Release Notes for HDF5 Release 1.8.13

Release Date: May 5, 2014

Contents

1. Introduction 2

2. New Features 3

2.1. Configuration 3

2.2. Library 3

2.3. Parallel Library 4

2.4. Tools 4

2.5. High-Level APIs 4

2.6. Fortran API 4

2.7. C++ API 4

3. Support for New Platforms, Languages, and Compilers 6

4. Bug Fixes since HDF5-1.8.12 7

4.1. Configuration 7

4.2. Library 8

4.3. Parallel Library 8

4.4. Performance 9

4.5. Tools 9

4.6. F90 API 9

4.7. C++ API 10

4.8. High-Level APIs 10

4.9. Fortran High-Level APIs 10

4.10. Testing 10

5. Supported Platforms 11

6. Tested Configuration Features Summary 13

7. More Tested Platforms 15

8. Known Problems 17

# Introduction

This document describes the differences between HDF5-1.8.12 and HDF5-1.8.13, and contains information on the platforms tested and known problems in HDF5-1.8.13. 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 1.8.13 source code, documentation, and additional materials can be found on the HDF5 web page at: <http://www.hdfgroup.org/products/hdf5/>.

The HDF5 1.8.13 release can be obtained from: <http://www.hdfgroup.org/HDF5/release/obtain5.html>.

User documentation for 1.8.13 can be accessed directly at this location: <http://www.hdfgroup.org/HDF5/doc/>.

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: <http://www.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 1.8.13 (current release) versus Release 1.8.12": <http://www.hdfgroup.org/HDF5/doc/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).

If you have any questions or comments about the release notes, please send them to [docs@hdfgroup.org](mailto:docs@hdfgroup.org).

# New Features

## Configuration

* Autotools: Automake updated to 1.14.1 (ADB - 2014/04/08)
* CMake: Moved minimum CMake version to 2.8.11 which enables better library include processing. (ADB - 2014/03/26)
* When configuring a thread-safe HDF5 Library it is no longer necessary to specify --enable-threadsafe with --with-pthreads if the Pthreads library is in a standard location. (DER - 2014/04/11 HDFFV-8693)

## Library

* Added an H5free\_memory API call. This should be used to free memory that has been allocated by HDF5 API calls. H5Tget\_member\_name and H5Pget\_class\_name are two examples. The main motivation for this call is Windows, where it is common for application code and the HDF5 Library to be using different C run-time libraries (CRT). Using the new call ensures that the same CRT handles both the allocation and free. This new function can also be useful in any case where the library uses a different memory manager than the application, such as when a debug memory manager is in use or when the HDF5 Library is wrapped for use in a managed language like Python or Java. Fixes HDFFV-7710, 8519, and 8851. (DER - 2014/04/11)
* The Core VFD (aka Memory VFD) can now be configured to track dirty regions in the file and only write out the changed regions on flush/close. Additionally, a "page aggregation" size can be set that will aggregate small writes into larger writes. For example, setting a 1 MiB page aggregation size will logically partition the in-memory file into 1 MiB pages that will be written out in their entirety if even a single byte is dirtied. The feature is controlled via the new H5Pset/get\_core\_write\_tracking() API call. A new "core\_paged" target has been added to the check-vfd target in test/Makefile.am that exercises the feature over all HDF5 VFD-aware tests. (DER - 2014/04/12)

## Parallel Library

* Removed MPI-POSIX VFD, as it wasn't helping anyone and was just generating support questions. Application developers performing parallel I/O should always use the MPI-IO VFD. (QAK - 2014/03/28 HDFFV-8659)
* Improved parallel I/O support to allow collective I/O on point selections. (QAK - 2014/03/15)

## Tools

* None

## High-Level APIs

* None

## Fortran API

* Wrappers h5pset\_file\_image\_f and h5pget\_file\_image\_f were added to the library. (MSB - 2014/1/2014)

## C++ API

The following new features are added:

* Wrappers to class H5Object to get an object's name (HDFFV-8548).

ssize\_t getObjName(char \*obj\_name, size\_t buf\_size = 0)

ssize\_t getObjName(H5std\_string& obj\_name, size\_t len = 0)

H5std\_string getObjName()

* Wrappers to class H5CommonFG to get a child object's type from a group or file (HDFFV-8367).

