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

This table lists all new features in HDF5 Release 1.8.0. Features presented in a regular typeface may create forward-compatibility conflicts; objects created or modified with these features may be inaccessible via HDF5 Libraries prior to Release 1.8.0. Features presented in italics are designed to create no backward- or forward-compatibility conflicts.

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.

Feature columns (numbered):

General
1. Object creation or modification with "use latest format" property

Groups and links
2. Configurable compact-or-indexed link storage (compact and large groups; new group implementation)
3. User-defined (UD) and external links
4. Link creation order tracking and indexing
5. Link APIs (H5L) unrelated to UD and external links

Attributes and object headers
6. Enhanced attribute handling
7. Faster access to large numbers of attributes
8. Large attributes (over ~64k)
9. Attribute creation order tracking and indexing
10. Shared object header messages (SOHM)
11. Enhanced local heap size management (meta data for group)

Miscellaneous
12. UNICODE support (UTF-8; non-ASCII character set encoding)
13. Object API (H5O)
14. Object creation or copying with "create intermediate groups" property
15. Object copy properties
16. Improved object information retrieval
17. Name of referenced object
18. Anonymous object creation
19. New-style object open and object creation functions
20. User-defined identifiers
21. Other identifier API (H5I) enhancements
22. Compression filters: N-bit and scale-offset
23. User-defined datatype conversion callback functions
24. Datatype compiler conversion check
25. Integer-to-floating-point conversion support (H5Tconvert) (3)
26. Dataset array size reduction (H5Dset_extent)
27. Dataspace equivalence (H5Sextent_equal)
28. NULL dataspace (H5Screate) (3)
29. Direct I/O, bypassing system cache
30. Parallel I/O optimization (collective chunk I/O)
31. Enhanced error handling (H5E)
32. Meta data cache management (6)
33. File read/write status (H5Fget_intent)
34. Arithmetic data transform on I/O
35. Dataspace and datatype serial conversion

High level APIs
36. Two-way conversion between datatype and text description of datatype (H5LT)
37. New attribute datatype: long long (H5LT) (7)
38. Packet Table API (H5PT) (5)
39. Dimension Scale API (H5DS) (5)
40. Fortran APIs for HDF5 Light, Image, and Table interfaces (H5LT, H5IM, and H5TB)

Objects × Features grid:

Objects	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
File (superblock)	●				✔					●	✔		✔		✔	✔																✔								
Root group — File	●/○	●	●	■	✔		■	■	■		✔	■	✔	○	✔	✔	✔	○		✔	✔											✔								
Root group — Links														○				○																						
Group	●	●	●	■	✔		■	■	■		✔	■	✔	○	✔	✔	✔	○		✔	✔											✔								
Group — Links		●												○				○																						
Link (1)	✔	○	■	■	✔						✔	■	✔		✔			○											✔	✔		✔								
Link — Parent group												○																												
Link — Target object												●																												
Dataset	●				✔		■	■	■	●			✔	○	✔	✔	✔	○		✔	✔	●			✔	✔	✔	■	✔	✔		✔		✔	✔	✔		✔	✔	
Named datatype (2) — Data object	●				✔		■	■	■		✔		✔	○	✔	✔	✔	○					✔	✔	✔							✔		✔	✔	✔	✔	✔	✔	
Attribute	●					✔	■	■	■	●	✔	■			✔			○						✔					✔	✔		✔				✔		✔	✔	
Attribute — Object						✔	■	■	■	●																														
Object header message (4)	●				✔	✔	■	■	■	●	✔		✔		✔			○											✔	✔		✔								
Datatype (2) — Data object	●									●	✔	■	✔		✔			○					✔	✔	✔				✔	✔		✔			✔	✔	✔	✔	✔	
Dataspace	●									●	✔		✔		✔			✔								✔	✔	■	✔	✔		✔		✔			✔			

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.
(7) An API that was simply missing; old libraries always could read the datatype.