

# HDF5 1.10.3

## Release Information

Version	HDF5 1.10.3
Release Date	2018-08-22
Download	<a href="#">Download</a>
Release Notes	<a href="#">Release Notes</a>
Tested Platforms and Configuration Features	<a href="#">Platforms and Config Features</a>
Compatibility Report	<a href="#">Compatibility Report</a>
Newsletter	<a href="#">Release Announcement</a>

## Files

File	Type	Comment	Md5 Checksum
<a href="#">hdf5-1.10.3.tar.gz</a>	Source release	Gzipped source tar file	<a href="#">hdf5-1.10.3.md5</a>
<a href="#">hdf5-1.10.3.tar.bz2</a>	Source release	Bzipped source tar file	"
<a href="#">hdf5-1.10.3.zip</a>	Source release	Windows zip file	"
<a href="#">CMake-hdf5-1.10.3.tar.gz</a>	CMake source release	File to build HDF5 with CMake on Unix. See <a href="#">Build Instructions</a> .	"
<a href="#">CMake-hdf5-1.10.3.zip</a>	CMake source release	File to build HDF5 with CMake on Windows. See <a href="#">Build Instructions</a> .	"

## Tested Platforms and Configuration Features

NOTE: HDF5-1.10.3 *requires* MPI 3.

<a href="#">release_docs / RELEASE.txt [454:610]</a>	<a href="#">hdf5-1_10_3</a>	<a href="#">HDF5V/hd</a>
<a href="#">f5</a>		
<p>Supported Platforms =====</p> <pre>Linux 2.6.32-696.16.1.el6.ppc64 gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18) #1 SMP ppc64 GNU/Linux          g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18) (ostrich)                       GNU Fortran (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18)                                  IBM XL C/C++ V13.1                                 IBM XL Fortran V15.1  Linux 3.10.0-327.10.1.el7        GNU C (gcc), Fortran (gfortran), C++ (g++) #1 SMP x86_64 GNU/Linux          compilers: (kituo/moohan)                   Version 4.8.5 20150623 (Red Hat 4.8.5-4)                                    Version 4.9.3, Version 5.2.0                                    Intel(R) C (icc), C++ (icpc), Fortran (icc)                                    compilers:                                    Version 17.0.0.098 Build 20160721                                    MPICH 3.1.4 compiled with GCC 4.9.3</pre>		

SunOS 5.11 32- and 64-bit (emu)	Sun C 5.12 SunOS_sparc Sun Fortran 95 8.6 SunOS_sparc Sun C++ 5.12 SunOS_sparc
Windows 7	Visual Studio 2015 w/ Intel Fortran 16 (cmake)
Windows 7 x64	Visual Studio 2012 w/ Intel Fortran 15 (cmake) Visual Studio 2013 w/ Intel Fortran 15 (cmake) Visual Studio 2015 w/ Intel Fortran 16 (cmake) Visual Studio 2015 w/ Intel C, Fortran 2017 (cmake)
Windows 10	Visual Studio 2015 w/ MSMPI 8 (cmake)
Windows 10 x64	Visual Studio 2015 w/ Intel Fortran 18 (cmake) Visual Studio 2017 w/ Intel Fortran 18 (cmake)
Mac OS X Yosemite 10.10.5 64-bit (osx1010dev/osx1010test)	Apple clang/clang++ version 6.1 from Xcode 7.0 gfortran GNU Fortran (GCC) 4.9.2 Intel icc/icpc/ifort version 15.0.3
Mac OS X El Capitan 10.11.6 64-bit (osx1011dev/osx1011test)	Apple clang/clang++ version 7.3.0 from Xcode 7.3 gfortran GNU Fortran (GCC) 5.2.0 Intel icc/icpc/ifort version 16.0.2
Mac OS Sierra 10.12.6 64-bit (swallow/kite)	Apple LLVM version 8.1.0 (clang/clang++-802.0.42) gfortran GNU Fortran (GCC) 7.1.0 Intel icc/icpc/ifort version 17.0.2

#### Tested Configuration Features Summary

=====

In the tables below

y = tested  
 n = not tested in this release  
 C = Cluster  
 W = Workstation  
 x = not working in this release  
 dna = does not apply  
 ( ) = footnote appears below second table  
 <blank> = testing incomplete on this feature or platform

