To: Distribution

From: Gary C. Dixon

Date: November 8, 1974

Subject: New Library Tools: library_map, library_print, library_descriptor, and multics_libraries_

In another MTB, the design principles used to implement the new library maintenance tools were developed, and the strategy for centralizing the library organization information and search procedures was outlined.

This MTB describes the command interfaces for two new tools, library_map (lm) and library_print (lpr). library_map replaces the msl_short_format and msl_global_format commands by providing detailed status information for all entries in the Multics System Libraries. library_print performs a new library maintenance function, that of gathering together a group of printable library entries (like info segments, peruse text segments, bind lists, source segments, etc) into an MSF for printing offline.

The MTB also describes the library_descriptor command, which can be used to set the name of the default library descriptor and to return information about the contents of a library descriptor. And finally, a description of the multics_libraries_ library descriptor is included.

These commands and the library descriptor will be included in a design review of the new library tools, to be held in the near future. Note that the command which replaces msl_info has not been completely designed yet, and will be described by a subsequent MTB.

(This page intentionally left blank)

library_map

Special Command
Administrative/User Ring
11/08/74

Name: library_map, lm

The library_map command selects entries from a library, and writes the status of these entries into a file suitable for dprinting. The entries in the file are alphabetized by primary name.

A full range of status information can be included in the output by using the various output arguments. Besides the information returned by the status command, the status can include object segment attributes reported by object_info_ and library-dependent information.

The command uses a library descriptor data base and search program to define the structure, contents, and naming conventions of the library. Refer to the writeup for the library_descriptor_compiler for more information about library descriptors.

When no output arguments are specified, the choice of status information is controlled by the library search program for the particular library being mapped. For the Multics Offline Libraries (Hardcore, Salvager, BOS, and 355), the information includes: the names on each library entry; it entry type; its date of modification and path name; and (for Hardcore and Salvager Libraries) the id of the system in which the entry was last installed. For other Multics Libraries, the information includes: the names on each library entry; its entry type; its dates of modification and dumping; its path name and the path name of a link's target; and its current length, bit count, and ring brackets.

Usage

library_map -search_name1-...-search_namen-
-ctl_arg1-...-ctl_argn- -output1-...-outputn-

The search names, control arguments, and output arguments described below appear in the command in any order.

1) search_name1
is an entry name which identifies the library entries to be output. The Multics star convention may be used to identify a group of entries. If no search names are specified, then the default search names specified in the library descriptor are used.

2) ctl_arg1 may be any of the following control arguments.

-search_name search_name
-snm search_name
is an entry name which identifies the library entries to be output. The Multics star convention may be used to identify a group of entries. This control argument must be used when search_name begins with a minus (-) to distinguish the search_name from a control argument. The -search_name control argument may be used several times in the same command to specify several different search names.

-library lib
-lb lib lib is a name which identifies the particular library or group of libraries which are to be searched when the library descriptor describes more than one library. The Multics star convention may be used to identify a group of libraries. The list of acceptable library names is defined by the library descriptor. More than one -library control argument may be specified to identify several groups of libraries. If the -library control argument is not specified, then the default library names specified in the library descriptor are used.

-output_file file
-of file file is the path name of the output file in which the library map is to be generated. It may be a relative or absolute path name. If it does not end in a suffix of .map, then one is assumed. If the -output_file control argument is not specified, then the map is generated in the library.map file created in the working directory.

-header heading
-he heading

heading is a character string which is placed on the header page of the output file to identify the contents of the file. If the string contains blanks, then it must be enclosed in quotes. Only the first 120 characters of the string will be used. If the -header control argument is not specified, then a default heading is placed on the heading page. (See Notes below.)

-footer footing
-fo footing

footing is a character string which is placed in the footing line of each output page to identify the library being mapped. If the string contains blanks, then it must be enclosed in quotes. Only the first 45 characters of the string are used. If the -footer control argument is not specified, then a default footing line is used. (See Notes below.)

-parent, -par

specifies that status information should be output for the parent of each library entry. For example, information about an archive will be output when one of its components matches a search name. Normally, the output includes status for only those library entries which match a search name.

-components
-cmp

specifies that status information should be output for all components of library archives which are identified by a search name. Normally, the output includes status for only those library entries which match a search name.

-retain specifies that library entries which await deletion from the library (as determined by the library search program) should be output. Normally, such entries are excluded.

