

Building HDF4 with CMake

Overview

A CTest script and platform configuration file can be used to simplify building with CMake. The script below will build HDF4 with SZIP, ZLIB, and the JPEG external libraries.

- [Preconditions](#)
- [Build Instructions](#)
- [Troubleshooting](#)

Preconditions:

- **CMake MUST** be installed. The minimum required version is CMake 3.10 and the recommended version is 3.15. CMake 3.15 is *required* for VS 2019.
- Blank spaces **MUST NOT** be used in directory path names as this will cause the build to fail.
- (Windows) **NSIS** or **WiX** should be installed in order to create an install image with CPack. NSIS will create a .exe installer. WiX will create a .msi installer.

Visual Studio Express users will not be able to package HDF into an install image executable. Turn off packaging by doing the following:

- Edit `HDF4options.cmake`.
- Add the line: `set(ADD_BUILD_OPTIONS "${ADD_BUILD_OPTIONS} -DNO_LOCAL_PACKAGE:BOOL=ON")`
- Uncomment the line: `set(ADD_BUILD_OPTIONS "${ADD_BUILD_OPTIONS} -DHDF4_NO_PACKAGES:BOOL=ON")`
- Comment out the line: `set(ADD_BUILD_OPTIONS "${ADD_BUILD_OPTIONS} -DHDF4_PACKAGE_EXTLIBS:BOOL=ON")`

Build Instructions

1. Create a working directory.
2. Uncompress the HDF4 CMake source code file into the working directory. It will contain a `CMake-hdf-N` directory (where N is the release version).
3. From the [command line](#), go into the `CMake-hdf-N` directory, which contains:

<code>build*.sh (.bat)</code>	Build Script(s)
<code>CTestScript.cmake</code>	ctest Command
<code>hdf-N/</code>	HDF4 Source Code
<code>HDF4config.cmake</code>	Configuration file
<code>HDF4options.cmake</code>	User modifiable options
<code>JPEG8d.tar.gz</code>	External library for JPEG Compression
<code>SZip.tar.gz</code>	External library for SZIP Compression
<code>ZLib.tar.gz</code>	External library for ZLIB Compression

4. By default HDF4 will be built:
 - Without Fortran
 - With SZIP compression enabled
 - With ZLIB compression enabled
 - With static libraries enabled

Users can change the options that HDF4 is built with by adding options to the build command (see the batch files/test script below) or by modifying the `HDF4options.cmake` file. The `HDF4options.cmake` file will override any options set in the configuration file. For more information see the [How to Change HDF4 CMake Build Options](#) page.

REQUIRED: Visual Studio Express users must change the build options to turn off packaging or the build will fail.

5. Execute the batch file or shell script containing the ctest command for your platform. (See [Troubleshooting](#) if you do not see your platform).

Example build files are shown below:

Platform	Batch File / Shell Script	ctest command
Windows 64-bit VS 2015	build-VS2015-64.bat	ctest -S HDF4config.cmake,BUILD_GENERATOR=VS201564 -C Release -V -O hdf4.log
Windows 32-bit VS 2015	build-VS2015-32.bat	ctest -S HDF4config.cmake,BUILD_GENERATOR=VS2015 -C Release -V -O hdf4.log
Unix	build-unix.sh	ctest -S HDF4config.cmake,BUILD_GENERATOR=Unix -C Release -V -O hdf4.log

Where the ctest command is using these options:

- The -S option uses the script version of ctest.
- The -C option specifies the build configuration which matches CTEST_BUILD_CONFIGURATION in the configuration file.
- The -V option indicates verbose. **-VV option indicates more verbose.**
- The -O option saves the output to a log file, *hdf4.log*.

6. Locate the built binary.

The built binary will be placed in the build directory and will have the format:

`hdf-4.2.N-<platform.<zip or tar.gz>`

On Windows, a .msi or .exe file will also be created, depending on the installer. On Unix, an additional .sh file will be created.

If the built binary is not there, then see [Troubleshooting](#) for help.

7. Check what is included with your built binaries

You will find the `libhdf4.settings` file in the build directory. It contains information on how the binaries were built.

Troubleshooting

My platform/compiler is not included. Can I still use the configuration files?

Yes, you can but you will have to edit the `HDF4config.cmake` file and update the variable:

```
CTEST_CMAKE_GENERATOR
```

Other variables may be updated for informational purposes but are not required (for example, `SITE_OS_BITS`).

The generators for your platform can be seen by typing:

```
cmake --help
```

What do I do if the build fails?

If the build works properly, then you will find the built and compressed binary in your `CMake-hdf-4.N` directory. If you do not find it then check the log files. You will find those in the build directory under `CMake-hdf-4.N`. For example, on Unix the log files will be in:

```
CMake-hdf-4.N/build/Testing/Temporary/
```

There are log files for the configure, test, and build.

What if the binary does not get packaged properly?

If the binary did not get packaged properly, look at the `cpack.log` file in the build directory under `CMake-hdf-4.N`. For example, on Unix look here:

CMake-hdf-4.N/build/cpack.log

What do I do if I need to rebuild the software?

If you have to rebuild HDF4, remove the build directory first.

The library was built but there are no binaries. What do I do?

To install or package the binaries, run either `make install` or `cpack` in the `build/` directory.