

HDF5-5881: dual binary releases and cmake changes

Change how the CMake files build the libraries of all HDF products.

Currently, the HDF CMake implementation builds either all static or all dynamic libraries and tools. At product release time, two release binaries are created for each platform and compiler we support. Each release binary is independent and cannot be installed at the same time without considerable effort. The support required for building and testing the release binaries could be significantly reduced and reduce the complexity of the windows products for users.

The proposal is to build the libraries both statically and dynamically for all HDF products and external projects, which will require only one release binary to be created for each platform and compiler. In addition, any distributable tools and utilities always will be built statically. The packaging will then include support for the user to choose the component and build type required, using CMake configuration files distributed within the binaries. The improvements to building binaries with CMake will address long term issues with the naming of dynamic executables on Windows, the inability to install more than one binary package at a time, the inconsistent process with respect to the Autotools packaging, and user support.

Most of the work for these changes will be in the documentation for building from source and there will be some documentation changes for using the libraries. The simplification of the web pages will help the users with download options.

Proposal

New and improved CMake 3 features enabled this proposal by improving the support for import libraries and packaging. More properties have been added to the exported library configuration which describe how the libraries were built and how they can be linked. The `find_package` function has improved support for components for use by application builders. The HDF5 1.8.15 release introduced the concept of components into the binary configuration. By default, CMake `find_package` command will return the C and HL libraries to the calling application CMake build. Other libraries can be added to the link process by adding the `COMPONENTS` option to the `find_package` command. The valid components are:

- C
- CXX
- Fortran
- HL
- CXX_HL
- Fortran_HL
- Tools

The dual binary changes will add static and shared to the component list, with static libraries as the default. Future additions to the component list could be 32/64 bit and debug/release selections. The distributable tools will always be built statically so that they are independent of the location of the library. Therefore the static libraries will always be built by default and building the shared libraries will be optional with the use of `"BUILD_SHARED_LIBS"` option, by default this will be "ON", the user can disable building shared/dynamic libraries by setting this option to "OFF".

Non-distributable test executables will need to be built shared as well as static if `"BUILD_SHARED_LIBS"` is "ON". This will then require the tests to execute both as static and as shared. In order to keep the results separated, each set of tests should be executed in a subfolder of the build test folder.

Changes needed

Eliminate the `"lib_type"` variable defined by the CMake flag of `"BUILD_SHARED_LIBS"`. `"BUILD_SHARED_LIBS"`

now indicates if dynamic libraries should be built in addition to static. If shared libraries are built only tests are built shared in addition to static tests. All distributable tools and utilities are only built static, this will simplify the import process in user projects as well. All distributable libraries are built static and only built dynamic if "BUILD_SHARED_LIBS" is "ON". On Windows, the link time libraries will be installed into a subfolder of the lib directory, with the run-time "dll" libraries installed into the "bin" directory. Other platforms will install both the shared and static libraries into the lib directory.

CMake binary configuration files must support component selection in the "find_package" call, this includes static or shared as well. For HDF5, the components will be; static, shared, C, CXX, Fortran, HL, CXX_HL, Fortran_HL, Tools. With the default being just the static C and HL library. All CMake targets (not actual filenames) in the CMake files must be appended with "-static" or "-shared" with the actual filename created by the use of the HDF function, "TARGET_NAMING" for executables and "SET_LIB_OPTIONS" for libraries.

Eliminate any include file definitions concerning static or shared state. This includes the "H5_BUILT_AS_{*}_LIB" defines in the H5pubconf.h generated file. Only the "H5_BUILT_AS_DYNAMIC_LIB" define is used and is moved to the interface properties of the imported library and is only used by the H5_adpt_api.h file for the import/export decorations. This H5_adpt_api.h file is cleaned of other defines and relies on the {library}_shared_EXPORTS definition for selecting the correct import/export decoration.

Users who do not use CMake to build applications or libraries based on HDF libraries, will need to define "H5_BUILT_AS_DYNAMIC_LIB" when building dynamically. (Users on WINDOWS must define this to set the correct import properties when using the HDF5 library)

CMake support

CMake 3.1+ has enhanced the interface properties of imported libraries w.r.t. compile and link properties. These enhancements are required for implementing this new packaging. In addition, all supporting external libraries built with the HDF library must support this dual-binary approach. (CMake 3.2.1 release has fixes for Windows UTF-8 support and OS X support.)

External Libraries

The external libraries, szip, zlib, and jpeg, will need the CMake support updated to produce dual-binary libraries. These changes require the same changes to macros and find_package calls, which build the library statically with the option for dynamic libraries dependent on the BUILD_SHARED_LIBS option.

Releated JIRA Issues

HDFFV-5881 - The Windows CMake installers do not support installing more than one package.

HDFFV-7849 - Consolidate the Windows installers.

HDFFR-1473 - Windows installer use the same name for both static and dynamic packages.

HDFFV-8074 - Synchronize autotools configure and CMake configuration checks and options. Each build system should build similar binaries given the same options on the same machine.