-library_descriptor refname
-lds refname
refname is the reference name of the library descriptor which describes the libraries to be searched. The descriptor identified by the refname will be found by using the search rules, which are documented in MPM Section 3.2. If the -library_descriptor control argument is not specified, then the default library descriptor is used. (See Notes below.)

-chase requests that any links which exist between a library link and its eventual target be omitted from the output. Normally, these intermediate links are included.

-check_archive, -ckac
specifies that each library segment and archive component is to be checked to see if it is an archive segment, or an archived archive. If the -check_archive control argument is not specified, checking for archives will be performed at the option of the library search program.

-check_character, -ckch
specifies that each library entry is to be checked to see if its contents is unprintable (i.e., contains non-ASCII characters). In addition, unprintable segments are checked to determine if they are peruse_text object segments. If the -check_character control argument is not specified, character checking will be performed at the option of the library search program.

-check_object, -ckob
specifies that each library segment and archive component is to be checked to see if it is an object segment. If the -check_object control argument is not specified, object checking will be performed at the option of the library search program.

-check, -ck
specifies that all there types of checking (archive, character, and object) are to be performed.

3) output_i may be any of the following output arguments. These arguments specify which status information is to be returned for each library entry which appears in the map. If no output arguments are specified, then default information is output for each library entry, under control of the library search program.

-default, -dft
requests that the default information for each library entry be output, in addition to output requested by any other output arguments.

-all, -a requests that all available status information be output.

-name, -nm requests that all of the names on each library entry be output.

-first, -ft requests that the first name on each library entry be output.

-match requests that the names on each library entry which match any of the search names given in the library_map command be output.

-type, -tp requests that the type of each library entry (link, segment, directory, archive, archive component, multi-segment file, or msf component) be output.

-parent_path, -pp
requests that the path name of the parent of each library entry be output.

-link, -lk requests that the path name of the target of a library link be output.

-date, -dt requests that the date modified, date used, date entry modified, and date dumped for each library entry be output. For an archive component, the date entry modified corresponds to the date component updated.

-date_modified, -dtm
requests that the date on which each library entry was last modified be output.

-date_used, -dtu
requests that the date on which each library entry was last used be output.

-date_entry_modified, -dtem
requests that the date on which the entry for a library link, segment, directory, archive, or multi-segment file was last modified, or the date on which a library archive component was last updated into its archive, be output.

-date_dumped, -dtd
requests that the date on which a library entry was last dumped onto a backup tape be output.

-length, -ln
requests that the current length, records used, and bit count or msf indicator of each library entry be output. The records used are included only when different from the current length.

-current_length, -cln
requests that the current length of each library entry be output.

-records_used, -ru
requests that the number of records occupied by each library entry be output. For a multi-segment file, the value includes only those records used by the msf directory. Those occupied by the msf components are included in the map entry for each component. If both the current length and records used have been requested, then the records used will be omitted from the output if equal to the current length.

-bit_count, -bc
requests that the bit count of each library segment, archive, archive component, and msf component be output, along with the msf indicator of each library multi-segment file.

-access, -acs
requests that the user's access mode to each library entry, and each library entry's ring brackets be output.

-mode, -md requests that the user's access mode to each library entry be output.

-ring_brackets, -rb
requests that the ring brackets of each library entry be output.

-contents, -ct
requests that the contents of each library entry be checked to determine whether the entry is an archive, an object entry (i.e., an object segment or archive component), a peruse text object entry, or another type of unprintable entry. For each object entry, the following information is output: date compiled or bound; compiler or binder version number; compiler options; and object entry attributes. The compiler version number and printable character length of each peruse text object entry is output. An indication is output for each unprintable entry. (See Notes below.)

-date_compiled, -dtc
requests that the date compiled be output for each library (bound or unbound) object entry. (See Notes below.)

-compiler_version -cv
requests that name and version information be output for the compiler of each object entry. (See Notes below.)

-compiler_options, -co
requests that the options used when compiling each object entry be output. (See Notes below.)

-object_info, -oi
requests that the attributes of each object entry be output. Attributes include: bound object indicator; old object format indicator; non-standard object format indicator; and the octal value of any call limiter. (See Notes below.)

-character, -ch
requests that an indication be output for each library entry whose contents is unprintable (i.e., contains non-ASCII characters). (See Notes below.)

