CMake Scripts for Building Applications

Overview

This page provides information on using CMake to build applications with simple scripts:

- How to Build an Application with CMake
- Detailed Information on the Steps
- Simple Scripts

How to Build an Application with CMake

These are the basic steps to build an application with a CMake script and the HDF5 libraries built with CMake. More information on these steps is provided after.

Steps

1. *Build HDF5 with CMake according to the instructions on the Building HDF5 with CMake page.* (Set options in `HDF5options.cmake` as needed.)
2. Set the HDF5_DIR environment variable to the location of the `hdf5-config.cmake` file in the `cmake` directory of the HDF5 built binaries.
3. Create a source directory and put your script (`CMakeLists.txt`) and application in it.
4. Create a build directory and build the application:
   ```
   cmake [ -G <generator> ] <path>/<your source directory>
   cmake --build . --config <Release|Debug>
   ```
5. Run the application `runapp`.

Detailed Information on the Steps to Build an Application

Set the HDF5_DIR environment variable

Set the HDF5_DIR environment variable to the location of the CMake files in the built HDF5 binary distribution. They are located in the `share/cmake/hdf5/` directory on Unix and in the `cmake` directory on Windows.

Unix:

The HDF5-1.N.N-Linux.tar.gz file is created in the CMake-hdf5-1.N.N directory when HDF5 is built on Linux with CMake. Uncompress the file. It contains the `share/cmake/hdf5/` directory. From a command line, set the HDF5_DIR environment variable to the location of this directory:

```
  setenv HDF5_DIR <mypath>/HDF5-1.N.N-Linux/HDF_Group/HDF5/1.N.N/share/cmake/hdf5)
```

Windows 64-bit:

The HDF5-1.N.N-win64.zip is created in the CMake-hdf5-1.N.N directory when HDF5 is built on Windows 64-bit. Uncompress the file. It contains a directory with cmake files in `cmake\`. Set the HDF5_DIR environment variable to the location of this directory.

Setting an Environment Variable on Windows 7:

Left click on Start at the bottom left and then right-click on Computer and select Properties from the menu that pops up. The Control Panel Home window opens. Select "Advanced system settings" on the left to bring up the "System Properties". Select the "Environment Variables" button near the bottom, from which you can add User and System variables.

Create a source directory for the build script and application

1. Create the directory: `mkdir` *source*
2. Place the build script (`CMakeLists.txt`) and the application in the source directory.
   The Scripts use `find_package`, which returns both the HDF5 libraries and properties.

Create a build directory and build and run the script

1. Create the directory: `mkdir` *`build`*
2. Go into the build directory: `cd` *`build`*
3. Run cmake with the path to the source directory:
   On Unix: `cmake <path>/`*`source`*
   On WIndows include the generator. For example: `cmake -G "Visual Studio 12 2013 Win64" <path>/`*`source`*

4. Build the application: `cmake --build . --config <Release|Debug>`
5. Run the created executable:
   Unix: `runapp`
   Windows:
   ```
   cd Release
   .\runapp.exe
   ```

Change the script (optional)

NOTE that you can change the script to include the HDF5_DIR environment variable.

Also note that if you have multiple versions of HDF5 on your machine, then you must specify the HDF5 version in the `find_package` call:

```
find_package (HDF5 1.10.5 NAMES "hdf5" COMPONENTS ${FIND_HDF_COMPONENTS})
```

Scripts

Following are simple scripts for compiling applications. The scripts can be used for other languages or libraries (in parenthesis), by editing the script and changing the application to build near the bottom.

C, shared (and C++)

- The README.txt file describes how to build shared and release HDF5 binaries and how to compile an application.
- The CMakeLists.txt file is a script for compiling C and C++ shared libraries.
- The application, h5_write.c, is an application taken from the C examples in the HDF5 source code.

Fortran, shared (and C, C++)

- The README.txt file describes how to build shared and release HDF5 binaries and how to compile an application.
- The CMakeLists.txt file is a script for compiling C, C++ and Fortran shared libraries.
- The application, rwdset_fortran2003.f90, is an application taken from the FORTRAN examples in the HDF5 source code.

C++, debug, static (and C, HL, HL C++)

- The README.txt file describes how to build HDF5 with static and debug and how to compile an application.
- The CMakeLists.txt file is a script for compiling C, C++, HL, and HL C++ static libraries.
- The application, writedata.cpp, is an application taken from the C++ examples in the HDF5 source code.