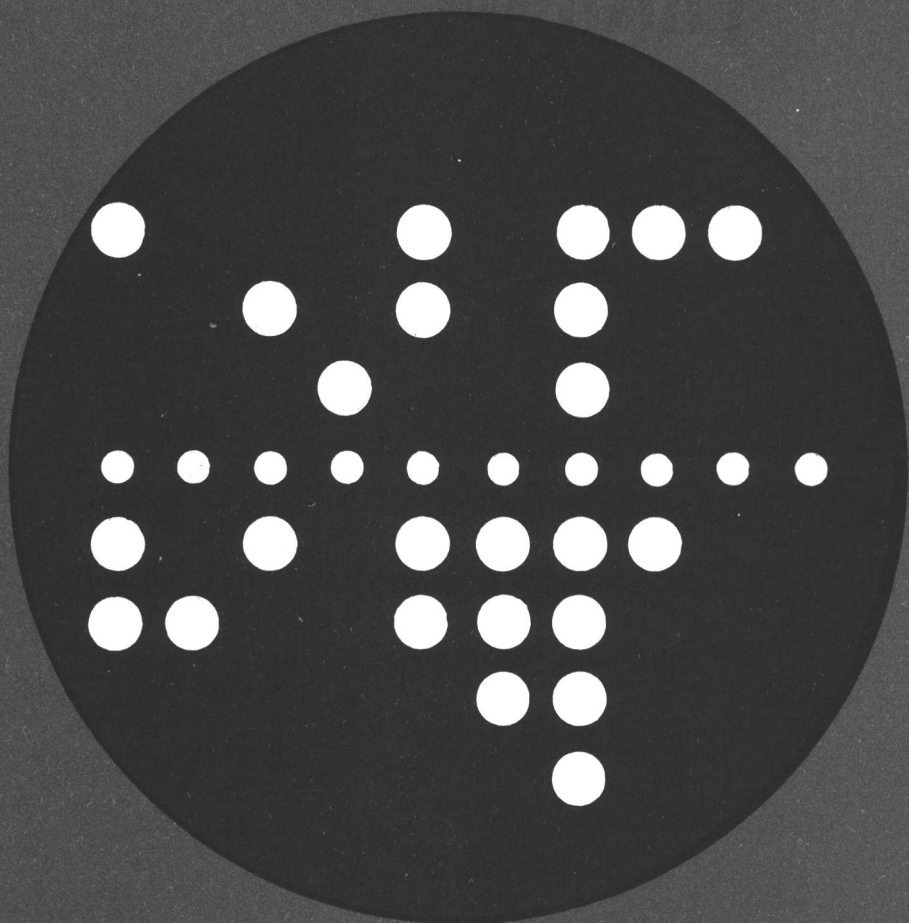


COMPUTING CENTRE NEWSLETTER

THE LIBRARIAN



Commission of the European Communities

**JOINT
RESEARCH
CENTRE**

Ispira Establishment

SPECIAL ISSUE



2

THE LIBRARIAN

**Handbook on the use of the Librarian
System as installed at the J.R.C.
Computing Centre, Ispra**

R. Meelhuysen, H.I. de Wolde

version 1, April 1978

INTRODUCTION

The Librarian is a dataset retrieval and maintenance system designed to facilitate the writing, maintaining, testing and documenting of programs. *

Source programs, object decks, test data and any other information traditionally stored on cards, may be stored through the Librarian system on disk. Updating and manipulations may be performed by a set of simple commands.

This paper describes the most common use of the Librarian. Detailed information for more complicated tasks is available at the unit "Support to Computing".

The basic element

The basic element is a record (length 80 bytes) containing a cardimage, without restrictions to the type of information contained. Each record is supplied with a sequence number in user specified columns. This may also be external to the records, e.g. columns 81-88, thus leaving all 80 columns available for information.

Module

A module consists of one or more records which have been grouped under an alphanumerical name chosen by the user.

Masterfile

A masterfile is a reserved disk space location containing one or more modules possibly from several users. The masterfile is known by the masterfile name.

