

2500 A.D. Librarian Description

The 2500 A.D. Librarian is used to create a library of user specified object modules to be linked together with a program created by a 2500 A.D. assembler. The linker will search the specified libraries and only include the referenced library modules.

The Librarian will process any size file, as long as enough disk space is available. A library is limited to 256 separate modules. The Librarian requires enough disk space for an existing library and space for a temporary file the size of the existing library plus any modules being added to the library. A temporary file is used to create a library to minimize the possibility of damage to an existing library. The Librarian also requires enough memory to store all the global symbol records of the library modules, and the module directory list. The global symbols are checked for multiply defined symbols.

The 1..'brarian command line parser only recognizes two operand separators space and tab. The command descriptions show the full command name and the allowed abbreviations. The operands allowed by each command are also shown. The system defaults used by the Librarian are :

obj	object filename extension
pak	packed object filename extension
lib	library filename extension
tmp	temporary filename extension

The Librarian displays the modules contained in a library on the screen. The current working module is displayed with highlighting. The current working module can be changed by scrolling up or down the library directory list. To scroll up the directory list type k or K and to scroll down the directory list type j or J. The directory list can be displayed 16 modules at a time. If a module is added to the list and is outside the currently displayed modules, the screen is adjusted to display the added module. The added module is always displayed as the current module. A

module may be a single object file or a module in a packed object file. A packed object file is several object files concatenated together. See the MODULE and ENDMOD directive descriptions in the Assembly Mode section of this manual for examples of creating a packed object file. A packed object file may be used for an operation on a single module or all modules within the file. The operand **all** or **ALL** may only be used with a packed object file, and will cause all the modules within the file to be added to a library or replace the modules in the library.

Note to VMS users:

VMS users must type carriage return after using 'j' or 'J' as well as 'k' or 'K'.

;

3 - 2

Librarian Installation

The 2500AD Librarian is included on the distribution media for the Assembler. The filename for the librarian on the UNIX or ULTRIX operating system is LIB. The filename for the librarian on the MSDOS or VMS operating system is LIB.EXE. Installation of the librarian is dependent on the host operating system. Users of the UNIX or ULTRIX operating systems must perform step 1 of the installation directions. Users of the MSDOS or VMS operating systems must also perform the appropriate section of step 2 of the installation directions.

STEP 1: Copy the file lib or lib.exe to the desired directory and disk drive.

NOTE: The file LIB.EXE will need to be renamed on the VMS operating system so the 2500AD librarian will not conflict with the VMS librarian utility.

STEP 2: This only applies to users of the MSDOS or VMS operating systems. Please refer to the section appropriate to your host operating system.

MSDOS:

The librarian requires the use of the ANSI device driver. The systems config.sys file must be modified to cause the MSDOS operating system to load the device driver. The config.sys file is located in the root directory, if the config.sys file does not exist one must be created. Please refer to your MSDOS manual for information on creating a config.sys file. Add the following line to the config.sys file.

```
DEVICE=ANSI.SYS
```

NOTE: The system will need to be rebooted for the MSDOS operating system to load the ANSI device driver.

VMS:

The librarian will need to be defined as a command to the VMS operating system, if you wish to execute the librarian without using the RUN command. The following line must be added to the LOGIN.COM file.

librarian == "\$Diski :librarian.exe"

The filename for the librarian shown in the previous line is dependent upon the name the librarian was given when it was copied from the distribution media.

Librarian Operating Instructions

The Librarian will accept one command line argument, the filename of an existing library or a library to be created.

To run the Librarian and read an existing library type :

fib filename

Where the filename is an existing library.

To run the Librarian and create a new library type :

lib filename

Where the filename is the new library name.

The Librarian may also be invoked with no command line arguments and an existing library or a new library specified from the Librarian command line.

To run the Librarian with no file specified type :

lib

The Librarian will respond with the prompt :

Enter Command:

The Librarian commands will be described in the next section. The Librarian is designed to be used interactively and cannot be used with an Msdos batch file or Unix shell file.

