

VMS

---

digital

VMS Librarian Utility Manual

Order Number AA-LA61A-TE

# VMS Librarian Utility Manual

Order Number: AA-LA61A-TE

**April 1988**

This manual explains how to create and maintain VMS libraries.

**Revision/Update Information:** This manual supersedes the *VAX/VMS Librarian Reference Manual, Version 4.0*.

**Software Version:** VMS Version 5.0

**digital equipment corporation  
maynard, massachusetts**

---

**April 1988**

---

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

---

Copyright ©1988 by Digital Equipment Corporation

All Rights Reserved.  
Printed in U.S.A.

---

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	
DECUS	RSTS	
DECwriter	RSX	

**digital**™

ZK4547

---

**HOW TO ORDER ADDITIONAL DOCUMENTATION  
DIRECT MAIL ORDERS**

**USA & PUERTO RICO\***

Digital Equipment Corporation  
P.O. Box CS2008  
Nashua, New Hampshire  
03061

**CANADA**

Digital Equipment  
of Canada Ltd.  
100 Herzberg Road  
Kanata, Ontario K2K 2A6  
Attn: Direct Order Desk

**INTERNATIONAL**

Digital Equipment Corporation  
PSG Business Manager  
c/o Digital's local subsidiary  
or approved distributor

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

\* Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473.

---

---

## Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by DIGITAL. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use DIGITAL-supported devices, such as the LN03 laser printer and PostScript<sup>®</sup> printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.



---

# Contents

---

PREFACE	vii
NEW AND CHANGED FEATURES	ix
<b>LIBRARIAN Description</b>	<b>LIB-1</b>
1 TYPES OF LIBRARIES	LIB-1
2 STRUCTURE OF LIBRARIES	LIB-2
3 CHARACTER CASE OF LIBRARY KEYS	LIB-2
4 SHAREABLE IMAGE LIBRARIES	LIB-3
5 HELP LIBRARIES	LIB-3
5.1 Creating Help Files	LIB-4
5.2 Formatting Help Files	LIB-5
5.3 Help Text Example	LIB-6
5.4 Retrieving Help Text	LIB-8
6 LIBRARIAN ROUTINES	LIB-10
<b>LIBRARIAN Usage Summary</b>	<b>LIB-11</b>
<b>LIBRARIAN Qualifiers</b>	<b>LIB-13</b>
/BEFORE	LIB-14
/COMPRESS	LIB-15
/CREATE	LIB-17
/CROSS_REFERENCE	LIB-19
/DATA	LIB-20
/DELETE	LIB-21
/EXTRACT	LIB-22
/FULL	LIB-23
/GLOBALS	LIB-24
/HELP	LIB-25
/HISTORY	LIB-26
/INSERT	LIB-27
/LIST	LIB-28

## Contents

/LOG	LIB-30
/MACRO	LIB-31
/MODULE	LIB-32
/NAMES	LIB-33
/OBJECT	LIB-34
/ONLY	LIB-35
/OUTPUT	LIB-36
/REMOVE	LIB-38
/REPLACE	LIB-39
/SELECTIVE_SEARCH	LIB-40
/SHARE	LIB-41
/SINCE	LIB-42
/SQUEEZE	LIB-43
/TEXT	LIB-44
/WIDTH	LIB-45

---

## INDEX

---

## EXAMPLES

LIB-1	Help Text for LIBRARY Command _____	LIB-7
LIB-2	HELP LIBRARY Display _____	LIB-8

---

## TABLES

LIB-1	LIBRARY Command Qualifier Compatibilities _____	LIB-13
-------	---	--------

---

# Preface

---

## Intended Audience

This document is intended for all VMS users.

---

## Document Structure

This document consists of the following three sections:

- Description—Provides a full description of the Librarian Utility (LIBRARIAN).
- Usage Summary—Outlines the following information:
  - Invoking the utility
  - Exiting the utility
  - Directing output
- Qualifiers—Describes LIBRARIAN qualifiers, including format, parameters, and examples.

---

## Associated Documents

For related information about this utility, see the following documents:

- *VMS DCL Dictionary*
- *Guide to VMS Programming Resources*

# Preface

---

## Conventions

Convention	Meaning
<code>RET</code>	In examples, a key name (usually abbreviated) shown within a box indicates that you press a key on the keyboard; in text, a key name is not enclosed in a box. In this example, the key is the RETURN key. (Note that the RETURN key is not usually shown in syntax statements or in all examples; however, assume that you must press the RETURN key after entering a command or responding to a prompt.)
<code>CTRL/C</code>	A key combination, shown in uppercase with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination CTRL/C indicates that you hold down the key labeled CTRL while you press the key labeled C. In examples, a key combination is enclosed in a box.
<code>\$ SHOW TIME</code> <code>05-JUN-1988 11:55:22</code>	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
<code>\$ TYPE MYFILE.DAT</code> . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data you would enter is shown.
<code>input-file, . . .</code>	In examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.
<code>[logical-name]</code>	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe is used to refer to a single quotation mark (').

---

---

## **New and Changed Features**

This version of the Librarian Utility (LIBRARIAN) contains no significant technical changes.



---

# LIBRARIAN Description

This section describes how to use the Librarian Utility (LIBRARIAN).

---

## 1 Types of Libraries

You can use the LIBRARIAN to maintain the following types of libraries:

- Object libraries, which contain the object modules of frequently called routines. The VMS Linker searches specified object module libraries when it encounters a reference it cannot resolve in one of its input files. For more information about how the linker uses libraries, see the *VMS Linker Utility Manual*.

An object library has a default file type of OLB and defaults the file type of input files to OBJ.

- Macro libraries, which contain macro definitions used as input to the assembler. The assembler searches specified macro libraries when it encounters a macro that is not defined in the input file. See the *VAX MACRO and Instruction Set Reference Manual* for information about defining macros.

A macro library has a default file type of MLB and defaults the file type of input files to MAR.

- Help libraries, which contain modules of help text that provide user information about a program. You can retrieve help text at DCL level by executing the DCL command HELP, or in your program by calling the appropriate LIBRARIAN routines. For information about creating help modules for insertion into help libraries, see Section 5.

A help library has a default file type of HLB and defaults the file type of input files to HLP.

- Text libraries, which contain any sequential record files that you want to retrieve as data for a program. For example, program source code can be stored in text libraries. Each text file inserted into the library corresponds to one library module. Your programs can retrieve text from text libraries by calling the appropriate LIBRARIAN routines.

A text library has a default file type of TLB and defaults the file type of input files to TXT.

- Shareable image libraries, which contain the symbol tables of shareable images used as input to the linker. For information on how to create a shareable image library, see Section 4.

A shareable image library has a default type of OLB and defaults the file type of input files to EXE.

You can create library files that do not have the default file type. For example, you can create the object library LIB.xxx by typing the following:

```
$ LIBRARY/CREATE/OBJECT LIB.xxx *.obj
```

You can then access the object library by typing the following:

```
$ LIBRARY/LIST LIB.xxx
```

# LIBRARIAN Description

---

## 2 Structure of Libraries

Every library contains a library header that describes the contents of the library, for example, its type, size, version number, creation date, and number of indexes.

Similarly, each module in the library has a module header that contains information about the module, including its type, attributes, and date of insertion into the library.

All libraries contain an index called the module name table (MNT); the keys in the MNT are the names of the modules in the library. Object module libraries also contain an index called the global symbol table (GST); the keys in the GST are the names of the global symbols defined in each of the library modules.

Note that the MNT catalogs modules by module name, rather than by the name of the input file that contained the inserted module. The only exception to this procedure occurs with text libraries, for which the file name of the input file containing the text automatically becomes the module name (unless you use the /MODULE qualifier).

For more information about the structure of libraries, see the *VMS Utility Routines Manual*.

---

## 3 Character Case of Library Keys

The character case of module names and global symbols in libraries depends on the library type:

- Help libraries—Module names remain in the format in which they were entered; that is, individual uppercase and lowercase characters are preserved. However, a second, identically spelled module name (but of a different or mixed character case) to the same library replaces the previous module name, and character case is ignored for match operations; for example, *help Sample* and *help SAMPLE* access the same module.
- Object libraries—Module names and global symbols are in the format in which they were entered. A second, identical keyword (but of a different or mixed character case) to the same library initiates a separate keyword; the previous keyword is not replaced. Match operations require the character case to be identical; for example, for *SAMPLE*, you must enter *SAMPLE*, not *Sample* or *sample*.
- Text and macro libraries—All module names are converted to uppercase characters; for example, *Sample* becomes *SAMPLE*. Likewise, for match operations, either *Sample* or *sample* matches *SAMPLE*.

