

NCR

MS-LIB Library Manager



MS-LIB

CONTENTS

Introduction

Features and Benefits of MS-LIB	
Overview of MS-LIB Operation	4

Chapter 1 RUNNING MS-LIB

1.1	Invoking MS-LIB	1-1
1.1.1	Method 1: LIB	1-2
	Summary of Command Prompts	1-2
	Summary of Command Characters	1-2
1.1.2	Method 2: LIB <library> <operations>, <listing>	1-3
1.1.3	Method 3: LIB @ <filespec>	1-5
1.2	Command Prompts	1-7
1.3	Command Characters	1-9
	+ - append	1-9
	- - delete	1-9
	* - extract	1-10
	; - default remaining prompts	1-10
	& - continuation	1-11
	Control-C - program abort	1-11

Chapter 2 ERROR MESSAGES

CONTENTS

Page

1. Introduction 1

2. Theoretical Framework 2

3. Methodology 3

4. Results 4

5. Discussion 5

6. Conclusion 6

7. References 7

8. Appendix 8

9. Index 9

INTRODUCTION

Features and Benefits

MS-LIB creates and modifies library files that are used with Microsoft's MS-LINK Linker Utility. MS-LIB can add object files to a library, delete modules from a library, or extract modules from a library and place the extracted modules into separate object files.

MS-LIB provides a means of creating either general or special libraries for a variety of programs or for specific programs only. With MS-LIB you can create a library for a language compiler, or you can create a library for one program only, which would permit very fast linking and possibly more efficient execution.

You can modify individual modules within a library by extracting the modules, making changes, then adding the modules to the library again. You can also replace an existing module with a different module or with a new version of an existing module.

The command scanner in MS-LIB is the same as the one used in Microsoft's MS-LINK, MS-Pascal, MS-FORTRAN, and other 16-bit Microsoft products. If you have used any of these products, using MS-LIB is familiar to you. Command syntax is straightforward, and MS-LIB prompts you for any of the commands it needs that you have not supplied. There are no surprises in the user interface.

Overview of MS-LIB Operation

MS-LIB performs two basic actions: it deletes modules from a library file, and it changes object files into modules and appends them to a library file. These two actions underlie five library manager functions:

- delete a module
- extract a module and place it in a separate object file
- append an object file as a module of a library
- replace a module in the library file with a new module
- create a library file

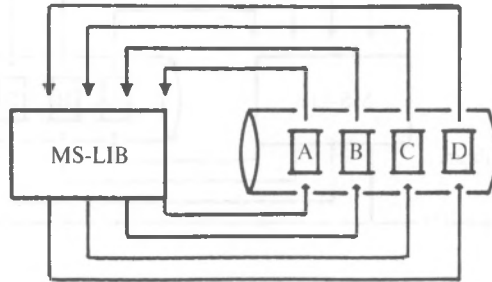
During each library session, MS-LIB first deletes or extracts modules, then appends new ones. In a single operation, MS-LIB reads each module into memory, checks it for consistency, and writes it back to the file. If you delete a module, MS-LIB reads in that module but does not write it back to the file. When MS-LIB writes back the next module to be retained, it places the module at the end of the last module written. This procedure effectively "closes up" the disk space to keep the library file from growing larger than necessary. When MS-LIB has read through the whole library file, it appends any new modules to the end of the file. Finally, MS-LIB creates the index, which MS-LINK uses to find modules and symbols in the library file, and outputs a cross reference listing of the PUBLIC symbols in the library, if you request such a listing. (Building the library index may take some extra time, up to 20 seconds in some cases.)

For example:

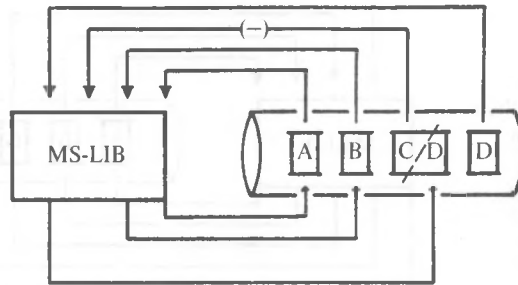
```
LIB PASCAL+HEAP-HEAP;
```

first deletes the library module HEAP from the library file, then adds the file HEAP.OBJ as the last module in the library. This order of execution prevents confusion in MS-LIB when a new version of a module replaces a version in the library file. Note that the replace function is simply the delete-append functions in succession. Also note that you can specify delete, append, or extract functions in any order; the order is insignificant to the MS-LIB command scanner.

