBEDIT is a system routine which can be used for the record management of a program library. The LIBEDIT processor is called by a control card which contains parameters that select files and options for LIBEDIT input and output.

The input to the LIBEDIT program consists of some combination of the following three files:

(1) Old Program Library
This file contains a previously generated library of programs which is to be edited in some way.

(2) Directives
This file contains the editing instructions which are to be processed by the LIBEDIT program.

(3) Replacement Library
This file contains decks which are to replace or to be inserted between records in the old program library to produce a new program library.

The output from a LIBEDIT job consists of the following two files:

(1) New Program Library
This file contains the library of programs which was produced by the LIBEDIT directives acting on the old program library. The new program library is usually stored and then used later as an old program library when another LIBEDIT job or a MODIFY job is required.

(2) List Output
This file contains a record of all actions taken by the LIBEDIT processor. All successful editing operations are noted and, if necessary, error diagnostics are listed for any directive which could not be followed.

2. Deleting Decks from a Library

The following job illustrates how to delete decks from a program library:

```
12345,SMITH,CM25000,T8.
PFILES(GET,NPL,X=OLD)
LIBEDIT(B=0)
PFILES(PUT,NPL,X=NEW)
7-8-9 (end-of-record)
*DELETE OPL/DATA
6-7-8-9 (end-of-information)
```

This job edits a MODIFY program library which is stored in a permanent file named NPL. The "B=0" parameter on the LIBEDIT control card specifies that there is no replacement file for this job. By default the old program library is OLD, the directive file is INPUT, the new program library is the file NEW and list output is on the file OUTPUT. See appendix A for a list of all possible control card options.

There is only one LIBEDIT directive on the INPUT file. That DELETE instruction causes the record named DATA in OPL format to be deleted in the new program library. All other records are transferred to the new program library from the old program library and a MODIFY directory deck is automatically appended. See appendix B for a list of all possible LIBEDIT directives.

3. Adding and Replacing Decks in a Library

The following job illustrates how to add and replace decks in a program library:

```
12345,SMITH,CM25000,T8.
COPYBF(INPUT,SOURCE)
MODIFY(F,P=0,N=LGO)
PFILES(GET,NPL,X=OLD)
LIBEDIT.
PFILES(PUT,NPL,X=NEW)
7-8-9 (end-of-record)
SUB2
      SUBROUTINE ROOT5(F,X)
      E=1.0/5.0
      F=ABS(X)**E
      IF(X.LT.0.0)F=-F
      RETURN
      END
7-8-9 (end-of-record)
DATA
1SMITH,JOSEPH      117-34-1398    26.9    35    X20
1JONES,JOHN        654-22-7603    12.0    19    X20
1BAKER,FRANK       026-33-4545    24.5    27    X20
6-7-9 (end-of-file)
*REWIND,SOURCE
*CREATE,SOURCE
7-8-9 (end-of-record)
*TYPE OPL
*INSERT MAIN,SUB2-DATA
6-7-8-9 (end-of-information)
```

This job edits a MODIFY program library which is stored in a permanent file named NPL. Two decks named SUB2 and DATA are first processed by MODIFY and put into file LGO in program library format. (See document VO-MODIFY for a discussion of the MODIFY program.) Then LIBEDIT is called using all default parameters, i.e. the replacement file is LGO, the old program library is OLD, the directive file is INPUT, the new program library is NEW and list output is on OUTPUT.

There are two LIBEDIT directives on the INPUT file. The *TYPE card specifies that all decks processed should be assumed to be in OPL (program library) format unless explicitly instructed otherwise by some following directive. The *INSERT card causes all decks on the replacement file between SUB2 and DATA to be inserted after the deck named MAIN in the new program library. If any of the inserted decks was already in the old program library, then that previous version is automatically deleted from the new program library. All other decks are transferred intact from the old program library to the new program library.

