

# HDF5 1.8.21

## Release Information

Version	HDF5 1.8.21
Release Date	2018-06-05
Download	<a href="#">Download</a>
Tested Platforms and Configuration Features	<a href="#">Platforms and Config Features</a>
Release Notes	<a href="#">Release Notes</a>
Compatibility Report	<a href="#">Interface Compatibility Report</a>
Newsletter	<a href="#">Release Announcement</a>

## Files

Location of Software: <https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8/hdf5-1.8.21/>

File	Type	Compilers	Comments	Install Instructions	MD5 Checksum
<a href="#">hdf5-1.8.21.tar.gz</a>	Source release		Gzipped source tar file	See <a href="#">release_docs/</a>	<a href="#">hdf5-1.8.21.md5</a>
<a href="#">hdf5-1.8.21.tar.bz2</a>	Source release		Bzipped source tar file	See <a href="#">release_docs/</a>	"
<a href="#">hdf5-1.8.21.tar</a>	Source release		Source tar file	See <a href="#">release_docs/</a>	"
<a href="#">hdf5-1.8.21.zip</a>	Source release		Windows zip file		"
<a href="#">CMake-hdf5-1.8.21.tar.gz</a>	CMake source release		File to build HDF5 with CMake on Unix	See <a href="#">Build Instructions.</a>	"
<a href="#">CMake-hdf5-1.8.21.zip</a>	CMake source release		File to build HDF5 with CMake on Windows	See <a href="#">Build Instructions.</a>	"
<a href="#">hdf5-1.8.21-Std-centos7-x86_64-shared_64.tar.gz</a>	Linux 3.10 CentOS 7 x86_64 binary	gcc, g++, gfortran 4.8.5			<a href="#">hdf5-1.8.21-Std-centos7-x86_64-</a>
<a href="#">hdf5-1.8.21-Std-win7_64-vs14.zip</a>	Windows 64-bit binary	CMake VS 2015 C, C++, IVF 16		To compile and run the included examples you <b>must</b> set INSTALLDIR to the location of HDF5 when running ctest. For example:  <pre>ctest -S HDF518_Examples.cmake, INSTALLDIR=MyLocation  -C Release -V -O test.log</pre>	<a href="#">hdf5-1.8.21-Std-win7_64-vs14.zi</a>

## Tested Platforms and Configuration Features

f5

## Supported Platforms

=====

The following platforms are supported and have been tested for this release. They are built with the configure process unless specified otherwise.

Linux 2.6.32-573.22.1.el6 #1 SMP x86_64 GNU/Linux (platypus/may11)	GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.4.7 20120313 Versions 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 20160721 MPICH 3.1.4 compiled with GCC 4.9.3 OpenMPI 2.0.1 compiled with GCC 4.9.3
Linux 2.6.32-573.18.1.el6 #1 SMP ppc64 GNU/Linux (ostrich) 4.4.7-16)	gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16) g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16) GNU Fortran (GCC) 4.4.7 20120313 (Red Hat  IBM XL C/C++ V13.1 IBM XL Fortran V15.1
Linux 3.10.0-327.10.1.el7 #1 SMP x86_64 GNU/Linux (kituo/moohan/jelly)	GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.8.5 20150623 (Red Hat 4.8.5-4) Versions 4.9.3, 5.3.0, 6.2.0 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 NAG Fortran Compiler Release 6.1(Tozai) Build 6116
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)	Visual Studio 2015 w/ MSMPI 8 (cmake)
Windows 10	Visual Studio 2015 w/ Intel Fortran 16 (cmake)
Windows 10 x64	Visual Studio 2015 w/ Intel Fortran 16 (cmake) Visual Studio 2017 w/ Intel Fortran 18 (cmake)

