

VAX/VMS Librarian Reference Manual

Order Number: AA-Z419A-TE

September 1984

This manual explains how to create and maintain VAX/VMS libraries.

Revision/Update Information:

This is a new manual.

Software Version:

VAX/VMS Version 4.0

**digital equipment corporation
maynard, massachusetts**

September 1984

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 ©1984 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

ZK-2305

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by T_EX, the typesetting system developed by Donald E. Knuth at Stanford University. T_EX is a registered trademark of the American Mathematical Society.

Librarian Contents

PREFACE	vii
----------------	------------

NEW AND CHANGED FEATURES	ix
---------------------------------	-----------

FORMAT	LIB-1
---------------	--------------

DESCRIPTION	LIB-3
--------------------	--------------

1	TYPES OF LIBRARIES	LIB-3
2	STRUCTURE OF LIBRARIES	LIB-4
3	CHARACTER CASE OF LIBRARY KEYS	LIB-5
4	SHAREABLE IMAGE LIBRARIES	LIB-5
5	HELP LIBRARIES	LIB-6
5.1	Creating Help Files	LIB-6
5.2	Formatting Help Files	LIB-7
5.3	Help Text Example	LIB-8
5.4	Retrieving Help Text	LIB-10
6	LIBRARIAN ROUTINES	LIB-12

COMMAND QUALIFIERS	LIB-13
---------------------------	---------------

/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

Librarian Contents

/HELP	LIB-25
/HISTORY	LIB-26
/INSERT	LIB-27
/LIST	LIB-28
/LOG	LIB-29
/MACRO	LIB-30
/MODULE	LIB-31
/NAMES	LIB-32
/OBJECT	LIB-33
/ONLY	LIB-34
/OUTPUT	LIB-35
/REMOVE	LIB-37
/REPLACE	LIB-38
/SELECTIVE_SEARCH	LIB-39
/SHARE	LIB-40
/SINCE	LIB-41
/SQUEEZE	LIB-42
/TEXT	LIB-43
/WIDTH	LIB-44

INDEX

EXAMPLES

LIB-1	Help Text for LIBRARY Command	LIB-9
LIB-2	HELP LIBRARY Display	LIB-11

TABLES

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

Preface

Intended Audience

This document is intended for use by all VMS users.

Structure of This Document

This document is composed of three major sections.

The Format Section is an overview of the Librarian Utility and is intended as a quick reference guide. The format summary contains the DCL command that invokes the Librarian Utility, and includes all command qualifiers, positional qualifiers and parameters. The usage summary describes how to invoke and exit from the Librarian Utility, how to direct output, and any restrictions you should be aware of.

The Description Section explains how to use the Librarian Utility.

The Qualifier Section describes each DCL command qualifier. Qualifiers appear in alphabetical order.

Associated Documents

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

- *VAX/VMS DCL Dictionary*
- *Guide to Programming on VAX/VMS*

Conventions Used in This Document

Convention	Meaning
<code>RET</code>	A symbol with a one- to three-character abbreviation indicates that you press a key on the terminal, for example, <code>RET</code> .
<code>CTRL/x</code>	The phrase CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O.
<code>\$ SHOW TIME</code> <code>05-JUN-1985 11:55:22</code>	Command examples show all output lines or prompting characters that the system prints or displays in black letters. All user-entered commands are shown in red letters.
<code>\$ TYPE MYFILE.DAT</code> . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.

Preface

Convention	Meaning
file-spec,...	Horizontal ellipsis indicates that additional parameters, values, or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square 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

A new qualifier, /DATA, allows you to store a library in data-reduced format, or expand a library previously stored in data-reduced format.

LIBRARIAN

The Librarian Utility 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, respectively. For more information on file maintenance, see the *VAX/VMS DCL Dictionary*.