ADD filename
A

The ADD command adds a module to a library, or replaces an existing module in a library. The command accepts one operand, the filename of a module to be added to a library or the operand all or ALL. The operand all or ALL is only allowed with a packed object file. The ADD command will not allow a module to be added unless a library has been open or created with the **NEW** command. The Librarian will always prompt for the object filename containing the module to be added. If the file specified is a packed object file the file is searched for the module, and if found the module is added to the library. If the maximum module limit is reached the module will not be added. If the operand is not present the Librarian will prompt for the module name.

Command Examples:

add printf

The Librarian will respond with the prompt:

Enter Name of Object File:

Assume the name mylib was entered at the prompt. The Librarian will search for the file mylib.obj, and if the file is not found the Librarian will search for the file mylib.pak.

add all

The Librarian will respond with the prompt:

Enter Name of Object File:

2500 A.D. Librarian
2500 A.D. Librarian

2500 A.D. Librarian

Assume the filename is mylib.pak. The Librarian will add all the modules contained in the file mylib.pak to the library.

add

The Librarian will respond with the prompt :

Enter Name of Module:

After the module name is entered the Librarian will prompt for the name of the file containing the module as in the previous examples.

3 - 7

DEL module

The DEL command deletes the named module or the current working module from the library. The current working module is the highlighted module on the screen. If the operand is not present the current working module is deleted. If the operand is present the library directory list is searched for the named module. It is an error to delete a module if no library has been opened, the library contains no modules or the named module cannot be found in the directory list.

Command Examples:

del printf

The Librarian will search for the directory entry printf and delete the entry from the list removing the module from the library. The next module in the directory list becomes the current working module, and if the deleted module was the last module in the list the previous module becomes the current working module.

del

The Librarian will delete the current working module from the directory list removing the module from the library.

EXIT

The EXIT command will save a library, if a library is open and contains at least one module, and return control to the operating system.

Command Examples:

```
exit
```

If a library is open and contains at least one module the Librarian will display the message :

```
Saving Library : library name
```

FIND module

The FIND command maybe used to locate a module in a large library. The command accepts one operand and the name of the module to find. The module, if found, will become the current or highlighted module. An error message will be displayed if the module cannot be found in the library.

Command Example:

```
find printf  
HELP    command name
```

The HELP command displays a summary of all the Librarian commands, or a description of the command specified by the optional operand. The directory display is cleared and the help text is displayed in place of the directory list. The Librarian prompts for any character to be typed to continue. The directory list is restored and the Librarian will wait for the next command.

Command Example:

```
HELP
```

The main help screen describing how to use the HELP command is displayed. The Librarian will respond with the prompt:

Press Any Key To Continue:

help ADD

The help screen describing how to use the ADD command is displayed and the Librarian responds with the prompt :

Press Any Key To Continue:

LIST module

The list command will list the global symbols of the specified module to the screen or to a file. The global'symbols are listed by name in sorted order. The command may have two operands. The first operand the module name or the word ALL or all. The operand ALL will list the global symbols for all modules in the library. The second operand is optional and must be the word disk or DISK. The second operand specifies sending the global symbol listing to a disk file .

Command Examples:

LIST mod2
LIST ALL disk

NEW filename

The **NEW** command creates a new library or opens an existing library. If a library was previously open and contained at least one module the previous library will be saved before the new library is opened. The command requires one operand the filename of the library to create or open; if the operand is not present the Librarian will prompt for the filename. A library must be open before modules can be added, deleted or replaced.

Command Example:

```
new mylib
```

The Librarian will search for an existing library with the filename mylib.lib. If the library exists the library directory is displayed on the screen. If the library does not exist a library with the name mylib.lib will be created.

QUIT

The **QUIT** command causes the Librarian to terminate and return control to the operating system. If a library is open and contains at least one module the Librarian will ask whether the library is to be saved. Typing y or Y will cause the library to be saved, any other character will result in the library being abandoned.