---

## 4 Shareable Image Libraries

A shareable image library is made up of the symbol tables of shareable images, which serve as input to the VMS Linker. To create a shareable image library, use the LIBRARY command with the /SHARE qualifier, as follows:

```
$ LIBRARY/CREATE/SHARE MYSHARLIB MYSHRIMG1,MYSHRIMG2,SHRIMG3
```

You can then specify the library in the LINK command exactly as if it were an object library.

```
$ LINK/MAP/FULL MYPROG, MYSHARLIB/LIBRARY
```

The linker includes whatever shareable images are needed from MYSHARLIB.

To explicitly include a shareable image, use the /INCLUDE qualifier.

```
$ LINK/MAP/FULL MYPROG, MYSHARLIB/INCLUDE=(MYSHRIMG1)/LIBRARY
```

For each shareable image found that either contains a necessary symbol or was specifically requested with the /INCLUDE qualifier, the linker looks up the image file module (default file type is EXE) and processes it as if it had been specified in an options file.

Unless the search is disabled with the /NOSYSSHR qualifier, the linker also searches the library SYS\$LIBRARY:IMAGELIB.OLB after processing any user default libraries (LNK\$LIBRARY). Modules found in IMAGELIB.OLB are opened with a default file specification of SYS\$LIBRARY:.EXE.

The default file type for the LIBRARY/SHARE command is OLB for the shareable image symbol table library and EXE for the input shareable image files.

The LIBRARIAN uses the GSMATCH of the shareable image as the module ID in the library. The linker issues a warning message if the GSMATCH of the library module is not equal to the GSMATCH found in the corresponding shareable image. A warning message is also issued if the creation date or times found in the library differ from the shareable image. For more information about GSMATCH processing, see the description of the linker in the *VMS Linker Utility Manual*.

You should note that a module inserted into a shareable image library contains only the module header and end-of-module record, which are extracted from the global symbol table of the input shareable image. Consequently, although it is not an illegal action, there is little reason to extract modules from a shareable image library.

---

## 5 HELP Libraries

Help text is a convenient means of providing specific information about a program to an interactive user. The help text is stored as modules in help libraries. You can access the help modules by using the DCL command HELP or by calling the appropriate LIBRARIAN routines described in the *VMS Utility Routines Manual*. In this way, a program can quickly retrieve relevant information about how to use your program.

# LIBRARIAN Description

You create help libraries the same way you create object, macro, and text libraries, using the LIBRARY/CREATE command. However, before you can insert modules into a help library, you must format the input file so that the LIBRARIAN can catalog its individual modules. Sections 5.1 and 5.2 describe how to create input files containing help modules.

---

## 5.1 Creating Help Files

The input files that you insert into help libraries are text files that you build with a program or a text editor. Each input file may contain one or more help modules. A help module is the lines of help text that relate to the same topic, or key.

Each module within a help library contains a group of related keys, or topics, numbered key 1 through key 9. Each key represents a level within the hierarchy of the module. The key-1 name identifies the main topic of help information, for example, the name of a command in your program that requires explanation. The key-2 through key-9 names identify subtopics that are related to the key-1 name, for example, the command's parameters or qualifiers or both. This organization enables users of your program to find general information describing how to use the command, and then optionally to select subtopics that provide additional information about the command's parameters and qualifiers. The maximum length of a key-1 name is determined by the keysize option of the /CREATE qualifier.

Index keywords remain in the format in which they were entered, that is, uppercase and lowercase characters. A second keyword to the same library, identically spelled but of a different or mixed character case, replaces the previous preserved keyword. However, character case is ignored for match operations; for example, *help Sample* and *help SAMPLE* access the same file.

The key names for help topics and subtopics may include any printable ASCII characters except those used by the LIBRARIAN as either delimiters (space, horizontal tab, and comma) or comments (exclamation point).

DIGITAL recommends that you restrict key names to the following characters:

- Uppercase and lowercase letters (A,a,B,b...Z,z)
- Digits (0,1,2...9)
- Dollar sign ( \$ )
- Underscore ( \_ )
- Hyphen ( - )

DIGITAL also recommends that you avoid—especially as the first character of a key name—certain characters that have special meaning to the LIBRARIAN retrieval routines. If you use these characters in key names, you may not be able to specify them explicitly for retrieval.

The characters you should not use are as follows:

- Asterisk ( \* )
- Percent sign ( % )
- Ellipsis ( ... )
- At sign ( @ )

- Slash (/)
- Question mark (?)
- Left parenthesis (()) used as a first character
- Apostrophe (') and quotation marks (")

If a key contains any of these characters, you may be able to retrieve its help text only by abbreviating the key to avoid the special characters, or by using wildcard characters in their place. For information about using wildcard characters, see the *VMS DCL Concepts Manual*.

Also, note that you cannot abbreviate your retrieval key if it contains wildcard characters.

## 5.2 Formatting Help Files

Each line in the module consists of the key number in the first column, followed by the name of the key. The subtopic lines following, key 2 through key 9, consist of the subkey number followed by the name of the subkey. For example, a help module for a command might have the following two key lines:

```
1 Command name
.
.
.
    help text
.
.
.
2 Parameters
```

Each help source file can contain several modules. The LIBRARIAN recognizes a group of key-1 and subkey lines, and their associated text as a module. A module is terminated either by another key-1 line or by an end-of-file record.

A help source file has the following format:

```
1 key-1 name
.
.
.
    help text
.
.
.
2 key-2 name
.
.
.
    help text
.
.
.
n key-n name
.
.
.
1 key-1 name
```

# LIBRARIAN Description

The LIBRARIAN stores the key-1 name in its module name table; therefore, the name of the module is the same as the key-1 name. The subsequent numbers in the first column indicate that the line is a subkey. A module can have several subkeys with the same number. For example, a help module describing a command might have the following key-2 lines:

```
2 parameters
2 arguments
```

You can insert comments anywhere in a module. When the LIBRARIAN encounters an exclamation point as the first character on a line, it assumes that the line consists of comments. Comment lines that follow a key-1 line are included in the module. However, when your program retrieves help text, the LIBRARIAN does not display the comments.

The help text may be any length; the only restriction to the text is that it cannot contain a number or a slash (/) in the first column of any line. A number in the first column of a line indicates that the line is a key. A slash in the first column indicates a qualifier line.

A qualifier line is similar to a key line, except that the LIBRARIAN returns a list of all the qualifier lines when you request help either on a key-1 topic or on the key containing the qualifiers (usually a key 2 named "Qualifiers"). Therefore, if your help module describes a command that has qualifiers, the LIBRARIAN provides a list of all the command's qualifiers whenever you request help on the command.

---

## 5.3 Help Text Example

The help module in Example LIB-1 shows the organization of some of the help text for the DCL LIBRARY command.

# LIBRARIAN Description

## Example LIB-1 Help Text for LIBRARY Command

---

### 1 LIBRARY

Invokes the Librarian Utility to create, modify, or describe an object, macro, help, text, or shareable image library.

Format:

```
LIBRARY library-file-spec [input-file-spec[...]]
```

### 2 Parameters

library-file-spec

Specifies the name of the library you want to create or modify.

If the file specification does not include a file type, the LIBRARY command assumes a default type of OLB, indicating an object library.

input-file-spec[...]

Specifies the names of one or more files that contain modules you want to insert or replace in the specified library.

If you specify more than one input file, separate the file specifications with commas. The input-file-spec parameter is required when you specify /REPLACE, which is the LIBRARY command's default operation, or /INSERT, which is an optional qualifier.

When you use the /CREATE qualifier to create a new library, the input-file-spec parameter is optional. If you include an input file specification with /CREATE, the LIBRARY command first creates a new library, and then inserts the contents of the input files into the library.

### 2 Command\_Qualifiers

/BEFORE

```
/BEFORE[=time]
```

Used in conjunction with the /LIST qualifier to specify that only those modules dated earlier than a particular time be listed. You can specify an absolute time or a combination of absolute and delta times.

If you omit the /BEFORE qualifier, all modules are listed regardless of date. If you specify /BEFORE without a date or time, all modules created before today are listed by default.

