Microsoft® LIB

Library Manager

for 8086 and 8088 Microprocessors

Microsoft Corporation

System Requirements

The Microsoft LIB Library Manager requires:

38K bytes of memory minimum:
- 28K bytes for code
- 10K bytes for run space

Disk drive(s):

One disk drive if and only if output is sent to the same physical disk from which the input was taken. The Microsoft LIB Library Manager does not allow time to swap disks during operation on a one-drive configuration. Therefore, two disk drives is a more practical configuration.

Contents

CHAPTER 1

INTRODUCTION

1.1 FEATURES OF MS-LIB

Microsoft LIB is a library manager. With MS-LIB, you can:

- Create and modify library files that are used with Microsoft's MS-LINK linker utility
- Add object files to a library
- Delete modules from a library
- Extract modules from a library and place the extracted modules into separate object files

MS-LIB can create either general or special libraries, for a variety of programs or for specific programs. With MS-LIB you can create a library, or you can create a library for one program only. The result is fast linking and more efficient execution for a language compiler or for one program.

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 also used in Microsoft MS-LINK, MS-Pascal, MS-FORTRAN, and other 16-bit Microsoft products. If you have used any of these products, using MS-LIB should be familiar to you. Command syntax is straightforward, and MS-LIB prompts you for commands that you have not supplied.

1.2 OVERVIEW OF MS-LIB OPERATION

MS-LIB performs five library manager functions:

Deletes modules

Extracts a module and places it in a separate object file

Appends an object file as a module of a library

Replaces a module in the library file with a new module

Creates a library file

During each library session, MS-LIB deletes or extracts modules, then appends new ones to the library file. 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 that module into memory but does not write it back to the file. When MS-LIB writes back the next module to be retained, it places that module at the end of the last module written. This procedure effectively "closes up" the disk space to keep the library file from growing too large.

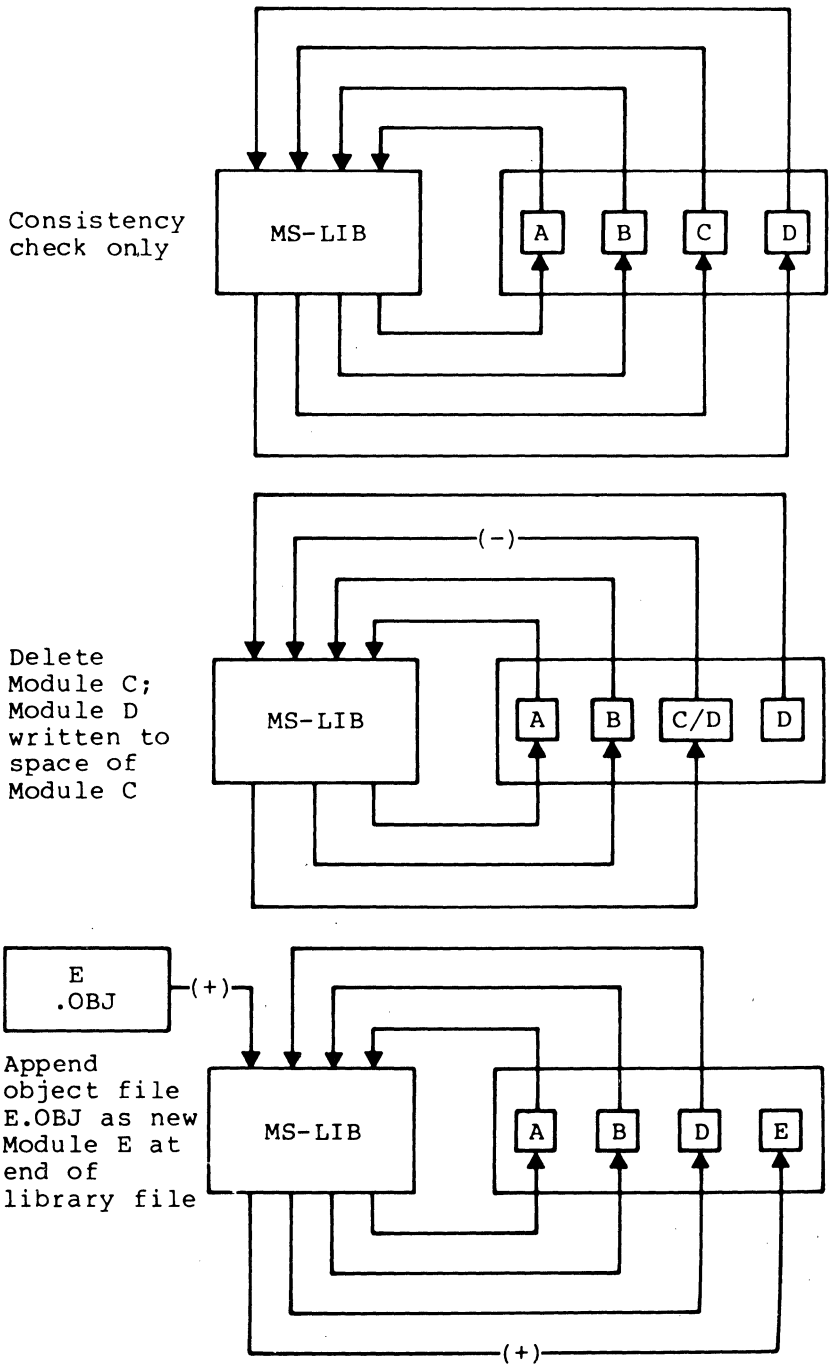
When MS-LIB has read the library file, it appends any new modules to the end of the file. Finally, MS-LIB creates the index, which MS-LINK uses to find modules and symbols in the library file. MS-LIB will output a cross-reference listing of the PUBLIC symbols in the library, if you request such a listing.