Platform	C	F90/ parallel	F90 F2003 parallel	C++	zlib	SZIP
Solaris2.11 32-bit	n	y/y	n	y	y	y
Solaris2.11 64-bit	n	y/n	n	y	y	y
Windows 7	y	y/y	n	y	y	y
Windows 7 x64	y	y/y	y	y	y	y
Windows 7 Cygwin	n	y/n	n	y	y	y
Windows 7 x64 Cygwin	n	y/n	n	y	y	y
Windows 10	y	y/y	n	y	y	y
Windows 10 x64	y	y/y	n	y	y	y
Mac OS X Mavericks 10.9.5 64-bit	n	y/y	n	y	y	y
Mac OS X Yosemite 10.10.5 64-bit	n	y/y	n	y	y	y
Mac OS X El Capitan 10.11.6 64-bit	n	y/y	n	y	y	y
Mac OS Sierra 10.12.6 64-bit	n	y/y	n	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 PGI	n	y/y	n	y	y	y

CentOS 7.2 Linux 2.6.32 x86_64 GNU	y	y/y	y	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 Intel	n	y/y	n	y	y	y
Linux 2.6.32-573.18.1.el6.ppc64	n	y/y	n	y	y	y

Platform	Shared C libs	Shared F90 libs	Shared C++ libs	Thread- safe
Solaris2.11 32-bit	y	y	y	y
Solaris2.11 64-bit	y	y	y	y
Windows 7	y	y	y	y
Windows 7 x64	y	y	y	y
Windows 7 Cygwin	n	n	n	y
Windows 7 x64 Cygwin	n	n	n	y
Windows 10	y	y	y	y
Windows 10 x64	y	y	y	y
Mac OS X Mavericks 10.9.5 64-bit	y	n	y	y
Mac OS X Yosemite 10.10.5 64-bit	y	n	y	y
Mac OS X El Capitan 10.11.6 64-bit	y	n	y	y
Mac OS Sierra 10.12.6 64-bit	y	n	y	y
CentOS 7.2 Linux 2.6.32 x86_64 PGI	y	y	y	n
CentOS 7.2 Linux 2.6.32 x86_64 GNU	y	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 Intel	y	y	y	n
Linux 2.6.32-573.18.1.el6.ppc64	y	y	y	n

Compiler versions for each platform are listed in the preceding "Supported Platforms" table.

#### More Tested Platforms

=====

The following platforms are not supported but have been tested for this release.

Linux 2.6.32-573.22.1.el6 #1 SMP x86_64 GNU/Linux (mayll/platypus)	GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.4.7 20120313 Version 4.9.3, 5.3.0, 6.2.0 PGI C, Fortran, C++ for 64-bit target on x86-64; Version 17.10-0 Intel(R) C (icc), C++ (icpc), Fortran (icc) compilers: Version 17.0.4.196 Build 20170411 MPICH 3.1.4 compiled with GCC 4.9.3
Linux 3.10.0-327.18.2.el7 #1 SMP x86_64 GNU/Linux (jelly)	GNU C (gcc) and C++ (g++) compilers Version 4.8.5 20150623 (Red Hat 4.8.5-4) with NAG Fortran Compiler Release 6.1(Tozai) GCC Version 7.1.0 OpenMPI 3.0.0-GCC-7.2.0-2.29, 3.1.0-GCC-7.2.0-2.29 Intel(R) C (icc) and C++ (icpc) compilers Version 17.0.0.098 Build 20160721 with NAG Fortran Compiler Release 6.1(Tozai)
Linux 3.10.0-327.10.1.el7 #1 SMP x86_64 GNU/Linux (moohan)	MPICH 3.2 compiled with GCC 5.3.0
Linux 2.6.32-573.18.1.el6.ppc64	MPICH mpich 3.1.4 compiled with

```
#1 SMP ppc64 GNU/Linux          IBM XL C/C++ for Linux, V13.1
(ostrich)                        and IBM XL Fortran for Linux, V15.1

Debian 8.4 3.16.0-4-amd64 #1 SMP Debian 3.16.36-1 x86_64 GNU/Linux
gcc, g++ (Debian 4.9.2-10) 4.9.2
GNU Fortran (Debian 4.9.2-10) 4.9.2
(cmake and autotools)

Fedora 24 4.7.2-201.fc24.x86_64 #1 SMP x86_64 x86_64 x86_64 GNU/Linux
gcc, g++ (GCC) 6.1.1 20160621
(Red Hat 6.1.1-3)
GNU Fortran (GCC) 6.1.1 20160621
(Red Hat 6.1.1-3)
(cmake and autotools)

Ubuntu 16.04.1 4.4.0-38-generic #57-Ubuntu SMP x86_64 GNU/Linux
gcc, g++ (Ubuntu 5.4.0-6ubuntu1~16.04.2)
5.4.0 20160609
GNU Fortran (Ubuntu 5.4.0-6ubuntu1~16.04.2)
```