H5O\_type\_t childObjType(const H5std\_string& objname)

H5O\_type\_t childObjType(const char\* objname)

H5O\_type\_t childObjType(hsize\_t index,

H5\_index\_t index\_type=H5\_INDEX\_NAME,

H5\_iter\_order\_t order=H5\_ITER\_INC, const char\* objname=".")

* Wrappers to class DSetMemXferPropList for setting/getting a transform property list (HDFFV-7907).

DSetMemXferPropList(const char\* expression);

void setDataTransform(const char\* expression)

void setDataTransform(const H5std\_string& expression)

ssize\_t getDataTransform(char\* exp, size\_t buf\_size=0)

H5std\_string getDataTransform()

* Wrapper to CompType for setting size to compound datatype (HDFFV-8642).

void setSize(size\_t size)

* Overloaded functions to provide prototypes that declare constant arguments const (HDFFV-3384). These include:

DataSet::fillMemBuf

DataSet::getVlenBufSize

DataSpace::extentCopy

DataType::commit

FileAccPropList::setSplit

H5File::getVFDHandle

* Additional overload to class H5Location to get a comment as a char\*

ssize\_t getComment(const char\* name, size\_t buf\_size, char\* comment)

* Additional overloads to class Attribute to get an attribute's name for convenience:

ssize\_t getName(char\* attr\_name, size\_t buf\_size=0)

ssize\_t getName(H5std\_string& attr\_name, size\_t buf\_size=0)

(BMR, 2014/04/15)

* A static wrapper to Exception for printing the error stack without an instance of Exception

static void printErrorStack(FILE\* stream = stderr,

hid\_t err\_stack = H5E\_DEFAULT);

(BMR, 2014/04/25)

# Support for New Platforms, Languages, and Compilers

* Mac OS X 10.6 Snow Leopard is not supported by Apple any more. In view of the added support of Mac OS X 10.9, Mac OS X 10.6 is retired from HDF5 supported platforms. (AKC - 2014/03/14 HDFFV-8704)
* Mac OS X 10.9 Mavericks is supported. (AKC - 2014/03/04 HDFFV-8694)

# Bug Fixes since HDF5-1.8.12

## Configuration

* CMake: When CMake commands are executed individually on the command line and the external filters are being built, the CMAKE\_BUILD\_TYPE define must be set to the same value as the configuration (-DCMAKE\_BUILD\_TYPE:STRING=Release if using -C Release). This is needed by the the szip and zlib filter build commands. (ADB - HDFFV-8695)
* CMake: Removed use of the XLATE\_UTILITY program. (ADB - 2014/03/28 HDFFV-8640)
* CMake: Added missing quotes in setting the CMAKE\_EXE\_LINKER\_FLAGS for the MPI option. (ADB - 2014/02/27 HDFFV-8674)
* CMake: Configuration of the HDF5 C++ or Fortran libraries with the thread-safety feature.

C++ and/or Fortran + thread-safe is enforced as a non-supported configuration. This matches the autotools. (DER - 2014/04/11)

* CMake: Configuration of static HDF5 C library with the thread-safety feature.

Static + thread-safe + Win32 threads is not a supported configuration due to the inability to automatically clean up thread-local storage. This is expected to be fixed in a future release. In the meantime, a work-around that uses internal functionality may allow the combination to be used without resource leaks. Contact the help desk for more information. (DER - 2014/04/11)

* Autotools: Several changes were done to configure and installcheck.

An export of LD\_LIBRARY\_PATH=<szip library location> was removed from configure; make installcheck was revised to run scripts installed in share/hdf5\_examples to use the installed h5cc, etc. to compile and run example source files also installed there.

