

Microsoft LIB-80 Library Manager

CP/M® Version

Software Reference Manual

for HEATH/ZENITH 8-bit digital computer systems

Table of Contents

LIB-80, Library Manager

Overview	1-1
LIB-80 Command Strings	1-2
Modules	1-3
LIB-80 Switches	1-4
LIB-80 Listings	1-5
Sample LIB Session	1-6

LIB-80 LIBRARY MANAGER

OVERVIEW

LIB-80 is the object time library manager for the CP/M version of FORTRAN-80, COBOL-80 and the BASIC Compiler. It is used to create and modify libraries which are then linked with compiled programs.

WARNING

Read this document carefully and make a back-up copy of your libraries before using LIB. It is not difficult to destroy a library with LIB-80.

LIB-80 COMMAND STRINGS

To run LIB-80, type LIB and press RETURN. The LIB-80 library manager will be loaded into memory and executed. LIB-80 will return the prompt “*” indicating it is ready to accept commands. Each command in LIB-80 either lists information about a library or adds new modules to the library under construction.

Commands to LIB-80 consist of an optional designation file name which sets the name of the library being created, followed by an equal sign, followed by module names separated by commas. The default designation file name is FORLIB.REL.

Examples:

```
*NEWLIB=FILE1<MOD2>, FILE3, TEST
```

```
*SIN, COS, TAN, ATAN
```

Any command specifying a set of modules concatenates the modules selected onto the end of the last destination file name given. Therefore,

```
*FILE1, FILE2<BIGSUB>, TEST
```

is equivalent to:

```
*FILE1
```

```
*FILE2<BIGSUB>
```

```
*TEST
```

MODULES

A module is typically a FORTRAN or COBOL subprogram or main program, or a MACRO-80 assembly program that contains ENTRY statements.

The primary function of LIB-80 is to concatenate modules in .REL files to form a new library. In order to extract modules from previous libraries or .REL files, a powerful syntax has been devised to specify ranges of modules within a .REL file.

The simplest way to specify a module within a file is simply to use the name of the module. For example:

```
SIN
```

But a relative quantity plus or minus 255 may also be used. For example:

```
SIN+1
```

specifies the module after SIN and:

```
SIN-1
```

specifies the one before it.

You may also specify ranges of modules by using two dots:

```
SIN.. means all modules from and including SIN to the end of the file.  
SIN..COS means SIN and COS and all modules in between.
```

Ranges of modules and relative offsets may also be used in combination:

```
SIN+1..COS-1
```

To select a given module from a file, use the name of the file followed by the module(s) specified enclosed in angle brackets and separated by commas.

Examples:

```
FORLIB<SIN..COS>  
MYLIB.REL<TEST>  
BIGLIB.REL<FIRST,MIDDLE,LAST>
```

If no modules are selected from a file, then all the modules in the file are selected.

LIB-80 SWITCHES

NOTE

/E will destroy the current library if there is no new library under construction. Exit LIB-80 using Control-C if the library is not being revised.

A number of switches are used to control the operation of LIB-80. These switches are always preceded by a slash:

- /O Octal — Set octal typeout mode for /L command.
- /H Hex — Set hex typeout mode for /L command (default).
- /U List the symbols which would remain undefined on a search through the file specified.
- /L List the modules in the files specified and the symbol definitions they contain.
- /C (Create) Throw away the library under construction and start over.
- /E Exit to CP/M. The library under construction (.LIB) is revised to .REL and any previous copy is deleted.
- /R Rename — same as /E but does not exit to CP/M on completion.

LIB-80 LISTINGS

To list the contents of a file in cross reference format, use /L.

```
*FORLIB/L
```

When you are building libraries, it is important to order the modules such that any intermodule references are "forward". The module containing the global reference should physically appear ahead of the module containing the entry point. Otherwise, LINK-80 may not satisfy all global references on a single pass through the library.

Use /U to list the symbols which could be defined in a single pass through a library. If a module in the library makes a backward reference to a symbol in another module, /U will list that symbol.

Example:

```
SYSLIB/U
```

NOTE: Since certain modules in the standard FORTRAN and COBOL systems are always force-loaded, they will be listed as undefined by /U but will not cause a problem when loading FORTRAN or COBOL programs.

Listings are currently always sent to the terminal; use control-P to send the listing to the printer.

SAMPLE LIB SESSION

Building a library:

```
A>LIB
*TRANLIB=SIN,COS,TAN,ATAN,ALOG
*EXP
*/E
A>
```

Listing a library:

```
A>LIB
*TRANLIB.LIB/U
*TRANLIB.LIB/L
.
.
.
(list of symbols in TRANLIB.LIB)
.
.
.
*Control-C
A>
```