

Building GMT on Mahuika

1. Load all necessary system modules on mahuika

```
ssh mahuika

melody.zhu@mahuika02 ~-> vim machine_env.sh
# content of machine_env.sh
# all needed system modules for building GMT. Maybe overkill but no harm.
module load Python/2.7.14-gimkl-2017a
module load GCCcore/7.1.0
module load FFTW/3.3.5-gimkl-2017a
module load GEOS/3.6.1-gimkl-2017a
module load GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1
module load PROJ/4.9.3-gimkl-2017a
module load HDF/4.2.13-gimkl-2017a
module load HDF5/1.8.18-gimkl-2017a
module load netcdf/gcc/64/4.4.0
module load libpng/1.6.28-gimkl-2017a
module load libjpeg-turbo/1.5.1-gimkl-2017a
module load libreadline/6.3-gimkl-2017a
module load libpmi
module load lapack/gcc/64/3.7.0
module load OpenJPEG/2.3.0-gimkl-2017a
module load KEALib/1.4.6-gimkl-2017a
module load PostgreSQL/9.6.2-gimkl-2017a
module load gcc/4.9.1

# load all modules
melody.zhu@mahuika02 ~-> source machine_env.sh

melody.zhu@mahuika02 ~-> module list
Currently Loaded Modules:
  1) craype-broadwell          6) FFTW/3.3.5-gimkl-2017a      11) KEALib/1.4.6-gimkl-2017a   16) GEOS
/3.5.1-gimkl-2017a            21) XZ/5.2.3-gimkl-2017a      26) HDF/4.2.13-gimkl-2017a
  2) craype-network-infiniband 7) expat/2.2.0-gimkl-2017a    12) LibTIFF/4.0.7-gimkl-2017a 17)
ncurses/6.0-gimkl-2017a      22) libxml2/2.9.4-gimkl-2017a 27) HDF5/1.8.18-gimkl-2017a
  3) slurm/17.11.7            8) libjpeg-turbo/1.5.1-gimkl-2017a 13) PROJ/4.9.3-gimkl-2017a 18)
libreadline/6.3-gimkl-2017a 23) PCRE/8.40-gimkl-2017a     28) netcdf/gcc/64/4.4.0
  4) gimpi/2017a              9) OpenJPEG/2.3.0-gimkl-2017a 14) libgeotiff/1.4.2-gimkl-2017a 19)
SQLite/3.16.2-gimkl-2017a    24) PostgreSQL/9.6.2-gimkl-2017a 29) lapack/gcc/64/3.7.0
  5) gimkl/2017a             10) Szip/2.1-gimkl-2017a      15) FreeXL/1.0.2-gimkl-2017a 20)
libspatialite/4.3.0a-gimkl-2017a-GEOS-3.5.1 25) GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1 30) Python/2.7.14-gimkl-
2017a
  31) gcc/4.9.1
```

2. Download GMT <http://gmt.soest.hawaii.edu/projects/gmt/wiki/BuildingGMT#Building-GMT-from-source>

(1) Get the latest GMT source code from [Github](#)

```
melody.zhu@mahuika02 ~-> git clone https://github.com/GenericMappingTools/gmt.git
```

(2) Get GSHHG (coastlines, rivers, and political boundaries; filename: gshhg-gmt-x.x.x.tar.gz) from the [download page](#) or <ftp://ftp.soest.hawaii.edu/gshhg> and extract the files.

(3) Get DCW (country polygons; filename: dcw-gmt-x.x.x.tar.gz) from the [download page](#) or <ftp://ftp.soest.hawaii.edu/dcw> and extract the files.