Make installcheck will now fail when a shared szip or other external lib file cannot be found in the same manner that executables compiled and linked with h5cc will fail to run when those lib files cannot be found after install. Make installcheck should pass after setting LD\_LIBRARY\_PATH to the szip location. (LRK - 2014/04/16)

## Library

* A Gnu Make directive (.NOTPARALLEL) is added to fortran/test/Makefile.

AIX native make does not support this directive and would fail if parallel make (e.g. make -j4) is used to build the library. AIX users either do not use parallel make or install Gnu Make to build the library. (AKC 2014/04/08 HDFFV-8738)

* H5R.c: H5Rget\_name gave an assertion failure if the "name" parameter was NULL.

Fixed H5Rget\_name to return the size of the buffer needed to read a name of the referenced object in this case. The size doesn't include the NULL terminator. H5Rget\_name returns negative on failure. (MSB - 2014/01/22 HDFFV-8620)

* H5Z.c: H5Zfilter\_avail didn't check if a filter was available as a dynamically loaded filter. The error manifested itself in the h5repack tool when removing user-defined dynamically loaded filter.

Added a code to find the filter among the dynamically loaded filters after the function fails to find it among the registered filters. (ADB - 2014/03/03 HDFFV-8629)

* Memory leak: a memory leak was observed in conjunction to the H5TS\_errstk\_key\_g thread-local variable allocated in the H5E\_get\_stack function in H5E.c.

The shared HDF5 thread-safe library now no longer leaks thread-local storage resources on Windows with Win32 threads. Currently, there is no solution for this problem when HDF5 is statically built. We disabled the build of the static HDF5 thread-safe library with Win32 threads. (DER - 2014/04/11 HDFFV-8518)

* H5Dio.c: Improved handling of NULL pointers to H5Dread/H5Dwrite calls. Credit to Jason Newton (nevion@gmail.com) for the original patch.

H5Dwrite/read failed when a NULL pointer was passed for a data buffer and 0 elements were selected. Fixed. (QAK - 2014/04/16 HDFFV-8705)

* Deprecated API (1\_6 API): Improved handling of closing the library and re-accessing it with a deprecated routine.

When a program used a deprecated API (for example, H5Gcreate1), closed the library, and reopened it again to access a group, dataset, datatype, dataspace, attribute, or property list, HDF5 failed to provide an identifier for the object. Fixed. (NAF, QAK - 2014/04/16 HDFFV-8232)

## Parallel Library

* Fixed a missing H5F\_Provisional module in HDF5mpio.f90 (MSB - 2014/2/7 HDFFV-8651)

## Performance

* None

## Tools

* The h5diff tool would report that a datafile compared with an exact copy of the same datafile had differences. This was due to the issue below of reading un-written chunks. This problem is also fixed. (AKC - 2014/05/01 HDFFV-8637)
* The h5dump and h5diff utilities occasionally produced different output between Linux and Windows systems. This has been fixed.

This happened to datasets that used chunked storage, with default fill values, and some of the chunks had not been written. When the dataset was read, the library failed to write the default fill values to parts of the use buffer that were associated with the unwritten chunks. (JP - 2014/05/01 HDFFV-8247)

* The compress option is retired from bin/release. (AKC - 2014/04/25 HDFFV-8755)
* bin/release has a new option "zip" that produces a release zip file for the Windows platform. (AKC - 2014/04/24 HDFFV-8433)
* h5diff: Several failures relating to handling of strings attributes are fixed.

The tool crashed or gave an error message when one of the strings had fixed size type and another variable-length size type. h5diff now flags such strings as "not comparable". We plan to enhance the tool to handle strings of the different types in the future releases. (AKC - 2014/04/18 HDFFV-8625, 8639, 8745)

* h5repack: h5repack would not remove user-defined filters. Fixed by modifying h5repack to check if the filter is registered or can be dynamically loaded. (ADB - 2014/03/03 HDFFV-8629)

## F90 API

* H5D\_CHUNK\_CACHE\_NSLOTS\_DFLT\_F and H5D\_CHUNK\_CACHE\_NBYTES\_DFLT\_F were changed from the default KIND for INTEGER to INTEGER of KIND size\_t. (MSB - 2014/3/31 HDFFV-8689)