5.4.0 20160609  
(cmake and autotools)

## Release Notes

HDF5 version 1.10.3 released on 2018-08-21

=====

### INTRODUCTION

This document describes the differences between this release and the previous HDF5 release. It contains information on the platforms tested and known problems in this release. For more details check the HISTORY\*.txt files in the HDF5 source.

Note that documentation in the links below will be updated at the time of each final release.

Links to HDF5 documentation can be found on The HDF5 web page:

<https://portal.hdfgroup.org/display/HDF5/HDF5>

The official HDF5 releases can be obtained from:

<https://www.hdfgroup.org/downloads/hdf5/>

Changes from Release to Release and New Features in the HDF5-1.10.x release series can be found at:

<https://portal.hdfgroup.org/display/HDF5/HDF5+Application+Developer%27s+Guide>

If you have any questions or comments, please send them to the HDF Help Desk:

[help@hdfgroup.org](mailto:help@hdfgroup.org)

### CONTENTS

- New Features
- Bug Fixes since HDF5-1.10.2
- Supported Platforms
- Tested Configuration Features Summary
- More Tested Platforms
- Known Problems
- CMake vs. Autotools installations

### New Features

=====

#### Library

-----

- Moved the H5Dread/write\_chunk() API calls to H5Dread/write\_chunk()

The functionality of the direct chunk I/O calls in the high-level library has been moved to the H5D package in the main library. This will allow using those functions without building the high-level library. The parameters and functionality of the H5D calls are identical to the H5DO calls.

The original H5DO high-level API calls have been retained, though they are now just wrappers for the H5D calls. They are marked as deprecated and are only available when the library is built with deprecated functions. New code should use the H5D calls for this reason.

As a part of this work, the following symbols from H5Dpublic.h are no longer used:

```
H5D_XFER_DIRECT_CHUNK_WRITE_FLAG_NAME
H5D_XFER_DIRECT_CHUNK_WRITE_FILTERS_NAME
H5D_XFER_DIRECT_CHUNK_WRITE_OFFSET_NAME
H5D_XFER_DIRECT_CHUNK_WRITE_DATASIZE_NAME
H5D_XFER_DIRECT_CHUNK_READ_FLAG_NAME
H5D_XFER_DIRECT_CHUNK_READ_OFFSET_NAME
H5D_XFER_DIRECT_CHUNK_READ_FILTERS_NAME
```

And properties with these names are no longer stored in the dataset transfer property lists. The symbols are still defined in H5Dpublic.h, but only when the library is built with deprecated symbols.

(DER - 2018/05/04)

Configuration:

-----

- Add missing USE\_110\_API\_DEFAULT option.

Option USE\_110\_API\_DEFAULT sets the default version of versioned APIs. The bin/makevers perl script did not set the maxidx variable correctly when the 1.10 branch was created. This caused the versioning process to always use the latest version of any API.

(ADB - 2018/08/17, H5FFV-10552)

- Added configuration checks for the following MPI functions:

MPI\_Mprobe - Used for the Parallel Compression feature  
MPI\_Imrecv - Used for the Parallel Compression feature

MPI\_Get\_elements\_x - Used for the "big Parallel I/O" feature  
MPI\_Type\_size\_x - Used for the "big Parallel I/O" feature

(JTH - 2018/08/02, H5FFV-10512)

- Added section to the libhdf5.settings file to indicate the status of the Parallel Compression and "big Parallel I/O" features.

(JTH - 2018/08/02, H5FFV-10512)

- Add option to execute swmr shell scripts from CMake.

Option TEST\_SHELL\_SCRIPTS redirects processing into a separate ShellTests.cmake file for UNIX types. The tests execute the shell scripts if a SH program is found.