3. Build and use custom curl, zlib, libpng instead of mahuika's system modules; install custom Sphinx

```
melody.zhu@mahuika02 ~> mkdir bin
melody.zhu@mahuika02 ~> cd bin
melody.zhu@mahuika02 ~/bin> mkdir curl
melody.zhu@mahuika02 ~/bin> mkdir zlib
melody.zhu@mahuika02 ~/bin> mkdir libpng

melody.zhu@mahuika02 ~/bin> cd ~

# install curl
melody.zhu@mahuika02 ~> git clone https://github.com/curl/curl.git
melody.zhu@mahuika02 ~> cd curl
melody.zhu@mahuika02 ~/curl> vim GIT-INFO
melody.zhu@mahuika02 ~/curl> ./buildconf
melody.zhu@mahuika02 ~/curl> ./configure --prefix=/home/melody.zhu/bin/curl
melody.zhu@mahuika02 ~/curl> make
melody.zhu@mahuika02 ~/curl> make install

# download zlib source code from https://zlib.net/
melody.zhu@mahuika02 ~> tar -xvzf zlib-1.2.11.tar.gz
melody.zhu@mahuika02 ~> cd zlib-1.2.11
melody.zhu@mahuika02 ~/zlib-1.2.11> ./configure --prefix=/home/melody.zhu/bin/zlib
melody.zhu@mahuika02 ~/zlib-1.2.11> make
melody.zhu@mahuika02 ~/zlib-1.2.11> make install

# install libpng
melody.zhu@mahuika02 ~> wget ftp://ftp-osl.osuosl.org/pub/libpng/src/libpng16/libpng-1.6.35.tar.gz
melody.zhu@mahuika02 ~> tar -xvzf libpng-1.6.35.tar.gz
melody.zhu@mahuika02 ~> cd libpng-1.6.35
melody.zhu@mahuika02 ~/libpng-1.6.35> LDFLAGS=-L/home/melody.zhu/bin/zlib/lib LIBS=-lz CPPFLAGS=-I/home/melody.zhu/bin/zlib/include ./configure --prefix=/home/melody.zhu/bin/libpng
melody.zhu@mahuika02 ~/libpng-1.6.35> make check
melody.zhu@mahuika02 ~/libpng-1.6.35> make install

# install Sphinx
melody.zhu@mahuika02 ~> pip install --install-option="--prefix=/home/melody.zhu/bin/" --user Sphinx

# update PATH
melody.zhu@mahuika02 ~> PATH=$PATH:/home/melody.zhu/bin
```

4. Now build GMT

(1) Edit ConfigUser.cmake

```
melody.zhu@mahuika02 ~> cd gmt
melody.zhu@mahuika02 ~/gmt> cp cmake/ConfigUserTemplate.cmake cmake/ConfigUser.cmake
melody.zhu@mahuika02 ~/gmt> vim cmake/ConfigUser.cmake
#
#
# Copyright (c) 1991-2018 by P. Wessel, W. H. F. Smith, R. Scharroo, J. Luis and F. Wobbe
# See LICENSE.TXT file for copying and redistribution conditions.
#
# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by the
```

```

# Free Software Foundation; version 3 or any later version.
#
# This program is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
# FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License
# for more details.
#
# Contact info: gmt.soest.hawaii.edu
# -----
# Use this file to override variables in 'ConfigDefault.cmake' on a per-user
# basis. First copy 'ConfigUserTemplate.cmake' to 'ConfigUser.cmake', then
# edit 'ConfigUser.cmake'. 'ConfigUser.cmake' is not version controlled
# (currently listed in svn:ignore property)
#
# Note: CMake considers an empty string, "FALSE", "OFF", "NO", or any string
# ending in "-NOTFOUND" to be false (this happens to be case-insensitive, so
# "False", "off", "no", and "something-NotFound" are all false). Other values
# are true. Thus it does not matter whether you use TRUE and FALSE, ON and
# OFF, or YES and NO for your booleans.
##
## Section 1: Installation paths
##
# =====
# Basic setup begins here. All settings are optional. In most cases, setting
# CMAKE_INSTALL_PREFIX should be all you need to do in order to build GMT with
# reasonable defaults enabled. Note: If you need to specify directory names
# with spaces (e.g., on Windows) then you must put them in quotes.
# =====
# Installation path (usually defaults to /usr/local) [auto]:
#set (CMAKE_INSTALL_PREFIX "/home/melody.zhu/bin/gmt")
# Set install name suffix used for directories and gmt executables
# [undefined]:
#set (GMT_INSTALL_NAME_SUFFIX "suffix")
# Install into traditional directory structure. Disable to install a
# distribution type directory structure (doc and share separated) [on]:
#set (GMT_INSTALL_TRADITIONAL_FOLDER_NAMES OFF)
# Install convenience links for GMT modules. Disable to install only the main
# gmt program and access modules as "gmt modulename options" [TRUE]:
#set (GMT_INSTALL_MODULE_LINKS FALSE)
# Make executables relocatable on supported platforms (relative RPATH) [FALSE]:
#set (GMT_INSTALL_RELOCATABLE TRUE)
# Exclude optional GDAL, PCRE, PCRE2, FFTW3, LAPACK, BLAS, ZLIB dependencies even if you have them installed
[FALSE]
#set (GMT_EXCLUDE_GDAL TRUE)
#set (GMT_EXCLUDE_PCRE TRUE)
#set (GMT_EXCLUDE_PCRE2 TRUE)
#set (GMT_EXCLUDE_FFTW3 TRUE)
#set (GMT_EXCLUDE_LAPACK TRUE)
#set (GMT_EXCLUDE_BLAS TRUE)
#set (GMT_EXCLUDE_ZLIB TRUE)
# =====
# Advanced configuration begins here. Usually it is not necessary to edit any
# settings below. You should know what you are doing if you do though. Note:
# installation paths are relative to ${CMAKE_INSTALL_PREFIX} unless absolute
# path is given.
# =====
# Set binary installation path [bin]:
#set (GMT_BINDIR "bin")
# Set library installation path [lib or lib64]:
#set (GMT_LIBDIR "lib")
# Set include installation path [include/gmt${GMT_INSTALL_NAME_SUFFIX}]:
#set (GMT_INCLUDEDIR "include/gmt")
# Set share installation path [share or share/gmt${GMT_INSTALL_NAME_SUFFIX}]:
#set (GMT_DATADIR "share/gmt")
# Set doc installation path [share/doc or
# share/doc/gmt${GMT_INSTALL_NAME_SUFFIX}]:
#set (GMT_DOCDIR "share/doc/gmt")
# Set manpage installation path [share/man or
# share/doc/gmt${GMT_INSTALL_NAME_SUFFIX}/man]:
#set (GMT_MANDIR "share/doc/gmt/man")
# Install documentation files from this external location instead of creating