## C++ API

* Added throw() to all exception destructors. Credit to Jason Newton (nevion@gmail.com) for the patch. (BMR - 2014/4/15 HDFFV-8623)
* Changed the default value for H5Location::getComment from 256 to 0 to conform to C function and because it makes more sense. (BMR - 2014/4/15)

## High-Level APIs

* None

## Fortran High-Level APIs

* None

## Testing

* testhdf5 now exits with EXIT\_SUCCESS(0) if no errors, else EXIT\_FAILURE(1). (AKC - 2014/01/27 HDFFV-8572)
* The big test now pays attention to the HDF5\_DRIVER environment variable. Previously, it would run all tests with the family, stdio, and sec2 virtual file drivers (VFDs) for each VFD in the check-vfd make target, regardless of the variable setting. It now checks the variable and either runs the appropriate VFD-specific tests or skips as needed. This saves much testing time. Fixes HDFFV-8554. (DER - 2014/04/11)

# Supported Platforms

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

AIX 5.3 xlc 10.1.0.5

(NASA G-ADA) xlC 10.1.0.5

xlf90 12.1.0.6

gmake v3.82

Linux 2.6.18-308.13.1.el5PAE GNU C (gcc), Fortran (gfortran), C++ (g++)

#1 SMP i686 i686 i386 compilers for 32-bit applications;

(jam) Version 4.1.2 20080704 (Red Hat 4.1.2-54)

Version 4.8.2

PGI C, Fortran, C++ Compilers for 32-bit

applications;

Version 13.7-0

Intel(R) C, C++, Fortran Compiler for 32-bit

applications;

Version 14.0.2 (Build 20140120)

Linux 2.6.18-371.6.1.el5 GNU C (gcc), Fortran (gfortran), C++ (g++)

#1 SMP x86\_64 GNU/Linux compilers for 64-bit applications;

(koala) Version 4.1.2 20080704 (Red Hat 4.1.2-54)

Version 4.8.2

PGI C, Fortran, C++ for 64-bit target on

x86-64;

Version 13.7-0

Intel(R) C, C++, Fortran Compilers for

applications running on Intel(R) 64;

Version 14.0.2 (Build 20140120)

Linux 2.6.32-431.11.2.el6 GNU C (gcc), Fortran (gfortran), C++ (g++)

#1 SMP x86\_64 GNU/Linux compilers:

(platypus) Version 4.4.7 20120313

Version 4.8.2

PGI C, Fortran, C++ for 64-bit target on

x86-64;

Version 13.7-0

Intel(R) C (icc), C++ (icpc), Fortran (icc)

compilers:

Version 14.0.2 (Build 20140120)

Linux 2.6.32-431.11.2.el6.ppc64 gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-3)

#1 SMP ppc64 GNU/Linux g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-3)

(ostrich) GNU Fortran (GCC) 4.4.7 20120313 (Red Hat

4.4.7-3)

IBM XL C/C++ V11.1

IBM XL Fortran V13.1

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 2008 w/ Intel Fortran 14 (cmake)

Visual Studio 2010 w/ Intel Fortran 14 (cmake)

Visual Studio 2012 w/ Intel Fortran 14 (cmake)

Cygwin(CYGWIN\_NT-6.1 1.7.28(0.271/5/3)

gcc(4.8.2) compiler and gfortran)

(cmake and autotools)

Windows 7 x64 Visual Studio 2008 w/ Intel Fortran 14 (cmake)

Visual Studio 2010 w/ Intel Fortran 14 (cmake)

Visual Studio 2012 w/ Intel Fortran 14 (cmake)

Windows 8.1 Visual Studio 2012 w/ Intel Fortran 14 (cmake)

Windows 8.1 x64 Visual Studio 2012 w/ Intel Fortran 14 (cmake)