Mac OS X Mavericks 10.9.5 64-bit (wren/quail)	Apple LLVM version 6.0 (clang-600.0.57) gfortran GNU Fortran (GCC) 4.9.2 Intel icc/icpc/ifort version 15.0.3
Mac OS X Yosemite 10.10.5 64-bit (osx1010dev/osx1010test)	Apple LLVM version 6.1 (clang-602.0.53) 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 (VM osx1011dev/osx1011test)	Apple LLVM version 7.3.0 (clang-703.0.29) gfortran GNU Fortran (GCC) 5.2.0 Intel icc/icpc/ifort version 16.0.2
Mac OS Sierra 10.12.6 64-bit (kite)	Apple LLVM version 8.1 (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 parallel	F90/ F2003	F90 parallel	C++	zlib	SZIP
SunOS 5.11 32-bit	n	Y/Y	n	Y	Y	Y
SunOS 5.11 64-bit	n	Y/Y	n	Y	Y	Y
Windows 7	Y	Y/Y	n	Y	Y	Y
Windows 7 x64	Y	Y/Y	n	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 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
AIX 6.1 32- and 64-bit	n	Y/n	n	Y	Y	Y
CentOS 6.7 Linux 2.6.32 x86_64 GNU	Y	Y/Y	Y	Y	Y	Y
CentOS 6.7 Linux 2.6.32 x86_64 Intel	n	Y/Y	n	Y	Y	Y
CentOS 6.7 Linux 2.6.32 x86_64 PGI	n	Y/Y	n	Y	Y	Y
CentOS 7.1 Linux 3.10.0 x86_64 GNU	Y	Y/Y	Y	Y	Y	Y
CentOS 7.1 Linux 3.10.0 x86_64 Intel	n	Y/Y	n	Y	Y	Y
Linux 2.6.32-573.18.1.el6.ppc64	n	Y/n	n	Y	Y	Y

Platform	Shared C libs	Shared F90 libs	Shared C++ libs	Thread- safe
SunOS 5.11 32-bit	Y	Y	Y	Y
SunOS 5.11 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 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
AIX 6.1 32- and 64-bit	Y	n	n	Y
CentOS 6.7 Linux 2.6.32 x86_64 GNU	Y	Y	Y	Y
CentOS 6.7 Linux 2.6.32 x86_64 Intel	Y	Y	Y	Y
CentOS 6.7 Linux 2.6.32 x86_64 PGI	Y	Y	Y	Y
CentOS 7.1 Linux 3.10.0 x86_64 GNU	Y	Y	Y	Y
CentOS 7.1 Linux 3.10.0 x86_64 Intel	Y	Y	Y	Y
Linux 2.6.32-573.18.1.el6.ppc64	Y	Y	Y	Y

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 g95 (GCC 4.0.3 (g95 0.94!))  
 #1 SMP x86\_64 GNU/Linux  
 (may11)

Debian8.4.0 3.16.0-4-amd64 #1 SMP Debian 3.16.36-1 x86\_64 GNU/Linux  
 gcc (Debian 4.9.2-10) 4.9.2  
 GNU Fortran (Debian 4.9.2-10) 4.9.2  
 (cmake and autotools)

Fedora24 4.7.2-201.fc24.x86\_64 #1 SMP x86\_64 x86\_64 x86\_64 GNU/Linux  
 gcc (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)

CentOS 7.2 3.10.0-327.28.2.el7.x86\_64 #1 SMP x86\_64 x86\_64 x86\_64 GNU/Linux  
 gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)  
 GNU Fortran (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)  
 (cmake and autotools)

Ubuntu 16.04 4.4.0-38-generic #62-Ubuntu SMP x86\_64 GNU/Linux  
 gcc (Ubuntu 5.4.0-6ubuntu1~16.04.2) 5.4.0

## Release Notes

HDF5 version 1.8.21 released on 2018-06-04

=====

### INTRODUCTION

=====

This document describes the differences between HDF5-1.8.20 and HDF5-1.8.21, and contains information on the platforms tested and known problems in HDF5-1.8.21.

For more details, see the files HISTORY-1\_0-1\_8\_0\_rc3.txt and HISTORY-1\_8.txt in the release\_docs/ directory of the HDF5 source.