(ADB - 2018/07/16)

C++ Library:

-----

- New wrappers

Added the following items:

```
+ Class DSetAccPropList for the dataset access property list.

+ Wrapper for H5Dget_access_plist to class DataSet
  // Gets the access property list of this dataset.
  DSetAccPropList getAccessPlist() const;

+ Wrappers for H5Pset_chunk_cache and H5Pget_chunk_cache to class
DSetAccPropList
  // Sets the raw data chunk cache parameters.
  void setChunkCache(size_t rdcc_nslots, size_t rdcc_nbytes, double rdcc_w0)

  // Retrieves the raw data chunk cache parameters.
  void getChunkCache(size_t &rdcc_nslots, size_t &rdcc_nbytes, double
&rdcc_w0)

+ New operator!= to class DataType (HDFV-10472)
  // Determines whether two datatypes are not the same.
  bool operator!=(const DataType& compared_type)

+ Wrappers for H5Oget_info2, H5Oget_info_by_name2, and H5Oget_info_by_idx2
(HDFV-10458)

  // Retrieves information about an HDF5 object.
  void getObjinfo(H5O_info_t& objinfo, unsigned fields = H5O_INFO_BASIC)
const;

  // Retrieves information about an HDF5 object, given its name.
  void getObjinfo(const char* name, H5O_info_t& objinfo,
    unsigned fields = H5O_INFO_BASIC,
    const LinkAccPropList& lapl = LinkAccPropList::DEFAULT) const;
  void getObjinfo(const H5std_string& name, H5O_info_t& objinfo,
    unsigned fields = H5O_INFO_BASIC,
    const LinkAccPropList& lapl = LinkAccPropList::DEFAULT) const;

  // Retrieves information about an HDF5 object, given its index.
  void getObjinfo(const char* grp_name, H5_index_t idx_type,
    H5_iter_order_t order, hsize_t idx, H5O_info_t& objinfo,
    unsigned fields = H5O_INFO_BASIC,
    const LinkAccPropList& lapl = LinkAccPropList::DEFAULT) const;
  void getObjinfo(const H5std_string& grp_name, H5_index_t idx_type,
    H5_iter_order_t order, hsize_t idx, H5O_info_t& objinfo,
    unsigned fields = H5O_INFO_BASIC,
    const LinkAccPropList& lapl = LinkAccPropList::DEFAULT) const;
```

(BMR - 2018/07/22, HDFV-10150, HDFV-10458, HDFV-1047)

Java Library:

-----

- Java HDFLibraryException class

Change parent class from Exception to RuntimeException.

(ADB - 2018/07/30, HDFS-10534)

- JNI Read and Write

Refactored variable-length functions, H5DreadVL and H5AreadVL, to correct dataset and attribute reads. New write functions, H5DwriteVL and H5AwriteVL, are under construction.

(ADB - 2018/06/02, HDFS-10519)

Bug Fixes since HDF5-1.10.2 release

=====

Library

-----

- Performance issue with H5Oget\_info

H5Oget\_info family of routines retrieves information for an object such as object type, access time, number of attributes, and storage space etc. Retrieving all such information regardless is an overkill and causes performance issue when doing so for many objects.

Add an additional parameter "fields" to the the H5Oget\_info family of routines indicating the type of information to be retrieved. The same is done to the H5Ovisit family of routines which recursively visits an object returning object information in a callback function. Both sets of routines are versioned and the corresponding compatibility macros are added.

The version 2 names of the two sets of routines are:

- (1) H5Oget\_info2, H5Oget\_info\_by\_idx2, H5Oget\_info\_by\_name2
- (2) H5Ovisit2, H5Ovisit\_by\_name2

(VC - 2018/08/15, HDFS-10180)

- Test failure due to metadata size in test/vds.c

The size of metadata from test\_api\_get\_ex\_dcpl() in test/vds.c is not as expected because the latest format should be used when encoding the layout for VDS.

Set the latest format in a temporary fapl and pass the setting to the routines that encode the dataset selection for VDS.

(VC - 2018/08/14 HDFS-10469)

- Java HDF5LibraryException class

The error minor and major values would be lost after the constructor executed.

Created two local class variables to hold the values obtained during execution of the constructor. Refactored the class functions to retrieve the class values rather than calling the native functions. The native functions were renamed and called only during execution of the constructor.  
Added error checking to calling class constructors in JNI classes.

(ADB - 2018/08/06, HДФFV-10544)

- Added checks of the defined MPI\_VERSION to guard against usage of MPI-3 functions in the Parallel Compression and "big Parallel I/O" features when HDF5 is built with MPI-2. Previously, the configure step would pass but the build itself would fail when it could not locate the MPI-3 functions used.