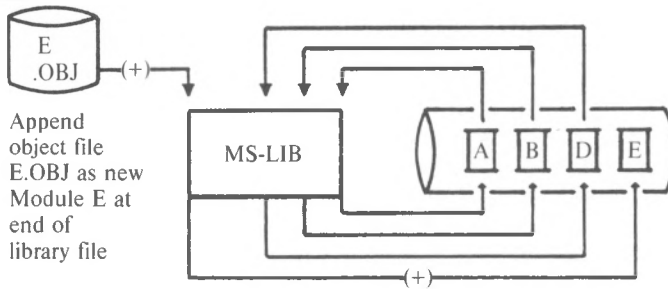
Consistency
Check only

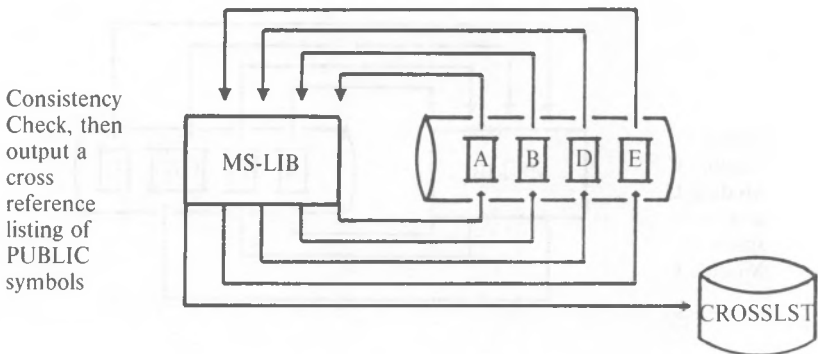
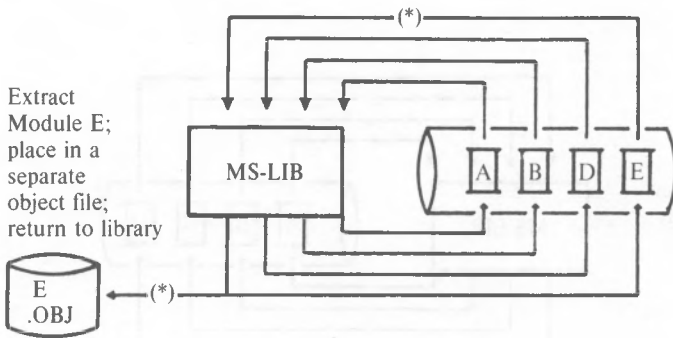


Delete
Module C;
Module D
written to
space of
Module C



Append
object file
E.OBJ as new
Module E at
end of
library file





CHAPTER 1

RUNNING MS-LIB

Running MS-LIB requires two types of commands: a command to invoke MS-LIB and answers to command prompts. Usually you will enter all the commands to MS-LIB on the terminal keyboard. As an option, answers to the command prompts may be contained in a Response File. Some special command characters exist. Some are used as a required part of MS-LIB commands. Others assist you while entering MS-LIB commands.

1.1 INVOKING MS-LIB

MS-LIB may be invoked three ways. By the first method, you enter the commands as answers to individual prompts. By the second method, you enter all commands on the line used to invoke MS-LIB. By the third method, you create a Response File that contains all the necessary commands.

Summary of Methods to invoke MS-LIB

Method 1	LIB
Method 2	LIB <library> <operations>, <listing>
Method 3	LIB @ <filespec>

1.1.1 Method 1: LIB

Enter:

LIB

MS-LIB will be loaded into memory. Then, MS-LIB returns a series of three text prompts that appear one at a time. You answer the prompts as commands to MS-LIB to perform specific tasks.

The Command Prompts and Command Characters are summarized here. The Command Prompts and Command Characters are described fully in Sections 1.2 and 1.3.