Links to the HDF5 1.8.21 source code, documentation, and additional materials can be found on the HDF5 web page at:

<https://support.hdfgroup.org/HDF5/>

The HDF5 1.8.21 release can be obtained from:

<https://support.hdfgroup.org/HDF5/release/obtain518.html>

User documentation for 1.8.21 can be accessed directly at this location:

<https://support.hdfgroup.org/HDF5/doc1.8/>

New features in the HDF5-1.8.x release series, including brief general descriptions of some new and modified APIs, are described in the "What's New in 1.8.0?" document:

<https://support.hdfgroup.org/HDF5/doc/ADGuide/WhatsNew180.html>

All new and modified APIs are listed in detail in the "HDF5 Software Changes from Release to Release" document, in the section "Release 1.8.21 (current release) versus Release 1.8.20

<https://support.hdfgroup.org/HDF5/doc1.8/ADGuide/Changes.html>

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
- Support for New Platforms, Languages, and Compilers
- Bug Fixes since HDF5-1.8.20
- Supported Platforms
- Supported Configuration Features Summary

- More Tested Platforms
- Known Problems

## New Features

=====

### Configuration

-----

- CMake

Change minimum version to 3.10.

This change removes the need to support a copy of the FindMPI.cmake module, which has been removed, along with its subfolder in the config/cmake\_ext\_mod location.

(ADB - 2018/03/09)

- CMake

Add pkg-config file generation

Added pkg-config file generation for the C, C++, HL, and HL C++ libraries. In addition, builds on linux will create h5cXXX scripts that use the pkg-config

files. This is a limited implementation of a script like autotools h5cc.

(ADB - 2018/03/08, HDFV-4359)

- CMake

Refactor use of CMAKE\_BUILD\_TYPE for new variable, which understands the type of generator in use.

Added new configuration macros to use new HDF\_BUILD\_TYPE variable. This variable is set correctly for the type of generator being used for the build.

(ADB - 2018/01/08, HDFV-10385, HDFV-10296)

### C++ API

-----

- The following C++ API wrappers have been added to class H5Location

+ H5Lcreate\_soft:

```
// Creates a soft link from link_name to target_name.  
void link(const char *target_name, const char *link_name,...)  
void link(const H5std_string& target_name,...)
```

+ H5Lcreate\_hard:

```
// Creates a hard link from new_name to curr_name.  
void link(const char *curr_name, const Group& new_loc,...)  
void link(const H5std_string& curr_name, const Group& new_loc,...)
```

```
// Creates a hard link from new_name to curr_name in the same location.  
void link(const char *curr_name, const hid_t same_loc,...)  
void link(const H5std_string& curr_name, const hid_t same_loc,...)
```

Note: previous version CommonFG::link will be deprecated.

```

+ H5Lcopy:
  // Copy an object from a group of file to another.
  void copyLink(const char *src_name, const Group& dst,...)
  void copyLink(const H5std_string& src_name, const Group& dst,...)

  // Copy an object from a group of file to the same location.
  void copyLink(const char *src_name, const char *dst_name,...)
  void copyLink(const H5std_string& src_name,...)

```

```

+ H5Lmove:
  // Rename an object in a group or file to a new location.
  void moveLink(const char* src_name, const Group& dst,...)
  void moveLink(const H5std_string& src_name, const Group& dst,...)

  // Rename an object in a group or file to the same location.
  void moveLink(const char* src_name, const char* dst_name,...)
  void moveLink(const H5std_string& src_name,...)

```

Note: previous version CommonFG::move will be deprecated.

```

+ H5Ldelete:
  // Removes the specified link from this location.
  void unlink(const char *link_name,
              const LinkAccPropList& lapl = LinkAccPropList::DEFAULT)
  void unlink(const H5std_string& link_name,
              const LinkAccPropList& lapl = LinkAccPropList::DEFAULT)

```

Note: An additional parameter is added to CommonFG::unlink and it is moved to H5Location.

(BMR - 2018/05/11 - H5FFV-10445)

- New property list subclasses

Property list subclasses StrCreatPropList, LinkCreatPropList, and AttrCreatPropList are added for the C property list classes H5P\_STRING\_CREATE, H5P\_LINK\_CREATE, and H5P\_ATTRIBUTE\_CREATE.

