Subject: MCR10104, Default Search Libraries for library_fetch

Author: Gary Dixon

Date: December 8, 2021

Ticket: https://multics-trac.swenson.org/ticket/259

The main function of the library_fetch (lf) command is to locate and extract source files from the Multics Libraries so developers can easily review and/or modify those files. Source files include both source programs written in a specific programming language, and include files that insert declarations and code snippets into such source programs.

The current library_fetch defaults search only the source directories for the online and hardcore libraries; the include directories are not searched. The library_descriptor (lds) command below shows these defaults.

```
library_descriptor default library_fetch

command:  library_fetch
  library names:         online_libraries.source
                         supervisor_library.source
  search names:          (none)
```

Multics Ticket #259 requests that the sss.include directory be added to these default library names.

You may be wondering why only the sss.include library directory should be added. What about include files for the other Multics Libraries?

The Multics Libraries hold software source files, compiled objects from those sources, executable bound and unbound objects, and info segments describing those programs. The libraries are divided into 6 major software categories:

- supervisor (hardcore): software for Multics ring 0 supervisor
- communications (mcs): software for the Multics communications front-end network processor (FNP)
- standard_service (sss): most often used commands, active functions and subroutines software
- tools_library (tools): most administrative, operator-specific and less-used commands, active functions, and subroutine software
- unbundled_library (unb): software for separately-priced products
- obsolete_library (obs): software programs that have been replaced by later Multics programs.

In general, each library category (or just library) includes its own directory holding source programs (source); objects compiled from each source (object); executable bound and unbound segments (execution); listings from source compilations and bind maps (listings).

A library.directory reference often identifies a single directory: hardcore.source identifies only the >ldd>hardcore>source directory.

Some directories are shared by several libraries. There is a single directory containing include files (source declarations and code snippets that are included by several different source programs) known by its most prominent library name (sss.include), but which also has names for the other libraries sharing that directory (e.g., hardcore.include, tools.include, unb.include, etc.).

So, a reference to sss.include causes this shared directory to be included in the default search libraries used by library_fetch, when fetching items from the Multics Libraries.

Proposed Changes

Defaults for the library support tools (library_fetch, library_info, library_print, library_map, library_cleanup, etc.) are stored in the library descriptor file describing a set of libraries. The descriptor for the Multics Libraries is called multics_libraries_.ld. The proposed addition is shown in green text below.

```
Descriptor:                    multics_libraries_;

Define:                        commands;
 …
    command:                   library_fetch;
          library names:        online_libraries.source
                                supervisor_library.source sss.include;
```

A second change is to add a new “log” directory type to hold the special segments maintained by update_seg that describe changes made to the libraries. These special segments are the Installations.(log info lock) segments. Under this proposal, these files would be maintained in a new library directory: >ldd>log.

```
Root:                           (standard_library std sss
                                 languages_library lang
                                 tools_library tools t
                                 supervisor_library sup hardcore hard h
                                 unbundled_library unb
                                 obsolete_library obsolete obs
                                 offline_libraries offline off
                                 online_libraries online on "" ).(log "");
     path:                      >ldd>log;
     search procedure:          multics_library_search_$list_info_dirs
```

This new log directory type allows installers using the update_seg command to reference its log directory via: [library_descriptor pathname *.log]

The build script describing this change is shown below.

```
Description:
Add sss.include to libraries searched by default by library_fetch.

See: https://multics-trac.swenson.org/ticket/259

Installation_directory:  >udd>m>gd>w>ml;

 Build_script:     ml.mb;

 Unbound_obj:      multics_libraries_      IN: tools  REPLACE;
     source:       multics_libraries_.ld    REPLACE            compiler: ldc;
```

Testing

After changing and recompiling the library descriptor file, the resulting descriptor object may be used in the process to test the change. The following library_descriptor (lds) command show selecting the modified descriptor and its defaults for library_fetch.

```
lds set [hd]>bin>multics_libraries_
r 07:47 0.072 12

lds default lf
command:  library_fetch
  library names:          online_libraries.source
     sss.include          supervisor_library.source
  search names:           (none)
```

Documentation

No documentation changes are needed. library_fetch.info advises users to display the default values for their selected descriptor using the library_descriptor command (as shown above in Testing).

