Maintaining the HDF5 Reference Manual

Ruth Aydt (aydt@hdfgroup.org)
Frank Baker (fbaker@hdfgroup.org)

The responsibility for writing and maintaining the entries in the HDF5 Reference Manual is shared by the development and documentation teams in The HDF Group. The process for keeping the Reference Manual entries up-to-date is outlined in this document.

1 Introduction

The HDF5 Reference Manual is of utmost importance to both the HDF5 user community and to The HDF Group technical staff. The responsibility for keeping the Reference Manual contents current, complete, and correct is shared between HDF5 developers and the documentation team (currently Frank and Ruth).

The process for supporting this shared responsibility has historically been undocumented. Moving forward, we are 1) taking steps to clearly define and explain the process, and 2) providing instructions, templates, and examples to help authors write high-quality Reference Manual entries.

Our goal is to make everyone aware of their roles and responsibilities, and to provide them with the information and tools they need to contribute effectively and efficiently to the final product.

We are on a journey toward this goal, and seek input from all involved on how we can work together to provide the best possible Reference Manual for our users and ourselves. Comments on this document, as well as on the supporting instructions, templates, and examples are welcome. Undoubtedly there are things we have left out that will need to be addressed. The process and the particulars will continue to evolve, and we will very likely embrace new technologies along the way. For now, take the first step in that direction with us.

2 Background

The HDF5 Reference Manual is currently coded in HTML, with each function description in a separate HTML file and a single index file for each API. For example, the function H5Acreate is described in the file RM/H5A/H5Acreate.htm and all of the H5A functions are linked from the file RM/RM_H5A.html. These files are kept in an SVN repository, maintained primarily by Frank. The HTML source files are uploaded to the website periodically, and are also used to generate a PDF version of the manual. These tasks are also carried out by Frank.

The one-function-per-file organization makes it somewhat easier for developers to work on a single entry as edits to one function or the addition of a new function no longer carry the risk of causing problems in other portions of the document.

While "raw" HTML is good from the perspective that it can be edited directly using any editor, it is bad from the perspective that it can be difficult to add and update content when you are not familiar with the various HTML tags. We anticipate moving to something other than HTML in the future, but that transition is unlikely to happen for quite some time.

Additional features, such as improved links and navigation, are also anticipated at a future date. An overarching long-term goal is not to duplicate content in multiple places, but to make it easily accessible from all relevant parts of the documentation set through links, navigation bars, and other technologies. We expect to have the Reference Manual entry documentation embedded in the source code at some point, and to use a documentation generator—such as *doxygen*—to extract the documentation and perform formatting.

3 Responsibilities

3.1 Developer Responsibilities

Developers write the functions and tools that are documented in the Reference Manual, and therefore are the experts on what the functions and tools do. Furthermore, the primary audiences for the Reference Manual are developers or technical users, so the development staff often has the best understanding of what the audience needs from a manual entry to use the function or tool. That said, the developers suffer from the "curse of knowledge"—their intimate knowledge of the function or tool, or of HDF5 as a whole, can make them blind to the aspects that are not obvious to someone without their background.

Each time a developer writes or updates an HDF5 function or tool, it is their responsibility to write or update the corresponding Reference Manual entry so that it is current, complete, and correct. In addition to making changes directly related to the work they have done on the function or tool, it is also their responsibility to bring the entry up to the latest standards and to correct any errors or deficiencies that may exist. The phrase "current, complete, and correct" corresponds not only to the format of the Reference Manual entry, but to the content as well. Terms should be used consistently, and "similar" entries should have "similar" phrasing.

The developers should keep the reader in mind and ask themselves "If I didn't know anything about this function/tool would this Reference Manual entry provide me with the information I need to use it correctly?" A reader must not be forced to go to the source code or tests to figure what the function actually does, out how to call the function, how to use the tool, what errors may occur, or what parameters really mean. There should be no surprises.

The developers are responsible for using the resources provided by the documentation team to perform their tasks, and for letting the documentation team know when the resources are unclear or insufficient. The documentation team can work with the developer on the specific entry, and improve the resources for the future. The developers should also ask other staff for help when they are not certain that a manual entry is clear and correct. Think of this as a "documentation review" – akin to a "code review."

A complete reference manual entry for a function should typically be written during the design phase, before coding begins. This practice, somewhat in the spirit of test-driven-design, can expose problems or ambiguities earlier in the development cycle when they are more easily addressed.

3.2 Documentation Team Responsibilities

The documentation team is responsible for providing the developers with the information they need to be successful in their task. This information can be written or "hands-on." The documentation team has the responsibility to improve its templates, guidelines, and processes, based on feedback from the developers. The information provided should cover not only the formatting of the Reference Manual entries, but also details about consistent phrasing and consistent use of terms. This information will evolve and become more extensive over time.

