

RFC: Maintaining the HISTORY.txt and RELEASE.txt files in HDF5
or
Setting the History straight
Elena Pourmal, Albert Cheng
February 6, 2008

This document describes the current status of the HISTORY.txt and RELEASE.txt documentation files in the “release_docs” directory of the HDF5 1.8 branch source code and proposes a solution for their future maintenance.

Background information

Log of the on going and past HDF5 development and support work (new features, bug fixes, supported platforms, known problems, etc.) is documented in the RELEASE.txt and HISTORY.txt files in the ‘release_docs’ directory of the HDF5 main source <http://svn.hdfgroup.uiuc.edu/hdf5/trunk>

Current RELEASE.txt contains all development changes and support information between the v1.6.0 and v1.8.0-rc3 releases. It has more than 1400 lines of text.

Current HISTORY.txt contains release information for all releases between HDF5 1.0.0 alpha and HDF5 1.6.5. The file has more than 6000 lines of text.

Both files are very hard to maintain due to the growing sizes and lack of guidelines how to handle them when a new branch is created. We propose a solution that should address those concerns, preserve the history of the HDF5 development and will make maintenance of the documents simpler.

Proposed solution

For this proposal, a *branch* means either a separated branch or the main Trunk.

We propose that, starting with v1.8, each branch of hdf5_M_m (where M and m are the Major and minor version numbers respectively¹) shall contain the following two files:

- RELEASE.txt contains information about the development done since the last release of this branch, platforms and compilers for which the branch was tested, known problems, etc. *Note: some software packages use “CHANGES.txt” or “changes.log” names, but we decided to stick with the name we have been using for the last 10 years; also, we already have Changes.html that contains modifications in the APIs from one release to another.*

¹ See <http://www.hdfgroup.org/HDF5/doc/TechNotes/Version.html> for discussion of versioning in HDF5

- HISTORY_M_m.txt contains development history in the HDF5 source of all released versions M.m.* of this branch. One exception is the file HISTORY_1_0-1_8_0_rc3.txt which holds all history from version 1.0 through version 1.8.0rc3.

How the two files are maintained:

- Developers should put any information like changes, support platform, known problems, etc in the RELEASE.txt file.
- Right after a formal release is made on this branch, the RELEASE.txt is copied to the beginning of the HISTORY_M_N file and the RELEASE.txt is reset to clean state.
- Snapshots, alpha, beta releases are usually not considered as a formal release, therefore, no need to copy the release file into the history file. Exceptions are allowed, e.g., the trunk version has gone on a long time of changes and come to a watershed point, and the RELEASE.txt file can be added to the HISTORY_M_m.txt file and then reset for a clean slate.

Examples:

Imagine we have just made a release of v1.10.1, the following branches shall consist of these files in their corresponding “release_docs” directory,

- Branch hdf5_1_10:
 - RELEASE.txt (should be empty);
 - HISTORY_1_10.txt (contains v1.10.0 & v1.10.1 RELEASE.txt contents);
 - HISTORY_1_9.txt;
 - HISTORY_1_0-1_8_0_rc3.txt.
- Main trunk (v1.11):
 - RELEASE.txt (may have some text of changes already);
 - HISTORY_1_11.txt (may or may not be empty);
 - HISTORY_1_9.txt;
 - HISTORY_1_0-1_8_0_rc3.txt.
- Branch hdf5_1_8:
 - RELEASE.txt (may have some text of changes);
 - HISTORY_1_8.txt (contains all released v1.8.* RELEASE.txt content);
 - HISTORY_1_0-1_8_0_rc3.txt.