```

```

# new PDF and HTML documents from scratch [${GMT_SOURCE_DIR}/doc_release]:
#set (GMT_INSTALL_EXTERNAL_DOC OFF)
# Install manual pages from this external location instead of creating the
# manpages from scratch [${GMT_SOURCE_DIR}/man_release]:
#set (GMT_INSTALL_EXTERNAL_MAN OFF)
###
## Section 2: Build dependencies (should only be needed if CMake cannot
## automatically detect the rights version or path.)
##
# Set URL to GMT Data server [auto]:
#set (GMT_DATA_URL "data_server_url")
# Set path to GSHHG Shoreline Database [auto]:
set (GSHHG_ROOT "/home/melody.zhu/gshhg-gmt-2.3.7")
# Copy GSHHG files to $/coast [FALSE]:
set (COPY_GSHHG TRUE)
# Set path to DCW Digital Chart of the World for GMT [auto]:
set (DCW_ROOT "/home/melody.zhu/dcw-gmt-1.1.4")
# Copy DCW files to $/dcw [FALSE]:
set (COPY_DCW TRUE)
# FOR WINDOWS ONLY
# Set path to location where the gmtmex is located.
#set (GMTMEX_PATH "C:/progs_cygw/GMTdev/gmtmex/${GMTver}")
# Set location of NetCDF (can be root directory, path to header file or path
# to nc-config) [auto]:
set (NETCDF_ROOT "/cm/shared/apps/netcdf/gcc/64/4.4.0/lib/pkgconfig")
# Set location of GDAL (can be root directory, path to header file or path to
# gdal-config) [auto]:
#set (GDAL_ROOT "gdal_install_prefix")
# Set location of PCRE (can be root directory, path to header file or path to
# pcre-config) [auto]:
#set (PCRE_ROOT "pcre_install_prefix")
# Alternatively, set location of PCRE2 (can be root directory, path to header file or path to
# pcre2-config) [auto]:
#set (PCRE2_ROOT "pcre2_install_prefix")
# Set location of single precision FFTW (can be root directory or path to
# header file) [auto]:
#set (FFTW3_ROOT "fftw_install_prefix")
# Set location of ZLIB (can be root directory or path to header file) [auto]:
set (ZLIB_ROOT "/home/melody.zhu/bin/zlib/lib")
# Set location of CURL (can be root directory or path to header file) [auto]:
set (CURL_DIR "/home/melody.zhu/bin/curl/lib/libcurl.so")
# Set location of GLIB component gthread [auto]. This is an optional (and
# experimental) option which you need to enable:
#set (GMT_USE_THREADS TRUE)
# If pkg-config is not installed (e.g. on Windows) you need to specify these:
#set (GLIB_INCLUDE_DIR c:/path/to/glib-dev/include/glib-2.0)
#set (GLIB_LIBRARIES c:/path/to/glib-dev/lib/glib-2.0.lib)
# Set location of pthreads-win32 (only Windows); needs GMT_USE_THREADS:
#set (PTHREADS_WIN32_ROOT c:/prefix/of/pthreads-win32)
#set (GMT_USE_THREADS TRUE)
# Set LAPACK location. Use this when want to link with LAPACK and it's not found automatically
#set (LAPACK_LIBRARY "V:/lapack-3.5.0/build/lib/liblapack.lib")
#set (BLAS_LIBRARY "V:/lapack-3.5.0/build/lib/libblas.lib")
###
## Section 3: GMT features
##
# Enforce GPL or LGPL conformity. Use this to disable routines that cannot be
# redistributed under the terms of the GPL or LGPL such as Shewchuk's
# triangulation (valid values are GPL, LGPL and off) [off]:
#set (LICENSE_RESTRICTED GPL)
# Allow building of OpenMP if compiler supports it
# set (GMT_ENABLE_OPENMP TRUE)
# Configure default units (possible values are SI and US) [SI]:
#set (UNITS "US")
# Enable building of shared libraries [TRUE] (disable to use static libraries;
# not recommended; on non-x86 architectures uncomment the next option as well):
#set (BUILD_SHARED_LIBS FALSE)
# Create position independent code on all targets [auto] (needed for static
# build on non-x86:
#set (CMAKE_POSITION_INDEPENDENT_CODE TRUE)
# Build GMT shared lib with supplemental modules [TRUE]:

```

```

#set (BUILD_SUPPLEMENTS FALSE)
# Build/Install GMT Developer include files [TRUE]:
# This installs the extra include files and configured files needed by 3rd-party
# developers. Until we build a separate gmt-devel we include them in the main
# Distribution.
#set (BUILD_DEVELOPER FALSE)
##
## Section 4: Advanced tweaking
##
#
# Testing and development
#
# Enable running examples/tests with "ctest" or "make check" (out-of-source).
# Need to set either DO_EXAMPLES, DO_TESTS or both and uncomment the following
# line.
#enable_testing()
#set (DO_EXAMPLES TRUE)
#set (DO_TESTS TRUE)
#set (DO_ANIMATIONS TRUE)
# Auto-convert classic scripts to modern during testing:
#set (MODERNIZE_TESTS TRUE)
# Number of parallel test jobs with "make check":
#set (N_TEST_JOBS 4)
# Location of optional third-party files used by test suite available from
# svn://gmtserver.soest.hawaii.edu/gmt-data [${GMT_SOURCE_DIR}/test/data]
#set (GMT_TEST_DATA "test_data_path")
# Enable this option to run GMT programs from within ${GMT_BINARY_DIR} without
# installing or setting GMT_SHAREDIR and GMT_USERDIR first. This is required
# for testing [OFF]:
#set (SUPPORT_EXEC_IN_BINARY_DIR ON)
# List extra sub-dirs of 'src' with a CMakeList.txt to build non-module codes
# that link against the full gmt libs (not just the API; for building codes
# that only need the GMT API, see the gmt-custom project).
#set (EXTRA_BUILD_DIRS apidemo)
#set (DO_API_TESTS ON)
# Directory in which to install the release sources per default
# [${GMT_BINARY_DIR}/gmt-${GMT_PACKAGE_VERSION}]:
#set (GMT_RELEASE_PREFIX "release-src-prefix")
# If set to false, image conversion from PS images to PNG and PDF does
# not depend on the gmt binary target. Note: "make gmt" is then required
# before docs_depends [TRUE].
#set (GMT_DOCS_DEPEND_ON_GMT FALSE)
#
# Debugging
#
# Set build type can be: empty, Debug, Release, RelWithDebInfo or MinSizeRel
# [Release]:
#set (CMAKE_BUILD_TYPE Debug)
# Extra debugging for developers:
#if ( CMAKE_GENERATOR STREQUAL "Xcode" )
##      So Xcode can find the supplemental plug-ins during debug sessions
#      add_definitions(-DXCODER)
#      add_definitions(-DDEBUG_MODERN)          # To set PPID == 0 during Xcode test
#      message("Add Xcode definition for GMT")
#endif()
#add_definitions(-DDEBUG)
#add_definitions(-DMEMDEBUG) # Turn on memory tracking see gmt_support.c for extra info
#set (CMAKE_C_FLAGS "-Wall -Wdeclaration-after-statement") # recommended even for release build
#set (CMAKE_C_FLAGS "-Wextra ${CMAKE_C_FLAGS}")           # extra warnings
#set (CMAKE_C_FLAGS_DEBUG -ggdb3)                        # gdb debugging symbols
#set (CMAKE_LINK_DEPENDS_DEBUG_MODE TRUE)                # debug link dependencies
if (HAVE_OPENMP)
    set (CMAKE_C_FLAGS_RELEASE "-ggdb3 -O2 -Wuninitialized -flax-vector-conversions") # check
uninitialized variables
else (HAVE_OPENMP)
    set (CMAKE_C_FLAGS_RELEASE "-ggdb3 -O2 -Wuninitialized") # check uninitialized variables
endif (HAVE_OPENMP)
#
# System specific tweaks
#
# This is for GCC on Solaris to avoid "relocations remain against allocatable