/COMPRESS

```
/COMPRESS[=(option[...])]
```

Recovers space that had been occupied by modules deleted from the library. When you specify /COMPRESS, the LIBRARY command by default creates a new version of the library in your current default directory. You can use options to the /COMPRESS qualifier to make some specifications in the new version of the library different from the original library.

/CREATE

```
/CREATE[=(option[...])]
```

Creates a new library. When you specify /CREATE, you can optionally specify a file or a list of files that contains modules to be placed in the library.

---

Example LIB-1 Cont'd. on next page

# LIBRARIAN Description

## Example LIB-1 (Cont.) Help Text for LIBRARY Command

---

By default, the LIBRARY command creates an object module library; specify /SHARE, /MACRO, /HELP, or /TEXT to change the default library type.

---

## 5.4 Retrieving Help Text

---

You can retrieve help text at DCL level by using the DCL command HELP, or in your program by calling the appropriate LIBRARIAN routines described in the *VMS Utility Routines Manual*.

By default, the HELP command retrieves help text from the system help library SYS\$HELP:HELPLIB.HLB, and from user help libraries associated with the logical names HLP\$LIBRARY, HLP\$LIBRARY\_1, HLP\$LIBRARY\_2, and so forth. Using the /LIBRARY qualifier with the HELP command allows you to search another library instead of the default libraries. For more information, see the description of the HELP command in the *VMS DCL Dictionary*.

When you retrieve help text, you specify the key-1 topic, followed by any subtopics that contain appropriate help information. The LIBRARIAN returns the help text associated with the key path you specify. For example, the help text for the LIBRARY command is stored in the default system help library; thus, to retrieve the LIBRARY command's key-1 help information, you would type the DCL command HELP LIBRARY. The LIBRARIAN would return the associated help text, followed by the message "Additional information available:" and a list of all the key-2 names in the module. In this case, the LIBRARIAN also returns a list of all the qualifiers specified in the qualifier lines. Example LIB-2 displays the text returned by the HELP LIBRARY command.

## Example LIB-2 HELP LIBRARY Display

---

### LIBRARY

Invokes the Librarian Utility to create, modify, or describe an object, macro, help, text, or shareable image library.

Format:

LIBRARY library-file-spec [input-file-spec[,...]]

Additional information available:

Parameters	Command_Qualifiers					
/BEFORE	/COMPRESS	/CREATE	/CROSS_REFERENCE	/DATA	/DELETE	
/EXTRACT	/FULL	/GLOBALS	/HELP	/HISTORY	/INSERT	/LIST
/LOG	/MACRO	/NAMES	/OBJECT	/ONLY	/OUTPUT	/REMOVE
/REPLACE	/SELECTIVE_SEARCH		/SHARE	/SINCE	/SQUEEZE	/TEXT
/WIDTH						
Positional_Qualifier						
/MODULE						
Examples						

---

# LIBRARIAN Description

Note that you could not retrieve the key-2 level "Parameters" by typing HELP PARAMETERS. The LIBRARIAN searches for a subkey only after finding the higher-level keys. In other words, if you want to retrieve key-3 text, you have to specify the key-1 and key-2 lines that form a path to the key-3 line.

Note that you can provide information on a qualifier that has more than one form by associating two or more qualifier lines with the same help text. That is, the text associated with the qualifiers follows the two or more qualifier lines. For example:

```
$ HELP LIBRARY/GLOBALS
```

```
LIBRARY
```

```
  /GLOBALS
```

```
  /GLOBALS
```

```
  /NOGLOBALS
```

Controls, for object module libraries, whether the names of global symbols in modules being inserted in the library are included in the global symbol table.

By default, the LIBRARY command places all global symbol names in the global symbol table. Use /NOGLOBALS if you do not want the global symbol names included in the global symbol table.

When the LIBRARIAN successfully searches the key path to the requested key, it displays all the key names in that path, followed by the help text associated with the last specified key. For example:

```
$ HELP LIBRARY/HELP
```

```
LIBRARY
```

```
  /HELP
```

Indicates that the library is a help library. When you specify the /HELP qualifier, the library file type defaults to HLB and the input file type defaults to HLP.

If you try to retrieve help text for a key that is not in the module name table, the LIBRARIAN issues a message. For example:

```
$ HELP FIRE
```

```
Sorry, no documentation on FIRE
```

```
Additional information available:
```

This message is followed by a list of all the module names in the module name table.

If you have correctly specified the key-1 line, but have requested a subkey that does not exist, the LIBRARIAN issues a message. For example:

```
$ HELP LIBRARY/FIRE
```

```
Sorry, no documentation on LIBRARY/FIRE
```

```
Additional information available:
```

```
Parameters Command_Qualifiers  
/BEFORE     /COMPRESS
```

# LIBRARIAN Description

The message includes a list of all the subkeys associated with the last key that was specified correctly .

## 6

---

### LIBRARIAN Routines

The LIBRARIAN provides routines that your programs can call to do the following:

- Initialize a library.
- Open a library.
- Look up a key in a library.
- Insert a new key in a library.
- Return the names of the keys.
- Delete a key and its associated data.
- Read and write records.

Your programs can call the LIBRARIAN routines using the VAX standard calling sequence provided in all languages that produce VAX native-mode instructions. When your program calls the LIBRARIAN routines, it must provide whatever arguments the routine requires. When the routine completes execution, it returns control to your program.

The *VMS Utility Routines Manual* describes, in detail, each of the LIBRARIAN routines.

---

# LIBRARIAN Usage Summary

The Librarian Utility (LIBRARIAN) allows you easy access to libraries. Libraries are files in which you can store frequently used modules of code or text.

The LIBRARIAN consists of two parts: the DIGITAL Command Language (DCL) command LIBRARY, and a collection of LIBRARIAN routines (see Section 6) you can call from a program. You use the LIBRARY command or the LIBRARIAN routines to create a library, maintain the modules in a library, or display information about a library and its modules.

Note that libraries are files, so you can use DCL commands to manipulate libraries in their entirety; for example, the DELETE, COPY, and RENAME commands delete, copy, and rename libraries. For more information about file maintenance, see the *VMS DCL Dictionary*.

---

**FORMAT**            **LIBRARY** *library-file-spec [input-file-spec[,...]]*

---

**PARAMETERS**    ***library-file-specification***

The name of the library you want to create or modify. This parameter is required. If you do not specify a library file, you are prompted for one, as follows:

\_Library:

No wildcard characters are allowed in the library file specification.

If the file specification does not include a file type and if the command string does not indicate one, the LIBRARY command assumes a default type of OLB, indicating an object library. You can change the default library file type by specifying the appropriate qualifier, as follows.

---

Qualifier	Default File Type
/HELP	HLB
/MACRO	MLB
/OBJECT	OLB
/TEXT	TLB
/SHARE	OLB

---

**Note:** If you attempt to modify a library that was created by the VAX-11 Version 1.0 LIBRARIAN, the library is automatically compressed into the new format introduced with Version 2.0. The compression occurs before the requested modification. Furthermore, libraries created before Version 2.0 that have not been modified or compressed will appear in a different format when listed by the /LIST qualifier.

# LIBRARIAN Usage Summary

## ***input-file-spec[,...]***

The names of one or more files that contain modules you want to insert into the specified library. If you specify more than one input file, separate the file specifications with commas.

The input file specification is required when you specify `/REPLACE`, which is the LIBRARY command's default operation, or `/INSERT`, which is an optional qualifier. If you do not specify an input file when you use these qualifiers, you are prompted for it, as follows:

\_File:

When you use the `/CREATE` qualifier to create a new library, the input file specification is optional. If you include an input file specification with the `/CREATE` qualifier, the LIBRARY command first creates a new library, and then inserts the contents of the input files into the library.

Note that the `/EXTRACT` qualifier does not accept an input file specification.

If any file specification does not include a file type and if the command string does not indicate one, the LIBRARY command assumes a default file type of OBJ, designating an object file. You can control the default file type by specifying the appropriate qualifier, as follows.

Qualifier	Default File Type
<code>/HELP</code>	HLP
<code>/MACRO</code>	MAR
<code>/OBJECT</code>	OBJ
<code>/TEXT</code>	TXT
<code>/SHARE</code>	EXE

Note also that the file type you specify with the library file specification determines the default file type of the input file specification, provided that you do not specify the `/CREATE` qualifier. For example, if the library file type is HLB, MLB, OLB, or TLB, the input file type default will be HLP, MAR, OBJ, or TXT, respectively. (If you specify the `/CREATE` qualifier, and you are not creating an object library, you must use the appropriate file type qualifier.)