Additional Information

For reference, the mbuild request lib_names provides the following list of preferred library.directory names and their associated directory pathname.

```
mbuild:  lib_names

  PREFERRED LIBRARY     ROOT PATHNAME
--------------------  ----------------------------------------
       sss.source        >ldd>sss>source
       sss.object        >ldd>sss>object
      sss.listings       >ldd>listings>sss
     sss.execution       >sss
       unb.source        >ldd>unb>source
       unb.object        >ldd>unb>object
      unb.listings       >ldd>listings>unbundled_1
      unb.listings       >ldd>listings>unbundled_2
     unb.execution       >unb
      tools.source       >ldd>tools>source
```

```
      tools.object       >ldd>tools>object
     tools.listings      >ldd>listings>tools
    tools.execution      >tools
      hard.source        >ldd>hard>source
     hard.execution      >ldd>hard>execution
      hard.object        >ldd>hard>object
     hard.listings       >ldd>listings>hard
         hard.i          >ldd>hard>info
       mcs.source        >ldd>mcs>source
       mcs.object        >ldd>mcs>object
         mcs.i           >ldd>mcs>info
      mcs.listings       >ldd>listings>comm
      sss.include        >ldd>include
        sss.info         >doc>info
       priv.info         >doc>privileged
    accounting.info      >doc>subsystem>accounting
        azm.info         >doc>subsystem>analyze_multics
        bce.info         >doc>subsystem>bce
        dfm.info         >doc>subsystem>deckfile_manager
       dfast.info        >doc>subsystem>dfast
     dial_out.info       >doc>subsystem>dial_out
       emacs.info        >doc>subsystem>emacs
      xforum.info        >doc>subsystem>executive_forum
       xmail.info        >doc>subsystem>executive_mail
       fast.info         >doc>subsystem>fast
       forum.info        >doc>subsystem>forum
     io_daemon.info      >doc>subsystem>io_daemon
      kermit.info        >doc>subsystem>kermit
       linus.info        >doc>subsystem>linus
        rdm.info         >doc>subsystem>mail_system>read_mail
        rdm.info         >doc>subsystem>mail_system>read_mail>forward_requests
        sdm.info         >doc>subsystem>mail_system>send_mail
        sdm.info         >doc>subsystem>mail_system>send_mail>original_requests
    operator.info        >doc>subsystem>operator
  r1_initializer.info    >doc>subsystem>r1_initializer
   report_writer.info    >doc>subsystem>report_writer_info_dirs
       rmdb.info         >doc>subsystem>rmdb
        ssu.info         >doc>subsystem
        am.info          >doc>iis
       obs.source        >ldd>obs>source
       obs.object        >ldd>obs>object
     obs.execution       >obs
        obs.info         >doc>obs
```

The library_descriptor (lds) command can always be used to identify which particular directories are selected by a given library identifier. For example:

```
lds pathname -lb online.source
  >ldd>sss>source
  >ldd>unb>source
  >ldd>tools>source


lds pn offline.source
  >ldd>hard>source
  >ldd>mcs>source
```

```
lds pn sss.include
  >ldd>include
```

library_descriptor can also display all names by which any given directory can be referenced. For example, the sss.include directory can be referenced by any of the following names:

```
lds rt -lb sss.include -nm

standard_library.include                type:  directory
standard_library.incl                   path:  >ldd>include
standard_library             search procedure:  multics_library_search_$list_info_dirs
std.include
std.incl
std
sss.include
sss.incl
sss
languages_library.include        obsolete_library.include
languages_library.incl           obsolete_library.incl
languages_library                obsolete_library
lang.include                     obsolete.include
lang.incl                        obsolete.incl
lang                             obsolete
tools_library.include            obs.include
tools_library.incl               obs.incl
tools_library                    obs
tools.include                    offline_libraries.include
tools.incl                       offline_libraries.incl
tools                            offline_libraries
t.include                        offline.include
t.incl                           offline.incl
t                                offline
supervisor_library.include       off.include
supervisor_library.incl          off.incl
supervisor_library               off
sup.include                      online_libraries.include
sup.incl                         online_libraries.incl
sup                              online_libraries
hardcore.include                 online.include
hardcore.incl                    online.incl
hardcore                         online
hard.include                     on.include
hard.incl                        on.incl
hard                             on
h.include                        include
h.incl                           incl
h
unbundled_library.include
unbundled_library.incl
unbundled_library
unb.include
unb.incl
unb
```

Version History

Date	Revision	Author	Comment
2021-11-27	1.0	Gary Dixon	Initial version of this MCR.
2021-11-29	1.1	Gary Dixon	Additional information suggested by Eric Swenson.
2021-12-08	1.2	Gary Dixon	Add description of new “log” directory type defined in multics_libraries_.ld.