Mac OS X Lion 10.7.3 Apple clang/clang++ version 3.0 from Xcode 4.6.1

64-bit gfortran GNU Fortran (GCC) 4.6.2

(duck) Intel icc/icpc/ifort version 13.0.3

Mac OS X Mt. Lion 10.8.5 Apple clang/clang++ version 5.0 from Xcode

5.0.2

64-bit gfortran GNU Fortran (GCC) 4.8.2

(swallow/kite) Intel icc/icpc/ifort version 14.0.2

Mac OS X Mavericks 10.9.2 Apple clang/clang++ version 5.1 from Xcode 5.1

64-bit gfortran GNU Fortran (GCC) 4.8.2

(wren/quail) Intel icc/icpc/ifort version 14.0.2

# 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/ F90 C++ zlib SZIP

parallel F2003 parallel

Solaris2.11 32-bit n y/y n y y y

Solaris2.11 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/y n y y y

Windows 8.1 n y/y n y y y

Windows 8.1 x64 n y/y n y y y

Mac OS X Lion 10.7.3 64-bit n y/y n y y y

Mac OS X Mountain Lion 10.8.1 64-bit n y/y n y y y

Mac OS X Mavericks 10.9.1 64-bit n y/y n y y y

AIX 5.3 32- and 64-bit n y/n n y y y

CentOS 5.9 Linux 2.6.18-308 i686 GNU y y/y y y y y

CentOS 5.9 Linux 2.6.18-308 i686 Intel n y/y n y y y

CentOS 5.9 Linux 2.6.18-308 i686 PGI n y/y n y y y

CentOS 5.9 Linux 2.6.18 x86\_64 GNU n y/y n y y y

CentOS 5.9 Linux 2.6.18 x86\_64 Intel n y/y n y y y

CentOS 6.4 Linux 2.6.32 x86\_64 GNU y y/y y y y y

CentOS 6.4 Linux 2.6.32 x86\_64 Intel n y/y n y y y

CentOS 6.4 Linux 2.6.32 x86\_64 PGI n y/y n y y y

Linux 2.6.32-431.11.2.el6.ppc64 n y/n n y y y

OpenVMS IA64 V8.4 n y/n n y y n

Platform Shared Shared Shared Thread-

C libs F90 libs C++ libs 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 8.1 y y y y

Windows 8.1 x64 y y y y

Mac OS X Lion 10.7.3 64-bit y n y y

Mac OS X Mountain Lion 10.8.1 64-bit y n y y

Mac OS X Mavericks 10.9.1 64-bit y n y y

AIX 5.3 32- and 64-bit y n n y

CentOS 5.9 Linux 2.6.18-308 i686 GNU y y y y

CentOS 5.9 Linux 2.6.18-308 i686 Intel y y y n

CentOS 5.9 Linux 2.6.18-308 i686 PGI y y y n

CentOS 5.9 Linux 2.6.18 x86\_64 GNU y y y y

CentOS 5.9 Linux 2.6.18 x86\_64 Intel y y y n

CentOS 6.4 Linux 2.6.32 x86\_64 GNU y y y n

CentOS 6.4 Linux 2.6.32 x86\_64 Intel y y y n

CentOS 6.4 Linux 2.6.32 x86\_64 PGI y y y n

Linux 2.6.32-431.11.2.el6.ppc64 y y y n

OpenVMS IA64 V8.4 n n n 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 2.6.18-308.13.1.el5PAE MPICH mpich 3.1 compiled with

#1 SMP i686 i686 i386 gcc 4.8.2 and gfortran 4.8.2

(jam) g95 (GCC 4.0.3 (g95 0.94!)

Linux 2.6.18-431.11.2.el6 MPICH mpich 3.1 compiled with

#1 SMP x86\_64 GNU/Linux gcc 4.8.2 and gfortran 4.8.2

(platypus) g95 (GCC 4.0.3 (g95 0.94!)