(BMR - 2018/05/11 - H5FFV-10445)

- Another argument, LinkCreatPropList& lcpl, is added to the following functions for the use of link creation property list.

```

  Group createGroup(const char* name, size_t size_hint = 0,
                   const LinkCreatPropList& lcpl = LinkCreatPropList::DEFAULT)
  Group createGroup(const H5std_string& name, size_t size_hint = 0,
                   const LinkCreatPropList& lcpl = LinkCreatPropList::DEFAULT)

```

(BMR - 2018/05/11 - H5FFV-10445)

Support for New Platforms, Languages, and Compilers

=====

- Added support for Visual Studio 2017 w/ Intel Fortran 18 on Windows 10 x64.

Bug Fixes since HDF5-1.8.20

=====

- If an HDF5 file contains a filter pipeline message with a 'number of filters' field that exceeds the maximum number of allowed filters, the error handling code will attempt to dereference a NULL pointer.

This issue was reported to The HDF Group as issue #CVE-2017-17505.

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.

This problem arose because the error handling code assumed that the 'number of filters' field implied that a dynamic array of that size had already been created and that the cleanup code should iterate over that array and clean up each element's resources. If an error occurred before the array has been allocated, this will not be true.

This has been changed so that the number of filters is set to zero on errors. Additionally, the filter array traversal in the error handling code now requires that the filter array not be NULL.

(DER - 2018/02/06, HDEFFV-10354)

- If an HDF5 file contains a filter pipeline message which contains a 'number of filters' field that exceeds the actual number of filters in the message, the HDF5 C library will read off the end of the read buffer.

This issue was reported to The HDF Group as issue #CVE-2017-17506.

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 problem was fixed by passing the buffer size with the buffer and ensuring that the pointer cannot be incremented off the end of the buffer. A mismatch between the number of filters declared and the actual number of filters will now invoke normal HDF5 error handling.

(DER - 2018/02/26, HDEFFV-10355)

- 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)

- If an HDF5 file contains a malformed compound type which contains a member of size zero, a division by zero error will occur while processing the type.

This issue was reported to The HDF Group as issue #CVE-2017-17508.

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.

Checking for zero before dividing fixes the problem. Instead of the division by zero, the normal HDF5 error handling is invoked.

(DER - 2018/02/26, HDFS-10357)

- If an HDF5 file contains a malformed symbol table node that declares it contains more symbols than it actually contains, the library can run off the end of the metadata cache buffer while processing the symbol table node.

This issue was reported to The HDF Group as issue #CVE-2017-17509.

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.

Performing bounds checks on the buffer while processing fixes the problem. Instead of the segmentation fault, the normal HDF5 error handling is invoked.

(DER - 2018/03/12, HDFS-10358)

## Configuration

-----

### - Library

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)

### - CMake

Update CMake commands configuration.

A number of improvements were made to the CMake commands. Most changes simplify usage or eliminate unused constructs. Also, some changes support better cross-platform support.

(ADB - 2018/02/01, HDFS-10398)

- CMake

Correct usage of CMAKE\_BUILD\_TYPE variable.

The use of the CMAKE\_BUILD\_TYPE is incorrect for multi-config generators (Visual Studio and XCode) and is optional for single config generators. Created a new macro to check

GLOBAL PROPERTY -> GENERATOR\_IS\_MULTI\_CONFIG

Created two new HDF variable, HDF\_BUILD\_TYPE and HDF\_CFG\_BUILD\_TYPE. Defaults for these variables is "Release".

(ADB - 2018/01/10, HDFSFFV-10385)

- CMake

Add replacement of fortran flags if using static CRT.

Added TARGET\_STATIC\_CRT\_FLAGS call to HDFUseFortran.cmake file in config/cmake\_ext\_mod folder.

(ADB - 2018/01/08, HDFSFFV-10334)

Library

-----

- Utility function can not handle lowercase Windows drive letters

Added call to toupper function for drive letter.

(ADB - 2017/12/18, HDFSFFV-10307)

Tools

-----

- h5repack

h5repack changes the chunk parameters when a change of layout is not specified and a filter is applied.

HDFSFFV-10297, HDFSFFV-10319 reworked code for h5repack and h5diff code in the tools library. The check for an existing layout was incorrectly placed into an if block and not executed. The check was moved into the normal path of the function.