```

```
# but non-writable sections" problems:
#set (USER_GMTLIB_LINK_FLAGS -mimpure-text)
# This may be needed to enable strdup and extended math functions with GCC and
# Suncc on Solaris:
#set (CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -D__EXTENSIONS__")
# Do not warn when building with Windows SDK or Visual Studio Express:
#set (CMAKE_INSTALL_SYSTEM_RUNTIME_LIBS_NO_WARNINGS ON)
# Manually select runtime library when compiling with Windows SDK or Visual
# Studio Express:
#set (CMAKE_INSTALL_SYSTEM_RUNTIME_LIBS c:/Windows/System32/msvcr100.dll)
# If your NetCDF library is static (not recommended, applies to Windows only)
#set (NETCDF_STATIC TRUE)
# If want to rename the DLLs to something else than the default (e.g. to
# append the bitness - Windows only)
# WARNING: if using this option it is mandatory that the suffix starts with an underscore.
#if (WIN32)
# set (BITAGE 32)
# # Detect if we are building a 32 or 64 bits version
# if (CMAKE_SIZEOF_VOID_P EQUAL 8)
#   set (BITAGE 64)
# endif ()
# set (GMT_DLL_RENAME gmt_w${BITAGE})
# set (PSL_DLL_RENAME psl_w${BITAGE})
#endif(WIN32)
# On Windows Visual C 2012 needs _ALLOW_KEYWORD_MACROS to build
#if(MSVC11)
#   add_definitions(/D_ALLOW_KEYWORD_MACROS)
#endif(MSVC11)
# vim: textwidth=78 noexpandtab tabstop=2 softtabstop=2 shiftwidth=2
```

(2) known errors and solutions

```
melody.zhu@mahuika02 ~/gmt> mkdir build
melody.zhu@mahuika02 ~/gmt> cd build
melody.zhu@mahuika02 ~/gmt/build> cmake ..
```

Error:

cannot find CURL_INCLUDE_DIR error

Solution:

check if you install your custom curl

edit ccmake..

```
melody.zhu@mahuika02 ~/gmt/build> ccmake ..
# hardcode the path for following variables. Enforce using custom curl and zlib. Editing ConfigUser.cmake will
not solve the issue.
CURL_INCLUDE_DIR                /home/melody.zhu/bin/curl
/include

CURL_LIBRARY                    /home/melody.zhu/bin/curl/lib/libcurl.so

ZLIB_INCLUDE_DIR                /home/melody.zhu/bin/zlib
/include

ZLIB_LIBRARY                    /home/melody.zhu/bin/zlib/lib/libz.so
```