FreeBSD 8.2-STABLE i386 gcc 4.5.4 [FreeBSD] 20110526

(loyalty) gcc 4.6.1 20110527

g++ 4.6.1 20110527

gfortran 4.6.1 20110527

FreeBSD 8.2-STABLE amd64 gcc 4.5.4 [FreeBSD] 20110526

(freedom) gcc 4.6.1 20110527

g++ 4.6.1 20110527

gfortran 4.6.1 20110527

Debian7.1.0 3.2.0-4-amd64 #1 SMP Debian 3.2.51-1 x86\_64 GNU/Linux

gcc (Debian 4.7.2-5) 4.7.2

GNU Fortran (Debian 4.7.2-5) 4.7.2

(cmake and autotools)

Fedora20 3.11.10-301.fc20.x86\_64 #1 SMP x86\_64 x86\_64 x86\_64 GNU/Linux

gcc (GCC) 4.8.2 20131212 (Red Hat 4.8.2-7)

GNU Fortran (GCC) 4.8.2 20130603 (Red Hat 4.8.2-7)

(cmake and autotools)

SUSE 13.1 3.11.6-4-desktop #1 SMP PREEMPT x86\_64 x86\_64 x86\_64 GNU/Linux

gcc (SUSE Linux) 4.8.1

GNU Fortran (SUSE Linux) 4.8.1

(cmake and autotools)

Ubuntu 13.10 3.11.0-13-generic #20-Ubuntu SMP x86\_64 GNU/Linux

gcc (Ubuntu/Linaro 4.8.1-10ubuntu8) 4.8.1

GNU Fortran (Ubuntu/Linaro 4.8.1-10ubuntu8) 4.8.1

(cmake and autotools)

Cray Linux Environment (CLE) PrgEnv-pgi/4.0.46

hopper.nersc.gov pgcc 12.5-0 64-bit target on x86-64 Linux -tp shanghai

pgf90 12.5-0 64-bit target on x86-64 Linux -tp shanghai

pgCC 12.5-0 64-bit target on x86-64 Linux -tp shanghai

# Known Problems

* When reading or writing a dataset (H5Dread/H5Dwrite) with a large selection size (e.g., 2GB ~= 500 million of 4 bytes integers or floating point numbers), some I/O systems may not be able to process it correctly. We advise users to find out system limits before using large selections. If I/O size limits exist, application should use HDF5 partial I/O capabilities (e.g., H5Sselect\_hyperslab(...)) to divide large requests into smaller sizes. In this case we also advise users not to use chunk storage sizes larger that 2GB since the HDF5 library performs I/O on the entire chunk. We will work on the HDF5 library to divide large data requests to smaller I/O requests. (AKC 2014/05/02 HDFFV-8479)
* Due to an Intel compiler bug introduced in version 14.0.1, the HDF5 FORTRAN wrappers do not work with configure option --enable-fortran2003. However, the option --enable-fortran works with Intel 14.0.1. The compiler bug was fixed in Intel version 14.0.2 and resolved the issue. (MSB - 2014/4/15)
* Due to a PGI compiler bug introduced in versions before 13.3 and versions after 14.2, the FORTRAN test 'Testing get file image' will fail. (MSB - 2014/4/15)
* On CYGWIN, when building the library dynamically, testing will fail on dynamically loaded filters. The test process will build dynamic filter libraries with the \*.dll.a extension, and the HDF5 Library will be looking for \*.so libraries. Entered as issue HDFFV-8736. (ADB - 2014/04/14)
* A Gnu Make directive (.NOTPARALLEL) is added to fortran/test/Makefile. AIX native make does not support this directive and would fail if parallel make (e.g. make -j4) is used to build the library. AIX users either do not use parallel make or install Gnu Make to build the library. (AKC 2014/04/08 HDFFV-8738)
* CLANG compiler with the options -fcatch-undefined-behavior and -ftrapv catches some undefined behavior in the alignment algorithm of the macro DETECT\_I in H5detect.c. Since the algorithm is trying to detect the alignment of integers, ideally the flag -fcatch-undefined-behavior should not to be used for H5detect.c. In the future, we can separate flags for H5detect.c from the rest of the library. (SLU - 2013/10/16 HDFFV-8147)
* Make provided by Solaris fails in "make check". Solaris users should use gmake to build and install the HDF5 software. (AKC - 2013/10/08 - HDFFV-8534)
* On OpenVMS, two soft conversion functions (H5T\_\_conv\_i\_f and H5T\_\_conv\_f\_i) have bugs. They convert data between floating-point numbers and integers. But the library's default is hard conversion. The user should avoid explicitly enabling soft conversion between floating-point numbers and integers. (Issue VMS-8; SLU - 2013/09/19)
* On OpenVMS, ZLIB 1.2.8 library doesn't work properly. ZLIB 1.2.5 works fine. So please use ZLIB 1.2.5 to build HDF5 library. (Issue VMS-5; SLU 2013/09/19)
* When building using the Cray compilers on Cray machines, HDF5 configure mistakenly thinks the compiler is an intel compiler and sets the -std=c99 flag which breaks configure on Cray. To build HDF5 properly on a Cray machine, please consult with the instructions in INSTALL\_parallel for building on Hopper. (MSC - 2013/04/26 - HDFFV-8429)
* The 5.9 C++ compiler on Sun failed to compile a C++ test ttypes.cpp. It complains with this message:

"/home/hdf5/src/H5Vprivate.h", line 130: Error: \_\_func\_\_ is not defined.

The reason is that \_\_func\_\_ is a predefined identifier in C99 standard. The HDF5 C library uses it in H5private.h. The test ttypes.cpp includes H5private.h (H5Tpkg.h<-H5Fprivate.h<-H5Vprivate.h<-H5private.h). Sun's 5.9 C++ compiler doesn't support \_\_func\_\_, thus fails to compile the C++ test. But Sun's 5.11 C++ compiler does. To check whether your Sun C++ compiler knows this identifier, try to compile the following simple C++ program:

#include<stdio.h>

int main(void)

{

printf("%s\n", \_\_func\_\_);

return 0;

}

(SLU - 2012/11/5)

* The C++ and FORTRAN bindings are not currently working on FreeBSD with the native release 8.2 compilers (4.2.1), but are working with gcc 4.6 from the ports (and probably gcc releases after that). (QAK - 2012/10/19)
* The following h5dump test case fails in BG/P machines (and potentially other machines that use a command script to launch executables):

h5dump --no-compact-subset -d "AHFINDERDIRECT::ah\_centroid\_t[0] it=0 tl=0" tno-subset.h5

This is due to the embedded spaces in the dataset name being interpreted by the command script launcher as meta-characters, thus passing three arguments to h5dump's -d flag. The command passes if run by hand, just not via the test script. (AKC - 2012/05/03)

* On hopper, the build failed when RUNSERIAL and RUNPARALLEL are set to aprun -np X, because the H5lib\_settings.c file was not generated properly. Not setting those environment variables works, because configure was able to automatically detect that it's a Cray system and used the proper launch commands when necessary. (MSC - 2012/04/18)
* The data conversion test dt\_arith.c fails in "long double" to integer conversion on Ubuntu 11.10 (3.0.0.13 kernel) with GCC 4.6.1 if the library is built with optimization -O3 or -O2. The older GCC (4.5) or newer kernal (3.2.2 on Fedora) doesn't have the problem. Users should lower the optimization level (-O1 or -O0) by defining CFLAGS in the command line of "configure" like:

CFLAGS=-O1 ./configure

This will overwrite the library's default optimization level. (SLU - 2012/02/07 - HDFFV-7829)

This issue is no longer present on Ubuntu 12.10 (3.5.0 kernel) with gcc 4.7.2.

