TO: Distribution

FROM: Arlene Scherer

DATE: 03/03/75

SUBJECT: New library_fetch command

Attached to this MTB is the description of the proposed new library_fetch command. It is a functional replacement for the current get_library_segment command.

library_fetch will use the library descriptor concept to gather information about where to find entries which are requested by the user. It will use the -library control argument as a replacement for the "-sys" option, eliminating the need to maintain gls control segments in the libraries.

It will be capable of accepting the Multics star convention in the searchname argument. It will be capable of accepting the Multics equals convention for renaming the entries which it fetches for the user.

It will be capable of fetching entries into a path, as well as into the user's working directory.

It will provide an option to create a printable record of the requested action, and a set of output arguments to control the information to be printed. This output file can be suitable for attaching to a Multics Change Request form, as a record of the date and time the entries were taken from the system libraries.

It will have a subroutine entry callable from a procedure, which will return an error code.

Comments or suggestions should be directed to Arlene Scherer via Multics mail.

library_fetch

Command
User Ring
03/03/75

Name: library_fetch, lf

The library_fetch command copies or extracts segments from the Multics system libraries or from a user-defined library structure into the user's working directory, or into the path specified by the user.

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

The library_fetch command may also be used to fetch entries from a subsystem library, or an entire subsystem library structure using the -library_descriptor (-lds) control argument with a subsystem-defined library descriptor.

Usage:

```
library_fetch  -search_name1-...-search_namen-
     -ctl_arg1-...-ctl_argn  -output1-...-outputn-
```

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

1) search_name_i
is an entry name which identifies the library entries to be fetched. More than one search name may be used in a command. The Multics star convention may be used to identify a group of entries.

2) ctl_arg_i
may be any or none of the following control arguments.

-into path
path is the absolute or relative pathname of the target entry which library_fetch will create. If the entry name of path is not the same as the searchname given, the entry will be renamed to the specified entry name. Note that only one "-into" path argument may be used in a command, but one or more searchnames may be used. The Multics equals convention may be used to rename a group of entries. If the "-into" path control argument is not used, the user's current working directory is the default path, and the original name of the entry is the default entry name.

-library lib
-lb lib
lib is a name which identifies the particular library or group of libraries which are to be searched for the given entries. 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.

-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.

-search_name search_name
-snm search_name
is an entry name which identifies the library entries to be fetched. 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.

-name
-nm
requests that all of the names on the requested segment(s) be copied. If no control arguments relating to external names are specified, the default options will be set by the search program for library_fetch. The default names are first name and matching name. If any of the name options are specified, and the output_file option is also specified, then the requested name information will be included in file.

-first
-ft
requests that only the first (primary) name of the entry will be attached to the entry which matches the searchname.

-match
specifies that only the names which match the searchnames are to be attached to the entry or entries requested by the user.

-parent
-par
requests that the parent of each library archive component which matches the search_name(s) be fetched, rather than the component itself. When the component of an archive matches the search_name(s) given, the -parent option will cause the parent archive to be copied.

-components
-cmp

specifies that those components of a library archive which match the search name are to be fetched. If the searchname matches the name of the archive itself, all components of the archive are fetched.

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

-chase
requests that any matching links which exists between a library entry and its eventual target be followed, and the target returned. The default is not to follow links unless -chase is specified.

-long
-lg
specifies that the originating path and name of the entries being fetched are to be printed on the terminal for the information of the user.

-output_file file
-of file
file is the path name of the output file in which the library fetch information is to be generated. It may be a relative or absolute path name. If it does not end in a suffix of ".fetch" then one is assumed. If the -output_file control argument is specified, then the output arguments listed below may be used to control the amount and type of information that will be included in file.

3) output_i
may be any of the following output arguments. These arguments control only the information which is to be included in the printable segment when the output_file option has been specified. These arguments do not affect the action which library_fetch will take regarding the extraction of entries from the libraries. If none of the arguments below are specified, then default information established by the library search program is output for each library entry.

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

-all
-a
requests that all available entry information be provided.

-type
-tp
requests that the type of each fetchable library entry (link, segment, archive or archive component) be included in file.

-parent_path
-pp
requests that the pathname of the parent be included whenever applicable. This output argument applies only to archive components.

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

-date_modified
-dtm

specifies that the date last modified be included in the output information.

-date_used
-dtu
specifies that the date last used be included in the output information.

-date_entry_modified
-dtem
requests the inclusion of the date on which the entry for a library link, segment, archive or archive component was last modified, or the date on which a component was last updated into its parent archive.

-date_dumped
-dtd
requests that the date last dumped be included if the -output_file option was also specified.

-length
-ln
specifies that the current length, records used, and bit count for each entry be included in file. The records used are included only when different from the current length.

-current_length
-cln
requests that the current length of each entry be included in file.

-records_used
-ru
requests that the number of records occupied by each library entry be included in file. 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 be included in the information in file.

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

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

-mode
-md
requests that the user's access mode be included for each library entry.

-contents
-ct
specifies that the contents of each library entry be checked for printability. If the entry is not printable, this information is included in file. If object segments are present in the group of requested entries, then object information is included as described in the section on the "-object_info" option.

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

-offset
specifies that the offset from the beginning of its archive be printed for the archive components requested.

-object_info
-oi
specifies that information relevant to object entries be included for each object entry requested. Such information includes time compiled and compiler Version.

-compiler_version
-cv

specifies that the compiler Version be included in the file output information.

-date_compiled
-dtc
specifies that the date compiled be included in file for each object entry requested.

-compiler_options
-co
requests that the options used in the compilation of each object entry be added to file.

Notes

The default library name(s) and search name(s) used for the library_fetch command are defined in the library descriptor for the libraries being searched. 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_fetch
```

The library_descriptor command can also be used to set and reset the name of the default library descriptor. Refer to the writeup on the library_descriptor command for more details.

The following two paragraphs describe the implementation of library_fetch as defined by the standard multics_libraries_ descriptor.

All types of segments in the Multics system libraries may be fetched:

archive components (source or object)
unarchived source or object segments
source or object archives
bind segments (.bind segments)
list segments (.list segments)
executable segments in the online libraries
info segments
peruse_text segments
include segments

Exceptions are executable segments in the hardcore library (system_library_1).

Examples

The command

```
library_fetch if.pl1  -into   >udd>m>user>new_if.=
```

will fetch the source segment if.pl1 into the directory >udd>m>user by the name new_if.pl1.

The command

```
library_fetch *.alm -lib unb.s -lib tools.s
```

will fetch all two-component entries having the suffix ".alm" in the source directories of the libraries unb and tools into the user's working directory.

The command

```
library_fetch *.alm -lib unb -lib tools
```

will accomplish the same thing, since the starname *.alm will only match entries in the source directories of the libraries unb and tools.

The command

```
library_fetch **.info -lib info.*  -into >udd>m>user>==  -nm
```

will fetch all info segments in the library matching the starname info.* into the directory >udd>m>user by their original names, with all additional names attached.

The command

```
lf  bound_runoff_.** -library online.*
```

will fetch all entries which match the starname bound_runoff_.** in the libraries classified as online.*.

The command

```
lf *.bind *.list -lds RDMS_libraries_ -library **
```

will fetch all entries having the suffixes .bind and .list in the set of libraries which match the starname ** in the library descriptor whose reference name is RDMS_libraries_.

(END)