Error: /cm/shared/apps/intel/compilers_and_libraries/2017.6.256/linux/mkl/lib/intel64/libmkl_gf_lp64.so: error: undefined reference to 'mkl_lapack_*

Solution:

Remove mkl module

```
melody.zhu@mahuika02 ~/gmt/build> module rm imkl/2017.6.256-gimpi-2017a
```

edit link.txt manually and remove lmkl_gf_lp64 flag

```
$find . -name "link.txt" | xargs grep lmkl_gf_lp64 | cut -d: -f1 | xargs -I % sed -i -e 's?-lmkl_gf_lp64??g' %
```

Error:

```
undefined reference to 'dsyev_'
```

this error often happens when LAPACK is not linked properly.

Solution:

edit the link.txt for gmt and manually add in the liblapack.so

NOTE: the lapack modules provided in modules on Mahuika is built using f77 instead of f90. using this module may result in crashes. try to use the one located in /usr/lib64 first.

see below for an example.

```
$vim src/CMakeFiles/gmt.dir/link.txt

/opt/nesi/mahuika/GCCcore/5.4.0/bin/cc -std=gnu99 -ggdb3 -O2 -Wuninitialized CMakeFiles/gmt.dir/gmt.c.o -
o gmt -rdynamic libgmt.so.5.4.4 /cm/shared/apps/netcdf/gcc/64/4.4.0/lib/libnetcdf.so /nesi/project/nesi00213/opt
/mahuika/curl/7.63.0-DEV-gcc-5.4.0/lib/libcurl.so /usr/lib64/liblapack.so.3.4.2 -lgdal -lpcres -lfftw3f -
lfftw3f_threads -lblas -lpthread -lm libpostscriptlight.so.5.4.4 /nesi/project/nesi00213/opt/mahuika/zlib/1.
2.11/lib/libz.so -lm -ldl -Wl,-rpath,/home/ykh22/deploy/gmt/build/src:/cm/shared/apps/netcdf/gcc/64/4.4.0/lib:
/nesi/project/nesi00213/opt/mahuika/curl/7.63.0-DEV-gcc-5.4.0/lib:/nesi/project/nesi00213/opt/mahuika/zlib/1.
2.11/lib:
```

Error:

```
/opt/nesi/mahuika/binutils/2.28-GCCcore-7.1.0/bin/ld: warning: libopenjp2.so.7, needed by /opt/nesi/mahuika/GDAL
/2.2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so, not found (try using -rpath or -rpath-link)
/opt/nesi/mahuika/binutils/2.28-GCCcore-7.1.0/bin/ld: warning: libpq.so.5, needed by /opt/nesi/mahuika/GDAL/2.
2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so, not found (try using -rpath or -rpath-link)
/opt/nesi/mahuika/binutils/2.28-GCCcore-7.1.0/bin/ld: warning: libkea.so.1.4.6, needed by /opt/nesi/mahuika/GDAL
/2.2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so, not found (try using -rpath or -rpath-link)
/opt/nesi/mahuika/GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so: undefined reference to `kealib::KEAImageIO::
setImageBandClrInterp(unsigned int, kealib::KEABandClrInterp)'
/opt/nesi/mahuika/GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so: undefined reference to `PQputCopyEnd'
/opt/nesi/mahuika/GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1/lib/libgdal.so: undefined reference to `opj_create_compress'
```

Solution:

Check modules:OpenJPEG/2.3.0-gimkl-2017a, KEALib/1.4.6-gimkl-2017a and PostgreSQL/9.6.2-gimkl-2017a are loaded.

Check the above modules are in PATH; Their libs are in LD_LIBRARY_PATH

```
melody.zhu@mahuika02 /scale_wlg_persistent/filesets/project/nesi00213/share> module show KEALib/1.4.6-gimkl-
2017a
prepend_path("LD_LIBRARY_PATH", "/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/lib")
prepend_path("LIBRARY_PATH", "/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/lib")
prepend_path("PATH", "/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/bin")

# Add missing libs to LD_LIBRARY_PATH: (use 'module show' to see each module path)
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/nesi/mahuika/PostgreSQL/9.6.2-gimkl-2017a/lib
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/nesi/mahuika/OpenJPEG/2.3.0-gimkl-2017a/lib
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/lib
```