* The STDIO VFD does not work on some architectures, possibly due to 32/64 bit or large file issues. The basic STDIO VFD test is known to fail on 64-bit SunOS 5.10 on SPARC when built with -m64 and 32-bit OS X/Darwin 10.7.0. The STDIO VFD test has been disabled while we investigate and a fix should appear in a future release. (DER - 2011/10/14 - HDFFV-8235)
* h5diff can report inconsistent results when comparing datasets of enum type that contain invalid values. This is due to how enum types are handled in the library and will be addressed in a future release. (DER - 2011/10/14 - HDFFV-7527)
* The links test can fail under the stdio VFD due to some issues with external links. This will be investigated and fixed in a future release. (DER - 2011/10/14 - HDFFV-7768)
* After the shared library support was fixed for some bugs, it was discovered that "make prefix=XXX install" no longer works for shared libraries. It still works correctly for static libraries. Therefore, if you want to install the HDF5 shared libraries in a location such as /usr/local/hdf5, you need to specify the location via the --prefix option during configure time. E.g, ./configure --prefix=/usr/local/hdf5 ... (AKC - 2011/05/07 - HDFFV-7583)
* The parallel test, t\_shapesame, in testpar/, may run for a long time and may be terminated by the alarm signal. If that happens, one can increase the alarm seconds (default is 1200 seconds = 20 minutes) by setting the environment variable, $HDF5\_ALARM\_SECONDS, to a larger value such as 3600 (60 minutes). Note that the t\_shapesame test may fail in some systems (see the "While working on the 1.8.6 release..." problem below). If it does, it will waste more time if $HDF5\_ALARM\_SECONDS is set to a larger value. (AKC - 2011/05/07)
* The C++ and FORTRAN bindings are not currently working on FreeBSD. (QAK - 2011/04/26)
* Shared Fortran libraries are not quite working on AIX. While they are generated when --enable-shared is specified, the fortran and hl/fortran tests fail. We are looking into the issue. HL and C++ shared libraries should now be working as intended, however. (MAM - 2011/04/20)
* The --with-mpe configure option does not work with Mpich2. (AKC - 2011/03/10)
* While working on the 1.8.6 release of HDF5, a bug was discovered that can occur when reading from a dataset in parallel shortly after it has been written to collectively. The issue was exposed by a new test in the parallel HDF5 test suite, but had existed before that. We believe the problem lies with certain MPI implementations and/or file systems.

We have provided a pure MPI test program, as well as a standalone HDF5 program, that can be used to determine if this is an issue on your system. They should be run across multiple nodes with a varying number of processes. These programs can be found at: <http://www.hdfgroup.org/ftp/HDF5/examples/known_problems/> (NAF - 2011/01/19)

* All the VFL drivers aren't backward compatible. In H5FDpublic.h, the structure H5FD\_class\_t changed in 1.8. There is new parameter added to get\_eoa and set\_eoa callback functions. A new callback function get\_type\_map was added in. The public function H5FDrealloc was taken out in 1.8. The problem only happens when users define their own driver for 1.6 and try to plug in 1.8 library. Because there's only one user complaining about it, we (Elena, Quincey, and I) decided to leave it as it is (see bug report #1279). Quincey will make a plan for 1.10. (SLU - 2010/02/02)
* The --enable-static-exec configure flag will only statically link libraries if the static version of that library is present. If only the shared version of a library exists (i.e., most system libraries on Solaris, AIX, and Mac, for example, only have shared versions), the flag should still result in a successful compilation, but note that the installed executables will not be fully static. Thus, the only guarantee on these systems is that the executable is statically linked with just the HDF5 library. (MAM - 2009/11/04)
* Parallel tests failed with 16 processes with data inconsistency at testphdf5 / dataset\_readAll. Parallel tests also failed with 32 and 64 processes with collective abort of all ranks at t\_posix\_compliant / allwrite\_allread\_blocks with MPI IO. (CMC - 2009/04/28)
* On an Intel 64 Linux cluster (RH 4, Linux 2.6.9) with Intel 10.0 compilers, use -mp -O1 compilation flags to build the libraries. A higher level of optimization causes failures in several HDF5 library tests.
* A dataset created or rewritten with a v1.6.3 library or after cannot be read with the v1.6.2 library or before when the Fletcher32 EDC filter is enabled. There was a bug in the calculation of the Fletcher32 checksum in the library before v1.6.3; the checksum value was not consistent between big- endian and little-endian systems. This bug was fixed in Release 1.6.3. However, after fixing the bug, the checksum value was no longer the same as before on little-endian system. Library releases after 1.6.4 can still read datasets created or rewritten with an HDF5 library of v1.6.2 or before. (SLU - 2005/06/30)