

Section 3

ZiLOG Z8 Librarian User's Guide

L
i
b
r

L
i
b
r

16.1 Introduction

An object librarian maintains collections of object modules generated by the assembler or compiler. The ZiLOG Z8 Librarian (**Z8LIB**) allows you to add, remove, replace, list, and extract modules. Once a library is created it may be used by the linker just as a module is used. However, only the modules residing in the library that are required by the linker are incorporated into the executable file. In this way collections of modules that may, or may not be required by a program, can be made available at link time.

16.2 Runtime Interface

The following describe the command line syntax of the librarian:

Z8LIB [options]<libraryname>[commands]

or

Z8LIB @<control file>

In the second form <control file> is a file which contains any number of options and commands. Commands in the control file can be spread across multiple lines. When multiple lines are used, each line, except for the last line, should end with a backslash (\). A command line can not exceed 3096 bytes. Lines that begin with a semicolon (;) are treated as comments.

NOTE: The librarian always makes a new library file (<filename>.LIB) when adding, deleting, or replacing modules. The old library is copied to <filename>.BAK.

Extracting modules, or listing public symbols does not change the library.

Module names may be specified without the **.OBJ** extension. **.OBJ** files are maintained using their internal names (written to the **.OBJ** header by the assembler). Therefore, to avoid confusing the librarian, do not rename **.OBJ** files.

16.3 Librarian Options

The following section describes the options allowed by the librarian. All options must be preceded by a - or /. Options may be abbreviated but the abbreviation must be unique; otherwise an error results.

16.3.1 HELP

The **HELP** option displays the librarian help screen.

Format:

-HELP

16.3.2 NOWARN

The **NOWARN** option tells the librarian to suppress warning messages.

Format:

-NOWARN

16.3.3 WARN

The **WARN** option tells the librarian to produce warning messages. The librarian writes warnings to the screen or to a user-specified warning file.

Format:

-WARN[=<warn filename>]

16.4 Librarian Commands

The following describes the commands supported by the librarian.

16.4.1 <libraryname>

If the librarian detects a library name, it expects certain variables:

- **<libraryname>** is the name of the library on which to perform the commands. If the library specified does not exist then it is created. The default extension **.LIB** is used for **<libraryname>**.
- **<modulename>** is an object file or a library file to be extracted, added, removed, or replaced in the library specified. If the **<modulename>** extension is not specified, **.OBJ** is assumed.
- **[commands]** are **ADD(+)**, **REMOVE (-)**, and **EXTRACT (*)**. These may be specified in any order and any number of times.

Format:

```
<libraryname> [= [<command>] <modulename> [ ,
                  [<command>] <modulename> ]]
```

Examples:

```
mylib =- module1
mylib =+ module1
mylib =* module1
mylib =+ module1, *module2
mylib =+ module1 *module2
```

16.4.2 ADD (+)

The **ADD** command inserts a copy of **<modulename>** into the library file. The file specified by **<modulename>** is not modified in any way. The **ADD** command is performed after any **REMOVE** command operating on the **<modulename>**.

A **<modulename>** may be an object file or a library file. If the **<modulename>** extension is not specified, **.OBJ** is assumed.

Format:

```
+<modulename>
```

Example:

mylib =+ module1

16.4.3 EXTRACT (*)

The **EXTRACT** command locates the module specified in the library file and creates an **.OBJ** file in the current directory.

WARNING: If an **.OBJ** file of the same name is already present in your current directory, THE FILE IS OVERWRITTEN.

Format:

*** <modulename>**

Example:

mylib =* module1

16.4.4 LIST

The **LIST** command lists all modules and their public symbols contained in the specified library.

Format:

LIST

Example:

mylib LIST

16.4.5 REBUILD

The **REBUILD** command forces the librarian to recreate a specified library. When a module is removed, the header block remains marked for removal. The space occupied by the module itself is free for reuse but the block header remains. **REBUILD** recreates the library, copying only active blocks and modules.

When a library is created or rebuilt, <n> number of header block positions are allocated. When these header blocks are used up, the librarian forces a **REBUILD**.

Format:

REBUILD [<n>]

Examples:

```
mylib REBUILD
mylib REBUILD 100
```

16.4.6 REMOVE(-)

The **REMOVE** command marks the module specified as removed. All removes are performed before any **ADD** of the same module. When you rebuild the library the removed modules are not copied over to the new library.

Format:

```
- <modulename>
```

Example:

```
mylib -= module1
```

16.4.7 REPLACE (-+)

The **REPLACE** command is treated as a **REMOVE/ADD** command. Each module is first removed, then added.