The main capabilities of Librarian are listed below. This list is not exhaustive, neither are all these options discussed in this manual.

- Updating of modules
- Preparing modules for use in a consecutive step
- Temporary updating: the source module remains unchanged
- Cobol syntax checker
- Storing biographical information
- Storing output listings
- Character string search
- Character string search and modification
- Introduction of specified data in specified column
- Masterfile back-ups
- Creating new modules out of parts from other modules
- Duplicating modules to cards, disk or tape
- Condensed storage of information

THE LIBRARIAN PROCEDURE

The Librarian procedure has been catalogued under the name "LIBRAP" as follows:

```
//LIBRAP      PROC A='MASTER',E='DISPACK',K='DISK',CC=4,
//           EO=' ,EVEN'
//INS        EXEC PGM=$$URIAN,PARM='DEPTH=50,NJTA,NRJS',
//           COND=( (&CC,LT) &EO)
//STEPLIB    DD DSN=LIBRA75,DISP=(SHR,KEEP),UNIT=DISK,
//           VOL=SER=COPICB
//MASTER     DD DSN=&A,UNIT=&K,DISP=(SHR,KEEP,KEEP),
//           VOL=(PRIVATE,RETAIN,SER=&E)
//SYSPRINT   DD SYSOUT=A,DCB=(RECFM=FBM,BLKSIZE=1330)
//INDEX      DD SYSOUT=A,DCB=(RECFM=FBM,BLKSIZE=1330)
//LIST       DD SYSOUT=A,DCB=(RECFM=FBM,BLKSIZE=1330)
//OSJOB      DD DISP=(NEW,PASS,DELETE),DCB=BLKSIZE=800,
//           UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSAF02    DD UNIT=SYSDA,SPACE=(CYL,(1,1)),
//           DISP=(NEW,DELETE),SEP=(OSJOB)
//SYSAF03    DD UNIT=SYSDA,SPACE=(CYL,(1,1)),
//           DISP=(NEW,DELETE),SEP=(OSJOB,SYSAF02)
//SYSPUNCH   DD SYSOUT=B,DCB=BLKSIZE=800
//           PEND
```

The Librarian program

The actual Librarian task is performed through the execution of program \$\$URIAN which is stored in library LIBRA75.

The masterfile

The masterfile is defined by MASTER DD statement. At the moment of the execution one must define the actual name of the masterfile and the residing diskpack.

Output

The SYSPRINT, INDEX and LIST DD statements describe the different output types.

Intermediate data sets

The OSJOB DD statement describes the module which is passed to a consecutive jobstep.

Working files

The SYSAF02 and SYSAF03 statements define the working files.

Punched output

SYSPUNCH defines the file for punched output, however it may also be used to create a second output module for consecutive steps, as will be demonstrated later.

The Librarian procedure is invoked by the control cards:

```
//STEPN      EXEC LIBRAP,A='....',E='....'  
//SYSIN DD *
```

Librarian input cards

/*

In which A gives the masterfile name and E supplies the residing diskpack.

A thorough knowledge of this procedure is not necessary for current use, however more complicated tasks need sometimes JCL substitutions.

THE MASTERFILE MANAGER

Many users may use the same masterfile and thus simplify the back-up procedures. In practice it is very useful to have a masterfile manager who is responsible for the total dataset without interfering with the single modules.

The tasks of such a masterfile manager are:

- The creation of the masterfile
- The checking of the dataset expiration date
- The production of tape back-ups, at least weekly
- The execution of the relocation procedure eventually
- The production of index listings

THE CREATION OF A MASTERFILE

To initialize a new masterfile on a user's disk, one has to execute the job:

```
//JOB LIB DD DSN=LIBRA75,DISP=(SHR,KEEP),UNIT=DISK,
//      VOL=SER=COPICB
//STEP1 EXEC PGM=$$URIAN
//SYS PRINT DD SYSOUT=A
//MASTER DD UNIT=3330,VOL=SER=USERnn,DSN=dsname,
//      DCB=(BLKSIZE=6444,DSORG=DA),DISP=(NEW,KEEP,DELETE),
//      SPACE=(CYL,(k))
//SYSIN DD *
-OPT INIT,DISK,options
/*
//STEP2 EXEC EURUDR,U=3330,V=USERnn
//GO.SYSIN DD *
```

reservation card

/*

in which: nn indicates a diskpack of the USER series
dsname is the name of your masterfile (max 8 char.)
k is the number of cylinders. According to our experiences one cylinder may contain roughly 6000 records of mixed nature.