Wildcard characters are allowed in the input file specifications.

## **usage summary**

The DCL command LIBRARY invokes the Librarian Utility. After the operations specified by the LIBRARY command have completed, the Librarian Utility exits. If you use the `/LIST` qualifier to request information about a library, the output is directed to the file specification associated with `/LIST` or, if you do not supply a file specification, to `SYS$OUTPUT`.

# LIBRARIAN

## LIBRARIAN Qualifiers

### LIBRARIAN QUALIFIERS

When using the LIBRARY command, you can specify qualifiers that request more than one function in a single command, with some restrictions. Generally, you cannot specify multiple qualifiers that request incompatible functions. The qualifiers that perform library functions, related qualifiers, and qualifier incompatibilities are summarized in Table LIB-1.

**Table LIB-1 LIBRARY Command Qualifier Compatibilities**

Qualifier	Related Qualifiers	Incompatible
/COMPRESS	/OUTPUT	/CREATE, /EXTRACT
/CREATE <sup>1</sup>	/SQUEEZE, <sup>2</sup> /GLOBALS <sup>3</sup> /SELECTIVE_SEARCH <sup>3</sup>	/COMPRESS, /EXTRACT
/CROSS_REFERENCE	/ONLY	/EXTRACT, /LIST
/DATA	/COMPRESS	—
/DELETE	—	/EXTRACT
/EXTRACT	/OUTPUT	/COMPRESS, /CREATE /DELETE, /INSERT /LIST, /REMOVE /REPLACE
/INSERT	/SQUEEZE, <sup>2</sup> /GLOBALS <sup>3</sup> /SELECTIVE_SEARCH <sup>3</sup>	/EXTRACT
/LIST	/FULL, /NAMES, <sup>3</sup> /ONLY /HISTORY, /BEFORE /SINCE	/EXTRACT /CROSS_REFERENCE
/REMOVE <sup>3</sup>	—	/EXTRACT
/REPLACE	/SQUEEZE, <sup>2</sup> /GLOBALS <sup>3</sup> /SELECTIVE_SEARCH <sup>3</sup>	/EXTRACT
/MODULE <sup>4</sup>	/TEXT	/EXTRACT, /DELETE /REMOVE

<sup>1</sup>The /CREATE, /INSERT, and /REPLACE qualifiers are compatible; however, if you specify more than one, /CREATE takes precedence over /INSERT, and /INSERT takes precedence over /REPLACE. The related qualifiers for /CREATE are applicable only if you enter one or more input files.

<sup>2</sup>This qualifier applies only to macro libraries.

<sup>3</sup>This qualifier applies only to object libraries.

<sup>4</sup>This positional qualifier applies only to text libraries.

Note that all the qualifiers are command qualifiers except for /MODULE, which is a positional qualifier that modifies the input file specification parameter.

# LIBRARIAN

## /BEFORE

---

## /BEFORE

Specifies that only those modules inserted earlier than a particular time be listed.

---

**FORMAT**            **/BEFORE[=*time*]**

---

**QUALIFIER**        ***time***

**VALUE**

Limits the modules to be listed to those inserted in the library before a specified time.

You can specify an absolute time or a combination of absolute and delta times. For details on specifying times, see the *VMS DCL Dictionary*.

---

**DESCRIPTION**    This qualifier is used in conjunction with the /LIST qualifier. If you omit the /BEFORE qualifier, you obtain all the modules regardless of the dates. If you specify /BEFORE without a date or time, the default is to provide the modules inserted before today.

---

## EXAMPLE

```
$ LIBRARY/LIST/BEFORE=15-APR-:15 MATHLIB
```

This LIBRARY command lists the modules that were inserted into MATHLIB.OLB before 3 p.m. on April 15.

---

## **/COMPRESS**

Recovers space that was occupied by modules deleted from the library. When you specify /COMPRESS, the LIBRARY command creates a new library. You can use options to the /COMPRESS qualifier to make some specifications in the new version of the library different from the original library.

---

**FORMAT**            **/COMPRESS**[=(*option*[,...])]

---

**QUALIFIER**        *option*  
**VALUE**            An option (listed under the Description section) that alters the size or format of the library, overriding the values specified when the library was created.

---

**DESCRIPTION**    When you specify /COMPRESS, the LIBRARY command creates a new library. By default, the new library is created in your current default directory and has the same file name as the existing library, and a file type that is the default for the type of library created. You can use the /OUTPUT qualifier to specify an alternate file specification for the compressed library.

Specify one or more of the following options to alter the size or format of the library, overriding the values specified when the library was created (for the default values, see the description of the /CREATE qualifier):

- BLOCKS:n**        Specifies the number of 512-byte blocks to be allocated for the library. By default, the LIBRARY command allocates 100 blocks for a new library.
- GLOBALS:n**       Specifies the maximum number of global symbols the library can contain initially. By default, the LIBRARY command sets a maximum of 512 global symbols for an object module library. (Macro, help, and text libraries do not have a global symbol directory; therefore, the maximum for these libraries defaults to 0.)
- HISTORY:n**       Specifies the maximum number of library update history records that the library is to maintain. The maximum number of library update records you can specify is 32,767. The default is 20.
- KEEP**            Copies library update history records and any additional user data in the module header to the compressed library.
- KEYSIZE:n**       Specifies the maximum name length of modules or global symbols. The maximum length you can specify for these names is 128 characters. The LIBRARY command assigns default name lengths of 15 characters for help modules, 31 characters for modules in object or macro libraries, and 39 characters for modules in text or shareable image libraries. When you specify a key size value, remember that VAX MACRO and the VMS Linker do not accept module names or global symbol names in excess of 31 characters.

# LIBRARIAN

## /COMPRESS

- MODULES:n** Specifies the maximum number of modules the library can contain. By default, the LIBRARY command sets an initial maximum of 128 modules for all library types.
- A library's size can grow past its initial allocation. However, for optimum performance, it is best to allocate the maximum number of modules you expect to use.
- VERSION:n** Specifies that the library is to be stored in VAX/VMS Version 2.0 library format, if *n* is 2; or VAX/VMS Version 3.0 format, if *n* is 3.

If you specify more than one option, separate them with commas and enclose the list in parentheses.

---

### EXAMPLE

```
$ LIBRARY/COMPRESS=(KEYSIZE:40,MODULES:80)/TEXT SOURCE
```

This LIBRARY command creates a new version of the text library SOURCE.TLB. Space left after modules were deleted from the old version is recovered in the new version. The new version may contain up to 80 modules; the maximum length of module names in the new version is 40.

---

## **/CREATE**

Requests the LIBRARY command to create a new library. When you specify /CREATE, you can optionally specify a file or a list of files that contains modules to be placed in the library.

---

**FORMAT**            **/CREATE**[=(*option*[,...])]

---

**QUALIFIER**        *option*  
**VALUE**            An option (listed under the Description section) that overrides the system defaults to control the size of the library.

---

**DESCRIPTION**    By default, the /CREATE qualifier creates an object module library. To indicate that the library is a macro, help, text, or shareable image library, specify /MACRO, /HELP, /TEXT, or /SHARE.

Specify one or more of the following options to control the size of the library, overriding the system defaults:

**BLOCKS:n**        Specifies the number of 512-byte blocks to be allocated for the library. By default, the LIBRARY command allocates 100 blocks for a new library.

**GLOBALS:n**       Specifies the maximum number of global symbols the library can contain initially. By default, the LIBRARY command sets a maximum of 512 global symbols for an object module library. (Macro, help, and text libraries do not have a global symbol directory; therefore, the maximum for these libraries defaults to 0.)

**HISTORY:n**       Specifies the maximum number of library update history records that the library is to maintain. The maximum number you can specify is 32,767. The default is 20.

**KEYSIZE:n**       Specifies the maximum name length of modules or global symbols. The maximum length you can specify for these names is 128 characters. The LIBRARY command assigns default name lengths of 15 characters for help modules, 31 characters for modules in object or macro libraries, and 39 characters for modules in text or shareable image libraries. When you specify a key size value, remember that VAX MACRO and the VMS Linker do not accept module names or global symbol names in excess of 31 characters.