-peruse_text, -pt
requests that the compiler version number and printable character length be output for each peruse text object entry. (See Notes below.)

-device_id, -did
requests that the type of device on which the library entry is stored be output.

-copy_switch, -cs
requests that the copy switch setting for the library entry be output.

-offset
requests that the offset (from the beginning of its containing segment) of the contents of an archive component be output. The offset is an octal word count.

-unique_id, -uid
requests that the unique identifier of the library entry be output in octal.

-error, -er
requests that any error which occurred while obtaining status be indicated by the appropriate error message. The message appears in the map entry for the parent of the library entry in which the error occurred.

-level, -lv
requests that a level number be output for each library entry. The level number indicates the relationship between a library entry and its components. For example, the map entry for an archive might appear in the map at level 1, and the archive's components might appear at level 2. Normally, the level is indicated only by relative indentation of entry names.

-new_line, -nl
requests that a line be skipped before writing the status of each level 1 entry in the map. This line makes it easier to identify level 1 entries. Normally, no lines are skipped between entries.

Notes

If the output file already exists, it is truncated and rewritten. Thus, if several library_map commands are executed in the same working directory (by the same process, or by different processes) without including an -output_file control argument, then the output of all but the last command is overwritten. In such cases, the -output_file control argument should be specified to prevent the output of the last command from overwriting the output of preceding commands.

If the -header heading control argument is specified, then the heading character string is centered on the header page of the output file beneath the lines:

Map of the nn Entries

of the

The heading character string should be worded with this in mind. For example:

Map of the 35 Entries

of the

Standard Service System Bind Listing Library

If no -header control argument is specified, then a default heading line is constructed by concatenating the library names, as shown below:

Map of the 350 Entries

of the

Libraries

standard_service.list, unbundled.list,
tools.list, author_maintained.list, network.list

If the -footer footing control argument is specified, then the footing line appears at the lower left corner of each output page (except the header page), along with the name(s) of the level 1 entries appearing on that page, and the page number. If the -footer control argument is omitted, then the library names are concatenated together as with the default heading line, and used as the default footing line.

In order for some output arguments to take effect, information in addition to simple status must be available. Therefore, some output arguments automatically invoke certain of the control arguments. For example, the -contents output argument requires that archive, character, and object checking be performed, and therefore invokes the -check control argument. Similarly, -date_compiled, -compiler_version, -compiler_options, and -object_info output options invoke the -check_object control argument; and the -character and -peruse_text output arguments invoke the -check_character control argument. The -all output argument invokes the -parent, -default, and -check control arguments, as well as all of the output arguments.

The default library name(s) and search name(s) used for the library_map command are defined in the library descriptor for the libraries being mapped. The library_descriptor command can be used to print these default values. In particular, the default values for the default library descriptor can be printed by typing the command:

```
lds default library_map
```

Refer to the writeup on the library_descriptor command for more details.

When no -library_descriptor control argument is given in the command, the default library descriptor is used. The name of the default library descriptor can be set and printed with the library_descriptor command. The initial default library descriptor describes the Multics System Libraries.

Examples

The command

```
library_map -lb info.* -lb peruse_text.info **.info **.pt
-of documentation
```

creates the documentation.map file in the working directory, which contains a map of the entries in the info.* and peruse_text.* libraries which match the search names **.info or **.pt. The command

```
library_map -lb online.* ** -of online -dtd -dft
```

creates the online.map file which contains a map of all of the entries in the online.* libraries. Each map entry includes the date dumped, as well as whatever default information was specified by the library search program. The command

```
library_map
```

creates a map in the library.map file of the working directory which contains status for those entries in the default library (or libraries) which match the default search name(s). These default values are specified by the default library descriptor data base.

 (END)

(This page intentionally left blank)

library_print

Special Command
Administrative/User Ring
11/08/74

Name: library_print, lpr

The library_print command selects printable entries from a library, and writes the contents of these entries into a file suitable for printing. Printable library entries are those which contain only ASCII characters. The ASCII portion of peruse text object segments is also printable. Thus printable entries can include source segments, listings, bind files, info segments, peruse text object segments, exec_com control segments, printable multi-segment files, etc.

The entries in the print file are alphabetized by primary name. Each entry is preceded by a header which lists the status of the entry. An index of all entry names appears at the end of the file.