Appendix A - LIBEDIT Control Card Options

Format: LIBEDIT $(p_1,p_2,\ldots,p_n)$

where p_i = Parameters which may be in the following general formats:

a

a = fname

a = 0

Parameter	Action Taken
I = 0	Use no directive card input
I = fname	Use file fname for directive card input
P = 0	Use no old library file
P = fname	Use file fname for old library file
N = fname	Use file fname for new library file
L = 0	Suppress correction listing
LO = fname	Use file fname for correction listing
B = 0	Do not use default replacement file
B = fname	Use file fname for replacement file
C	Copy new library file to old library file after processing
R	Do not rewind old or new library file after processing
V	Call VFYLIB after the LIBEDIT operation

The LIBEDIT control card calls the LIBEDIT program and specifies additional information in the parameter list. Parameters can be in any order and are not mandatory. For any or all parameters omitted from the control card, LIBEDIT assumes a default value:

I = INPUT	LO = OUTPUT	C deselected
P = OLD	B = LGO	R deselected
N = NEW		V deselected

The LIBEDIT processor requires a minimum of about 25000 words and there is no automatic field length adjustment.

Appendix B - LIBEDIT Directives

LIBEDIT processes directives independent of their order of appearance except for the cards grouped by *TYPE, *FILE, and *NRANDOM and where several inserts take place in the same record. If LIBEDIT cannot process the specified combination of directives, it lists the conflicting directive cards, issues an error diagnostic message, and aborts the job.

Directives begin with an asterisk in column 1. All cards in the directive file that do not begin with an asterisk are treated as continuation cards to the most recent card that has an asterisk in column 1. If there are no directives, LIBEDIT replaces the records of the file OLD which have the same name and type as the records on the file LGO and writes a new library file on the file NEW. LIBEDIT directives are:

*FILE	*INSERT	*ADD	*NOREP
*NRANDOM	*BEFORE	*BUILD	*RENAME
*REWIND	*DELETE	*COMMENT	*REPLACE
*TYPE	*IGNORE	*DATE	*COPY

*FILE Card

Format: *FILE fname

where fname = Name of additional replacement file; subject to operating system restrictions on file names.

The *FILE card declares a secondary file to be an additional file that contains replacement records. LIBEDIT directives following a *FILE card refer to records on the declared replacement file. If fname is *, LIBEDIT uses the file names on the control card.

*NRANDOM Card

Format: *NRANDOM fname

where fname = Name of non-random replacement file; subject to operating system restrictions on file names.

The *NRANDOM card declares a secondary file to be a non-random replacement file. This means that all manipulation of the file must be done by read, skip, and backspace operations (and not by random read operations). LIBEDIT directives following a *NRANDOM card refer to records on the non-random replacement file.

*REWIND Card

Format: *REWIND fname

where fname = Name of file to rewind

For the *REWIND card, LIBEDIT rewinds the specified file before and after the editing process.

*TYPE Card

Format: *TYPE Type

where type = Specifies internal record structure:

Type	Structure	
PP	Peripheral processor program	
COS	Chippewa format CPU program	
REL	Relocatable CPU program	
OVL	SCOPE format CPU overlay	
ULIB	User library program	
OPL	Old program library deck	i.e. MODIFY deck format
OPLC	Old program library common deck	
ABS	Multiple entry point overlay	
TEXT	Unrecognizable as a program	

With the *TYPE card, the user can specify the default type of record to which subsequent LIBEDIT directives refer. A *TYPE card is in effect until the next *TYPE card appears. If there is no *TYPE card, the default type is TEXT.

Example:

```
*TYPE COS
*INSERT GEORGE, MARY
*DELETE HENRY - IDA
is equivalent to:
*INSERT COS/GEORGE, COS/MARY
*DELETE COS/HENRY - COS/IDA
```

*INSERT Card

Format: *INSERT $type_1/name_1, type_2/name_2$ or

*INSERT $type_1/name_1, type_2/name_2$ - $type_3/name_3$ or

*INSERT $name_1, name_2$ or

*INSERT $name_1, name_2$ - $name_3$

where $type_i$ = Same as type described with the *TYPE card, if omitted, LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of record on old library file after which to insert the specified record or records.