Options

-OPT is a Librarian command card. The options at initializing a masterfile are the default definitions for the whole file. At the consecutive loading of modules into the file one may change these default values and define new ones for the single module.

The most common form is:

```
-OPT INIT,DISK,NORESEQ,SEQ=/73,8,10,10/,NOLIST,NOPUNCH,NOEXEC
```


The records are numbered starting in column 73 with a field width of 8 columns. The first sequence number is 10 increased each time by 10. The modules of this masterfile are not automatically renumbered after each run. If you expect to have input records with 80 columns of information you may write: SEQ=/81,8,10,10/, but this may also be done at the module level.

The option NORESEQ defines no automatic updating of the recordsequence numbers. Using a card with 10 increase, the programmer can easily see which records have been added namely the cards numbered with no 10 multiple. Of course, if you insert more than 9 cards at a single place the system performs some resequencing.

Reservation of space on disk

The reservation card is composed as follows:

columns 1-7 the number of the "fiche d'activite"
9-12 the number of authorization
14-17 the number of the programmer (masterfile manager)
19-24 the expiration date
26-69 the masterfile name, left adjusted
8,13,18,25 must be left blank

Space on users disks may be reserved only half a year in advance. To renew the reservation it is sufficient to run the next job:

```
//STEP1 EXEC EURUDR,U=3330,V=USERnn  
//GO.SYSIN DD *
```

new reservation card

/*

ADD A MODULE

The Librarian commands start with a minus sign in column 1, followed by three or four characters and one blank.

Each input deck for a Librarian execution has to start with the -OPT control card.

In the next examples it is assumed that the masterfile manager has initialized a file with the name "PRIVATE" on diskpack USER05.

The next procedure is executed to add a module 'PROGA':

```
//          EXEC LIBRAP,A='PRIVATE',E='USER05'
//SYSIN DD *
-OPT
-ADD PROGA,LIST
-DESC max 30 characters description
-PGMR max 10 characters programmers name

        the module

-EMOD      indicates end of module
-END       indicates end of input deck
/*
```

The LIST option in the -ADD command produces the complete listing of the module with the sequence numbers.

The programmers name must be spelled always in the same manner without containing blanks.

More than one module may be added in one step providing the -EMOD card at the end of each module.

The -END card occurs only one time.

Adding a module directly from tape or disk is also possible providing that the source dataset is fixed blocked and has records of 80 bytes, the -AUX card refers to the related DD card.

```
//          EXEC LIBRAP,A='PRIVATE',E='USER05'
//DD1 DD... dataset definition with
                ...DCB=(RECFM=FB,LRECL=80,BLKSIZE....)
//SYSIN DD *
-OPT
-ADD PROGB,LIST,PUNCH
-DESC .....
-PGMR PINKOPALIN
-AUX DD1
-EMOD
-END
/*
```

Punching of a module

The additional option "PUNCH" in the -ADD command delivers in this case a punched output also. Normally this is not required of course.

MODULES FOR CONSECUTIVE EXECUTION

Suppose the user has put two modules into the masterfile "PRIVATE". The first module is a fortran program PROGA, which needs some updating. The second module are the input data stored under the name PROGDATA. The system has the disposition of only one output file, so two steps are needed to produce the necessary datasets:

```
//STEP1          EXEC LIBRAP,A='PRIVATE',E='USER05'  
//SYSIN DD *  
-OPT  
-SEL PROGA,EXEC  
-REP 20  
      .... Record(s) to replace old record 20  
-DEL 30,40  
-INS 90  
      .... Record(s) to be inserted after record 90  
-EMOD  
-END  
/*  
//STEP2          EXEC LIBRAP,A='PRIVATE',E='USER05'  
//SYSIN DD *  
-OPT  
-SEL PROGDATA,EXEC  
-EMOD  
-END  
/*  
//STEP3          EXEC FTGCLG  
//CMP.SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)  
//GO.SYSIN DD DSN=*.STEP2.INS.OSJOB,DISP=(OLD,DELETE)
```

The EXEC option in the select command, -SEL, places the module in the output file of the step.