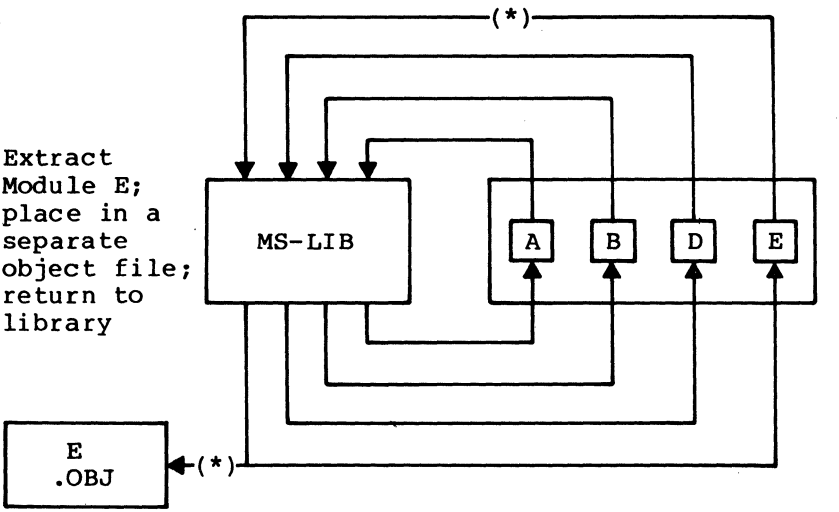
Example:

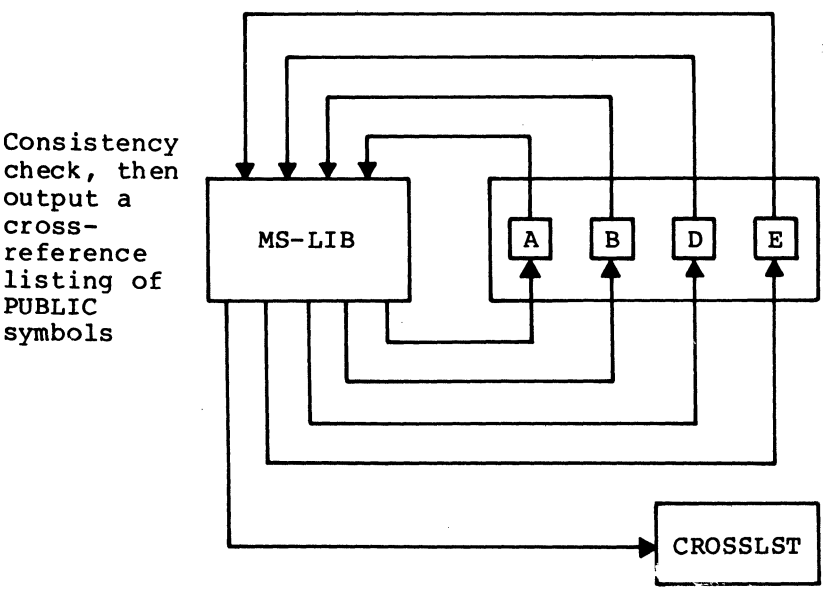
```
LIBx PASCAL+HEAP-HEAP;
```

This command first deletes the library module HEAP from the library file, then adds the file HEAP.OBJ as the last module in the library. 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. This order of execution prevents confusion in MS-LIB when a new version of a module replaces a version in the library file.

The following figure illustrates the MS-LIB operation.


Consistency check only
MS-LIB
A
B
C
D
(-)
Delete Module C; Module D written to space of Module C
MS-LIB
A
B
C/D
D
E .OBJ
(+)
Append object file E.OBJ as new Module E at end of library file
MS-LIB
A
B
D
E
(+)


Extract Module E; place in a separate object file; return to library
(*)
MS-LIB
A
B
D
E
E .OBJ
(*)


Consistency check, then output a cross-reference listing of PUBLIC symbols
MS-LIB
A
B
D
E
CROSSLST

CHAPTER 2

RUNNING MS-LIB

Running MS-LIB requires two types of commands: a command to start MS-LIB and answers to command prompts. Usually you will type all the commands to MS-LIB on a command line or in response to MS-LIB prompts. As an option, answers to the command prompts may be contained in a response file. Command characters can be used to assist you while giving commands to MS-LIB.

2.1 HOW TO START MS-LIB

There are three ways to start MS-LIB. With the first method, you type the commands as answers to individual prompts. By the second method, you type all commands on the line used to start MS-LIB. As a third option, you can create a response file that contains all the necessary commands.

Summary of Methods to Start MS-LIB

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

2.1.1 Method 1: Prompts

To start MS-LIB with method 1, type:

```
LIB
```

MS-LIB will be loaded into memory. MS-LIB will then display three text prompts that appear one at a time. You answer the prompts, commanding MS-LIB to perform specific tasks.

The command prompts are summarized here and described more fully in the Section 2.2, "Command Prompts."

Summary of Command Prompts

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

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 the default drive) and a filename extension. Object modules possess neither of these.

2.1.2 Method 2: Command Line

Type:

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, or asterisk (+, -, or *). 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> field does not exist, MS-LIB will prompt you:

Library file does not exist. Create?

Type Yes to create a new library file. Type No to abort the library session.

<operations> is a command to delete a module, append an object file as a module, or extract a module as an object file from the library file. Use the three command characters plus, minus, and asterisk to direct MS-LIB to append, delete, or extract modules and object files.

<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.

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

Examples:

```
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). The MS-LIB semicolon command character indicates that MS-LIB should use the default responses for the remaining prompts. Refer to Section 2.3, "Command Characters," for more information.

```
LIB PASCAL
```

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

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

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

If you have many operations to perform during a library session, use the ampersand (&) command character to extend the line so that you can type 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.

2.1.3 Method 3: Response File

Type:

LIB @<filespec>

where: <filespec> is the name of a response file. A response file contains answers to the MS-LIB prompts. Method 3 permits you to conduct the MS-LIB session without user responses to the MS-LIB prompts.

IMPORTANT

Before using method 3 to start MS-LIB, you must first create a response file.

A response file has one text line 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 you would for responses typed on the keyboard.

When the library session begins, each prompt will be displayed 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 will be made to the modules currently in the library file, and no cross-reference listing file will be created.)

If you type 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 type a library filename, a carriage return, a comma, and then a list filename, MS-LIB will perform its consistency check of the library file, then produce the cross-reference listing file.

Example:

```
PASCAL
+CURSOR+HEAP-HEAP*FOIBLES
CROSSLST
```

This response file causes MS-LIB to delete the module HEAP from the PASCAL.LIB library file; extract the module FOIBLES and place it 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.

2.2 COMMAND PROMPTS

You command MS-LIB by typing responses to three text prompts. After you have typed 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. You will see the operating system prompt when MS-LIB has finished the library session successfully. If the library session is unsuccessful, MS-LIB will display 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).

Command Prompts

Library File:

Type 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 type 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 type 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 displayed.

If the filename you type does not exist, MS-LIB will display the prompt:

Library file does not exist. Create?

You must type either Yes or No.

Operation:

Type one of the three command characters for manipulating modules (+, -, *); followed immediately (no space) by the module name or the object filename. The plus sign appends an object file as the last module in the library file (see further discussion under the description of plus sign in the next section). The minus sign deletes a module from the library file. The 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), type an ampersand (&) as the last character on the line. MS-LIB will repeat the Operation: prompt, which permits you to type additional module names and object filenames.

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

More information about modules is given in the description of each command character.

List file:

If you want a PUBLIC symbols cross-reference list for the modules in the library file, type the name of a file in which you want MS-LIB to place the cross-reference listing. If you do not type a filename, no cross-reference listing is generated.

The response to the List file: prompt is a file specification. You can specify a drive (or device) designation and a filename extension with the filename. 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 typing 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.

2.3 COMMAND CHARACTERS

MS-LIB provides six command characters. Three of the command characters are required in response to the Operation: prompt. The other three command characters provide you with helpful commands to MS-LIB.

Plus sign

Use the plus sign (+), followed by an object filename, to append 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
```

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

Minus sign

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

Asterisk

Use the asterisk, followed by a module name, to extract the module from the library file and place it into a separate object file. The module will still exist in the library. (The extraction process copies 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 reponse to the Operation: prompt of

```
*CURSOR
```

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

```
A: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.

Semicolon

Use a single semicolon (;), followed immediately by a carriage return at any time after responding to the first prompt (i.e., 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 typed, you can no longer respond to any of the prompts for that library session. Therefore, do not use the semicolon to skip over prompts. To skip prompts, use carriage return.

Example:

```
Library file:  FUN
Operation:  +CURSOR;
```

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

Ampersand

Use the ampersand to extend the current line. This command character is only used in response to the Operation: prompt. 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 by the number of modules in the library file.

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, and then you can type more responses. For example:

```
Library File:  FUN
Operation:  +CURSOR-HEAP+HEAP*FOIBLES&
Operation:  *INIT+ASSUME+RIDE;
```

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 that MS-LIB allows you to type your Operation: reponses in any order. You may use the ampersand character as

many times as needed.

CONTROL-C Use <CONTROL-C> to abort the library session at any time. If you type an incorrect response, such as the wrong filename or module name, or an incorrectly spelled filename or module name, you must press <CONTROL-C> to exit MS-LIB; then you must restart MS-LIB. If the error has been typed and you have not pressed the <RETURN> key, you may delete the erroneous characters for that line only.

Summary of Command Characters

Character	Action
+	Appends an object file as the last module
-	Deletes a module from the library
*	Extracts a module and places in an object file
;	Use default responses to remaining prompts
&	Extends current physical line; repeats command prompt
CONTROL-C	Aborts library session

CHAPTER 3

ERROR MESSAGES

The following are MS-LIB error messages:

<symbol> is a multiply defined PUBLIC. Proceed?
Cause: Two modules define the same public symbol. You are asked to confirm the removal of the definition of the old symbol.

Cure: Remove the PUBLIC declaration from one of the object modules and recompile or reassemble. If you respond No, the library will be left in an indeterminate state.

Allocate error on VM.TMP
Cause: Out of disk 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

MS-LIB cannot open VM.TMP
Cause: There is no room for VM.TMP in disk directory

Cannot write library file
Cause: Out of disk space

Close error on extract file
Cause: Out of disk space

Error: An internal error has occurred
Contact Microsoft Corporation

Fatal Error: Cannot open input file
Cause: You mistyped an object filename

Fatal Error: Module is not in the library
Cause: You tried 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 module and/or library

Library Disk is full
Cause: No more room on disk

Listing file write error
Cause: Out of disk 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 disk space

Write error on VM.TMP
Cause: Out of disk space

INDEX