Summary of Command Prompts

PROMPT	RESPONSES
Library file:	List filename of library to be manipulated (default: filename extension .LIB)
Operation:	List command character(s) followed by module name(s) or object filename(s) (default action: no changes. default object filename extension: .OBJ)
List file:	List filename for a cross reference listing file (default: NUL; no file)

Summary of Command Characters

Character	Action
+	Append an object file as the last module
-	Delete a module from the library
*	Extract a module and place in an object file
;	Use default responses to remaining prompts
&	Extend current physical line; repeat command prompt
Control-C	Abort library session.

1.1.2 Method 2: LIB <library> <operations>,<listing>

Enter:

```
LIB <library> <operations>,<listing>
```

The entries following LIB are responses to the command prompts. The **library** and **operations** fields and all operations entries must be separated by one of the command characters plus, minus, and asterisk (+, -, *). If a cross reference listing is wanted, the name of the file must be separated from the last operations entry by a comma.

where: **library** is the name of a library file. MS-LIB assumes that the filename extension is .OBJ, which you may override by specifying a different extension. If the filename given for the **library** fields does not exist, MS-LIB will prompt you:

```
Library file does not exist. Create?
```

Enter Yes (or any response beginning with Y) to create a new library file. Enter No (or any other response not beginning with Y) to abort the library session.

operations is deleting a module, appending an object file as a module, or extracting a module as an object file from the library file. Use the three command characters plus (+), minus (-), and asterisk (*) to direct MS-LIB what to do with each module or object file.

listing is the name of the file you want to receive the cross reference listing of PUBLIC symbols in the modules in the library. The list is compiled after all module manipulation has taken place.

To select the default for remaining field(s), you may enter the semicolon command character.

If you enter a Library filename followed immediately by a semicolon, MS-LIB will read through the library file and perform a consistency check. No changes will be made to the modules in the library file.

If you enter a Library filename followed immediately by a comma and a List filename, MS-LIB will perform its consistency check of the library file, then produce the cross reference listing file.

Example

```
LIB PASCAL-HEAP+HEAP;
```

This example causes MS-LIB to delete the module HEAP from the library file PASCAL.LIB, then append the object file HEAP.OBJ as the last module of PASCAL.LIB (the module will be named HEAP).

If you have many operations to perform during a library session, use the ampersand (&) command character to extend the line so that you can enter additional object filenames and module names. Be sure to always include one of the command characters for operations (+, -, *) before the name of each module or object filename.

Example

```
LIB PASCAL<CR>
```

causes MS-LIB to perform a consistency check of the library file PASCAL.LIB. No other action is performed.

Example

```
LIB PASCAL,PASCROSS.PUB
```

causes MS-LIB to perform a consistency check of the library file PASCAL.LIB, then output a cross reference listing file named PASCROSS.PUB.

1.1.3 Method 3: LIB @ <filespec>

Enter:

```
LIB @ <filespec>
```

where: **filespec** is the name of a Response File. A Response File contains answers to the MS-LIB prompts (summarized under method 1 for invoking and described fully in Section 1.2). Method 3 permits you to conduct the MS-LIB session without interactive (direct) user responses to the MS-LIB prompts.

IMPORTANT

Before using method 3 to invoke MS-LIB, you must first create the Response File.

A Response File has text lines, one for each prompt. Responses must appear in the same order as the command prompts appear.

Use Command Characters in the Response File the same way as they are used for responses entered on the terminal keyboard.

When the library session begins, each prompt will be displayed in turn with the responses from the response file. If the response file does not contain answers for all the prompts, MS-LIB will use the default responses (no changes to the modules currently in the library file for Operation, and no cross reference listing file created).

If you enter a Library filename followed immediately by a semicolon, MS-LIB will read through the library file and perform a consistency check. No changes will be made to the modules in the library file.

If you enter a Library filename then only a carriage return of Operations then a comma and a List filename, MS-LIB will perform its consistency check of the library file, then produce the cross reference listing file.

Example:

```
PASCAL<CR>  
+CURSOR+HEAP-HEAP*FOIBLES<CR>  
CROSSLST<CR>
```

This Response File will cause MS-LIB to delete the module HEAP from the PASCAL.LIB library file, extract the module FOIBLES and place in an object file named FOIBLES.OBJ, then append the object files CURSOR.OBJ and HEAP.OBJ as the last two modules in the library. Then, MS-LIB will create a cross reference file named CROSSLST.