The replace command, -REP, can also be of the form -REP 20,25 at which all the cards (20,25) are substituted by the new records.

The delete command, -DEL, can also be of the form -DEL 30, which causes the deletion of just one card.

The update commands have to be ordered according increasing sequence numbers.

Of course the data input may be modified in the same way as the source deck.

A JCL SUBSTITUTION FOR TWO OUTPUT FILES

The same exercise as the previous example may be executed with one step less, through a JCL substitution. The Librarian offers only one output file for a transfer to the next step, however the "PUNCH" file may also be transformed into an output file by a substitution of the related DD statement.

```
//STEP1      EXEC LIBRAP,A='PRIVATE',E='USER05'  
//SYSPUNCH DD DISP=(NEW,PASS,DELETE),DCB=BLKSIZE=800,  
//          UNIT=SYSDA,SPACE=(CYL,(1,1))  
//SYSIN     DD *  
-OPT  
-SEL PROGA,EXEC  
    ....updates  
-EMOD  
-SEL PROGDATA,PUNCH  
-EMOD  
-END  
/*  
//STEP2     EXEC FTGCLG  
//CMP.SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)  
//GO.SYSIN  DD DSN=*.STEP1.INS.SYSPUNCH,DISP=(OLD,DELETE)
```

In this case the PUNCH option in the -SEL command puts the module to the file described by the SYSPUNCH statement, ready for use in the next step.

MERGING MODULES BY UTILITY

Creating a new module from parts of existing ones is possible through the "UTILITY" option. This procedure exists out of two steps.

The first one retrieves the necessary records and the second step puts the new module into the masterfile.

```
//STEP1      EXEC LIBRAP,A='PRIVATE',E='USER05'  
//SYSIN DD *  
-OPT UTILITY  
-OPT  
-ADD NEWMOD,LIST  
-DESC .....  
-PGMR .....  
-INC  PROGA,seq1,seq2  
-INC  PROGB,seq3,seq4  
-EMOD  
-END  
/*  
//STEP2      EXEC LIBRAP,A='PRIVATE',E='USER05'  
//SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)
```

In this example the first step recognizes the -OPT UTILITY card, removes this card and continues to substitute only the include command, -INC, by the specified records, e.g. taking from module PROGA the records numbered seq1 through seq2. The other commands are not touched but directly transferred to the output file. The second step receives thus a complete input description for a Librarian execution. The masterfile name in the EXEC statement of the second step may differ from the name in the first step. In this way this procedure may be used to transfer modules or part of modules between two different masterfile.

TESTING LARGE PROGRAMS

At the testing of large programs it is good practice to store the parts of the programs on which you are not working in object form to save compilation time and output listings. Suppose you have a large system with a main program and a number of subroutines stored as one module PROGA in the masterfile named "MASTERF" on disk "USER02". A number of new modules are created out of the existing source module by the method as described in the previous section "Merging modules by Utility".

From PROGA is produced MOD1 which contains MAIN
and MOD2 which contains SUBRA
and SUBRB
and MOD3 which contains SUBRC

The input deck composition for this task is as follows:

```
//STEP1      EXEC LIBRAP,A='MASTERF',E='USER02'  
//SYSIN DD *  
-OPT UTILITY  
-OPT  
-ADD MOD1,LIST  
-DESC .....  
-PGMR .....  
-INC PROGA,1,17  
-EMOD  
-ADD MOD2,LIST  
-DESC .....  
-PGMR .....  
-INC PROGA,18,330  
-EMOD  
-ADD MOD3,LIST  
-DESC .....  
-PGMR .....  
-INC PROGA,331,580  
-EMOD  
-END  
/*  
//STEP2      EXEC LIBRAP,A='MASTERF',E='USER02'  
//SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)
```