As a result of these new checks, HDF5 can again be built with MPI-2, but the Parallel Compression feature will be disabled as it relies on the MPI-3 functions used.

(JTH - 2018/08/02, HДФFV-10512)

- User's patches: CVEs

The following patches have been applied:

CVE-2018-11202 - NULL pointer dereference was discovered in  
H5S\_hyper\_make\_spans in H5Shyper.c (HДФFV-10476)  
<https://security-tracker.debian.org/tracker/CVE-2018-11202>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=3DCVE-2018-11202>

CVE-2018-11203 - A division by zero was discovered in  
H5D\_\_btree\_decode\_key in H5Dbtree.c (HДФFV-10477)  
<https://security-tracker.debian.org/tracker/CVE-2018-11203>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=3DCVE-2018-11203>

CVE-2018-11204 - A NULL pointer dereference was discovered in  
H5O\_\_chunk\_deserialize in H5Ocache.c (HДФFV-10478)  
<https://security-tracker.debian.org/tracker/CVE-2018-11204>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=3DCVE-2018-11204>

CVE-2018-11206 - An out of bound read was discovered in  
H5O\_fill\_new\_decode and H5O\_fill\_old\_decode in H5Ofill.c  
(HДФFV-10480)  
<https://security-tracker.debian.org/tracker/CVE-2018-11206>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=3DCVE-2018-11206>

CVE-2018-11207 - A division by zero was discovered in  
H5D\_\_chunk\_init in H5Dchunk.c (HДФFV-10481)  
<https://security-tracker.debian.org/tracker/CVE-2018-11207>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=3DCVE-2018-11207>

(BMR - 2018/7/22, PR#s: 1134 and 1139,  
HДФFV-10476, HДФFV-10477, HДФFV-10478, HДФFV-10480, HДФFV-10481)

- H5Adelete

H5Adelete failed when deleting the last "large" attribute that is stored densely via fractal heap/v2 b-tree.

After removing the attribute, update the ainfo message. If the number of attributes goes to zero, remove the message.

(VC - 2018/07/20, HДФFV-9277)

- A bug was discovered in the parallel library which caused partial parallel reads of filtered datasets to return incorrect data. The library used the incorrect dataspace for each chunk read, causing the selection used in each chunk to be wrong.

The bug was not caught during testing because all of the current tests which do parallel reads of filtered data read all of the data using an H5S\_ALL selection. Several tests were added which exercise partial parallel reads.

(JTH - 2018/07/16, HДФFV-10467)

- A bug was discovered in the parallel library which caused parallel writes of filtered datasets to trigger an assertion failure in the file free space manager.

This occurred when the filter used caused chunks to repeatedly shrink and grow over the course of several dataset writes. The previous chunk information, such as the size of the chunk and the offset in the file, was being cached and not updated after each write, causing the next write to the chunk to retrieve the incorrect cached information and run into issues when reallocating space in the file for the chunk.

(JTH - 2018/07/16, HДФFV-10509)

- A bug was discovered in the parallel library which caused the H5D\_\_mpio\_array\_gather() function to allocate too much memory.

When the function is called with the 'allgather' parameter set to a non-true value, the function will receive data from all MPI ranks and gather it to the single rank specified by the 'root' parameter. However, the bug in the function caused memory for the received data to be allocated on all MPI ranks, not just the singular rank specified as the receiver. In some circumstances, this would cause an application to fail due to the large amounts of memory being allocated.

(JTH - 2018/07/16, HДФFV-10467)

- Error checks in h5stat and when decoding messages

h5stat exited with seg fault/core dumped when errors are encountered in the internal library.

Add error checks and --enable-error-stack option to h5stat.  
Add range checks when decoding messages: old fill value, old layout and refcount.

(VC - 2018/07/11, HДФFV-10333)

- If an HDF5 file contains a malformed compound datatype with a suitably large offset, the type conversion code can run off the end of the type conversion buffer, causing a segmentation fault.

This issue was reported to The HDF Group as issue #CVE-2017-17507.

NOTE: The HDF5 C library cannot produce such a file. This condition should only occur in a corrupt (or deliberately altered) file or a file created by third-party software.

THE HDF GROUP WILL NOT FIX THIS BUG AT THIS TIME

Fixing this problem would involve updating the publicly visible H5T\_conv\_t function pointer typedef and versioning the API calls which use it. We normally only modify the public API during major releases, so this bug will not be fixed at this time.