The command uses a library descriptor data base and search program to define the structure, contents, and naming conventions of the library. Refer to the writeup for the library_descriptor_compiler for more information about library descriptors.

When no output arguments are specified, the status information which is included in the header of each entry is controlled by the library search program for the particular library being printed. For the Multics System Libraries, the information includes: the names on the library entry; its entry type; its date of modification; its path name; and (for Hardcore and Salvager Libraries) the id of the system in which the entry was last installed.

Usage

library_print -search_name1-...-search_namen-
-ctl_arg1-...-ctl_argn- -output1-...-outputn-

The search names, control arguments, and output arguments described below appear in the command in any order.

1) search_namei
is an entry name which identifies the library entries to be output. The Multics star convention may be used to identify a group of entries. If no search names are specified, then the default

search names specified in the library descriptor are used.

2) ctl_argi may be any of the following control arguments.

-search_name search_name
-snm search_name
is an entry name which identifies the library entries to be output. The Multics star convention may be used to identify a group of entries. This control argument must be used when search_name begins with a minus (-) to distinguish the search_name from a control argument. The -search_name control argument may be used several times in the same command to specify several different search names.

-library lib
-lb lib lib is a name which identifies the particular library or group of libraries which are to be searched when the library descriptor describes more than one library. The Multics star convention may be used to identify a group of libraries. The list of acceptable library names is defined by the library descriptor. More than one -library control argument may be specified to identify several groups of libraries. If the -library control argument is not specified, then the default library names specified in the library descriptor are used.

-output_file file
-of file file is the path name of the output file in which the library print out is to be generated. It may be a relative or absolute path name. If it does not end in a suffix of .print, then one is assumed. If the -output_file control argument is not specified, then the print out is generated in the library.print file created in the working directory.

-header heading
-he heading

heading is a character string which is placed on the header page of the output file to identify the contents of the file. If the string contains blanks, then it must be enclosed in quotes. Only the first 120 characters of the string will be used. If the -header control argument is not specified, then a default heading is placed on the heading page. (See Notes below.)

-footer footing
-fo footing

footing is a character string which is placed in the footing line of each output page to identify the library being printed. If the string contains blanks, then it must be enclosed in quotes. Only the first 45 characters of the string are used. If the -footer control argument is not specified, then a default footing line is used. (See Notes below.)

-parent, -par

specifies that the contents of the parent of each library entry should be output, rather than the library entry itself. For example, the contents of an archive will be output when one of its components matches a search name. Normally, only those library entries which match a search name are output.

-components
-cmp

specifies that the contents of each component of a library archive which matches a search name is to be output individually. Normally, only matching components of the archive are output (or if no components match, the entire archive is output).

-retain specifies that library entries which await deletion from the library (as determined by the library search program) should be output. Normally, such entries are excluded.

-library_descriptor refname
-lds refname
refname is the reference name of the library descriptor which describes the libraries to be searched. The descriptor identified by the refname will be found by using the search rules, which are documented in MPM Section 3.2. If the -library_descriptor control argument is not specified, then the default library descriptor is used. (See Notes below.)

-chase requests that any links which exist between a library link and its eventual target be omitted from the status information in the header. Normally, these intermediate links are included.

-check_archive, -ckac
specifies that each library segment and archive component is to be checked to see if it is an archive segment, or an archived archive. If the -check_archive control argument is not specified, checking for archives will be performed at the option of the library search program.

-check_character, -ckch
specifies that each library entry is to be checked to see if its contents is unprintable (i.e., contains non-ASCII characters). In addition, unprintable segments are checked to determine if they are peruse_text object segments. If the -check_character control argument is not specified, character checking will be performed at the option of the library search program.

-check_object, -ckob
specifies that each library segment and archive component is to be checked to see if it is an object segment. If the -check_object control argument is not specified, object checking will be performed at the option of the library search program.

-check, -ck
specifies that all there types of checking (archive, character, and object) are to be performed.

3) outputi may be any of the following output arguments. These arguments specify which status information is to be returned in the header of each library entry which appears in the print out. If no output arguments are specified, then default information is output for each library entry, under control of the library search program.

-default, -dft
requests that the default information for each library entry be output, in addition to output requested by any other output arguments.

-all, -a requests that all available status information be output.

-name, -nm requests that all of the names on each library entry be output.

-first, -ft requests that the first name on each library entry be output.

