New Features in HDF5 Release 1.8.0
and Backward/Forward Format Compatibility Considerations

This table lists new features in HDF5 Release 1.8.0 that may create forward-compatibility conflicts.
Objects created or modified with any of these features may be inaccessible via HDF5 Libraries prior to Release 1.8.0.

Objects / Features	General	Groups and links			Attributes and object headers				Miscellaneous				
	Object creation or modification with "use latest format" property	Configurable compact-or-indexed link storage (compact and large groups; new group implementation)	User-defined (UD) and external links	Link creation order tracking and indexing	Faster access to large numbers of attributes	Large attributes (over ~64k)	Attribute creation order tracking and indexing	Shared object header messages (SOHM)	UNICODE support (UTF-8; non-ASCII character set encoding)	Object creation or copying with "create intermediate groups" property	Anonymous object creation	Compression filters: N-bit and scale-offset	NULL dataspace (H5Screate) (3)
File (superblock)	●							●					
Root group	●	●	●	■	■	■	■		■	○	○		
File	○												
Links										○	○		
Group	●	●	●	■	■	■	■		■	○	○		
Links		●								○	○		
Link (1)	✔	○	■	■					■		○		
Parent group										○			
Target object										●			
Dataset	●				■	■	■	●		○	○	●	■
Named datatype (2)	●				■	■	■			○	○		
Data object													
Attribute	●				■	■	■	●	■		○		
Object					■	■	■	●					
Object header message (4)	●				■	■	■	●			○		
Datatype (2)	●							●	■		○		
Data object													
Dataspace	●							●			✔		■

Key:
- ○   May affect backward/forward compatibility of object format.
- ●   Always/generally affects backward/forward compatibility of object format.
- ■   Used only with new-format objects; either a no-op on old formats or will fail.
- ✔   Does not affect backward/forward compatibility; either format neutral or run-time only (modifies format only while processing with no effect on file content).
- (blank)   Does not touch object in any way.

Notes:

(1) Under the new implementation, links may be stored as messages in an object header or in a type-2 B-tree in the group's local heap.

(2) By default, a datatype is stored as a message in the object header; a named datatype, which may be shared by several datasets, is stored as an independent object in the file.

(3) Several new features do not involve new or changed APIs.

(4) The following elements are stored (or can be stored; see notes 2 and 3) as messages in an object header.

(5) These APIs create coordinated sets of objects and metadata. Though individual elements are quite likely to be accessible through an older HDF5 library, that library will have no means of understanding the relationships among the various pieces.

(6) Compatibility problems will arise only if an application or the HDF5 Library crashed without properly closing a file with which this functionality was being used.