(DER - 2018/02/26, HDFS-10356)

#### Configuration

-----

- Applied patches to address Cywin build issues

There were three issues for Cygwin builds:

- Shared libs were not built.
- The -std=c99 flag caused a SIG\_SETMASK undeclared error.
- Undefined errors when building test shared libraries.

Patches to address these issues were received and incorporated in this version.

(LRK - 2018/07/18, HDFS-10475)

- The --enable-debug/production configure flags are listed as 'deprecated' when they should really be listed as 'removed'.

In the autotools overhaul several years ago, we removed these flags and implemented a new --enable-build-mode= flag. This was done because we changed the semantics of the modes and didn't want users to silently be exposed to them. The newer system is also more flexible and us to add other modes (like 'clean').

The --enable-debug/production flags are now listed as removed.

(DER - 2018/05/31, HDFS-10505)

- Moved the location of gcc attribute.

The gcc attribute(no\_sanitise), named as the macro HDF\_NO\_UBSAN, was located after the function name. Builds with GCC 7 did not indicate any problem, but GCC 8 issued errors. Moved the attribute before the function name, as required.

(ADB - 2018/05/22, HDFS-10473)

- Reworked java test suite into individual JUnit tests.

Testing the whole suite of java unit tests in a single JUnit run made it difficult to determine actual failures when tests would fail. Running each file set of tests individually, allows individual failures to be diagnosed easier. A side benefit is that tests for optional components

of the library can be disabled if not configured.

(ADB - 2018/05/16, HDFS-9739)

- Converted CMake global commands `ADD_DEFINITIONS` and `INCLUDE_DIRECTORIES` to use `target_*` type commands. This change modernizes the CMake usage in the HDF5 library.

In addition, there is the intention to convert to generator expressions, where possible. The exception is Fortran `FLAGS` on Windows Visual Studio. The HDF macros `TARGET_C_PROPERTIES` and `TARGET_FORTRAN_PROPERTIES` have been removed with this change in usage.

The additional language (C++ and Fortran) checks have also been localized to only be checked when that language is enabled.

(ADB - 2018/05/08)

#### Performance

-----

- Revamped internal use of DXPLs, improving performance

(QAK - 2018/05/20)

#### Fortran

-----

- Fixed issue with `h5fget_obj_count_f` and using a file id of `H5F_OBJ_ALL_F` not returning the correct count.

(MSB - 2018/5/15, HDFS-10405)

#### C++ APIs

-----

- Adding default arguments to existing functions

Added the following items:

+ Two more property list arguments are added to `H5Location::createDataSet`:  
`const DSetAccPropList& dapl = DSetAccPropList::DEFAULT`  
`const LinkCreatPropList& lcpl = LinkCreatPropList::DEFAULT`

+ One more property list argument is added to `H5Location::openDataSet`:  
`const DSetAccPropList& dapl = DSetAccPropList::DEFAULT`

(BMR - 2018/07/21, PR# 1146)

- Improvement C++ documentation

Replaced the table in main page of the C++ documentation from mht to htm format for portability.

(BMR - 2018/07/17, PR# 1141)