FORMAT

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

Command Qualifiers

<i>/BEFORE</i> [=time]	<i>See text.</i>
<i>/COMPRESS</i> [(option[,...])]	<i>See text.</i>
<i>/CREATE</i> [(option[,...])]	<i>See text.</i>
<i>/CROSS_REFERENCE</i> [(option[,...])]	<i>See text.</i>
<i>/DATA</i> =option	<i>None.</i>
<i>/DELETE</i> =(module[,...])	<i>None.</i>
<i>/EXTRACT</i> =(module[,...])	<i>None.</i>
<i>/FULL</i>	<i>See text.</i>
<i>/[NO]GLOBALS</i>	<i>/GLOBALS</i>
<i>/HELP</i>	<i>/OBJECT</i>
<i>/HISTORY</i>	<i>See text.</i>
<i>/INSERT</i>	<i>/REPLACE</i>
<i>/[NO]LIST</i> [(file-spec)]	<i>/NOLIST</i>
<i>/[NO]LOG</i>	<i>/NOLOG</i>
<i>/MACRO</i>	<i>/OBJECT</i>
<i>/[NO]NAMES</i>	<i>/NONAMES</i>
<i>/OBJECT</i>	<i>/OBJECT</i>
<i>/ONLY</i> =(module[,...])	<i>None.</i>
<i>/OUTPUT</i> =file-spec	<i>See text.</i>
<i>/REMOVE</i> =(symbol[,...])	<i>None.</i>
<i>/REPLACE</i>	<i>/REPLACE</i>
<i>/SELECTIVE_SEARCH</i>	<i>See text.</i>
<i>/SHARE</i>	<i>/OBJECT</i>
<i>/SINCE</i> [(time)]	<i>See text.</i>
<i>/[NO]SQUEEZE</i>	<i>/SQUEEZE</i>
<i>/TEXT</i>	<i>/OBJECT</i>
<i>/WIDTH</i> =n	<i>See text.</i>

Positional Qualifier

/MODULE=module-name

Default

None.

LIBRARIAN

Command 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 will be 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 control the default library file type by specifying the appropriate qualifier as indicated below.

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.

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 will be 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 will assume a default file type

of OBJ, designating an object file. You can control the default file type by specifying the appropriate qualifier as indicated below.

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

Note also that the file type you specify with the library file specification affects 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 do 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

Invoking

Enter the DCL command LIBRARY.

Exiting

The Librarian Utility exits after the operations specified by the LIBRARY command have completed.

Directing Output

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.

Privileges/Restrictions

Use of the librarian is not restricted.

DESCRIPTION

This section describes how to use the 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 VAX/VMS Linker searches specified object module libraries when it encounters a reference it cannot resolve in one of its input files. For more information on how the linker uses libraries, see the description of the linker in the *VAX/VMS Utilities Reference Volume*.

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 Reference Volume* for information on defining macros.

LIBRARIAN

Description

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 provides 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. See Section 2 for information about creating help modules for insertion into help libraries.

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. See Section 4 for information on how to create a shareable image library.

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

```
$ LIB/CREAT/OBJ LIB.xxx *.obj
```

and then access it by typing

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

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 *VAX/VMS Utility Routines Reference 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 are in the format in which they were entered, that is, individual uppercase and lowercase characters are entered that way. 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” will 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 that the character case be identical; for example, for SAMPLE, the user 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 VAX/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 string is .EXE) and processes it as if it had been specified in an options file.

Unless 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 string 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 files.

The librarian uses the GSMATCH of the shareable image as the module ident 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/times found in the library and the shareable image differ. For more

LIBRARIAN

Description

information about GSMATCH processing, see the description of the linker in the *VAX/VMS Utilities Reference Volume*.

Users 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 *VAX/VMS Utility Routines Reference Manual*. In this way, program can quickly retrieve relevant information about using your program.

You create help libraries in the same manner that 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. This section describes 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 key names, or topics, numbered key-1 through key-9. 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 and/or qualifiers. This organization enables users of your program to find a 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 are in the format in which they were entered, that is, individual uppercase and lowercase characters are entered that way. A second, identically spelled keyword (but of a different or mixed character case) to the same library replaces the previous keyword. However, character case is ignored for match operations, for example, "help Sample" and "help SAMPLE" will 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 name characters to the following:

- Uppercase and lowercase letters (A,a,B,b...Z,z)
- Digits (0,1,2...9)

- Dollar sign (\$)
- Underscore (_)
- Hyphen (-)

In addition, it is strongly recommended 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 explicitly specify them 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 mark (")

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.

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

5.2 Formatting Help Files

Each key-1 line in the module consists of the key number 1 in the first column, followed by the name of the key. Subsequent subkey lines, 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 an individual module as a group of key-1 and subkey lines, and their associated text. A module is terminated either by another key-1 line or an end-of-file record.

LIBRARIAN

Description

The format of a help source file is shown below.

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

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 mark as the first character on a line, it assumes that the line is for 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 output the comment lines.

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 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 will provide 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.

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.

(Continued on next page)

LIBRARIAN

Description

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

2 Command_Qualifiers

/BEFORE

/BEFORE[=time]

Used in conjunction with the /LIST qualifier to specify that only those files 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 files 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.

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 *VAX/VMS Utility Routines Reference Manual*.

The HELP command by default 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. The /LIBRARY qualifier for the HELP command allows you to search another library in place of the default libraries. For details, see the description of the HELP command in the *VAX/VMS DCL Dictionary*.