# LIBRARIAN

## /CREATE

- MODULES:n** Specifies the maximum number of modules the library can contain. By default, the LIBRARY command sets an initial maximum of 128 modules for all library types.
- A library's size can grow past its initial allocation. However, for optimum performance, it is best to allocate the maximum number of modules you expect to use.
- VERSION:n** Specifies that the library is to be stored in VAX/VMS Version 2.0 library format, if *n* is 2; or VAX/VMS Version 3.0 library format, if *n* is 3.

If you specify more than one option, separate them with commas and enclose the list in parentheses.

---

## EXAMPLES

**1** \$ LIBRARY/CREATE TESTLIB ERRMSG, STARTUP

This LIBRARY command creates an object module library named TESTLIB.OLB and places the files ERRMSG.OBJ and STARTUP.OBJ as modules in the library.

**2** \$ LIBRARY/MACRO/CREATE=(BLOCKS:40,MODULES:100) MYMAC TEMP  
\$ MACRO MYMAC/LIBRARY,CYGNUS/OBJECT

This LIBRARY command creates a macro library named MYMAC.MLB from the macros in the file TEMP.MAR. The new library has room for 100 modules in a 40-block file. If the input file contains multiple macros, each macro is entered in the new library.

---

## /CROSS\_REFERENCE

Requests a cross-reference listing of an object library.

---

**FORMAT**            **/CROSS\_REFERENCE**[(*option*[,...])] ]

---

**QUALIFIER VALUE**        *option*  
An option (listed under the Description section) that produces a cross-reference listing that is not limited to only symbols by name and symbols by value.

---

**DESCRIPTION**        If you omit this qualifier, cross-reference listings will not be provided. If you specify /CROSS\_REFERENCE without specifying an option, you will obtain cross-reference listings that contain symbols by name and symbols by value. By default, the listing file is created in your current default directory and has the same file name as the library and a file type of LIS. You can use the /OUTPUT qualifier to specify an alternate file specification for the listing file.

You can specify one or more of the following options:

ALL	All types of cross-references
MODULE	Cross-reference listing of both the global symbol references in the module and the global symbol definitions
NONE	No cross-reference listing
SYMBOL	Cross-reference listing by symbol name
VALUE	Cross-reference listing of symbols by value

If you specify more than one option, separate the options with commas and enclose the list in parentheses.

---

### EXAMPLE

```
$ LIBRARY/CROSS_REFERENCE=ALL/OUTPUT=SYS$OUTPUT LIBRAR
```

This LIBRARY command requests a cross-reference listing of the object library LIBRAR.OLB. The cross-reference listing is displayed at the terminal. The listing includes cross-references by symbol, by value, and by module.

# LIBRARIAN

## /DATA

---

## /DATA

Stores a library in data-reduced format, or expands a library previously stored in data-reduced format.

---

**FORMAT**            */DATA=option*

---

**QUALIFIER**        *option*  
**VALUE**            The option REDUCE, which stores a library in data-reduced format, and the option EXPAND, which expands a library previously stored in data-reduced format. There is no default; you must specify one of the options.

---

**DESCRIPTION**    When you specify /DATA, the LIBRARY command creates a new library. By default, the new library is created in your current default directory and has the same file name as the existing library, and a file type that is the default for the type of library created. You can use the /OUTPUT qualifier to specify an alternate file specification for the library.

You use the /DATA qualifier to control how data is stored in the library. If you specify /DATA=REDUCE, data in the library is stored in data-reduced format; less disk space is required for the library, but access to the library generally is slower.

If you omit this qualifier, data is stored in the library in standard, rather than data-reduced, format. You can change a data-reduced library back to the standard format by specifying /DATA=EXPAND.

When you use the /DATA qualifier (with either option), the LIBRARIAN also recovers space in the library that had been occupied by modules deleted from the library, just as if you specified /COMPRESS.

---

## EXAMPLE

```
$ LIBRARY/TEXT/DATA=REDUCE TEXTLIB
```

This LIBRARY command stores the data in the text library TEXTLIB.TLB in data-reduced format.

---

## **/DELETE**

Requests the LIBRARY command to delete (physically remove) one or more modules from a library.

---

**FORMAT**            **/DELETE=(*module*[,...])**

---

**QUALIFIER**        *module*  
**VALUE**            . The name of the module to be deleted.

---

**DESCRIPTION**    You must specify the names of the modules to be deleted. If you specify more than one module, separate the module names with commas and enclose the list in parentheses.

Wildcard characters are allowed in the module specification.

If you specify the /LOG qualifier with /DELETE, the LIBRARY command issues the following message:

```
%LIBRAR-S-DELETED, MODULE module-name DELETED FROM library-name
```

---

## **EXAMPLE**

```
$ LIBRARY/DELETE=FREEZE/LOG THAW
```

This LIBRARY command physically removes the module FREEZE from the object library THAW. A message is displayed to confirm that the module was deleted.

# LIBRARIAN

## /EXTRACT

---

## /EXTRACT

Copies one or more modules from a library into a file.

---

**FORMAT**            **/EXTRACT=(*module*[,...])**

---

**QUALIFIER**        ***module***  
**VALUE**            The name of the module to be copied.

---

**DESCRIPTION**    If you specify more than one module, separate the module names with commas and enclose the list in parentheses.

Wildcard characters are allowed in the module specification.

If you specify the /OUTPUT qualifier with /EXTRACT, the LIBRARY command writes the output into the file specified by the /OUTPUT qualifier. If you do not specify a directory, the file is written to your current default directory. If you specify /EXTRACT and do not specify /OUTPUT, the LIBRARY command writes the file into a file that has the same file name as the library and a file type of OBJ, EXE, MAR, HLP, or TXT, depending on the type of library.

---

## EXAMPLE

```
$ LIBRARY/EXTRACT=(ALLOCATE, APPEND)/OUTPUT=MYHELP SYS$HELP:HELPLIB.HLB
```

This LIBRARY command specifies that the modules ALLOCATE and APPEND are to be extracted from the help library HELPLIB.HLB and output to the file MYHELP.HLP in your current default directory.

---

**/FULL**

Requests a full description of each module in the module name table.

---

**FORMAT**            **/FULL**

---

**DESCRIPTION**

Use the /FULL qualifier with the /LIST qualifier to request a list of each library module in this format:

module-name [Ident nn]    Inserted dd-mmm-yyyy    hh:mm:ss    [n symbols]

where:

Ident nn            is the identification number of the module

The identification number and the number of symbols appear only in object libraries.

If an update history is maintained for the library, then /LIST/FULL /HISTORY lists the module name in the update history records.

---

**EXAMPLE**

\$ LIBRARY/LIST=MYMAC.LIS/FULL MYMAC.MLB

This LIBRARY command requests a full listing of the macro library MYMAC; the output is written to a file named MYMAC.LIS.

# LIBRARIAN

## /GLOBALS

---

## /GLOBALS

Controls, for object module libraries, whether the names of global symbols in modules being inserted in the library are included in the global symbol table.

---

**FORMAT**            **/GLOBALS**  
                      **/NOGLOBALS**

---

**DESCRIPTION**    By default, the LIBRARY command places all global symbol names in the global symbol table. Use /NOGLOBALS if you do not want the global symbol names included in the global symbol table.

---

### EXAMPLE

```
$ LIBRARY/INSERT/NOGLOBALS TOOLS SPELL
```

This LIBRARY command inserts the modules in SPELL.OBJ into the object library TOOLS, but global symbol names in the inserted modules are not included in the library's global symbol table.

---

**/HELP**

Indicates that the library specified is a help library.

---

**FORMAT**            **/HELP**

---

**DESCRIPTION**    When you use the /HELP qualifier, the library file type defaults to HLB and the input file type defaults to HLP.

---

**EXAMPLE**

`$ LIBRARY/HELP/CREATE ERRMSG EDITERRS`

This LIBRARY command creates a help library called ERRMSG.HLB. Help text from the file EDITERRS.HLP is inserted into the library.

# LIBRARIAN

## /HISTORY

---

## /HISTORY

Requests that update history record headers be listed (for libraries that contain a history) in the following format:

username operation n modules on dd-mmm-yyy hh:mm:ss

The operation referred to in the header has one of three values: replaced, inserted, or deleted.