(ADB - 2018/02/21, HDFSFFV-10412)

- h5dump

the tools library will hide the error stack during file open.

While this is preferable almost always, there are reasons to enable display of the error stack when a tool will not open a file. Adding an optional argument to the --enable-error-stack will provide this use case. As an optional argument it will not affect the operation of the --enable-error-stack. h5dump is the only tool to implement this change.

(ADB - 2018/02/15, HDFSFFV-10384)

- h5dump

h5dump would output an indented blank line in the filters section.

h5dump overused the h5tools\_simple\_prefix function, which is a function intended to account for the data index (x,y,z) option. Removed the function call for header information.

(ADB - 2018/01/25, HDAFFV-10396)

- h5repack

h5repack incorrectly searched internal object table for name.

h5repack would search the table of objects for a name, if the name did not match it tried to determine if the name without a leading slash would match. The logic was flawed! The table stored names(paths) without a leading slash and did a strstr of the table path to the name.

The assumption was that if there was a difference of one then it was a match, however "pressure" would match "/pressure" as well as "/pressure1", "/pressure2", etc. Changed logic to remove any leading slash and then do a full compare of the name.

(ADB - 2018/01/18, HDAFFV-10393)

- h5repack

h5repack failed to handle command line parameters for customer filters.

User defined filter parameter conversions would fail when integers were represented on the command line with character strings larger than 9 characters. Increased local variable array for storing the current command line parameter to prevent buffer overflows.

(ADB - 2018/01/17, HDAFFV-10392)

- h5diff

h5diff seg faulted if comparing VL strings against fixed strings.

Reworked solution for HDAFFV-8625 and HDAFFV-8639. Implemented the check for string objects of same type in the diff\_can\_type function by adding an if(tclass1 == H5T\_STRING) block. This "if block" moves the same check that was added for attributes to this function, which is used by all object types. This function handles complex type structures. Also added a new test file in h5diffgentest for testing this issue and removed the temporary files used in the test scripts.

(ADB - 2018/01/04, HDAFFV-8745)

C++ API

-----

- Removal of memory leaks

A private function was inadvertently called, causing memory leaks. This is now fixed.

(BMR - 2018/04/12 - User reported in email)

- Changes in exception classes

Some exception classes are reorganized to reflect the HDF5 object hierarchy and allow customization.

DataSetIException -> LocationException -> Exception

DataTypeIException -> LocationException -> Exception

GroupIException -> LocationException -> Exception

AttributeIException -> LocationException -> Exception

FileIException -> GroupIException -> LocationException -> Exception

Member functions in H5Location and H5Object now throw specific exceptions associated with the invoking objects.

(BMR - 2018/05/11)

- H5Location::closeObjId is made static

(BMR - 2018/05/11)

- H5A wrappers in H5Location are removed as they have been in H5Object.

(BMR - 2018/05/11)

Supported Platforms

=====

The following platforms are supported and have been tested for this release. They are built with the configure process unless specified otherwise.

Linux 2.6.32-573.22.1.el6 #1 SMP x86_64 GNU/Linux (platypus/may11)	GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.4.7 20120313 Versions 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 20160721 MPICH 3.1.4 compiled with GCC 4.9.3 OpenMPI 2.0.1 compiled with GCC 4.9.3
Linux 2.6.32-573.18.1.el6 #1 SMP ppc64 GNU/Linux (ostrich) 4.4.7-16)	gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16) g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16) GNU Fortran (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16)  IBM XL C/C++ V13.1 IBM XL Fortran V15.1
Linux 3.10.0-327.10.1.el7 #1 SMP x86_64 GNU/Linux (kituo/moohan/jelly)	GNU C (gcc), Fortran (gfortran), C++ (g++) compilers: Version 4.8.5 20150623 (Red Hat 4.8.5-4) Versions 4.9.3, 5.3.0, 6.2.0 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 NAG Fortran Compiler Release 6.1(Tozai) Build 6116
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 16 (cmake)

Windows 10 x64 Visual Studio 2015 w/ Intel Fortran 16 (cmake)
Visual Studio 2017 w/ Intel Fortran 18 (cmake)

