

DEC-11-ULBAA-A-D

DOS/BATCH
Librarian (LIBR)
Programmer's Manual

FOR THE DOS/BATCH OPERATING SYSTEM

Monitor Version V09

August 1973

For additional copies, order No. DEC-11-ULBAA-A-D from Digital Equipment Corporation, Software Distribution Center, Maynard, Massachusetts 01754.

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. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and considered when documenting subsequent manuals.

Copyright © 1973 by Digital Equipment Corporation

Associated Documents:

DOS/BATCH Monitor Programmer's Manual,
DEC-11-OMPMA-A-D

DOS/BATCH Assembler (MACRO)
Programmer's Manual, DEC-11-LASMA-A-D

DOS/BATCH FORTRAN IV Compiler and Object-Time System
Programmer's Manual, DEC-11-LFRTA-A-D

DOS/BATCH System Manager's Guide,
DEC-11-OSMGA-A-D

DOS/BATCH File Utility Package (PIP)
Programmer's Manual, DEC-11-UPPAA-A-D

DOS/BATCH Debugging Program (ODT-11R)
Programmer's Manual, DEC-11-UDEBA-A-D

DOS/BATCH Linker (LINK)
Programmer's Manual, DEC-11-ULKAA-A-D

DOS/BATCH Text Editor (EDIT)
Programmer's Manual, DEC-11-UEDAA-A-D

DOS/BATCH File Compare Program (FILCOM)
Programmer's Manual, DEC-11-UFCAA-A-D

DOS/BATCH Verification Program (VERIFY)
Programmer's Manual, DEC-11-UVERA-A-D

DOS/BATCH Disk Initialization Program (DSKINT)
Programmer's Manual, DEC-11-UDKIA-A-D

DOS/BATCH File Dump Program (FILDMP)
Programmer's Manual, DEC-11-UFLDA-A-D

Trademarks of Digital Equipment Corporation include:

DEC	PDP-11
DIGITAL (Logo)	COMTEX-11
DEctape	RSTS-11
UNIBUS	RSX-11

PREFACE

This manual describes the features and operation of the DOS/BATCH Librarian (LIBR). The reader should be familiar with the DOS/BATCH Monitor Programmer's Manual in order to fully understand the use of the Librarian.

In addition to the Monitor and the Librarian, DOS/BATCH software comprises the following:

- Assembler (MACRO-11)
- FORTRAN IV Compiler
- File Utility Package (PIP)
- Debugging Program (ODT-11R)
- Linker (LINK)
- Text Editor (EDIT)
- File Compare Program (FILCOM)
- Verification Program (VERIFY)
- Disk Initialization Program (DSKINT)
- File Dump Program (FILDMP)
- System Loader (SYSLOD)

NOTE

This software is furnished to purchaser under a license for use on a single computer system and can be copied (with inclusion of DEC's copyright notice) only for use in such system, except as may otherwise be provided in writing by DEC.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

CONTENTS

	<u>Page</u>
CHAPTER 1 INTRODUCTION	1-1
CHAPTER 2 OPERATING PROCEDURES	2-1
2.1 CALLING LIBR	2-1
2.2 COMMAND STRINGS	2-1
2.2.1 Creating a Library	2-1
2.2.2 Updating a Library	2-2
2.2.2.1 To Delete One or More Object Modules	2-2
2.2.2.2 To Insert One or More Object Modules	2-3
2.2.2.3 To Replace One or More Object Modules	2-3
2.2.3 Listing a Library	2-3
2.2.4 Naming Libraries	2-5
2.2.5 Legal File Specification Combinations	2-5
2.3 EXAMPLES	2-6
2.4 ERROR MESSAGES	2-7
2.5 PROGRAMMING NOTES	2-8
2.5.1 Creating	2-8
2.5.2 Updating	2-9
2.5.3 Listing	2-9
INDEX	

CHAPTER 1

INTRODUCTION TO LIBRARIAN

The DOS/BATCH Librarian (LIBR) is a system program that provides facilities for creating, modifying, deleting, and listing the contents of libraries. A library is a file consisting of one or more object modules. (An object module is the binary output of the DOS/BATCH Assembler or FORTRAN Compiler).

LIBR is a valuable program for the DOS/BATCH user because:

It eliminates having separate directory entries in a User File Directory (UFD) for each object module.

It expedites the linking process in conjunction with the Linker's library search capabilities.

It allows for standardization and controlled updating of frequently used routines, e.g., FORTRAN cosine routine.

The user controls the operation of LIBR through command strings typed on the keyboard. Specified in the command strings are such things as devices, library and object module names, and switches that indicate the LIBR operation desired. The user can direct LIBR to:

Create a library

Update a library

Insert one or more object modules in a library

Replace one or more object modules in a library

List the contents of a library

Delete one or more object modules from a library

Delete an entire library

A directory listing of the object modules of a library can be obtained by merely specifying the device on which the directory is to appear and the name of the library.