Command Example:

```
QUIT
```

If a library is open and contains at least one module the Librarian will respond with the prompt :

```
Do You Want to Save the Library (yin) :
```

Any character other than y or Y will cause the library to abandoned.

REP module

The **REP** command is used to replace an existing module in a library with a new version of the module. The command accepts one optional operand the module to replace, or the operand **all** or **ALL**. The **all** operand may only be used with packed object files. If the operand is present the library directory list is searched for the named module, and if the module is found it is replaced. If the module is not found an error message is displayed. If the operand is not present the current working module is replaced. The current working module is defined as the module highlighted on the screen. It is an error to replace a module if no library is open or the library contains no modules. The Librarian will always prompt for the object filename containing the updated version of the module to be replaced.

Command Examples:

```
rep printf
```

The Librarian will search the library directory list for the module printf, and if the module is found displays the prompt :

Enter Name of Object File:

Assume the filename mylib was entered. The Librarian will first search for the file mylib.obj. If the file mylib.obj is not found the Librarian will search for the file mylib.pak, and if the file is found search for the module printf.

```
rep
```

The Librarian will use the module name of the current working module (highlighted module) as the module to replace. The Librarian then displays the prompt :

Enter Name of Object File:

STAT

The **STAT** command displays the status of the current library. If no library is open the command displays a message that no library is open. If a library is open the total number of modules, externals, globals, and the library name is displayed. The directory display is overwritten and the status message is displayed, the Librarian then prompts for any key to be pressed to continue. The directory display is then restored.

Command Example:

```
stat
```

The library status is displayed, and the Librarian displays the prompt:

Press Any Key To Continue:

After a key is pressed the directory list display is restored and the Librarian waits for the next command to be entered.

TOP

The TOP command moves the current working directory highlighting to the first module in the directory list.

Command Example:

```
top  
BOT
```

The SOT command moves the current working directory highlighting to the last module in the directory list.

Command Example:

```
bat
```

Librarian Error Messages

This Section provides a list of the error messages output by the Librarian and an explanation of the error message. The command line is output with the error message unless otherwise specified.

- Error - Input Line Too Long
Meaning - The input line exceeded 80 characters.
- Error- Illegal Command
Meaning - The command entered was not a valid command or the ADD,DEL or REP command was used and a library was not open.
- Error- Read Error
Meaning - An error occurred while a file was being read. The name of the file is output with the error message.
- Error - Write Error
Meaning - An error occurred while a file was being updated. The name of the file is output with the error message.
- Error - Cannot Open File
Meaning - The file specified could not be accessed or opened. The name of the file is output with the error message.
- Error - Multiple Defined Module
Meaning - The module name being added to a library already exists in the library. Replace the module or rename the module to be added to the library. The name of the file is output with the error message.
- Error - Filename Too Long
Meaning - The filename specified was too long after the default extension was appended.

- Error - Undefined Module.
Meaning - The module being replaced or deleted from library does not exist in the library
- Error - Not Enough memory
Meaning - There was not enough memory for the buffer or symbol table space required
- Error - Seek Error
Meaning- An error occurred while the position within a file was being updated.
- Error- Cannot Delete Old Library
Meaning - An old library file could not be deleted after the new library was created or updated. The name of the file is output with the error message.
- Error - Cannot Create New Library
Meaning - The temporary library file could not be renamed. The name of the file is output with the error message.
- Error - Incompatible Object Module
Meaning - A library file was specified with the ADD command or an object module was specified with the NEW command.
- Error - Too Many Command Line Arguments
Meaning - The Librarian was invoked with more than one argument from the operating system command line.
- Error - Multiple Defined Global Symbol
Meaning - A global symbol in a module being added to a library contains a symbol name that exists in another module. The symbol name and the filename are output with the error message.
- Error - Maximum Module Count Exceeded
Meaning - The maximum module count of 256 modules was exceeded.
- Error - Must Be A Packed Object File
Meaning - The operand "all" or "ALL" was used for the ADD or REP commands, and the file specified was not a packed object file.