---

## DESCRIPTION

The /HISTORY qualifier is used in conjunction with the /LIST qualifier. Use the /HISTORY qualifier with /LIST/FULL to request that the names of updated modules be listed in addition to the update history headers.

---

## EXAMPLE

```
$ LIBRARY/LIST/HISTORY/MACRO SETUP
```

This LIBRARY command lists the headers of the update history records in the macro library SETUP.MLB.

---

## **/INSERT**

Requests the LIBRARY command to add the contents of one or more files to an existing library.

---

### **FORMAT            /INSERT**

---

#### **DESCRIPTION**

If an object module input file consists of concatenated object modules, the LIBRARY command creates a separate entry for each object module in the file; each module name table entry reflects an individual module name. If a macro or help file specified as input contains more than one definition, the LIBRARY command creates a separate entry for each one, naming the module name table entries according to the names specified in the .MACRO directives or in the key-1 name in the HELP format (see Section 5.1).

Unlike object, macro, and help libraries, the input file in text libraries contains data records of undefined contents. Therefore, the LIBRARIAN catalogs the entire input file as a single module using the file name (not the directory or file type) as the module name. If you want to rename the inserted module, use the /MODULE qualifier.

When you use the /INSERT qualifier to insert modules into an existing library, the Librarian Utility checks the module name table before inserting each module. If a module name or global symbol name already exists in the library, an error message is issued and the module or symbol is not added to the library.

To insert or replace a module in a library, regardless of whether a current entry exists with the same name, use the /REPLACE qualifier (the default operation).

---

#### **EXAMPLE**

```
$ LIBRARY/INSERT TESTLIB SCANLINE  
$ LINK TERMTEST,TESTLIB/LIBRARY
```

This LIBRARY command adds the module SCANLINE.OBJ to the library TESTLIB.OLB. The library is specified as input to the linker by the /LIBRARY qualifier on the LINK command. If the module TERMTEST.OBJ refers to any routines or global symbols not defined in TERMTEST, the linker searches the global symbol table of library TESTLIB.OLB to resolve the symbols.

# LIBRARIAN

## /LIST

---

## /LIST

Controls whether the LIBRARY command creates a listing that provides information about the contents of the library.

---

**FORMAT**            */LIST[=file-spec]*  
                      */NOLIST*

---

**QUALIFIER**        *file-spec*  
**VALUE**            The file specification of the file to be listed.

---

**DESCRIPTION**    By default, the LIBRARY command does not produce a listing. If you specify /LIST without a file specification, the LIBRARY command writes the output file to the current SYS\$OUTPUT device. If you include a file specification that does not have a file type, the LIBRARY command uses the default file type LIS.

If you specify /LIST with qualifiers that perform additional operations on the library, the LIBRARY command creates the listing after completing all other requests; thus, the listing reflects the status of the library after all changes have been made.

When you specify /LIST, the LIBRARY command provides, by default, the following information about the library:

```
Directory of OBJECT library _DBB0: [LIBRAR]LIBRAR.OLB;1 on 31-DEC-1988 10:08:28
Creation date: 12-DEC-1988 19:40:36        Creator: VAX-11 librarian V03-00
Revision date: 31-DEC-1988 16:04:58        Library format: 3.0
Number of modules:        15            Max. key length: 31
Other entries:            73            Preallocated index blocks: 35
Recoverable deleted blocks:    15            Total index blocks used: 12
Max. update history records: 10            Update history records: 5
```

---

### EXAMPLES

**1** \$ LIBRARY/LIST=MYMAC.LIS/FULL MYMAC.MLB

This LIBRARY command requests a full listing of the macro library MYMAC; the output is written to a file named MYMAC.LIS.

**2** \$ LIBRARY/LIST/NAMES/ONLY=\$ONE/WIDTH=80 SYMBOLIB

This LIBRARY command requests a full listing of the module \$ONE, contained in the object library SYMBOLIB.OLB. The /WIDTH qualifier requests that the global-symbol display be limited to 80 characters per line.

**3** \$ LIBRARY/INSERT/LIST ALLOBJECTS \*

This LIBRARY command inserts into ALLOBJECTS.OLB all object modules from all object files in the current directory. If any of the modules to be inserted have the same name as an existing module in the library, the existing module is replaced. The LIBRARY command then lists the resulting library on SYS\$OUTPUT.

# LIBRARIAN

**/LOG**

---

**/LOG**

Controls whether the LIBRARY command verifies each library operation.

---

**FORMAT**

**/LOG**  
**/NOLOG**

---

**DESCRIPTION**

If you specify /LOG, the LIBRARY command displays the module name, followed by the library operation performed, followed by the library file specification.

Other applications of the /LOG qualifier appear in the descriptions of /DELETE and /REPLACE.

---

**EXAMPLE**

```
$ LIBRARY/REMOVE=(LIB_EXTRCT_MODS,LIB_INPUT_MAC)/LOG LIBRAR
```

This LIBRARY command requests the removal of the global symbols LIB\_EXTRCT\_MODS and LIB\_INPUT\_MAC from the object library LIBRAR.OLB. The /LOG qualifier requests that the removal of the symbols be confirmed by messages.

---

**/MACRO**

Indicates that the library specified is a macro library.

---

**FORMAT**            **/MACRO**

---

**DESCRIPTION**    When you specify the /MACRO qualifier, the library file type defaults to MLB and the input file type defaults to MAR.

---

**EXAMPLE**

\$ LIBRARY/MACRO/INSERT MONTHS APRIL

This LIBRARY command inserts modules from APRIL.MAR into the macro library MONTHS.MLB.

# LIBRARIAN

## /MODULE

---

## /MODULE

Names a text module that you want to replace or insert into a text library. It also modifies the input file specification parameter.

---

**FORMAT**            */MODULE=module-name*

---

**QUALIFIER**        *module-name*  
**VALUE**            The name of the module to be inserted in the library.

---

**DESCRIPTION**    When you insert text modules into a library, the input file you specify is assumed to be a single module. Therefore, the file name of the input file specification becomes the module name. If you want the file you are inserting to have a module name different from its input file name, use the */MODULE* qualifier to name the added module.

You can also use the */MODULE* qualifier to enter a text module interactively. If you specify the logical name *SY\$INPUT* as the input file, and use the */MODULE* qualifier, the *LIBRARY* command inserts the text you enter from the terminal into the specified library module. To terminate the terminal input, press *CTRL/Z*.

Remember that the */MODULE* qualifier is an input file qualifier; it assumes that you are either replacing or inserting a new text module. Therefore, the qualifiers that remove modules—*/EXTRACT*, */DELETE*, */REMOVE*—are incompatible with */MODULE*.

Note that you must place the */MODULE* qualifier after the input file you specify.

---

## EXAMPLE

```
$ LIBRARY/INSERT/TEXT TSTRING SY$INPUT/MODULE=TEXT1
```

This *LIBRARY* command inserts a module named *TEXT1* into the text library *TSTRING.TLB*. The input is taken from *SY\$INPUT*.

---

## **/NAMES**

When /LIST is specified for an object module library, controls whether the LIBRARY command lists the names of all global symbols in the global symbol table as well as the module names in the module name table.

---

**FORMAT**            **/NAMES**  
                      **/NONAMES**

---

**DESCRIPTION**    The default is /NONAMES, which does not list the global symbol names. If you specify /NAMES, each module entry name is displayed in the following format:

```
module "module-name"  
global-symbol  global-symbol  global-symbol  global-symbol  
  .              .              .              .  
  .              .              .              .  
  .              .              .              .
```

If you specify /NAMES and the library is a macro, help, or text library, no symbol names are displayed.

---

## **EXAMPLE**

```
$ LIBRARY/LIST/NAMES/ONLY=$ONE/WIDTH=80 SYMBOLIB
```

This LIBRARY command requests a full listing of the module \$ONE, contained in the object library SYMBOLIB.OLB. The /WIDTH qualifier requests that the global symbol display be limited to 80 characters per line.

# LIBRARIAN

**/OBJECT**

---

## **/OBJECT**

Indicates that the library specified is an object module library.

---

**FORMAT**

**/OBJECT**

---

### **DESCRIPTION**

Libraries are assumed to be object module libraries unless you specify the `/SHARE`, `/MACRO`, `/TEXT`, or `/HELP` qualifier. The library file type for object module libraries defaults to `OLB` and the input file type defaults to `OBJ`.