The written information provided by the documentation team can be found under this subversion directory:

```
http://svn.hdfgroup.uiuc.edu/hdf5doc/trunk/hdf5doc_support
```

The documentation team is responsible for final formatting and integration of the content into the Reference Manual, and may make final edits. In addition, they will work with the developers to ensure content across entries has the same style, is clear and consistent, and uses terms uniformly.

The documentation team is not responsible for consulting source code or running commands to clarify what a function or tool does in order to document it sufficiently, and may return a Reference Manual entry to the developer for further revision if it is not current, complete, and correct.

The documentation team is responsible for investigating and adopting technologies that will make the job easier for the developers, and that will produce high-quality and easily-used documentation for the community.

4 Process: Adding a New Reference Manual Entry

When a new function or tool is created, the developer is responsible for creating a corresponding Reference Manual entry. The process for doing this is outlined here. Ask for help from Frank if you have any questions along the way.

At any point in time, either during or after this process is complete, the entry may be "returned for revision" if it does not meet documentation standards. The return may be initiated by any member of the documentation or development staff who feels the entry is not current, complete, and correct. It may also be initiated by a report from a user who does not understand the entry or finds problems with it.

To create a new entry

1. Check out the Reference Manual resources (guidelines, templates, and examples) from the SVN repository. Always do an update or a fresh checkout to make sure you have the latest version:

   ```
   svn checkout http://svn.hdfgroup.uiuc.edu/hdf5doc/trunk/ hdf5doc_trunk
   ```

   The function entry template will be this file:

   ```
   hdf5doc_trunk/hdf5doc_support/templates/H5function_entry.htm
   ```

   The tool entry template will be this file:

   ```
   hdf5doc_trunk/hdf5doc _support/templates/h5tool_entry.htm
   ```

2. Make a copy of the template and rename it to match your new function or tool. For example:

   ```
   H5Onew_function.htm
   ```

3. Edit the file using your favorite text editor. (But *do not* edit it with an editor in which you cannot control the HTML coding; the resulting HTML will almost certainly be unacceptable. Sorry.)
4. Follow the instructions in this document and in the .htm file that was created from the template to update the individual sections. Remove material that is not relevant for your function or tool.
5. View the document in a browser to see if it has the expected content and format; correct any errors.
6. If you have any doubt about the correctness, completeness, or clarity of the document, ask a colleague to review it for you.
7. If you have any doubt about word choice or grammar, ask Frank or Ruth to review it for you.
8. When you are satisfied, add the manual entry to the appropriate directory of the hdf5doc repository and commit it. For example, if your new function were `H5Onew_function`, the filename would be `H5Onew_function.htm` and it would be added in the directory `hdf5doc_trunk/html/RM/H5O/`. Please be sure to include an informative log message.
9. Frank will complete the integration of the new function into the document set, including linking it into the appropriate index file(s). Be sure to tell him what branch or branches it pertains to.
10. Also let Frank know if other documents such as the User's Guide need to be updated. He will work with you to make those changes.

5 Process: Updating an Existing Reference Manual Entry

When an existing function or tool is updated, the developer is responsible for updating the corresponding Reference Manual entry. The updates should include not only changes that are directly related to the changes the developer made in the function or tool, but also any changes that are needed to bring the manual entry up to the latest documentation standards. The process for updating an entry is outlined here. Ask for help from Frank if you have any questions along the way.

At any point in time, either during or after this process is complete, the entry may be "returned for revision" if it does not meet documentation standards. The return may be initiated by any member of the documentation or development staff who feels the entry is not current, complete, and correct. It may also be initiated by a report from a user who does not understand the entry or finds problems with it.

To update an existing entry

1. Check out the Reference Manual resources (guidelines, templates, and examples) from the SVN repository. Always do an update or a fresh checkout to make sure you have the latest version:

   ```
   svn checkout http://svn.hdfgroup.uiuc.edu/hdf5doc/trunk/ hdf5doc_trunk
   ```

   The function entry template will be this file:

   ```
   hdf5doc_trunk/hdf5doc_support/templates/H5function_entry.htm
   ```

The tool entry template will be this file: **(Note: The tool template is not yet ready.)**

```
hdf5doc_trunk/hdf5doc_support/templates/h5tool_entry.htm
```

2. Find the function entry file that you will revise here:
   ```
   hdf5doc_trunk/html/RM/H5*/
   ```
   For example, if you are updating H5Screate, you will be editing this file:
   ```
   hdf5doc_trunk/html/RM/H5S/H5Screate.htm
   ```
3. Edit the file using your favorite text editor. (But *do not* edit it with an editor in which you cannot control the HTML coding; the resulting HTML will almost certainly be unacceptable. Sorry.)
4. Follow the instructions in this document and in the template file checked out in step 1 to update the individual sections. Remember that you will be making updates relevant to your coding changes, and also bringing the manual entry into compliance with the latest documentation standards.
5. View the document in a browser to see if it has the expected content and format – correct any errors.
6. If you have any doubt about the correctness, completeness, or clarity of the document, ask a colleague to review it for you.
7. If you have any doubt about word choice or grammar, ask Frank or Ruth to review it for you.
8. When you are satisfied, commit the HTML file for the manual entry to the hdf5doc repository. All commits must include a meaningful log message that will help a later developer determine where a change took place.
9. Then merge and commit the changes to all relevant branches. (See below.)
10. Also let Frank know if other documents such as the User's Guide need to be updated. He will work with you to make those changes.