The flexibility of LIBR enables the user to specify certain combinations of operations in a single command string. For example, a library can be modified, renamed, and listed in one command string.

The switch options which direct LIBR operations are:

Switch	Operation
/D	Delete object module
/DL	Delete input library
/I	Insert object module
/LO	List object modules
/R	Replace object module

If the user types an illegal command string, e.g., illegal format, excessive switches, nonexistent file or object module, etc., LIBR prints an appropriate error message on the teleprinter.

The following discussion assumes that the reader is familiar with the DOS/BATCH Monitor, EDIT Text Editor, MACRO Assembler, ODT-11R Debugging Program, and LINK Linker.

CHAPTER 2

OPERATING PROCEDURES

2.1 CALLING LIBR

The Librarian is called into core by typing the RUN command in response to the Monitor's dot or dollar sign. The Librarian is stored as LIBR; when called it prints its name, version number, and a # sign, and then waits for the user to issue a command string. For example:

```
$RUN LIBR  
  
LIBR Vxxx  
#
```

2.2 COMMAND STRINGS

When the Librarian is in core and has printed the # sign, it is ready to accept a user command string. The format is:

```
output library,listing file<input library,input file(s)
```

LIBR performs two passes over all input files. For nonfile-structured devices (e.g., paper tape reader), the system informs the user to reload the device for the second pass. For file-structured devices, both passes are performed automatically without requiring user intervention.

2.2.1 Creating a Library

```
output library(,listing file)<,input file(s)
```

A library is created on the device specified in the output library specification and named as specified. The listing file specification is optional and, if it is present, the contents of the output library will be listed. The format of the listing is discussed under the heading "Listing a Library."

An input library need not appear, but the comma and one or more input files must appear (each of which contains one or more object modules). For example:

```
#DT1:FIL.LIB<,FIL.1,FIL.2
```

creates a library named FIL.LIB on DECTape 1. The library consists of all object modules in FIL.1 and FIL.2 in that order, and in the order in which the object modules appear in their respective input files.

A restriction placed on the user by the Librarian is that it accepts a command specification of only one line. It is therefore suggested when creating a Library from a large number of object modules, that the object modules be concatenated into a reasonable number of files.* An example of this follows.

```
$RU PIP
#FILE.001<FILE.A,FILE.B,FILE.C
#FILE.002<FILE.D,FILE.E,FILE.F
#FILE.003<FILE.G,FILE.H,FILE.I
#↑C
.KI
$RU LIBR
#NAME.LIB<,FILE.001,FILE.002,FILE.003
$RU PIP
#FILE.001,FILE.002,FILE.003/DE
```

2.2.2 Updating a Library

Libraries can be updated in one of three ways:

1. Delete one or more object modules.
2. Insert one or more object modules.
3. Replace one or more object modules.

2.2.2.1 To Delete One or More Object Modules

```
output library(,listing file)<input library/D:v(1):...v(n)
```

The output library is created as a result of deleting the object modules named v(1)...,v(n) from the input library. The listing file is optional.

The name associated with an object module is the symbol assigned to the module by the MACRO Assembler's .TITLE directive.

The object modules to be deleted must appear in the same order as they appear in the library; their order can be determined from the listing.

For example:

```
#DT1:LIBR.1<DT2:LIBR.0/D:M1:M2
```

creates a library named LIBR.1 on DECTape 1 as a result of deleting the object modules M1 and M2 from LIBR.0 on DECTape 2.

Insert and/or Replace operations cannot accompany a Delete request.

*Note that an input file of concatenated object modules differs from a library in that it does not have a directory of the object modules it contains.

2.2.2.2 To Insert One or More Object Modules

```
output library(,listing file)<input library,input file(s)/I(:v)
```

The output library is created as a result of inserting the object modules of the input file into the input library. If *v* is specified, the object module(s) in the input file are inserted starting at position *v*, otherwise, they are inserted at the end. *v* is treated as a decimal integer, and is always relative to the input library.

If more than one input file is specified for insertion, the positions at which the files are to be inserted must appear in non-descending order. For example:

```
#DT1:LIBR.1<DT2:LIBR.0,FIL.1/I:2,FIL.2/I
```

creates an output library on DECTape 1 as a result of inserting the object modules of FIL.1 into LIBR.0, beginning at position 2, and then inserting the object modules of FIL.2 into LIBR.0 at the end.

Insert and Replace operations can appear in the same command as long as the order restriction is observed.

2.2.2.3 To Replace One or More Object Modules

```
output library(,listing file)<input library,input file(s)/R
```

The output library is created as a result of replacing the object module(s) in the input library by those in the input file(s).