-match requests that the names on each library entry which match any of the search names given in the library_print command be output.

-type, -tp requests that the type of each library entry (link, segment, directory, archive, archive component, multi-segment file, or msf component) be output.

-parent_path, -pp
requests that the path name of the parent of each library entry be output.

-link, -lk requests that the path name of the target of a library link be output.

-date, -dt requests that the date modified, date used, date entry modified, and date dumped for each library

entry be output. For an archive component, the date entry modified corresponds to the date component updated.

-date_modified, -dtm
requests that the date on which each library entry was last modified be output.

-date_used, -dtu
requests that the date on which each library entry was last used be output.

-date_entry_modified, -dtem
requests that the date on which the entry for a library link, segment, directory, archive, or multi-segment file was last modified, or the date on which a library archive component was last updated into its archive, be output.

-date_dumped, -dtd
requests that the date on which a library entry was last dumped onto a backup tape be output.

-length, -ln
requests that the current length, records used, and bit count or msf indicator of each library entry be output. The records used are included only when different from the current length.

-current_length, -cln
requests that the current length of each library entry be output.

-records_used, -ru
requests that the number of records occupied by each library entry be output. For a multi-segment file, the value includes only those records used by the msf directory. Those occupied by the msf components are included in the header information for each component. If both the current length and records used have been requested, then the records used will be omitted from the output if equal to the current length.

-bit_count, -bc
requests that the bit count of each library

segment, archive, archive component, and msf component be output, along with the msf indicator of each library multi-segment file.

-access, -acs
requests that the user's access mode to each library entry, and each library entry's ring brackets be output.

-mode, -md requests that the user's access mode to each library entry be output.

-ring_brackets, -rb
requests that the ring brackets of each library entry be output.

-contents, -ct
requests that the contents of each library entry be checked to determine whether the entry is printable. The compiler version number and printable character length of each peruse text object entry is output. Other entries are merely checked for printability. (See Notes below.)

-peruse_text, -pt
requests that the compiler version number and printable character length be output for each peruse text object entry. (See Notes below.)

-device_id, -did
requests that the type of device on which the library entry is stored be output.

-copy_switch, -cs
requests that the copy switch setting for the library entry be output.

-offset requests that the offset (from the beginning of its containing segment) of the contents of an archive component be output. The offset is an octal word count.

-unique_id, -uid
requests that the unique identifier of the library entry be output in octal.

-error, -er

requests that any error which occurred while obtaining status be indicated by the appropriate error message. The message appears in the header information for the parent of the library entry in which the error occurred.

-level, -lv

requests that a level number be output for each library entry. The level number indicates the relationship between a library entry and its components. For example, the header of the entry for an archive might have the archive at level 1 and the archive's components at level 2. Normally, the level is indicated only by relative indentation of entry names.

Notes

If the output file already exists, it is truncated and rewritten. Thus, if several library_print commands are executed in the same working directory (by the same process, or by different processes) without including an -output_file control argument, then the output of all but the last command is overwritten. In such cases, the -output_file control argument should be specified to prevent the output of the last command from overwriting the output of preceding commands.

If the -header heading control argument is specified, then the heading character string is centered on the header page of the output file beneath the lines:

Print Out of the nn Entries

of the

The heading character string should be worded with this in mind. For example:

Print Out of the 35 Entries

of the

Standard Service System Bind Listing Library

If no -header control argument is specified, then a default heading line is constructed by concatenating the library names, as shown below:

Map of the 350 Entries

of the

Libraries

standard_service.list, unbundled.list,
tools.list, author_maintained.list, network.list

If the -footer footing control argument is specified, then the footing line appears at the lower left corner of each output page (except the header page), along with the name(s) of the level 1 entries appearing on that page, and the page number. If the -footer control argument is omitted, then the library names are concatenated together as with the default heading line, and used as the default footing line.

In order for some output arguments to take effect, information in addition to simple status must be available. Therefore, some output arguments automatically invoke certain of the control arguments. For example, the -contents output argument requires that archive, character, and object checking be performed, and therefore invokes the -check control argument. Similarly, the -peruse_text output argument invokes the -check_character control argument. The -all output argument invokes the -default and -check control arguments, as well as all of the output arguments.