---

### **EXAMPLE**

`$ LIBRARY/OBJECT/INSERT MONTHS APRIL`

This `LIBRARY` command inserts modules from `APRIL.OBJ` into the object library `MONTHS.OLB`. The `/OBJECT` qualifier is optional.

---

## **/ONLY**

Specifies the individual modules on which the LIBRARY command can operate.

---

**FORMAT**            */ONLY=(module-name[,...])*

---

**QUALIFIER**        *module-name*  
**VALUE**             The module on which the LIBRARY command can operate.

---

**DESCRIPTION**    When you use the */ONLY* qualifier, the LIBRARY command lists or cross-references only those modules specified.

If you specify more than one module, separate the module names with commas and enclose the list in parentheses.

The */ONLY* qualifier must be used with the */LIST* or */CROSS\_REFERENCE* qualifier.

Wildcard characters are allowed in the module name specification.

---

## **EXAMPLE**

`$ LIBRARY/LIST/NAMES/ONLY=$ONE/WIDTH=80 SYMBOLIB`

This LIBRARY command requests a full listing of the module \$ONE, contained in the object library SYMBOLIB.OLB. The */WIDTH* qualifier requests that the global symbol display be limited to 80 characters per line.

# LIBRARIAN

## /OUTPUT

---

## /OUTPUT

When used with the /EXTRACT, /COMPRESS, /CROSS\_REFERENCE, or /DATA qualifier, specifies the file specification of the output file.

---

**FORMAT**            */OUTPUT=file-spec*

---

**QUALIFIER VALUE**        *file-spec*  
The file specification of the output file.

---

**DESCRIPTION**        For /EXTRACT, the output file contains the modules extracted from a library; for /COMPRESS, the output file contains the compressed library; for /CROSS\_REFERENCE, the output file contains the cross-reference listing; for /DATA, the output file contains the data-reduced or data-expanded library.

No wildcard characters are allowed in the file specification.

If you omit the file type in the file specification, a default is used depending on the library function qualifier and, in some cases, the library type qualifier, as follows.

---

Qualifier	Library Type Qualifier	Default File Type
/COMPRESS or /DATA	/HELP	HLB
	/MACRO	MLB
	/OBJECT	OLB
	/TEXT	TLB
	/SHARE	OLB
/CROSS_REFERENCE	—	LIS
/EXTRACT	/HELP	HLP
	/MACRO	MAR
	/OBJECT	OBJ
	/TEXT	TXT
	/SHARE	EXE

---

---

## EXAMPLES

**1** \$ LIBRARY/EXTRACT=(ALLOCATE,APPEND)/OUTPUT=MYHELP SYS\$HELP:HELPLIB.HLB

This LIBRARY command specifies that the modules ALLOCATE and APPEND be extracted from the help library HELPLIB.HLB and output to the file MYHELP.HLP.

**2** \$ LIBRARY/CROSS\_REFERENCE=ALL/OUTPUT=SYS\$OUTPUT LIBRAR

This LIBRARY command requests a cross-reference listing of the object library LIBRAR.OLB. The cross-reference listing is displayed at the terminal. The listing includes cross-references by symbol, by value, and by module.

# LIBRARIAN

## /REMOVE

---

## /REMOVE

Requests the LIBRARY command to delete one or more entries from the global symbol table in an object library.

---

**FORMAT**            **/REMOVE=(symbol[,...])**

---

**QUALIFIER**        **symbol**  
**VALUE**            The symbol to be deleted from the global symbol table.

---

**DESCRIPTION**    If you specify more than one symbol, separate the symbols with commas and enclose the list in parentheses.

Wildcard characters are allowed in the symbol specification.

To display the names of the deleted global symbols, you must also specify the /LOG qualifier.

---

## EXAMPLE

```
$ LIBRARY/REMOVE=(LIB_EXTRCT_MODS,LIB_INPUT_MAC)/LOG LIBRAR
```

This LIBRARY command requests the removal of the global symbols LIB\_EXTRCT\_MODS and LIB\_INPUT\_MAC from the object library LIBRAR.OLB. The /LOG qualifier requests that the removal of the symbols be confirmed by messages.

---

## **/REPLACE**

Requests the LIBRARY command to replace one or more existing library modules with the modules specified in the input files.

---

### **FORMAT            /REPLACE**

---

### **DESCRIPTION**

The LIBRARY command first deletes any existing library modules with the same names as the modules in the input files. Then the new version of the module is inserted in the library. If a module contained in an input file does not have a corresponding module in the library, the LIBRARY command inserts the new module in the library.

The /REPLACE qualifier is the LIBRARY command's default operation. If you specify an input file parameter, the LIBRARY command either replaces or inserts the contents of the input file into the library. If you use the /LOG qualifier with the /REPLACE qualifier, the LIBRARY command displays, in the following format, the name of each module that it replaces or inserts:

%LIBRAR-S-REPLACED, MODULE module-name REPLACED IN library-file-spec

%LIBRAR-S-INSERTED, MODULE module-name INSERTED IN library-file-spec

---

### **EXAMPLE**

\$ LIBRARY/REPLACE/HELP HELPLIB NEWTEXT

This LIBRARY command inserts into the help library HELPLIB.HLB the help modules from the file NEWTEXT.HLP. If a help module in NEWTEXT.HLP has the same name as an existing help module in the library, the module from NEWTEXT.HLP replaces the existing module.

# LIBRARIAN

## /SELECTIVE\_SEARCH

---

## /SELECTIVE\_SEARCH

Defines the input modules being inserted into a library as candidates for selective searches by the linker.

---

### FORMAT            /SELECTIVE\_SEARCH

---

**DESCRIPTION**    If you specify /SELECTIVE\_SEARCH and the library is specified as a linker input file, the linker selectively searches the modules; the linker takes from the library, for the symbol table of its output image file, only those global symbols that have been referenced by other modules.

Note that the selective search operation applies only to those modules that were inserted in the library by a LIBRARY command that included the /SELECTIVE\_SEARCH qualifier; it does not apply to the library itself.

---

### EXAMPLE

```
$ LIBRARY/SELECTIVE_SEARCH/INSERT MATHLIB TRIG
```

This LIBRARY command inserts the modules in TRIG.OBJ into the library MATHLIB.OLB. The inserted modules are selectively searched when MATHLIB.OLB is specified as an input file to the VMS Linker.

---

**/SHARE**

Indicates that the library specified is a shareable image library.

---

**FORMAT**            **/SHARE**

---

**DESCRIPTION**    This LIBRARY command assumes a file type of OLB for the shareable image symbol table library and EXE for the input files. See Section 4 for additional information.

---

**EXAMPLE**

\$ LIBRARY/SHARE/CREATE SHARLIB

This LIBRARY command creates a shareable image library called SHARLIB.OLB.

# LIBRARIAN

## /SINCE

---

## /SINCE

Specifies that only those modules inserted later than a particular time be listed.

---

**FORMAT**            */SINCE[=*time*]*

---

**QUALIFIER**        *time*  
**VALUE**            Limits the modules to be listed to those inserted in the library since a specified time.

You can specify an absolute time or a combination of absolute and delta times. For details on specifying times, see the *VMS DCL Dictionary*.

---

**DESCRIPTION**    This qualifier is used in conjunction with the */LIST* qualifier. If you omit the */SINCE* qualifier, you obtain all the modules regardless of the date. If you specify */SINCE* without a date or time, the default provides the modules inserted since today.

---

## EXAMPLE

```
$ LIBRARY/HELP/LIST/SINCE=:12 ERRMSG
```

This LIBRARY command displays information about help modules added to ERRMSG.HLB since noon today.

---

## **/SQUEEZE**

Controls whether the LIBRARY command compresses individual macros before adding them to a macro library.

---

**FORMAT**            **/SQUEEZE**  
                      **/NOSQUEEZE**

---

**DESCRIPTION**    When you specify **/SQUEEZE**, which is the default, trailing blanks, trailing tabs, and comments are deleted from each macro before its insertion in the library.

Use **/SQUEEZE** only with the **/CREATE**, **/INSERT**, and **/REPLACE** qualifiers to conserve space in a macro library. If you want to retain the full macro, specify **/NOSQUEEZE**.

---

## **EXAMPLE**

**\$ LIBRARY/MACRO/NOSQUEEZE/INSERT MYMACS MYMACS**

This LIBRARY command inserts the macros in MYMACS.MAR into the library MYMACS.MLB. Trailing blanks, trailing tabs, and comments are not deleted from each macro before its insertion into the library.