When you retrieve help text, you specify the key-1 level, followed by any subkeys that contain appropriate help information. The librarian returns the help text associated with the key path you specified. 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 from 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						

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

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 will issue a message. For example:

\$ HELP FIRE

Sorry, no documentation on FIRE

Additional information available:

This message will be 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 will print 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 will include a list of all the subkeys associated with the last correctly specified key.

6

Librarian Routines

The librarian provides routines that your programs can call to

- 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 furnish whatever arguments the routine requires. When the routine completes execution, it returns control to your program.

The *VAX/VMS Utility Routines Reference Manual* describes, in detail, each of the librarian routines.

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

¹The /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.

²This qualifier applies only to macro libraries.

³This qualifier applies only to object libraries.

⁴This 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

Used with the /LIST qualifier to specify that only those modules dated earlier than a particular time be listed.

FORMAT	/BEFORE[=<i>time</i>]
---------------	------------------------------

qualifier value	<i>time</i>
------------------------	--------------------

Limits the modules to be listed to a particular time of generation.

You can specify an absolute time or a combination of absolute and delta times. For details on specifying times, see the *VAX/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 date. If you specify /BEFORE without a date or time, the default provides the files created before today.
--------------------	---

EXAMPLE

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

The LIBRARY command lists the modules that were inserted into MATH-LIB.OLB before 3 P.M. on April 15.

/COMPRESS

Recovers space that had been 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 value

option

An option (listed below) 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.
GLOBALS:n	Specifies the maximum number of global symbols the library can contain (for object module libraries only).
HISTORY:n	Specifies the maximum number of library update history records that the library will maintain.
KEEP	Copies library update history records and any additional user data in the module header, to the compressed library.
KEYSIZE:n	Changes the maximum length of module names or global symbol names.
MODULES:n	Specifies the maximum number of modules or macros the library can contain.
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.

LIBRARIAN

/COMPRESS

EXAMPLE

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

The 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 contain modules to be placed in the library.

FORMAT

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

qualifier value

option

An option (listed below) that overrides the system defaults to control the size of the library.

DESCRIPTION

By default, the LIBRARY command creates an object module library; specify /MACRO, /HELP, /TEXT, or /SHARE to indicate that the library is a macro, help, text, or shareable image library.

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 will maintain. The default maximum 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 VAX/VMS Linker do not accept module names or global symbol names in excess of 31 characters. |
| 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 directory 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.

LIBRARIAN

/CREATE

EXAMPLES

1 \$ LIBRARY/CREATE TESTLIB ERRMSG,STARTUP

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

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

The 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 will be 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 below) 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	Specifies that all types of cross-references are required
MODULE	Specifies a cross-reference listing of both the global symbol references in the module and the global symbol definitions
NONE	Specifies that no cross-reference listing is required
SYMBOL	Provides a cross-reference listing by symbol name
VALUE	Provides a 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

The **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 value

option

The options are REDUCE, which stores a library in data-reduced format, and EXPAND, which expands a library previously stored in data-reduced format. There is no default; one of the options must be specified.

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

The 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 value

module

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 will issue the message:

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

EXAMPLE

\$ LIBRARY/DELETE=FREEZE/LOG THAW

The 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 value ***module***
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 will write 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 will write 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

\$ LIBR/EXTR=(ALLOCATE,APPEND)/OUT=MYHELP SYS\$HELP:HELPLIB.HLB

The LIBRARY command specifies that the modules ALLOCATE and APPEND be extracted from the help library HELPLIB.HLB and output to the file MYHELP.HLP in the user's current default directory.

/FULL

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

FORMAT /FULL

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

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

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 will list the module name in the update history records.

EXAMPLE

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

The 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 **/[NO]GLOBALS**

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

The 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 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

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

LIBRARIAN

/HISTORY

/HISTORY

Used with the /LIST qualifier to request that update history record headers be listed (for libraries that contain a history) in the format

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

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

DESCRIPTION When used with /LIST/FULL, requests that the names of updated modules be listed in addition to the update history headers.

EXAMPLE

\$ LIBRARY/LIST/HISTORY/MACRO SETUP

The 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 will create 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 will create a separate entry for each one, naming the module name table entries according to the names specified on the .MACRO directives or in the key-1 name in the HELP format (see Section 5.1).