Error:

```
Linking C executable gmt /opt/nesi/mahuika/libpng/1.6.28-gimkl-2017a/lib/libpng16.so.16: undefined reference to
`inflateValidate@ZLIB_1.2.9
```

Solution:

check if you install your custom libpng using your custom zlib (see point 3)

check if you have the correct ZLIB_INCLUDE_DIR and ZLIB_LIBRARY in cmake
check src/CMakeFiles/gmt.dir/link.txt and edit so cmake can find libs for your custom curl and zlib

```
melody.zhu@mahuika02 ~/gmt/build> vim src/CMakeFiles/gmt.dir/link.txt
/usr/bin/cc -std=gnu99 -ggdb3 -O2 -Wuninitialized CMakeFiles/gmt.dir/gmt.c.o -o gmt -rdynamic libgmt.so.
6.0.0 /cm/shared/apps/netcdf/gcc/64/4.4.0/lib/libnetcdf.so /home/melody.zhu/bin/curl/lib/libcurl.so -lgdal -
lpcr -lfftw3f -lfftw3f_threads -lblas libpostscriptlight.so.6.0.0 /home/melody.zhu/bin/zlib/lib/libz.so -lm -
ldl -Wl,-rpath,/home/melody.zhu/gmt/build/src:/cm/shared/apps/netcdf/gcc/64/4.4.0/lib:/home/melody.zhu/bin/curl
/lib:/home/melody.zhu/bin/zlib/lib
```

When no error running cmake ..

Do

```
melody.zhu@mahuika02 ~/gmt/build> cmake ..
melody.zhu@mahuika02 ~/gmt/build> make
melody.zhu@mahuika02 ~/gmt/build> make install DESTDIR=/home/melody.zhu/bin
```

GMT will be installed to /home/melody.zhu/bin/usr/local/bin

Now add GMT to PATH and LD_LIBRARY_PATH

```
melody.zhu@mahuika02 ~/gmt> PATH=$PATH:/home/melody.zhu/bin/usr/local/bin
melody.zhu@mahuika02 ~/gmt> LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/melody.zhu/bin/usr/local/lib64
```

(3) Test

```
melody.zhu@mahuika02 ~-> gmt
GMT - The Generic Mapping Tools, Version 6.0.0_a7d5353-dirty [64-bit] [72 cores]
(c) 1991-2018 Paul Wessel, Walter H. F. Smith, Remko Scharroo, Joaquim Luis, and Florian Wobbe
Supported in part by the US National Science Foundation (http://www.nsf.gov/)
and volunteers from around the world (see http://gmt.soest.hawaii.edu/).
This program comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of this program under the terms of the
GNU Lesser General Public License (http://www.gnu.org/licenses/lgpl.html).
For more information about legal matters, see the file named LICENSE.TXT.
usage: gmt [options]
      gmt <module name> [<module-options>]
session management:
gmt begin      Initiate a new GMT session using modern mode [classic].
gmt end        Terminate the current GMT modern mode session.
gmt docs       Display the HTML documentation for selected module.
gmt figure     Set figure format specifics under a GMT modern mode session.
gmt subplot    Initiate a multi-panel figure.
gmt clear all | cache | cpt | conf | data | history | sessions
               Delete gmt.history, gmt.conf, session CPT, the user cache or data dir, or all of them.
options:
--help         List descriptions of available GMT modules.
--show-bindir  Show directory with GMT executables.
```

```

--show-cores      Print number of available cores.
--show-datadir    Show directory/ies with user data.
--show-modules    List all module names.
--show-library    Show path of the shared GMT library.
--show-plugindir  Show directory for plug-ins.
--show-sharedir   Show directory for shared GMT resources.
--version         Print GMT version number.
if <module-options> is '=' we call exit (0) if module exist and non-zero otherwise.

