GenLIB
v1.00

Reference guide.

CONTENTS

INTRODUCTION

GenLIB is used to maintain files of routines which you, as a programmer, don't want to rewrite for each application you are developing.

You may build libraries of subroutines which you use many times, or you may want to build a set of subroutines which you would distribute to other programmers. (You could build a set of graphics subroutines for the 9938 video processor which you might want to sell to other programmers.)

Subroutine libraries generally reduce the time it takes to develop new applications, though it may take a bit more effort on the first application you write to make the subroutines into a library file. Libraries also reduce the size of the source code for new applications, since the source code for a library is only written once, no matter how many applications use the library.

OVERVIEW

GenLIB uses four different types of files during the course of its execution, these will be described briefly now.

"LIBR" is the MDOS program file which reads the control file (or keyboard input), tagged object files, and updates library files.

Control file (or keyboard input), this file contains commands which must be executed by GenLIB in order to update library files. A control file (if keyboard input is not used) must be created by the user using a standard text editor which saves files in Dis/Var 80 format.

Tagged Object file(s), this type of file is usually created by an assembler (or compiler) and contains all of the information needed for REFerencing subroutines from other programs. It may include information to be passed on to a symbolic debugger.

Tagged Object library(s), this type of file is a collection of tagged object files which are updated with GenLIB. You will generally add many simple, often-used, subroutines to these using GenLIB.

USING GenLIB

GenLIB is executed from an MDOS command line or from within an MDOS batch file.

You must perform the following actions before using GenLIB:

First, MDOS must be able to find the file "LIBR" somewhere in your current command path (set with the "PATH" command in MDOS.)

Second, you must create all tagged object files which are to be included in the resultant library file.

Third, GenLIB must be able to write to the library file, or create if it doesn't already exist. This means that your destination disk can not be write protected, and that it must have enough free sectors to allow the files to be written.

Recommended (optional), you should create a Dis/Var 80 control file for the librarian.

Continuing with the assumption that the previous conditions have been met, GenLIB is invoked from MDOS with the following command format:

LIBR [control_file] (brackets indicate optional items)

If no control file was specified, GenLIB will prompt you for keyboard input. Otherwise, it will read one line at a time from the control file and process each line as if it had been typed from the keyboard.

CREATING A CONTROL FILE

Control files may be created with any editor which can save files in a Dis/Var 80 format. You may have one command per line in the file, and you may have comment lines. Each line in the control file must begin with a non-blank character for GenLIB to recognize it as a valid command.

Control files are most useful when you want to run the librarian in a completely automatic mode, from a batch file, or as part of an update batch file created by GenMAKE.

"EXIT" must be the last command executed in the control file for completely automatic operation. If GenLIB does not find an "EXIT" command before it finds the end of the control file, it will prompt you for keyboard input.

Commonly used commands

ADD

syntax: ADD object_filename

You will use the ADD command to include new subroutines into the library file selected with the MAINT command. All subroutines you wish to add must be DEFined in the object file.

The first function performed by the ADD command is to read the entire object file into memory, so that it can determine all of the names DEFined in the object module and the number of records in the object module.

For each name DEFined in the object module, the ADD command will search the library index to see if any duplicate names are present. If any name in the object module is the same as a name already in the library file, you will receive an error message and GenLIB will proceed to the next available command.

If none of the names in the object module were already in the library index, the ADD command will search the library's free space list for a block with enough free records to load the object module. If a block with enough records is found, the ADD command will remove the appropriate number of records from the block and copy the entire object file into library file.

After copying the object file into the library file, the ADD command will insert each DEFine from the object module into the library's index.

NOTE: An object module can have more than one subroutine in it, but all of the subroutines in an object module will be included in an image file by GenLINK, even if only one is referenced.

MAINT

syntax: MAINT "library_filename"

You must the MAINT command to tell GenLIB the name of the library file which will be updated with subsequent commands. If the library files does not yet exist, the MAINT command will create the file and initialize all of the header information in record number zero of the file.

SAVE

syntax: SAVE

This command causes GenLIB to close the library file currently in use and to make sure that all of the changes you have made to the library have been written to the disk.

General Commands

COMMENTS

syntax: *<any text you want until end of line>

Any line whose first character is an asterisk, "*", is ignored by GenLIB. COMMENTS are useful for documenting various part of a control file, so that you can remember later on why you did things in a certain way. COMMENTS are also useful for explaining the logic of what you did to someone you will never meet, if you plan to distribute your control files to other people.

DELETE

syntax: DELETE "symbol_name"

The DELETE command is used to remove a symbol from the library index. It does not remove the object code associated with the symbol from the library file. This command is useful if you want to replace a certain subroutine in a library without rebuilding the whole library from scratch. (The recommended procedure is to rebuild the whole library, with a control file, when you want to replace a certain subroutine.)

EXIT

syntax: EXIT

The EXIT command causes GenLIB to return to MDOS.

HELP

syntax: HELP

The HELP command provides a list of all commands recognized by GenLIB, and a brief summary of the correct syntax for the command.

MDOS

syntax: MDOS <mdos command string>

The MDOS command allows you to execute any internal command of the Command Line Interpreter. Functions such as DIR and DEL are most commonly used in this manner.

SYMTAB

syntax: SYMTAB

The SYMTAB command causes GenLIB to display the value and name of each symbol in the library index.

Object file format

See the documentation for GenLINK for complete information on this topic.

Library file format

See the documentation for GenLINK for complete information on this topic.

Example control file

```
*
* library control file for LIB_4A
*
* first, delete outdated versions
*
MDOS DEL LIB_4A
*
* create the library file <again>
*
MAINT LIB_4A
*
* add the video routines
*
ADD VIDEO1    vsbr
ADD VIDEO2    vsbw
ADD VIDEO3    vmbr
ADD VIDEO4    vmbw
ADD VIDEO5    vset_addr
ADD VIDREG    32 byte BSS for registers
ADD VDPREG    addresses of VDP registers
*
* add the DSR call routine
*
ADD DSRLINK
*
* add the GPL call routine
*
ADD GPLLINK
ADD GROMREG   addresses of GRAM/GROM registers
ADD XMLLINK
*
* add the object code loader
*
ADD LOADER
*
* add the various keyscan routines
*
ADD KSCAN1    normal keyscan
ADD KSCAN2    CRU keyscan
*
SAVE
*
EXIT
*
```