# LIBRARIAN

## /TEXT

---

## /TEXT

Indicates that the library specified is a text library.

---

## FORMAT        /TEXT

---

**DESCRIPTION**    When you use the /TEXT qualifier, the library file type defaults to TLB and the input file type defaults to TXT.

---

## EXAMPLES

**1**    \$ LIBRARY/INSERT/TEXT TSTRING SYS\$INPUT/MODULE=TEXT1

This LIBRARY command inserts a module named TEXT1 into the text library TSTRING.TLB. The input is taken from SYS\$INPUT.

**2**    \$ LIBRARY/INSERT/TEXT TSTRING TEXT2

This LIBRARY command inserts the contents of the file TEXT2.TXT into the text library TSTRING.TLB. The name of the inserted module is TEXT2.

---

## **/WIDTH**

Controls the screen display width (in characters) for listing global symbol names.

---

**FORMAT**            **/WIDTH=*n***

---

**QUALIFIER**        ***n***  
**VALUE**            The width of the screen display.

---

**DESCRIPTION**    Specify the /WIDTH qualifier with the /NAMES qualifier to limit the line length of the /NAMES display.  
  
The default display width is the width of the listing device. The maximum width is 132.

---

## **EXAMPLE**

**\$ LIBRARY/LIST/NAMES/ONLY=\$ONE/WIDTH=80 SYMBOLIB**

This LIBRARY command requests a full listing of the module \$ONE, contained in the object library SYMBOLIB.OLB. The /WIDTH qualifier requests that the global symbol display be limited to 80 characters per line.



---

# Index

---

---

## B

---

/BEFORE qualifier • LIB-14

---

---

## C

---

Character case • LIB-2

Comment lines

    in help files • LIB-6

/COMPRESS qualifier • LIB-15

    See also /DATA qualifier

    See also /SQUEEZE qualifier  
    using with /OUTPUT • LIB-36

/CREATE qualifier • LIB-12, LIB-17

/CROSS\_REFERENCE qualifier • LIB-19

    using with /ONLY • LIB-35

    using with /OUTPUT • LIB-36

---

---

## D

---

/DATA qualifier • LIB-20

    See also /COMPRESS qualifier  
    using with /OUTPUT • LIB-36

Default file type • LIB-1, LIB-11

Default library file type • LIB-11

/DELETE qualifier • LIB-21

---

---

## E

---

/EXTRACT qualifier • LIB-12, LIB-22

    using with /OUTPUT • LIB-36

---

---

## F

---

File

    insertion of • LIB-27

Format

    of LIBRARY command • LIB-11

---

/FULL qualifier • LIB-23

    using with /HISTORY • LIB-26

---

---

## G

---

/GLOBALS qualifier • LIB-24

Global symbol table (GST) • LIB-2

---

---

## H

---

Help files

    comment lines in • LIB-6

    creating • LIB-4 to LIB-5

    formatting • LIB-5

    qualifier lines in • LIB-6

    restrictions in • LIB-4

Help library • LIB-1, LIB-4

    character case in • LIB-2

    index keywords in • LIB-4

    key names in • LIB-4 to LIB-5

HELP LIBRARY command display • LIB-8 to LIB-10

/HELP qualifier • LIB-25

Help text

    example of • LIB-6 to LIB-8

    retrieving • LIB-8 to LIB-10

/HISTORY qualifier • LIB-26

---

---

## I

---

Index keywords

    in help libraries • LIB-4

Input file specification • LIB-11

    default file type • LIB-12

Insertion of files/modules • LIB-27

    See also /REPLACE qualifier

/INSERT qualifier • LIB-12, LIB-27

---

---

## K

---

Key • LIB-2, LIB-4, LIB-5

---

## Index

### Key (cont'd.)

See also Library key

### Key line

formatting • LIB-5

### Key name

character restrictions in • LIB-4

in help libraries • LIB-4 to LIB-5, LIB-9

### Key number • LIB-5

See also Module

### Keyword • LIB-4

---

## L

---

### LIBRARIAN routines • LIB-10

#### Librarian Utility (LIBRARIAN)

command qualifiers • LIB-13 to LIB-45

DCL qualifiers • LIB-14 to LIB-45

directing output • LIB-12

See also /LIST qualifier

See also /OUTPUT qualifier

exiting from • LIB-12

help files • LIB-4 to LIB-5

help libraries • LIB-4 to LIB-5

HELP LIBRARY command display •  
LIB-8 to LIB-10

help text example • LIB-6 to LIB-8

invoking • LIB-12

key lines in help files • LIB-5 to LIB-6

retrieval of help text • LIB-8 to LIB-10

### Libraries

reformatting • LIB-15, LIB-20

types of • LIB-1

### LIBRARY command • LIB-11

creating a new library using

/CREATE • LIB-17

cross-referencing

/CROSS\_REFERENCE qualifier • LIB-19

directing output • LIB-12

exiting • LIB-12

format of • LIB-11

input file specification • LIB-11

default file type • LIB-12

invoking • LIB-12

library file specification • LIB-11

default file type • LIB-11

qualifiers for • LIB-13 to LIB-45

restrictions on • LIB-12

specifying time in • LIB-14

### Library file specification • LIB-11

### Library file specification (cont'd.)

default file type • LIB-11

### Library header • LIB-2

### Library index • LIB-2

### Library key • LIB-2

### Library size

See /COMPRESS qualifier

See /CREATE qualifier

### Linker • LIB-1, LIB-3

### /LIST qualifier • LIB-12, LIB-28

using with /BEFORE • LIB-14

using with /FULL • LIB-23

using with /HISTORY • LIB-26

using with /NAMES • LIB-33

using with /ONLY • LIB-35

using with /SINCE • LIB-42

### /LOG qualifier • LIB-30

See also /DELETE

See also /REPLACE

---

## M

---

### Macro library • LIB-1

character case in • LIB-2

### /MACRO qualifier • LIB-31

### Match operations • LIB-2

### Module

creating • LIB-4

formatting • LIB-5

key number in • LIB-5

terminating • LIB-5

### Module header • LIB-2

### Module name table (MNT) • LIB-2

### /MODULE qualifier • LIB-32

using with /INSERT • LIB-32

---

## N

---

### /NAMES qualifier • LIB-33

### Naming help modules • LIB-4

---

## O

---

### Object library • LIB-1

character case in • LIB-2

/OBJECT qualifier • LIB-34  
 /ONLY qualifier • LIB-35  
 Output  
   directing • LIB-15  
 /OUTPUT qualifier • LIB-36  
   using with /COMPRESS • LIB-15  
   using with /CROSS\_REFERENCE • LIB-19  
   using with /EXTRACT • LIB-22

---

## P

---

Parameters • LIB-11

---

## Q

---

Qualifier lines  
   help files • LIB-6  
 Qualifiers • LIB-13 to LIB-45

---

## R

---

Reformatting libraries  
   with /COMPRESS qualifier • LIB-15  
   with /DATA qualifier • LIB-20  
 /REMOVE qualifier • LIB-38  
 /REPLACE qualifier • LIB-12, LIB-39  
 Restrictions • LIB-11  
   in help file keys • LIB-4  
 Routines • LIB-10

---

## S

---

/SELECTIVE\_SEARCH qualifier • LIB-40  
 Shareable image library • LIB-1, LIB-3  
   shareable image in • LIB-3  
 /SHARE qualifier • LIB-41  
 /SINCE qualifier • LIB-42  
 Special characters • LIB-5  
 /SQUEEZE qualifier • LIB-43  
 Subkeys • LIB-5, LIB-6  
 System help library • LIB-8

---

## T

---

Text library • LIB-1  
   character case in • LIB-2  
 /TEXT qualifier • LIB-44  
 Types of libraries • LIB-1

---

## W

---

/WIDTH qualifier • LIB-45  
 Wildcard characters • LIB-5



# Reader's Comments

VMS Librarian Utility  
Manual  
AA-LA61A-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

**I rate this manual's:**

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like best about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like least about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I am using **Version** \_\_\_\_\_ of the software this manual describes.

Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
\_\_\_\_\_ Phone \_\_\_\_\_

-- Do Not Tear - Fold Here and Tape

**digital**<sup>TM</sup>



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35 110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



-- Do Not Tear - Fold Here