```

Sample PATHs that will correctly compile GMT

```

PATH=/opt/nesi/mahuika/GCCcore/7.1.0/bin:/opt/nesi/mahuika/GEOS/3.6.1-gimkl-2017a/bin:/opt/nesi/mahuika/GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1/bin:/opt/gcc/4.9.1/bin:/opt/slurm/17.11.7/sbin:/opt/slurm/17.11.7/bin:/opt/nesi/mahuika/libreadline/6.3-gimkl-2017a/bin:/opt/nesi/mahuika/libpng/1.6.28-gimkl-2017a/bin:/cm/shared/apps/netcdf/gcc/64/4.4.0/bin:/opt/nesi/mahuika/HDF5/1.8.18-gimkl-2017a/bin:/opt/nesi/mahuika/HDF/4.2.13-gimkl-2017a/bin:/opt/nesi/mahuika/PROJ/4.9.3-gimkl-2017a/bin:/opt/nesi/mahuika/FFTW/3.3.5-gimkl-2017a/bin:/opt/nesi/mahuika/binutils/2.28-GCCcore-7.1.0/bin:/opt/nesi/mahuika/Python/2.7.14-gimkl-2017a/bin:/opt/nesi/mahuika/Tk/8.6.6-gimkl-2017a/bin:/opt/nesi/mahuika/Tcl/8.6.6-gimkl-2017a/bin:/opt/nesi/mahuika/SuiteSparse/4.5.4-gimkl-2017a/bin:/opt/nesi/mahuika/METIS/5.1.0-gimkl-2017a/bin:/opt/nesi/mahuika/netCDF/4.4.1-gimkl-2017a/bin:/opt/nesi/mahuika/cURL/7.52.1-gimkl-2017a/bin:/opt/nesi/mahuika/LLVM/4.0.0-gimkl-2017a/bin:/opt/nesi/mahuika/libxslt/1.1.29-gimkl-2017a/bin:/opt/nesi/mahuika/freetype/2.7.1-gimkl-2017a/bin:/opt/nesi/mahuika/bzip2/1.0.6-gimkl-2017a/bin:/opt/nesi/mahuika/PostgreSQL/9.6.2-gimkl-2017a/bin:/opt/nesi/mahuika/PCRE/8.40-gimkl-2017a/bin:/opt/nesi/mahuika/libxml2/2.9.4-gimkl-2017a/bin:/opt/nesi/mahuika/XZ/5.2.3-gimkl-2017a/bin:/opt/nesi/mahuika/SQLite/3.16.2-gimkl-2017a/bin:/opt/nesi/mahuika/ncurses/6.0-gimkl-2017a/bin:/opt/nesi/mahuika/libgeotiff/1.4.2-gimkl-2017a/bin:/opt/nesi/mahuika/LibTIFF/4.0.7-gimkl-2017a/bin:/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/bin:/opt/nesi/mahuika/OpenJPEG/2.3.0-gimkl-2017a/bin:/opt/nesi/mahuika/libjpeg-turbo/1.5.1-gimkl-2017a/bin:/opt/nesi/mahuika/expat/2.2.0-gimkl-2017a/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/ibutils/bin:/opt/nesi/bin:/home/melody.zhu/bin:/home/melody.zhu/bin/usr/local/bin:/scale_wlg_persistent/filesets/project/nesi00213/opt/mahuika:/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/bin:/scale_wlg_persistent/filesets/project/nesi00213/opt/mahuika/gmt/6.0.0/usr/local/bin

```

```

LD_LIBRARY_PATH=/opt/nesi/mahuika/GCCcore/7.1.0/lib/gcc/x86_64-pc-linux-gnu/7.1.0:/opt/nesi/mahuika/GCCcore/7.1.0/lib64:/opt/nesi/mahuika/GCCcore/7.1.0/lib:/opt/nesi/mahuika/GEOS/3.6.1-gimkl-2017a/lib:/opt/nesi/mahuika/GDAL/2.2.2-gimkl-2017a-GEOS-3.5.1/lib:/opt/gcc/4.9.1/snos/lib64:/opt/cray/pe/lib64:/opt/cray/pe/lib64/cce:/opt/cray/lib64:/opt/slurm/17.11.7/lib64/slurm:/opt/slurm/17.11.7/lib64:/opt/nesi/mahuika/KEALib/1.4.6-gimkl-2017a/lib:/opt/nesi/mahuika/OpenJPEG/2.3.0-gimkl-2017a/lib:/opt/nesi/mahuika/PostgreSQL/9.6.2-gimkl-2017a/lib:/scale_wlg_persistent/filesets/project/nesi00213/opt/mahuika/gmt/6.0.0/usr/local/lib64

```