$name_2$ = Name of record from replacement file to insert after record $name_1$. If $name_2$ = 0, LIBEDIT inserts a zero length record. If $name_2$ = *, LIBEDIT inserts all records of the specified or implied type.

$name_2$-$name_3$ = Ellipsis of group of records from replacement file to insert after record $name_1$. If $name_3$ = *, LIBEDIT inserts all records to an End-of-File mark.

The *INSERT card directs LIBEDIT to insert a record or group of records from the current replacement file after a specified old library record for transcription to the new library file. The current replacement file is the last replacement file specified by *FILE or LIBEDIT control card. As described above, asterisk and zero have special meaning when they appear in lieu of record names. Insertion of a record or records causes automatic deletion of the record(s) having the same name(s) and type(s) from the old library file.

Example:

```
*TYPE TEXT
*INSERT OPL/LEA, OSCAR - *
```

These cards direct LIBEDIT to insert after the OPL deck LEA on the old library file all TEXT type records from OSCAR to an End-of-File mark. If any of these TEXT records has the same name as a TEXT record that is already on the old library file, the old TEXT record is deleted for transcription to the new library file.

*BEFORE Card

Format: *BEFORE $type_1/name_1,type_2/name_2$ or

*BEFORE $type_1/name_1,type_2/name_2$ - $type_3/name_3$ or

*BEFORE $name_1,name_2$ or

*BEFORE $name_1,name_2$ - $name_3$

where $type_i$ = Same as type described with the *TYPE card; if omitted LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of record on old library file before which to insert the specified record or records. If $name_1$ = *, LIBEDIT inserts the records before the End-of-File mark.

$name_2$ = Name of record from replacement file to insert before record $name_1$. If $name_2$ = 0, LIBEDIT inserts a zero length record. If $name_2$ = *, LIBEDIT inserts all records of the specified or implied type.

$name_2$ - $name_3$ = Ellipsis of group of records from replacement file to insert before record $name_1$. If $name_3$ = *, LIBEDIT inserts all records to an End-of-File mark.

The *BEFORE card directs LIBEDIT to insert a record or group of records from the current replacement file before a specified old library record for transcription to the new library file. The current replacement file is the last replacement file specified by *FILE or LIBEDIT control card. As described above, asterisk and zero have special meaning when they appear in lieu of record names. As for *INSERT, insertion of a record or records causes automatic deletion of the old record(s) having the same name(s).

*DELETE Card

Format: *DELETE $type_1/name_1$ or

*DELETE $type_1/name_1$ - $type_2/name_2$ or

*DELETE $name_1$ or

*DELETE $name_1$ - $name_2$

where $type_i$ = Same as type described with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of record on old library file to delete. If $name_1$ = *, LIBEDIT deletes all records of the specified type.

$name_1$ - $name_2$ = Ellipsis of group of records on old library file to delete. If $name_2$ = *, LIBEDIT deletes records to an End-of-File mark.

The *DELETE card directs LIBEDIT to suppress transcription of a record or group of records from the old library file to the new library file.

Example:

```
*DELETE PP/LAD-REL/RUN
```

This card directs LIBEDIT to delete the existing peripheral processor program LAD, the relocation CPU program RUN, and all logical records between them.

*IGNORE Card

Format: *IGNORE $type_1/name_1$ or

*IGNORE $type_1/name_1$ - $type_1/name_2$ or

*IGNORE $name_1$ or

*IGNORE $name_1$ - $name_2$

where $type_i$ = Same as type described with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of record from replacement file to ignore during processing. If $name_1$ = *, LIBEDIT ignores all records of the specified type.

$name_1$ - $name_2$ = Ellipsis of group of records from replacement file to ignore during processing. If $name_2$ = *, LIBEDIT ignores records to an End-of-File mark.

The *IGNORE card directs LIBEDIT to ignore a record or group of records on the current replacement file during processing.

Example:

```
*FILE WOMAN
*IGNORE FRAN - *
```

LIBEDIT ignores the program FRAN of the current type and all following programs to an End-of-File mark on the replacement file WOMAN.