1.2 COMMAND PROMPTS

MS-LIB is commanded by entering responses to three text prompts. When you have entered your response to the current prompt, the next appears. When the last prompt has been answered, MS-LIB performs its library management functions without further command. When the library session is finished, MS-LIB exits to the operating system. When the operating system prompt is displayed, MS-LIB has finished the library session successfully. If the library session is unsuccessful, MS-LIB returns the appropriate error message.

MS-LIB prompts you for the name of the library file, the operation(s) you want to perform, and the name you want to give to a cross reference listing file, if any.

Library file:

Enter the name of the library file that you want to manipulate. MS-LIB assumes that the filename extension is .LIB. You can override this assumption by giving a filename extension when you enter the library filename. Because MS-LIB can manage only one library file at a time, only one filename is allowed in response to this prompt. Additional responses, except the semicolon command character, are ignored.

If you enter a library filename and follow it immediately with a semicolon command character, MS-LIB will perform a consistency check only, then return to the operating system. Any errors in the file will be reported.

If the filename you enter does not exist, MS-LIB returns the prompt:

Library file does not exist. Create?

You must enter either Yes or No, in either upper or lower (or mixed) case. Actually, MS-LIB checks the response of the letter Y as the first character. If any other character is entered first, MS-LIB terminates and returns to the operating system.

Operation:

Enter one of the three command characters for manipulating modules (+, -, *), followed immediately (no space) by the module name or the object filename. Plus sign appends an object file as the last module in the library file (see further discussion under the description of plus sign below). Minus sign deletes a module from the library file. Asterisk extracts a module from the library and places it in a separate object file with the filename taken from the module name and a filename extension .OBJ.

When you have a large number of modules to manipulate (more than can be typed on one line), enter an ampersand (&) as the last character on the line. MS-LIB will repeat the Operation prompt, which permits you to enter additional module names and object filenames.

MS-LIB allows you to enter operations on modules and object files in any order you want.

More information about order of execution and what MS-LIB does with each module is given in the descriptions of each Command Character.

List file:

If you want a cross reference list of the PUBLIC symbols in the modules in the library file after your manipulations, enter a filename in which you want MS-LIB to place the cross reference listing. If you do not enter a filename, no cross reference listing is generated (a NUL file).

The response to the List file prompt is a file specification. Therefore, you can specify, along with the filename, a drive (or device) designation and a filename extension. The List file is not given a default filename extension. If you want the file to have a filename extension, you must specify it when entering the filename.

The cross reference listing file contains two lists. The first list is an alphabetical listing of all PUBLIC symbols. Each symbol name is followed by the name of its module. The second list is an alphabetical list of the modules in the library. Under each module name is an alphabetical listing of the PUBLIC symbols in that module.

1.3 COMMAND CHARACTERS

MS-LIB provides six command characters: three of the command characters are required in responses to the Operation prompt; the other three command characters provide you additional helpful commands to MS-LIB.

- + The plus sign followed by an object filename appends the object file as the last module in the library named in response to the Library file prompt. When MS-LIB sees the plus sign, it assumes that the filename extension is .OBJ. You may override this assumption by specifying a different filename extension.

MS-LIB strips the drive designation and the extension from the object file specification, leaving only the filename. For example, if the object file to be appended as a module to a library is:

B:CURSOR.OBJ

a response to the Operation prompt of:

+B:CURSOR.OBJ

causes MS-LIB to strip off the B: and the .OBJ, leaving only CURSOR, which becomes a module named CURSOR in the library.

NOTE

The distinction between an object file and a module (or object module) is that the file possesses a drive designation (even if it is default drive) and a filename extension. Object modules possess neither of these.

- The minus sign followed by a module name deletes that module from the library file. MS-LIB then “closes up” the file space left empty by the deletion. This cleanup action keeps the library file from growing larger than necessary with empty space. Remember that new modules, even replacement modules are added to the end of the file, not stuffed into space vacated by deleting modules.

- * The asterisk followed by a module name extracts that module from the library file and places it into a separate object file. The module will still exist in the library (extract means, essentially, copy the module to a separate object file). The module name is used as the filename. MS-LIB adds the default drive designation and the filename extension .OBJ. For example, if the module to be extracted is:

CURSOR

and the current default disk drive is A:, a response to the Operation prompt of:

*CURSOR

causes MS-LIB to extract the module named CURSOR from the library file and to set it up as an object file with the file specification of:

default drive:CURSOR.OBJ

(The drive designation and filename extension cannot be overridden. You can, however, rename the file, giving a new filename extension, and/or copy the file to a new disk drive, giving a new filename and/or filename extension.)