STEP2 creates MOD1, MOD2 and MOD3; consequently the subroutines are compiled and the output of the compiler is loaded as a module with the name "OBJECTR".

```
//STEP1      EXEC LIBRAP,A='MASTERF',E='USER02'  
//SYSIN DD *  
-OPT  
-SEL MOD2,EXEC  
-INC MOD3  
-EMOD  
-END  
/*
```

```

//STEP2      EXEC FTGC
//CMP.SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)
//STEP3      EXEC LIBRAP,A='MASTERF',E='USER02'
//DD1        DD DSN=&LOADSET,DISP=(OLD,DELETE)
//SYSIN DD *
-OPT
-ADD OBJECTR
-DESC .....
-PGMR .....
-AUX DD1
-EMOD
-END
/*

```

In this particular case in -INC command statement is not used in combination with the UTILITY option. The command selects the requested module and adds these records to the existing output file.

The use of the -INC command in this sense does not allow to transfer parts of modules, thus sequence numbers must be omitted.

After the execution of this job the module OBJECTR contains the object decks of the subroutines SUBRA, SUBRB, SUBRC.

Suppose now the programmer wants to test modifications in the main program and subroutines SUBRC.

The next job will do this task:

```

//STEP1      EXEC LIBRAP,A='MASTERF',E='USER02'
//SYSIN DD *
-OPT
-SEL MOD1,EXEC,TEMP
      .... Updates specifications for MAIN
-EMOD
-SEL MOD3,EXEC,TEMP
      .... Update specifications for SUBRAC
-EMOD
-END
/*
//STEP2      EXEC LIBRAP,A='MASTERF',E='USER02'
//SYSIN DD *
-OPT
-SEL OBJECTR,EXEC
-EMOD
-END
/*
//STEP3      EXEC FTGCLG
//CMP.SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)
//LKED.SYSIN DD DSN=*.STEP2.INS.OSJOB,DISP=(OLD,DELETE)
//GO.SYSIN DD *
      .... Input data
/*

```

The modifications to the main program and SUBRC are only temporary because of the option TEMP. The source modules are not affected. If the testing results of the modifications are positive you have only to remove the option TEMP and the changes become definitive. Do not forget to add in this case the LIST option to receive the updated list of record sequence numbers.

The updated versions of MAIN and SUBRC are input to the compiler, which produces the new object deck. At the same time the output of STEP3 still contain the obsolete versions. However the linkage editor searches first the output of the compiler to resolve the external references. As a consequence the modification of MAIN and SUBRC will be used and not the obsolete versions.

It is clear that the module OBJECTR is not modified so after some time of introducing modifications the programmer must delete OBJECTR and create a new module from all the updated routines.

EDITING COMMANDS

The Librarian disposes of a series editing commands. The first example converts a module named PROGA, from BCD characters to EBCDIC.

```
//STEP1      EXEC LIBRAP,A='MASTER',E='USER05'  
//SYSIN DD *  
-OPT  
-SEL PROGA  
-EDIT *%*( *  
-EDIT * *)*  
-EDIT *#*=*  
-EDIT *a*'*  
-EDIT *ε*+*  
-EMOD  
-END  
/*
```

The asterisk is a separator character. Any special character may be used as such; Librarian defines the first symbol after the EDIT command as the separator.

The general form of the -EDIT command is

```
-EDIT *S1*S2* (seq1(,seq2)) (,STR=nn) (,END=kk)
```

in which

* is any special character
S1 is search string of maximal 35 characters
S2 is replacement string of maximal 35 characters
seq1 define part of the module by record numbers
seq2
nn defines starting column, default is 1
kk defines column beyond which no shifting may occur.
Default is 72.

The bracket indicate that the mentioned item is optional. The -SCAN command is used to detect and signal a string of specified characters through whole the module or part of it. The general form is:

```
-SCAN *S1* (seq1(,seq2)) (,STR=nn) (,END=kk)
```

The meaning of the lower case characters is as described for the -EDIT command.