In text libraries, unlike object, macro, and help libraries, the input file 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 the LIBRARY command inserts modules into an existing library, it 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 will be issued and the module or symbol will not be 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
```

The LIBRARY command adds the module SCANLINE.OBJ to the library TESTLIB.OLB. The library is specified as input to the linker by using 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 will search 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 value

file-spec

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 will write 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 will use the default file type of LIS.

If you specify /LIST with qualifiers that perform additional operations on the library, the LIBRARY command will create 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 _DBBO:[LIBRAR]LIBRAR.OLB;1 on 14-JUN-1984 10:08:28
Creation date: 12-JUN-1984 19:40:36      Creator: VAX-11 librarian V03-00
Revision date: 14-JUN-1984 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

The 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

The 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/LIST ALLOBJECTS *

The 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 has 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.

/LOG

Controls whether the LIBRARY command verifies each library operation.

FORMAT **/[NO]LOG**

DESCRIPTION If you specify /LOG, the LIBRARY command will display 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

The 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.

LIBRARIAN

/MACRO

/MACRO

Indicates that the library is a macro library.

FORMAT

/MACRO

DESCRIPTION

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

EXAMPLE

\$ LIBRARY/MACRO/INSERT MONTHS APRIL

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

/MODULE

Names a text module that you want to replace or insert into a text library.

FORMAT	/MODULE= <i>module-name</i>
---------------	------------------------------------

qualifier value	<i>module-name</i>
------------------------	---------------------------

The name of the module to be inserted.

DESCRIPTION	<p>When you insert text modules into a library, the input file that you specify is taken 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 the input file name, use the /MODULE qualifier to name the added module.</p>
--------------------	--

You can also use the /MODULE qualifier to enter a text module interactively. If you specify the logical name SYS\$INPUT as the input file, and issue the /MODULE qualifier, the LIBRARY command will insert 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.

EXAMPLE

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

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

LIBRARIAN

/NAMES

/NAMES

Controls, when /LIST is specified for an object module library, 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

/[NO]NAMES

DESCRIPTION

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

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

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

EXAMPLE

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

The 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.

/OBJECT

Indicates that the library 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

The LIBRARY command inserts modules from APRIL.OBJ into the object library MONTHS.OLB. The /OBJECT is optional; the librarian assumes the library is an object module library unless you specify otherwise.

LIBRARIAN

/ONLY

/ONLY

Used with /LIST or /CROSS_REFERENCE to specify the individual modules on which the LIBRARY command can operate.

FORMAT

/ONLY=(*module*[,...])

qualifier value

module

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.

Wildcard characters are allowed in the module name specification.

EXAMPLE

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

The 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.

/OUTPUT

Specifies, when used with the /EXTRACT, /COMPRESS, /CROSS_REFERENCE, or /DATA qualifiers, 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 will be used depending on the library function qualifier and, in some cases, the library type qualifier as shown below.

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

EXAMPLES

1 \$ LIBR/EXTR=(ALLOCATE,APPEND)/OUT=MYHELP SYS\$HELP:HELPLIB.HLB

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

LIBRARIAN

/OUTPUT

2 \$ LIBRARY/CROSS_REFERENCE=ALL/OUTPUT=SYS\$OUTPUT LIBRAR

The 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.

/REMOVE

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

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

qualifier value **symbol**
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

The 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.

LIBRARIAN

/REPLACE

/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 name 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.

/REPLACE is the LIBRARY command's default operation. If you specify an input file parameter, the LIBRARY command will either replace or insert the contents of the input file into the library. If you use the /LOG qualifier with the /REPLACE qualifier, the LIBRARY command will display, in the following form, the names 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

The 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 exiting module.