The object modules to be replaced must have the same name as those replacing them, and they must be in the same order. For example:

```
#DT1:LIBR.1<DT2:LIBR.0,FIL.1/R,FIL.2/R
```

creates the output library LIBR.1 on DECTape 1 as a result of replacing the object modules in the input library LIBR.0 with those in FIL.1 and FIL.2.

2.2.3 Listing a Library

```
#,listing file(/LO)<input library
```

The directory of the input library is listed. Optionally, the presence of the /LO switch directs the Librarian to produce an object module listing. This is a means of double-checking the accuracy of the library; the directory listing must correspond exactly to the object module listing.

The output library is listed if one was created; otherwise, the input library is listed. The format of the listing is:

```
Library Name & Extension
Decimal Order Number      Object Module Name (1st Module)
      .
      :
      .
      .
Decimal Order Number      Object Module Name (last module)
```

For example, if LIB.1 contains object modules M1, M2, and M3 in that order, the command:

```
#,LP:FIL.LST<DT1:LIB.1
```

produces on the line printer:

```
LIBR  Vxxx  (xxx is the LIBR version number)
FIL      .LST      date      time
SEQ.     NAME      VERSION
00001    M1        021
00002    M2        031
00003    M3        041
```

If the /LO switch appears, for example:

```
#,LP:FIL.LST/LO<DT1:LIB.1
```

the listing above is followed by a form feed and a similar table, except that the name of the second table is always OBJMOD.LST.

The library name that is printed at the head of the listing is the name specified in the listing file specification. For example:

```
#LIB.ABC,LP:NAME<,FIL.1,FIL.2
```

The listing is titled NAME, not the new created library LIB.ABC. When the listing file name is not specified, then the listing is titled with the name of the newly created file (LIB.ABC).

2.2.4 Naming Libraries

The output library can have the same name as the input library. In this case, however, the input library has an implied /DL; that is, the input library is deleted. For example:

```
#LIB.1<LIB.1/D:OM1
```

is the same as:

```
#LIBR.TMP<LIB.1/D:OM1/DL
```

and then rename LIB.TMP to LIB.1

NOTE

The user should never name a Library LIBR.TMP. This name is reserved for use by the Librarian.

2.2.5 Legal File Specification Combinations

In a command string, various combinations of file specifications are possible; legal combinations and their operation are shown below.

	Output Library	Listing	Input Library	Input File(s)	Operation	Note
(1)	P	P	P	P	Insert or Replace Object Modules; List Output Library	SE if /D on input library
(2)	P	P	P	NP	Delete Object Modules; List Output Library	SE if/D not on input library
(3)	P	P	NP	P	Create Library; List Output Library	SE if switch on input file
(4)	NP	P	P	NP	List Input Library	SE if /D on input library
(5)	P	NP	P	P	Same as (1) except no listing	Same as (1)
(6)	P	NP	P	NP	Same as (2) except no listing	Same as (2)
(7)	P	NP	NP	P	Same as (3) except no listing	Same as (3)

Legend: P = present
NP = not present
SE = syntax error

2.3 EXAMPLES

Assume FIL.1 contains object modules OM1, OM2, and OM3 in that order, FIL.2 contains OM4 and OM5 in that order, FIL.3 contains OM5 and OM3 in that order, and FIL.4 contains OM6. Then:

```
#LIB.1,LP:LIB.1<,FIL.1,FIL.2
```

creates a library named LIB.1 containing object modules OM1, OM2, OM3, OM4, and OM5 in that order. The listing appears on the line printer as:

```
LIBR  Vxxx

LIB      .1      date      time
SEQ.     NAME     VERSION
00001    OM1      025
00002    OM2      032
00003    OM3      041
00004    OM4      054
00005    OM5      027
```

Files FIL.1 and FIL.2 remain unaltered. The listing is produced after all other actions have been performed. Consequently,

```
#LIB.1,LIB.1<,FIL.1,FIL.2
```

produces an error message (file already exists) when an attempt is made to write the listing to the disk.

Using the assumption above:

```
#LIB.2<LIB.1/D:OM1:OM4
```

creates a library named LIB.2 containing object modules OM2, OM3, and OM5 in that order. No listing is produced and LIB.1 is not deleted.

```
#LIB.3<LIB.2/D:OM3:OM2
```

produces an error message because the modules to be deleted are not in the order in which they appear in the library.

The command string:

```
#,LP:LIB2.LS/LO<LIB.2
```

produces a listing on the line printer which appears as:

```
LIBR  Vxxx

LIB2    .LS      date      time
SEQ.     NAME     VERSION
00001    OM2      032
00002    OM3      041
00003    OM5      027
```

The command string:

```
#LIB.3<LIB.2/DL,FIL.4/I:2
```

creates a library named LIB.3 containing OM2, OM6, OM3 and OM5 in that order. No listing is produced and LIB.2 is deleted.