Mac OS X Mavericks 10.9.5 Apple LLVM version 6.0 (clang-600.0.57)
64-bit gfortran GNU Fortran (GCC) 4.9.2
(wren/quail) Intel icc/icpc/ifort version 15.0.3

Mac OS X Yosemite 10.10.5 Apple LLVM version 6.1 (clang-602.0.53)
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 LLVM version 7.3.0 (clang-703.0.29)
64-bit gfortran GNU Fortran (GCC) 5.2.0
(VM osx1011dev/osx1011test) Intel icc/icpc/ifort version 16.0.2

Mac OS Sierra 10.12.6 Apple LLVM version 8.1 (clang-802.0.42)
64-bit gfortran GNU Fortran (GCC) 7.1.0
(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	F90/ F2003	F90 parallel	C++	zlib	SZIP
SunOS 5.11 32-bit	n	y/y	n	y	y	y
SunOS 5.11 64-bit	n	y/y	n	y	y	y
Windows 7	y	y/y	n	y	y	y
Windows 7 x64	y	y/y	n	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 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
AIX 6.1 32- and 64-bit	n	y/n	n	y	y	y
CentOS 6.7 Linux 2.6.32 x86_64 GNU	y	y/y	y	y	y	y
CentOS 6.7 Linux 2.6.32 x86_64 Intel	n	y/y	n	y	y	y
CentOS 6.7 Linux 2.6.32 x86_64 PGI	n	y/y	n	y	y	y
CentOS 7.1 Linux 3.10.0 x86_64 GNU	y	y/y	y	y	y	y
CentOS 7.1 Linux 3.10.0 x86_64 Intel	n	y/y	n	y	y	y
Linux 2.6.32-573.18.1.el6.ppc64	n	y/n	n	y	y	y

Platform	Shared C libs	Shared F90 libs	Shared C++ libs	Thread- safe
SunOS 5.11 32-bit	Y	Y	Y	Y
SunOS 5.11 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 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
AIX 6.1 32- and 64-bit	y	n	n	y
CentOS 6.7 Linux 2.6.32 x86_64 GNU	y	y	y	y
CentOS 6.7 Linux 2.6.32 x86_64 Intel	y	y	y	y
CentOS 6.7 Linux 2.6.32 x86_64 PGI	y	y	y	y
CentOS 7.1 Linux 3.10.0 x86_64 GNU	y	y	y	y
CentOS 7.1 Linux 3.10.0 x86_64 Intel	y	y	y	y
Linux 2.6.32-573.18.1.el6.ppc64	y	y	y	y

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      g95 (GCC 4.0.3 (g95 0.94!))
#1 SMP x86_64 GNU/Linux
(mayll)

Debian8.4.0 3.16.0-4-amd64 #1 SMP Debian 3.16.36-1 x86_64 GNU/Linux
gcc (Debian 4.9.2-10) 4.9.2
GNU Fortran (Debian 4.9.2-10) 4.9.2
(cmake and autotools)

Fedora24 4.7.2-201.fc24.x86_64 #1 SMP x86_64 x86_64 x86_64 GNU/Linux
gcc (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)

CentOS 7.2 3.10.0-327.28.2.el7.x86_64 #1 SMP x86_64 x86_64 x86_64 GNU/Linux
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)
GNU Fortran (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)
(cmake and autotools)

Ubuntu 16.04 4.4.0-38-generic #62-Ubuntu SMP x86_64 GNU/Linux
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.2) 5.4.0
```

GNU Fortran (Ubuntu 5.4.0-6ubuntu1~16.04.2) 5.4.0  
(cmake and autotools)

#### Known Problems

=====

The dynamically loaded plugin test libraries require undefined references to HDF5 functions to be resolved at runtime in order to function properly. With autotools on CYGWIN this results in build errors, and we have not found a solution that satisfies both. Therefore the dynamically loaded plugin tests have been disabled on CYGWIN.

Mac OS X 10.13 added additional subdirectory structure in .libs for shared libraries. Consequently "make check" will fail testing java and dynamically loaded plugin test libraries attempting to copy files from the previous locations in .libs directories. This will be addressed in the next release when support for the Mac OS X 10.13 platform is added.

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.

## **Interface Compatibility Report**