Format:

```
-+ <modulename>
```

Example:

```
mylib -=+ module1
```

16.4.8 SEARCHPATH

The **SEARCHPATH** command establishes an additional search path to be specified in locating files.

The search order is:

1. current directory
2. environment path
3. search path

Format:

SEARCHPATH = “<path>”

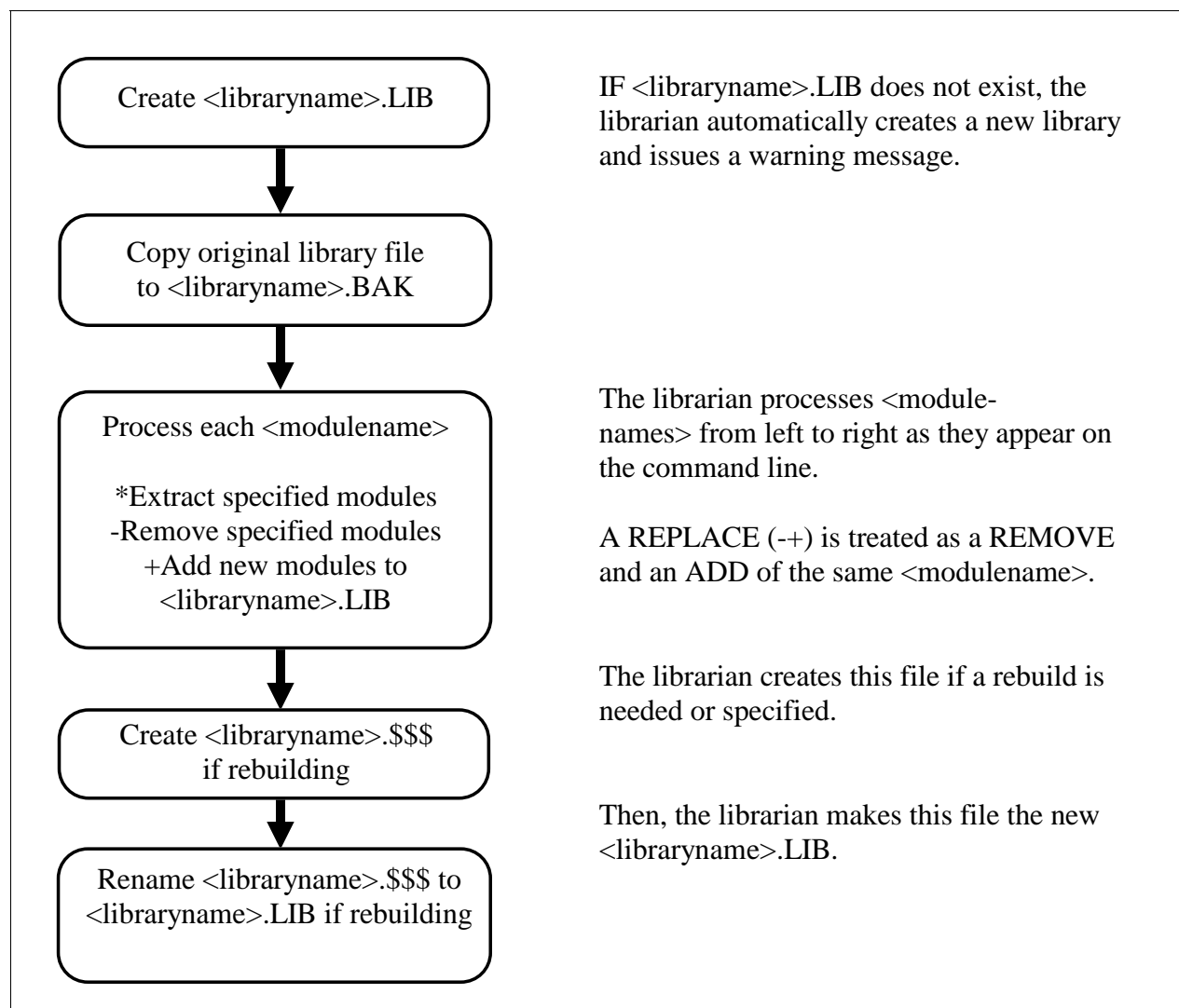
Example:

searchpath = “c:\project”

NOTE: The environment path is set by the DOS command:

SET Z8=<path>

which is usually placed in the **AUTOEXEC.BAT** file.



Librarian