#### Supported Platforms

=====

Linux 2.6.32-696.16.1.el6.ppc64	gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18)
#1 SMP ppc64 GNU/Linux	g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18)
(ostrich)	GNU Fortran (GCC) 4.4.7 20120313 (Red Hat
4.4.7-18)	
	IBM XL C/C++ V13.1
	IBM XL Fortran V15.1
Linux 3.10.0-327.10.1.el7	GNU C (gcc), Fortran (gfortran), C++ (g++)
#1 SMP x86_64 GNU/Linux	compilers:
(kituo/moohan)	Version 4.8.5 20150623 (Red Hat 4.8.5-4)
	Version 4.9.3, Version 5.2.0
	Intel(R) C (icc), C++ (icpc), Fortran (icc)
	compilers:
	Version 17.0.0.098 Build 20160721
	MPICH 3.1.4 compiled with GCC 4.9.3
SunOS 5.11 32- and 64-bit	Sun C 5.12 SunOS_sparc
(emu)	Sun Fortran 95 8.6 SunOS_sparc
	Sun C++ 5.12 SunOS_sparc
Windows 7	Visual Studio 2015 w/ Intel Fortran 16 (cmake)
Windows 7 x64	Visual Studio 2012 w/ Intel Fortran 15 (cmake)
	Visual Studio 2013 w/ Intel Fortran 15 (cmake)
	Visual Studio 2015 w/ Intel Fortran 16 (cmake)
	Visual Studio 2015 w/ Intel C, Fortran 2017
(cmake)	
	Visual Studio 2015 w/ MSMPI 8 (cmake)
Windows 10	Visual Studio 2015 w/ Intel Fortran 18 (cmake)
Windows 10 x64	Visual Studio 2015 w/ Intel Fortran 18 (cmake)
	Visual Studio 2017 w/ Intel Fortran 18 (cmake)
Mac OS X Yosemite 10.10.5	Apple clang/clang++ version 6.1 from Xcode 7.0
64-bit	gfortran GNU Fortran (GCC) 4.9.2
(osx1010dev/osx1010test)	Intel icc/icpc/ifort version 15.0.3
Mac OS X El Capitan 10.11.6	Apple clang/clang++ version 7.3.0 from Xcode 7.3
64-bit	gfortran GNU Fortran (GCC) 5.2.0
(osx1011dev/osx1011test)	Intel icc/icpc/ifort version 16.0.2
Mac OS Sierra 10.12.6	Apple LLVM version 8.1.0 (clang/clang++-802.0.42)
64-bit	gfortran GNU Fortran (GCC) 7.1.0
(swallow/kite)	Intel icc/icpc/ifort version 17.0.2

Tested Configuration Features Summary

=====

In the tables below

y = tested

n = not tested in this release

C = Cluster

W = Workstation

x = not working in this release

dna = does not apply

( ) = footnote appears below second table

<blank> = testing incomplete on this feature or platform

Platform	C parallel	F90/ F2003	F90 parallel	C++	zlib	SZIP
Solaris2.11 32-bit	n	y/y	n	y	y	y
Solaris2.11 64-bit	n	y/n	n	y	y	y
Windows 7	y	y/y	n	y	y	y
Windows 7 x64	y	y/y	y	y	y	y
Windows 7 Cygwin	n	y/n	n	y	y	y
Windows 7 x64 Cygwin	n	y/n	n	y	y	y
Windows 10	y	y/y	n	y	y	y
Windows 10 x64	y	y/y	n	y	y	y
Mac OS X Mavericks 10.9.5 64-bit	n	y/y	n	y	y	y
Mac OS X Yosemite 10.10.5 64-bit	n	y/y	n	y	y	y
Mac OS X El Capitan 10.11.6 64-bit	n	y/y	n	y	y	y
Mac OS Sierra 10.12.6 64-bit	n	y/y	n	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 PGI	n	y/y	n	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 GNU	y	y/y	y	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 Intel	n	y/y	n	y	y	y
Linux 2.6.32-573.18.1.el6.ppc64	n	y/y	n	y	y	y

Platform	Shared C libs	Shared F90 libs	Shared C++ libs	Thread- safe
Solaris2.11 32-bit	y	y	y	y
Solaris2.11 64-bit	y	y	y	y
Windows 7	y	y	y	y
Windows 7 x64	y	y	y	y
Windows 7 Cygwin	n	n	n	y
Windows 7 x64 Cygwin	n	n	n	y
Windows 10	y	y	y	y
Windows 10 x64	y	y	y	y
Mac OS X Mavericks 10.9.5 64-bit	y	n	y	y
Mac OS X Yosemite 10.10.5 64-bit	y	n	y	y
Mac OS X El Capitan 10.11.6 64-bit	y	n	y	y
Mac OS Sierra 10.12.6 64-bit	y	n	y	y
CentOS 7.2 Linux 2.6.32 x86_64 PGI	y	y	y	n
CentOS 7.2 Linux 2.6.32 x86_64 GNU	y	y	y	y
CentOS 7.2 Linux 2.6.32 x86_64 Intel	y	y	y	n
Linux 2.6.32-573.18.1.el6.ppc64	y	y	y	n

Compiler versions for each platform are listed in the preceding "Supported Platforms" table.

#### More Tested Platforms

=====

The following platforms are not supported but have been tested for this release.