The default library name(s) and search name(s) used for the library_print command are defined in the library descriptor for the libraries being printed. The library_descriptor command can be used to print these default values. In particular, the default values for the default library descriptor can be printed by typing the command:

```
ldd >defaults>library_print
```

Refer to the writeup on the library_descriptor command for more details.

When no -library_descriptor control argument is given in the command, the default library descriptor is used. The name of the default library descriptor can be set and printed with the library_descriptor command. The initial default library descriptor describes the Multics System Libraries.

Examples

The command

```
library_print -lb info.* -lb peruse_text.info **.info **.pt
-of documentation
```

creates the documentation.print file in the working directory, which contains a print out of the entries in the info.* and peruse_text.* libraries which match the search names **.info or **.pt. The command

```
library_print -lb online.object **.bind -of online -dtd -dft
```

creates the online.print file which contains a print out of all of the bind files in the online object libraries. Each entry includes a header with the date dumped, as well as whatever default status information was specified by the library search program. The command

```
library_print
```

creates a print out in the library.print file of the working directory which contains the contents of those entries in the default library (or libraries) which match the default search name(s). These default values are specified by the default library descriptor data base.

 (END)

library_descriptor

Command
Administrative/User Ring
11/08/74

Names: library_descriptor, lds

The library_descriptor command controls the use of library descriptors by library maintenance commands, and provides information about the contents of a library descriptor. The command: can print and set the name of the library descriptor which is used by default in library maintenance commands when no library descriptor is explicitly specified; can print the path names of the library roots which are associated with one or more library names; can print detailed information about one or more of the library roots defined by the descriptor; and it can print the default library and search names associated with each library command.

The library descriptor is a data base which, with its associated library search program, defines the structure, contents, and naming conventions of the library. Refer to the writeup for the library_descriptor_compiler for more information about library descriptors.

Usage

library_descriptor key -options-

The keys and their options are described in the sections which follow.

Key: name

The name key returns the name of the default library descriptor which is currently being used. library_descriptor may be invoked as an active function when the name key is used.

Usage

library_descriptor name

Key: set_name

The set_name key sets the name of the default library descriptor.

Usage

library_descriptor set_name refname

1) refname is the reference name of the new default library descriptor. The descriptor identified by refname is searched for according to the search rules, which are documented in MPM Section 3.2.

Key: path

The path key returns the path name of the library root(s) which are identified by one or more library names. library_descriptor may be invoked as an active function when the path key is used.

Usage

library_descriptor path lib_name1 ... lib_namen -ctl_arg1-

1) lib_name1 is a name which identifies the particular library or group of libraries whose paths are to be returned. The Multics star convention may be used to identify a group of libraries.

2) ctl_arg1 may be the following optional control argument.

-library_descriptor refname
-lds refname
refname is the reference name of the library descriptor which defines the library roots whose path names are to be returned. The descriptor identified by refname will be found by using the search rules, which are documented in MPM Section 3.2. If the -library_descriptor control argument is not specified, then the default library descriptor is used.

Key: defaults

The defaults key prints the default library name(s) and search name(s) associated with one or more of the library maintenance commands.

Usage

library_descriptor defaults -command1- ... -commandn- -ctl_arg1-

1) command*i* is the name of the library maintenance command whose default library and search names are to be printed. If no command names are given, the default for all of the library maintenance commands will be printed.

2) ctl_arg*i* may be any of the optional control arguments defined for the path key.

Key: roots

The roots key prints detailed information about one or more library roots on the user's terminal. The information includes the names on each library root, its path name, and its type.

Usage

library_descriptor roots lib_name*i* ... lib_name*n* -ctl_arg*i*- ... -ctl_arg*n*-

1) lib_name*i* is a name which identifies the particular library roots about which information is to be printed. The Multics star convention may be used to identify a group of libraries.

2) -ctl_arg*i* may be one of the following control arguments.

-name, -nm specifies that all of the names defined for the library root are to be printed. Usually, only the first name and names which match the lib_name*i* arguments are printed.

-library_descriptor refname
-lds refname
as above.

 (END)

(This page intentionally left blank)

multics_libraries_

Data Base
Administrative/User Ring
11/08/74

Name: multics_libraries_

This data base is the library descriptor for the Multics System Libraries. Like all library descriptors, it defines the roots of the Multics System Libraries, the names by which library roots can be referenced in the various library maintenance commands, and the default library names and search names used for each of the library maintenance commands.