If one wants to substitute the contents of certain columns through whole the deck or part of it, the command -FILL will do the job.

The general format is:

```
-FILL *S1* (seq1(,seq2)) (,COL=nn)
```

in which nn is starting column for the field; default is 73.
For example, to add characters ABC in column 73-75, it is
sufficient to write:

-FILL *ABC*

It must be noted that for each -SEL command the number of EDIT,
-SCAN and -FILL operations are maximal 10. If you have to
perform more than 10 operations add other -SEL command.

BACK-UP MASTERFILES

Many users may dispose of modules in one single masterfile. The advantage is that back-up procedures are simplified. The masterfile manager has the obligation to make at least once a week a back-up copy of the whole file. It is strongly advised to use at least two or three different tapes which are permutated. It may happen that a masterfile is damaged and through the back-up procedure also the file copy. Caution: Never make a back-up of a probably damaged masterfile to your regular tapes.

The procedure for a back-up is as follows:

```
//STEP1      EXEC LIBRAP,A='XXXX',E='YYYY'  
//BACKUP     DD UNIT=TP9,VOL=(PRIVATE,SER=ZZZZ),  
//           LABEL=(1,SL,,OUT),DSN=dsname,  
//           DISP=(NEW,PASS),  
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)  
//SYSIN DD *  
-OPT INDEX,BKUPTAPE  
-END  
/*
```

in which XXXX is the masterfile name
 YYYY is the disk unit
 ZZZZ is the tape label
 dsname is the tape file dataset name

A masterfile of 30 cylinders 3330 disk can easily be stored on a 1200 ft tape with density 1600 BPI.

The option INDEX in the -OPT card produces also a summary of the present modules. See the paragraph on index listings.

RESTORING A MASTERFILE

If a masterfile has been destroyed or damaged, it is necessary to delete and recreate the file. Any default option and the space reservation may be modified at this moment.

```
//JOB LIB DD DSN=LIBRA75,DISP=(SHR,KEEP),UNIT=DISK,
//      VOL=SER=COPICB
//STEP1 EXEC PGM=IEHPRGM
//SYS PRINT DD SYSOUT=A
//DD1 DD DSN=masterf,UNIT=3330,VOL=SER=USERnn,
//     DISP=(OLD,DELETE)
//STEP2 EXEC PGM=J$URIAN
//SYS PRINT DD SYSOUT=A
//MASTER DD UNIT=3330,VOL=SER=USERnn,DSN=masterf,
//      DCB=(BLKSIZE=6444,DSORG=DA),
//      DISP=(NEW,KEEP,DELETE),SPACE=(CYL,(k))
//SYS IN DD *
-OPT INIT,DISK,options
/*
//STEP3 EXEC EURUDR,U=3330,V=USERnn
//GO.SYS IN DD *
      reservation card
/*
//STEP4 EXEC PGM=J$URIAN
//SYS PRINT DD SYSOUT=A
//BACKUP DD UNIT=FP9,VOL=(PRIVATE,SER=zzzz),
//      DSN=dsname,LABEL=(1,SL,,IN),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//      DISP=(OLD,PASS)
//MASTER DD DSN=masterf,UNIT=3330,VOL=SER=USERnn,
//      DISP=(OLD,KEEP)
//SYS IN DD *
-OPT RESTORE,INDEX
-END
/*
```

In which masterf is the masterfilename
nn indicates the dispack of the USER series
k is the number of cylinders
options are already previously described

The delete and restore procedure is also used to reorganize the space occupation of the masterfile. Especially very active files have to be deleted and restored once in a few months to maintain an economical space occupation.