- ; Use a single semicolon (;) followed immediately by a carriage return at any time after responding to the first prompt (from Library file on) to select default responses to the remaining prompts. This feature saves time and overrides the need to answer additional prompts.

NOTE

Once the semicolon has been entered, you can no longer respond to any of the prompts for that library session. Therefore, do not use the semicolon to skip over some prompts. For this, use carriage return.

Example:

```
Library file: FUN <CR>  
Operation: +CURSOR;<CR>
```

The remaining prompt will not appear, and MS-LIB will use the default value (no cross reference file).

- & Use the ampersand to extend the current physical line. This command character will only be needed for the Operation prompt. MS-LIB can perform many functions during a single library session. The number of modules you can append is limited only by disk space. The number of modules you can replace or extract is also limited only by disk space. The number of modules you can delete is limited only by the number of modules in the library file. However, the line length for a response to any prompt is limited to the line length of your system. For a large number of responses to the Operation prompt, place an ampersand at the end of a line. MS-LIB will display the Operation prompt again, then enter more responses. You may use the ampersand character as many times as you need. For example:

```
Library file: FUN<CR>
Operation: +CURSOR-HEAP+HEAP*FOIBLES&
Operation: *INIT+ASSUME+RIDE;<CR>
```

MS-LIB will delete the module HEAP, extract the modules FOIBLES and INIT (creating two files, FOIBLES.OBJ and INIT.OBJ), then append the object files CURSOR, HEAP, ASSUME, and RIDE. Note, however, that MS-LIB allows you to enter your Operation responses in any order.

Control-C

Use Control-C at any time to abort the library session. If you enter an erroneous response, such as the wrong filename or module name, or an incorrectly spelled filename or module name, you must press CTRL-C to exit MS-LIB then reinvoked MS-LIB and start over. If the error has been typed but not entered, you may delete the erroneous characters, but for that line only.

The first part of the document is a letter to the editor. It discusses the importance of maintaining accurate records and the role of the organization in this regard.

The second part of the document is a report on the activities of the organization during the past year. It details the various projects undertaken, the challenges faced, and the achievements made. The report also includes a list of the organization's members and a summary of the financial statements.

The third part of the document is a list of the organization's members. It includes the names of all members, their contact information, and their respective roles within the organization. This list is intended to provide a comprehensive overview of the organization's membership.

The final part of the document is a list of the organization's financial statements. It includes a balance sheet, an income statement, and a cash flow statement. These statements provide a detailed overview of the organization's financial health and are intended to provide transparency to the members and the public.

CHAPTER 2

ERROR MESSAGES

<symbol> is a multiply defined PUBLIC. Proceed?

Cause: two modules define the same public symbol. The user is asked to confirm the removal of the definition of the old symbol. A No response leaves the library in an undetermined state.

Cure: Remove the PUBLIC declaration from one of the object modules and recompile or reassemble.

Allocate error on VM.TMP

Cause: out of space

Cannot create extract file

Cause: no room in directory for extract file

Cannot create list file

Cause: No room in directory for library file

Cannot nest response file

Cause: "@filespec" in response (or indirect) file

Cannot open VM.TMP

Cause: no room for VM.TMP in disk directory

Cannot write library file

Cause: Out of space

Close error on extract file

Cause: out of space

Error: An internal error has occurred.

Contact Microsoft, Inc.

Fatal Error: Cannot open input file

Cause: Mistyped object file name

Fatal Error: Module is not in the library

Cause: trying to delete a module that is not in the library

Input file read error

Cause: bad object module or faulty disk

Invalid object module/library

Cause: bad object and/or library

Library Disk is full

Cause: no more room on diskette

Listing file write error

Cause: out of space

No library file specified

Cause: no response to Library File prompt

Read error on VM.TMP

Cause: disk not ready for read

Symbol table capacity exceeded

Cause: too many public symbols (about 30K chars in symbols)

Too many object modules

Cause: more than 500 object modules

Too many public symbols

Cause: 1024 public symbols maximum

Write error on library/extract file

Cause: Out of space

Write error on VM.TMP

Cause: out of space