6 Process: Correcting an Existing Reference Manual Entry

Current reference manual entries do not conform to the latest guidelines. Moving forward we will need to update those entries, but the process for doing that has not yet been defined.

If a developer notices an error in an existing manual entry that makes the entry incorrect or difficult to understand, they should file a bug report and include the correction that needs to be made, being as specific as possible about what needs to be changed, and hopefully supplying the correct text (in plain words, not formatted).

The documentation team will be responsible for making the correction, but will not necessarily bring the entry into full compliance with the new standards immediately. They will, however, correct inconsistent terminology and add IN/OUT designations for all parameters as they work on the specific problem reported.

7 Merging work from the trunk to a branch or across branches

In general, any particular task should first be completed in the trunk then ported to the relevant branches. With reasonable caution, Subversion's merge command is quite adequate to this task.

The following is a general outline of the requirements and steps; if you need additional information, feel free to speak with Frank.

- The hdf5doc/ documentation repository generally maintains the same branch and tag structure as the hdf5/ code repository; documentation is be added to the same hdf5doc/ branches as the hdf5/ branches your code is ported to.
  - Work that is strictly in development and not targeted to a current release series is generally limited to the trunk or a feature branch.
  - If work is targeted to one or more current release series, both the code and the documentation must be merged to those release branches.
- Subversion's merge command is quite adequate to this task, but a few precautions are necessary.
  - Subversion provides several approaches to merging. This discussion assumes that merges are done in a working copy, reviewed, and committed.
  - The branch you are merging to must be up to date.
  - Always check for conflicts before merging; try to avoid them. Conflicts are occasionally unavoidable; just fix them before committing the merged change.
  - Make sure the log message from the trunk commit accompanies the relevant commit in a branch.
  - See Frank if you are unfamiliar with the details of Subversion's merge function.

NOTE: This discussion does not treat the use of *feature branches*. These are special branches, usually based on the trunk, used in developing features that are kept isolated from other code until they are sufficiently developed to minimize disrupting the rest of the product. Feature branches introduce additional merging requirements.

8 Draft Function Entries in RFCs

Historically, function descriptions have been drafted in an RFC (generally in MS Word), and then converted to HTML for the Reference Manual once a feature is complete. Developing function descriptions in the provided HTML template then copying them into the RFC file will eliminate the need for the conversion.

This is strongly recommended.

Tests have demonstrated that copy-and-paste from Firefox into a .doc file produces reasonable formatting on both Windows and Mac. (Sorry, but IE and other browsers were not tested; Firefox was sufficient to provide proof of principle.)

The procedure would be as follows:

- Draft the function entries using the provided template.
- View the completed entries in Firefox and copy-and-paste them into the RFC (presumably a Word file).
- As the API evolves, update the function entries in the HTML file and repeat the copy-and-paste.

9 Conclusion

Current, complete, and correct HDF5 Reference Manual entries are important to both the HDF5 user community and to The HDF Group technical staff. High-quality Reference Manual entries can only be achieved through the collaborative efforts of the development and documentation staff. We will continue to evolve the process outlined in this document, as well as the supporting instructions, templates, and examples.

If everyone does their part, The HDF Group will deliver documentation that we can be proud of.

10 Open Items

The following items are incomplete, on hold, or not yet addressed:

- Examples are incomplete

The following template files are place-holders and will be completed as the need arises:

- `H5param_enum_entry.htm`: This template will be prepared in parallel with the first free-standing ENUM entry, anticipated in August or September 2009.
- `H5param_struct_entry.htm`: This template will be prepared in parallel with the next free-standing struct entry.
- `H5param_long_insert.htm`: This template will be created when it becomes clear that it is actually needed.

Revision History

October 1, 2008:	Version 1 circulated for comment within a subset of The HDF Group.
October 8, 2008:	Version 2 incorporates feedback received on Version 1. Circulated to technical staff of The HDF Group.
April 21, 2009:	Version 3 reflects the revised underlying file structure of Reference Manual (individual files per function entry) and includes a discussion of merging.
August 7, 2009	Version 4 includes a light update, the correction of a prior omission regarding the revised RM file structure, and the addition of the “Draft Function Entries in RFCs” and “Open Items” sections.