*ADD Card

Format: *ADD lib, $type_1/name_1$ or

*ADD lib, $type_1/name_1$ - $type_2/name_2$ or

*ADD lib, $name_1$ or

*ADD lib, $name_1$ - $name_2$

where lib = Specifies the library or directory:

lib	Library or Directory
DDS	Dead start Library
RCL	Resident Chippewa Library
RSL	Resident Subroutine Library
RPL	Resident Peripheral Library
PLD	Peripheral Library Directory
CLD	Chippewa Library Directory
SLD	SCOPE Library Directory
ULD	User Library Directory

$type_i$ = Same as type descripted with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

name = Name of program to add to the specified library or directory. If $name_1$ = *, LIBEDIT adds all programs of the specified type.

$name_1$ - $name_2$ = Ellipsis of group of programs to add at the end of the specified library or directory. If $name_2$ = *, LIBEDIT adds all programs from $name_1$ to an End-of-File mark.

The *ADD card appends the specified program or programs to the specified library or directory for transcription to the new library file.

*BUILD Card

Format: *BUILD dname

where dname = Name of directory record

The *BUILD card requests LIBEDIT to construct and append a directory deck in MODIFY format to the new library file. Moreover, if the old library file has such a directory, LIBEDIT automatically generates a new directory deck without explicit *BUILD request.

*COMMENT Card

Format: *COMMENT type/name comment

where type = Same as type described with the *TYPE card; if ommitted, LIBEDIT assumes the last type given on a *TYPE card.

name = Name of program on replacement or old library file.

comment = Any string of 40_{10} characters or less that is suitable as a comment.

The *COMMENT card adds a comment to the Prefix (77) table for a program on a replacement or the old library file.

*DATE Card

Format: *DATE type/name comment

where type = Same as type described with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

name = Name of program on replacement or old library file.

comment = Any string of 40_{10} characters or less that is suitable as a comment.

The *DATE card adds the current data and submitted comment to the Prefix (77) table for a program on a replacement or the old library file.

*NOREP Card

Format: *NOREP fname

where fname = Name of replacement file to become a no-replace file

The *NOREP card declares the specified replacement file to be a no-replace file. LIBEDIT then does not replace all records of the old library file with records on the no-replace file having identical names, but replaces records from a no-replace file selectively according to *REPLACE, *INSERT, and *BEFORE cards.

*RENAME Card

Format: *RENAME type/$name_1$,$name_2$

where type = Same as type described with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of program on replacement or old library file to have a new name.

$name_2$ = New name of program.

The *RENAME card assigns a new name to a record on the old library or current replacement file for transcription to the new library file.

*REPLACE Card

Format: *REPLACE $type_1$/$name_1$ or

*REPLACE $type_1$/$name_1$ - $type_2$/$name_2$ or

*REPLACE $name_1$ or

*REPLACE $name_1$ - $name_2$

where $type_i$ = Same as type described with the *TYPE card; if omitted, LIBEDIT assumes the last type given on a *TYPE card.

$name_1$ = Name of record from replacement file to replace on old library file.

$name_1$ - $name_2$ = Ellipsis of group of records from replacement file to replace on old library file.

The *REPLACE card directs LIBEDIT to selectively replace records from a current replacement file that has been declared a no-replace file (see the *NOREP card description).

Example:

Suppose file FRUIT has records APPLE, CHERRY, GRAPE, and ORANGE. Records APPLE and CHERRY of the old library file are to be retained. The sequence:

```
*FILE FRUIT
*IGNORE APPLE - CHERRY
```

and the sequence:

```
*FILE FRUIT
*NOREP FRUIT
*REPLACE GRAPE - ORANGE
```

Will produce the same effect, LIBEDIT provides both methods for the situations where discreet choice can save the effort of punching countless correction cards.

*COPY

Format: *COPY*

The *COPY card directs LIBEDIT to copy the new library file to the old library file after it has processed all correction cards.