<p>Linux 2.6.32-573.22.1.el6 #1 SMP x86_64 GNU/Linux (mayll/platypus)</p>	<p>GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.4.7 20120313 Version 4.9.3, 5.3.0, 6.2.0 PGI C, Fortran, C++ for 64-bit target on x86-64; Version 17.10-0 Intel(R) C (icc), C++ (icpc), Fortran (icc) compilers: Version 17.0.4.196 Build 20170411 MPICH 3.1.4 compiled with GCC 4.9.3</p>
---	---

Linux 3.10.0-327.18.2.el7 GNU C (gcc) and C++ (g++) compilers  
#1 SMP x86\_64 GNU/Linux Version 4.8.5 20150623 (Red Hat 4.8.5-4)  
(jelly) with NAG Fortran Compiler Release 6.1(Tozai)  
GCC Version 7.1.0  
OpenMPI 3.0.0-GCC-7.2.0-2.29,  
3.1.0-GCC-7.2.0-2.29  
Intel(R) C (icc) and C++ (icpc) compilers  
Version 17.0.0.098 Build 20160721  
with NAG Fortran Compiler Release 6.1(Tozai)

Linux 3.10.0-327.10.1.el7 MPICH 3.2 compiled with GCC 5.3.0  
#1 SMP x86\_64 GNU/Linux  
(mochan)

Linux 2.6.32-573.18.1.el6.ppc64 MPICH mpich 3.1.4 compiled with  
#1 SMP ppc64 GNU/Linux IBM XL C/C++ for Linux, V13.1  
(ostrich) and IBM XL Fortran for Linux, V15.1

Debian 8.4 3.16.0-4-amd64 #1 SMP Debian 3.16.36-1 x86\_64 GNU/Linux  
gcc, g++ (Debian 4.9.2-10) 4.9.2  
GNU Fortran (Debian 4.9.2-10) 4.9.2  
(cmake and autotools)

Fedora 24 4.7.2-201.fc24.x86\_64 #1 SMP x86\_64 x86\_64 x86\_64 GNU/Linux  
gcc, g++ (GCC) 6.1.1 20160621  
(Red Hat 6.1.1-3)  
GNU Fortran (GCC) 6.1.1 20160621  
(Red Hat 6.1.1-3)  
(cmake and autotools)

Ubuntu 16.04.1 4.4.0-38-generic #57-Ubuntu SMP x86\_64 GNU/Linux  
gcc, g++ (Ubuntu 5.4.0-6ubuntu1~16.04.2)  
5.4.0 20160609  
GNU Fortran (Ubuntu 5.4.0-6ubuntu1~16.04.2)  
5.4.0 20160609  
(cmake and autotools)

#### Known Problems

=====

At present, metadata cache images may not be generated by parallel applications. Parallel applications can read files with metadata cache images, but since this is a collective operation, a deadlock is possible if one or more processes do not participate.

Three tests fail with OpenMPI 3.0.0/GCC-7.2.0-2.29:

testphdf5 (ecdsetw, selnone, cchunk1, cchunk3, cchunk4, and actualio)  
t\_shapesame (sscontig2)  
t\_pflushl/fails on exit

The first two tests fail attempting collective writes.

Known problems in previous releases can be found in the HISTORY\*.txt files in the HDF5 source. Please report any new problems found to [help@hdfgroup.org](mailto:help@hdfgroup.org).

CMake vs. Autotools installations

=====

While both build systems produce similar results, there are differences. Each system produces the same set of folders on linux (only CMake works on standard Windows); bin, include, lib and share. Autotools places the COPYING and RELEASE.txt file in the root folder, CMake places them in the share folder.

The bin folder contains the tools and the build scripts. Additionally, CMake creates dynamic versions of the tools with the suffix "-shared". Autotools installs one set of tools depending on the "--enable-shared" configuration option.

build scripts

-----

Autotools: h5c++, h5cc, h5fc

CMake: h5c++, h5cc, h5hlc++, h5hlcc

The include folder holds the header files and the fortran mod files. CMake places the fortran mod files into separate shared and static subfolders, while Autotools places one set of mod files into the include folder. Because CMake produces a tools library, the header files for tools will appear in the include folder.

The lib folder contains the library files, and CMake adds the pkgconfig subfolder with the hdf5\*.pc files used by the bin/build scripts created by the CMake build. CMake separates the C interface code from the fortran code by creating C-stub libraries for each Fortran library. In addition, only CMake installs the tools library. The names of the szip libraries are different between the build systems.

The share folder will have the most differences because CMake builds include a number of CMake specific files for support of CMake's find\_package and support

for the HDF5 Examples CMake project.

## Compatibility Report