RESTORING OF SINGLE MODULES

Single modules may be selected from a back-up tape and copied to a masterfile by the next procedure:

```
//STEP1 EXEC LIBRAP
//MASTER DD DUMMY
//MASTIN DD UNIT=TP9,VOL=(PRIVATE,SER=zzzz),
// LABEL=(1,SL,,IN),DSN=dsname,DISP=(OLD,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSAF01 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
-OPT UTILITY,NOCYCLE
-OPT INDEX
-COPY PROGA
-COPY PROGB
-EMOD
-END
/*
//STEP2 EXEC LIBRAP,A='xxxx',E='yyyy'
//SYSIN DD DSN=*.STEP1.INS.OSJOB,DISP=(OLD,DELETE)
/*
```

in which

```
xxxx is the masterfile name
yyyy is the disk unit
zzzz is the tape label
dsname is the tape file dataset name
PROGA,PROGB are modules to be restored from back-up tape zzzz
onto masterfile xxxx.
```

INDEX LISTINGS

The contents list of a masterfile and relevant information may be produced through the INDEX option in the -OPT command. Four types of indexes are available respectively produced by:

- OPT INDEX
- OPT INDEX(pgmr)
- OPT INDEX(S)
- OPT INDEX(M)

The operand INDEX(pgmr) in which pgmr is the programmers name as specified in the -PGMR card produces a list with one line for each module, including the module descriptions, the date added, the date and time of the list update and other pertinent information.

Omitting the (pgmr) specification produces the list for the modules.

The management variation of the index list is selected by -OPT INDEX(M) and produces an index listing of the entire masterfile with the standard module attributes and in addition counts representing the number of accesses, updates and selections executions. Total for the number of records, modules and available tracks are also provided.

The system variation of the index list is selected by -OPT INDEX(S). This option will produce an index listing of an entire masterfile containing the standard module attributes, a count of the number of extents occupied by a module and a count of the number of half-track or full-track blocks occupied by the module.

LIBRARIAN COMMANDS

The list of Librarian commands is not exhaustive, neither are all possible options mentioned. Only the most frequent used terms are specified.

The complete list may be consulted at the Computing Support Library.

THE OPT CONTROL CARD:

-OPT options. This card must be the first input card and, except in the case of UTILITY, there may only be one of this type.

Allowable options are:

INDEX(type) masterfile index of specified type
will be produced

UTILITY invokes UTILITY option

ADDING A NEW MODULE:

-ADD modname options

-DESC description of max 30 characters

-PGMR name of max 10 characters not blanks

Source deck or -AUX card

-EMOD

UPDATING A MODULE

-SEL modname,options

-HST optional columns 6-80 used to contain comments

-INS seq1 or -INS LAST

cards to be inserted after record seq1 or at the end of the module

-REP seq1(,seq2) or -REP ALL

cards to replace either seq1 through seq2 or all cards

-DEL seq1(,seq2)

-EMOD

OPTIONS FOR -ADD AND -SEL COMMANDS

EXEC module to be transferred to next step

LIST module to be listed

LISTH list the history cards (-HST)

PUNCH module to be punched

SEQ=/a,b,c,d/ sequence numbers begin in column a, are b columns long, have an increment of c and starting value d. The value of a may also be 81.

OPTIONS FOR -SEL COMMANDS

In addition to the previous listed options, the -SEL card may also contain:

TEMP updates are effective only for the output file,
the source module remains unchanged
COPY=newname creates a duplicate version of the module named
"newname"
RESEQ performs new record numbers for the module

DELETING A MODULE

-DLM modname no options permitted

EDITING OPERATIONS

-SEL name,options
-EDIT *S1*S2*(seq1(,seq2))(1,STR=nn)(,END=kk)
-SCAN *S1*S2*(seq1(,seq2))(1,STR=nn)(,END=kk)
-FILL *S1*(seq1(,seq2))(,COL=mm)
nn is starting column, default is 1
kk specifies ending column defaults to 72
mm is starting column for field, default is 73
maximum length of S1 and S2 is 35 characters
* is any special character
No more than 10 EDIT/SCAN/FILL operations per single
-SEL command are permitted.

COMMANDS WITH UTILITY

The next commands can only be used in combination with the -OPT UTILITY option:

-INC modname(,seq1)(,seq2)
referenced module or part of it is written to the
output file.
-PRINT modname(,seq1)(,seq2)
-PUNCH modname(,seq1)(,seq2)
referenced module or part of it is printed or
punched.
Warning: First record of output is the Librarian
command.

