Building MARS with CMake

Version support

MARS Version >= 7.0 - see Releases

Requirements

System

To **run** MARS you need your system to be and have:

- UNIX operating system with POSIX support
- POSIX threads

To **build** MARS from source, you need additionally:

- C/C++ compiler
- perl5
- lex & yacc (or flex & bison)

Software

Additionally you also need:

- CMake >= 2.8 - www.cmake.org (it may work with =2.6.4, but is untested)
- curses library - www.gnu.org/software/ncurses
- Grib API >=1.9.9 - GRIB API

Grib API build

Grib API (aka **grib_api**) must have been compiled with "-fPIC".

Building Grib API yourself

If the **grib_api** is not already installed in your system, please, download a version from the ECMWF web site, untar it, and follow the instructions for compilation and installation.

As a quick reference, here is how you can do it:

build grib_api

```
./configure --prefix=/path/to/install/location CFLAGS='-fPIC' --disable-jpeg --disable-fortran
make
make install
```

Optional

If you don't have these you will still be able to build MARS:

- GNU Readline library, www.gnu.org/s/readline, enhances user interaction.

Tape Management Systems

You may need or use one of the following tape management systems:

- HPSS
- ADSM (now called Tivoli Storage Manager, or TSM)
- HSM - a general hierchical storage manager where tapes are not directly accessed and data is written directly to disk.

Helper install dependencies script

In the scripts directory there is a script `cookup.pl`, see github, which may be used to automate the installation of external dependencies.

This script is provided *"as-is*" and is not maintained by **ECMWF**.

This will download, build and install `cmake` and `grib_api` as well as some other packages that you might require.

cookup

```
cd scripts
./cookup.pl --cook --verbose --package=cmake,grib_api --prefix=/path/to/install/location
--sandbox=/tmp
```

Building

1. Untar the source package (this creates a dir named MARS-YYYY.MM.DD-Source)

   ```
   tar zxvf MARS-2012.01.06-Source.tar.gz
   ```

2. Create a ***build directory*** and go into it.

   ```
   mkdir -p build/Production
   cd build/Production
   ```

3. Define the environment variable where MARS will run (optional but recommended).

   ```
   export DHSHOME=/path/to/dhshome
   ```

4. Configure the build system with your options.

   No option is mandatory to define. If no paths are defined, then software dependencies will be searched on standard system path locations (`/usr /usr/local`).

   ```
   cmake ../.. -DCMAKE_BUILD_TYPE=Production -DGRIB_API_PATH=/path/to/grib_api [options...]
   ```

5. Build the software

   ```
   make
   ```

6. Install links in the `$DHSHOME` to the built binaries and libraries.
   You can also run mars from the build directory `/bin` and `/lib` dirs.

   ```
   make links
   ```

7. **[Optionally]** Install MARS into the system (default `/usr/local`).

   ```
   make install
   ```

CMake Options

CMake Generic Options

-DCMAKE_BUILD_TYPE=[Release|Debug|RelWithDebInfo]

Defines the build type (default it RelWithDebInfo)

-DCMAKE_PREFIX_PATH=/path/to/package1:/path/to/package2

Defines a list of paths where to search packages that are installed in non-standard places.

For example: if Jasper is installed in **/usr/local/lib/metaps/lib/grib_api/jasper** instead of **/usr**.

-DCMAKE_INSTALL_PREFIX=/path/to/installation/dir

Path where to install the software (default: **/usr/local**).

In principle, this should be the **$DHSHOME**.

Binaries will be in **$DHSHOME/bin** and libs in **$DHSHOME/lib**.

-DCMAKE_C_COMPILER=gcc

Which C compiler to use.

By default it tries to find the system default (will usually find **cc** or **gcc**).

-DCMAKE_CXX_COMPILER=g++

Which C compiler to use.

By default it tries to find the system default (will usually find **c++** or **g++**).

MARS Specific Options

-DDHSHOME=/path/to/dhshome

Either passed as an option directly to cmake or defined as an environment variable, **DHSHOME** should define where MARS will run.

-DHPSS_PATH=/path/to/hpss

Path where to find the HPSS tape manager library. If not passed, it might still find it because it always searches in the standard system locations (**/usr**...).

-DADSM_PATH=/path/to/adms

Path where to find the ADSM/TSMtape manager library. If not passed, it might still find it because it always searches in the standard system locations (**/usr**...).

-DWITH_READLINE=ON

Use the GNU readline library to hold command history and line navigation.

-DREADLINE_PATH=/path/to/readline

In case it does not find the Readline library in the standard system locations (**/lib** or **/usr/lib**), use this variable to define the path directly.