The command string:

```
#LIB.4<LIB.3,FIL.4/R
```

creates a library named LIB.4, which is really LIB.3 with OM6 replaced (i.e., removed from LIB.3 before creating LIB.4).

```
#LIB.5<LIB.4,FIL.3/R
```

produces an error message because the object modules in FIL.3 are not in the same order as in LIB.4

The command string:

```
#LIB.5<LIB.3/DL,FIL.4/I
```

creates a library named LIB.5 containing OM2, OM6, OM3, OM5 and OM6 in that order. No listing is produced and LIB.3 is deleted. Note that a library can contain multiple copies of the same object module, e.g., two OM6 modules, above.

The command string:

```
#LIB.6<LIB.5/D:OM6
```

creates a library named LIB.6 containing OM2, OM3, OM5 and OM6 in that order. No listing is produced and LIB.5 is not deleted. When a library contains multiple copies of the same object module, the copies are deleted one at a time in their order of occurrence.

If the purpose of the previous example were to delete all occurrences of OM6, the command string would have been either:

```
#LIB.6<LIB.5/D:OM6:OM6
```

or

```
#LIB.6<LIB.5/D:OM6/D:OM6
```

2.4 Error Messages

Error messages issued by the Librarian are listed below. (See the DOS/BATCH Monitor Programmer's Manual for a listing of all DOS/BATCH error messages).

Error Code	Additional Information	Meaning
S202	File Name and Error Status Byte	Fatal I/O error; due to truncated line, checksum, character parity, or device parity error.
S203	File Name	Switch error or semantic error; due to illegal switch, too many switches on a file, or illegal combination of file specifications.
S204		Illegal file specification format; more than two output files specified.
S213	File Name	Error on input file; illegal object module format; first line not a GSD, or EOF prior to reading end module line.
S244	File Name	Out of order; already past requested position for Insert.
S245	File Name	Object module error; object module not found, or /R or /D out of order.
S246		Error on input library; illegal library format, first two lines incorrect.
S247		Listing error; output library cannot be read from output library device, i.e., PP:.

2.5 Programming Notes

2.5.1 Creating

Creating a library requires an output library specification and the input files to be placed in the library. The input files must be preceded by a comma; they are not considered to be an input library. Listing the newly created library is optional.

Because the length of the Librarian's command string is restricted to one (1) teletype line, creating libraries from many object modules must be accomplished with an intermediate step.

The step is to first concatenate the object modules in their desired order using PIP. Having done this, the command string to the Librarian is reduced to one input file, i.e., that containing the concatenated object modules.

The Librarian does not supply default extensions (e.g., .OBJ); the user must remember to supply them.

2.5.2 Updating

To update a library, both output and input library specifications are required, along with object modules or input files. Updating includes deletion of object modules, insertion of object modules, and replacing of one or more object modules. Listing the updated library is optional.

2.5.3 Listing

To list the contents of a library requires only a listing file specification and an input library specification. The listing file specification must be preceded by a comma, so that it is not interpreted as an output library.

INDEX

Calling LIBR, 2-1
Command string, 2-1
Creating a library, 2-1, 2-8

/D switch, 2-2
Delete object module, 2-2
/DL switch, 2-5

Error messages, 2-7
Examples, 2-6

File specifications, 2-5

/I switch, 2-3
Inserting object modules, 2-3

Library
 creating, 2-1, 2-8
 listing, 2-3, 2-9
 naming, 2-5
 updating, 2-2, 2-9

/LO switch, 2-3

Module name, 2-2

Naming libraries, 2-5

Programming notes
 creating, 2-8
 listing, 2-9
 updating, 2-9

/R switch, 2-3
Replace object module, 2-3

Switches
 /D, 2-2
 /DL, 2-5
 /LO, 2-3
 /I, 2-2
 /R, 2-3

Updating a library, 2-2, 2-3, 2-9

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 and PDP-12
DIGITAL Software News for the PDP-11
DIGITAL Software News for 18-bit Computers

These newsletters contain information applicable to software available from DIGITAL'S Software Distribution Center. Articles in DIGITAL Software News update the cumulative Software Performance Summary which is included 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. If no Software Specialist is available, please send a Software Performance Report form with details of the problems to:

Digital Equipment Corporation
Software Information Service
Software Engineering and Services
Maynard, Massachusetts 01754

These forms, which are provided in the software kit, should be fully completed and accompanied by terminal 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 manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest DIGITAL field office or representative. USA customers may order directly from the Software Distribution Center 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:

Digital Equipment Corporation
DECUS
Software Engineering and Services
Maynard, Massachusetts 01754

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form (see the HOW TO OBTAIN SOFTWARE INFORMATION page).

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

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

or
Country

If you do not require a written reply, please check here.

-----**Fold Here**-----

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Communications
P. O. Box F
Maynard, Massachusetts 01754