Refer to the writeup on library descriptors for a definition of the internal structure of this (and other) library descriptors. Refer to the writeup on the library_descriptor_compiler for the definition of the Library Description Language which is used to define the contents of the library descriptor.

The Multics Libraries

The Multics System is composed of the "logical libraries" listed below. Each of the libraries is, in turn, composed of several directories containing the different kinds of library segments (source, object, bind lists, info, include, peruse_text) which are stored in the libraries. A library maintenance command can reference an entire logical library by name, or one or more of its directories.

Note that the logical library organization defined below does not map directly onto the physical library organization in the Multics Storage System. However, the library maintenance tools can reference all of the physical libraries through their logical library names.

standard_library, std
: the library which contains most user commands and subroutines, and the system support routines for these commands and subroutines.

unbundled_library, unb
: the library which contains Honeywell programmed products and other unbundled software.

tools_library, tools
: the library which contains system maintenance and administrative commands and subroutines.

installation_library, inst
the library which contains installation-maintained software.

user_library, user
the library which contains user-maintained software.

network_library, net
the library which contains the software for linking the Multics System to the ARPA Network.

supervisor_library, sup
the library which contains the supervisor (ring 0) segments of the Multics System.

salvager_library, salv
the library which contains the Multics Storage System salvager commands and subroutines.

bootload_library, bos
the library which contains the commands and subroutines of the Bootload Operating System.

data_net_355_library, 355
the library which contains the commands and subroutines of the Data Net 355 Operating System.

Each of the above logical libraries contains one or more of the following logical directories.

source, s
the directory containing the source language segments which can be translated into the object segments of the library.

object, o
the directory containing the object segments produced by translating the source segments of the library.

lists, l
the directory containing the listings produced by binding several object segments together into a bound segment.

execution, x
the directory containing bound and unbound object segments and data bases used by users of Multics. These directories are generally included in the search rules of some or all users.

bound_comp, bc
: the directory containing the archives which may be bound into bound segments.

info, i
: the directory containing information segments which can be printed on the user's terminal under control of the help command. These segments describe the commands and subroutines included in the library, and outline library problems, upcoming changes, etc.

peruse_text, pt
: the directory containing the peruse_text object segments which describe the commands and subroutines of the library. Selected portions of these segments may be printed on the user's terminal under control of the peruse_text command.

include, incl
: the directory containing source segments which are included as part of several other source segments, under the control of a language translator.

Library Names

One or more libraries or directories may be referenced in a library maintenance command by giving the appropriate library or directory name.

- An entire library can be referenced by giving one of the library names listed above.
- A particular type of directory can be referenced across all libraries by giving one of the directory names listed above.
- A particular directory within a specific library can be referenced by giving a 2-component name of the form, library.directory. For example, standard_library.source or installation_library.info .
- The star convention may be used to identify several libraries or directories. For example, *.source or ** .
- Two groups of libraries can be referenced by the following names:

  online_libraries, on
  : standard_library, unbundled_library, tools_library, installation_library, user_library, network_library.

offline_libraries, off
supervisor_library, salvager_library, bootload_library, data_net_355_library.

Not all of the libraries listed above contain each type of directory. The following lists show which library.directory combinations are valid.

std.source	unb.source	tools.source
std.object	unb.object	tools.object
std.lists	unb.lists	tools.lists
std.execution	unb.execution	tools.execution
std.info	unb.info	tools.info
std.peruse_text	unb.peruse_text	tools.peruse_text
std.include	unb.include	tools.include
inst.source	user.source	net.source
inst.object	user.object	net.object
inst.lists	user.lists	net.lists
inst.execution	user.execution	net.execution
inst.info	user.info	net.info
inst.peruse_text	user.peruse_text	net.peruse_text
inst.include	user.include	net.include
sup.source	salv.source	
sup.bound_comp	salv.bound_comp	
sup.object	salv.object	
sup.include	salv.include	
bos.source	355.source	
bos.object	355.object	
bos.include		

Some examples of library names are:

online_libraries
off.source
standard_library.info
include
user.x
network_library.lists
pt
std.??????

Library Maintenance Command Defaults

The table below shows the default library names and search names used by each of the library maintenance commands. Commands which have no default values are not shown in the table.

Command	Default_Library_Name	Default_Search_Name
library_map	online_libraries	**
library_print	info	**.info
library_cleanup	online_libraries	!??????????????

 (END)