/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, the linker selectively searches the modules when the library is specified as a linker input file; 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 quality 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

The 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 VAX/VMS Linker.

LIBRARIAN

/SHARE

/SHARE

Indicates that the library is a shareable image library.

FORMAT

/SHARE

DESCRIPTION

The 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

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

/SINCE

Used with the /LIST qualifier to specify that only those modules dated later than a particular time be listed.

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

qualifier value ***time***

Limits the modules to be listed to a particular time of generation.

You can specify an absolute time or a combination of absolute and delta times. For details on specifying times, see the *VAX/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 date. If you specify /SINCE without a date or time, the default provides the files created since today began.

EXAMPLE

\$ **LIBRARY/HELP/LIST/SINCE=:12 ERRMSG**

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

LIBRARIAN

/SQUEEZE

/SQUEEZE

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

FORMAT

/[NO]SQUEEZE

DESCRIPTION

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

Use /SQUEEZE 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

The 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 insertion into the library.

/TEXT

Indicates that the library 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

The 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

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

LIBRARIAN
/WIDTH

/WIDTH

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

FORMAT **/WIDTH=*n***

qualifier value ***n***
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

The 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-5

Comment lines

 help files • LIB-8

/COMPRESS qualifier • LIB-15

 used with /OUTPUT • LIB-35

/CREATE qualifier • LIB-2, LIB-17

/CROSS_REFERENCE qualifier • LIB-19

 used with /ONLY • LIB-34

 used with /OUTPUT • LIB-35

D

/DATA qualifier • LIB-20

 used with /OUTPUT • LIB-35

/DELETE qualifier • LIB-21

Directing output • LIB-3

E

Exiting • LIB-3

/EXTRACT qualifier • LIB-2, LIB-22

 used with /OUTPUT • LIB-35

F

Format • LIB-1

/FULL qualifier • LIB-23

 used with /HISTORY • LIB-26

G

Global symbol table (GST) • LIB-4

/GLOBALS qualifier • LIB-24

H

Help files

 comment lines • LIB-8

 creating • LIB-6 to LIB-7

 formatting • LIB-7

 qualifier lines • LIB-8

 restrictions • LIB-6

Help library • LIB-4, LIB-6

 character case • LIB-5

 index keywords • LIB-6

 key names • LIB-6 to LIB-7

HELP LIBRARY command display • LIB-10 to LIB-12

/HELP qualifier • LIB-25

Help text

 example • LIB-8 to LIB-10

 retrieval • LIB-10 to LIB-12

/HISTORY qualifier • LIB-26

I

Index keywords

 in help libraries • LIB-6

Input-file-specification • LIB-2

 default file type • LIB-3

Insert files/modules • LIB-27

/INSERT qualifier • LIB-2, LIB-27

Invoking the librarian • LIB-3

Index

K

Key lines

formatting • LIB-7

Key names

in help libraries • LIB-6 to LIB-7, LIB-11

L

Librarian routines • LIB-12

calling • LIB-12

Libraries

types of • LIB-3

LIBRARY command

creating a new library

/CREATE • LIB-17

cross-referencing

/CROSS_REFERENCE qualifier • LIB-19

directing output • LIB-3

exiting • LIB-3

format • LIB-1

input-file-specification • LIB-2

default file type • LIB-3

invoking • LIB-3

library-file-specification • LIB-2

default file type • LIB-2

privileges • LIB-3

qualifier list • LIB-1

qualifiers • LIB-13 to LIB-44

restrictions • LIB-3

time

specifying • LIB-14

Library header • LIB-4

Library index • LIB-4

Library size

See /COMPRESS qualifier

Library size/format

See /CREATE

Library-file-specification • LIB-2

default file type • LIB-2

/LIST qualifier • LIB-28

used with /BEFORE • LIB-14

used with /FULL • LIB-23

used with /HISTORY • LIB-26

/LIST qualifier (cont'd.)

used with /NAMES • LIB-32

used with /ONLY • LIB-34

used with /SINCE • LIB-41

/LOG qualifier • LIB-29

M

Macro library • LIB-3

character case • LIB-5

/MACRO qualifier • LIB-30

Module header • LIB-4

Module name table (MNT) • LIB-4

/MODULE qualifier • LIB-31

N

/NAMES qualifier • LIB-32

O

Object library • LIB-3

character case • LIB-5

/OBJECT qualifier • LIB-33

/ONLY qualifier • LIB-34

/OUTPUT qualifier • LIB-35

used with /COMPRESS • LIB-15

used with /CROSS_REFERENCE • LIB-19

used with /EXTRACT • LIB-22

P

Privileges • LIB-3

Q

Qualifier lines

help files • LIB-8

Qualifiers • LIB-13 to LIB-44

R

Reformatting libraries

 /COMPRESS qualifier • LIB-15

 /REMOVE qualifier • LIB-37

 /REPLACE qualifier • LIB-2, LIB-38

Restrictions • LIB-3

 in help file keys • LIB-6

S

 /SELECTIVE_SEARCH qualifier • LIB-39

 /SHARE qualifier • LIB-40

Shareable image library • LIB-4, LIB-5

 /SINCE qualifier • LIB-4 1

 /SQUEEZE qualifier • LIB-42

T

Text library • LIB-4

 character case • LIB-5

 /TEXT qualifier • LIB-43

Types of libraries • LIB-3

 help library • LIB-4

 macro library • LIB-3

 object library • LIB-3

 shareable image library • LIB-4

 text library • LIB-4

W

 /WIDTH qualifier • LIB-44

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital



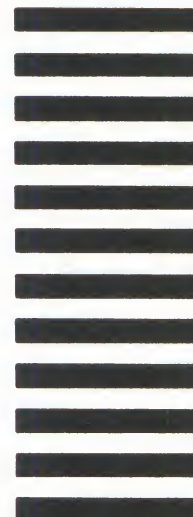
No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here

Cut Along Dotted Line