Contents (use only during writing of chapter)

1. Directions 2

1.1. Other Resources 2

2. HDF5 <Interface> 3

2.1. Feature Description: Introduction to <Interface> 3

2.1.1. How does it work? 4

2.1.2. Important Terms 4

2.1.3. Benefits 4

2.1.4. How is the feature connected to other features? 5

2.1.5. Examples 5

2.2. Programming 5

2.2.1. Programming Model 5

2.2.2. Commonly Used APIs 7

2.2.3. Best Practices, Special Issues 7

2.2.4. Function Listings 7

2.3. Tools 8

2.4. Related Topics 8

This table of contents is only included as a convenience while writing the chapter or topic; the table of contents can be deleted when the chapter or topic is imported into its final destination document.

# Directions

This document describes how to write documentation for a chapter in the *HDF5 User’s Guide* (UG) or for a smaller topic.

This template started with the more general design and RFC template Type-ProjectName-FeatureName-Identifier-DocVersionNumber.dotx which can be found in SVN at <https://svn.hdfgroup.org/hdf5doc/trunk/hdf5doc_support/templates>. Some of the differences between the general design template and this template include:

* No revision history chapter
  + This document may be revised by others, but once the initial work is done, the document will be incorporated into the UG or other document. Any further changes will be included in the UG’s “Update Status” section or the revision history of any other document.
* Only one chapter. Chapters in the UG have their title tagged with a Heading 1 or equivalent style. There is only one Heading 1 paragraph in each chapter in the UG. See the “HDF5 <Interface>” chapter on page 21 for more information.
* No Word Instructions. See “Other Resources” below.

## Other Resources

For more information on what should be included in a UG chapter, see “[The HDF5 User’s Guide Topic Template](https://confluence.hdfgroup.org/display/TECHDOC/The+HDF5+User's+Guide+Topic+Template)” in Confluence.

Other HDF Group writing help is available in Confluence on the [Dev Doc Start](https://confluence.hdfgroup.org/display/TECHDOC/Dev+Doc+Start) and [Documentation Style Sheet](https://confluence.hdfgroup.org/display/TECHDOC/Documentation+Style+Sheet) pages.

For help with Microsoft Word, see the Type-ProjectName-FeatureName-Identifier-DocVersionNumber.dotx template mentioned above and the [Building Expertise in Microsoft Word](https://confluence.hdfgroup.org/display/TECHDOC/Building+Expertise+in+Microsoft+Word) page in Confluence.

For general writing help, see the [Building Expertise in Written Communications](https://confluence.hdfgroup.org/display/TECHDOC/Building+Expertise+in+Written+Communications) page.

For a list of topics that can be worked on, see the [Possible UG Chapters and Topics](https://confluence.hdfgroup.org/display/TECHDOC/Possible+UG+Chapters+and+Topics) page in Confluence.

# HDF5 <Interface>

Chapter names take the format of **HDF5 <interface>** where **HDF5** is constant and **<interface>** is the name of the interface. For example, the chapter that describes HDF5 datasets is titled “HDF5 Datasets.”

The following is a basic outline of a UG chapter. These items are described in the sections below.

* Feature Description (see page 4)
  + What is it?
  + How does it work?
  + Important Terms
  + Benefits
  + How is it connected to other features?
* Programming (see page 7)
  + Model
  + Commonly Used APIs
  + Special Issues
  + Function Listings
* Tools (see page 11)
* Related Topics (see page 11)

## Feature Description: Introduction to <Interface>

This part of the chapter might also be called the Introduction. Titles might follow the format of **Introduction to <interface>** where **Introduction to** is constant and **<interface>** is the name of the interface. For the Datasets chapter this might be “Introduction to Datasets.”

See the [UG Template: Feature Description](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Feature+Description) page for more information.

What is it? The purpose of the feature description is to provide general information that will help users understand what the feature is and does. What the feature is or does can be broken down into smaller topics. To help users understand the feature, you might include the following sections: terms and definitions, feature workings, feature benefits, and bigger picture. These sub-topics are described further in sections below.

For short topics, all of the information described above (terms, definitions, bigger picture, and so on) might be included in a single introductory section.

For the high-level interfaces (H5LT, H5DM, H5PT, H5TB, H5IM), you might also make a general statement that describes how the interface simplifies programming. In other words, what operations does the interface simplify for developers?

Note that there may be overlap between the sections that describe features. For example, a section that talks about how a feature works might define some terms and might mention how the feature relates to other features. What matters most is that the information is described in a way that makes sense to readers.

### How does it work?

You can use comments about how something works to help define what the feature is. The following is a paragraph from the file image operations documentation that describes property lists and gives an idea of how they work.

The HDF5 property lists are a mechanism for passing values into HDF5 Library calls. They were created to allow calls to be extended with new parameters without changing the actual API or breaking existing code. They were designed based on the assumption that all new parameters would be “call by value” and not “call by reference.” Having “call by value” parameters means property lists can be copied, reused, and discarded with ease.

See the [UG Template: Feature Workings](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Feature+Workings) page for more information.

### Important Terms

The purpose of this section is to define the important terms related to the feature in this chapter readers should know.

After you have defined a term, consider adding it to the [*HDF5 Glossary*](https://www.hdfgroup.org/HDF5/doc/Glossary.html).

See the [UG Template: Terms and Definitions](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Terms+and+Definitions) page for more information on writing good definitions.

Assign developers to write definitions after terms that need to be defined have been collected.

### Benefits

The purpose of this section is to describe any benefits the feature might bring to users. In other words, why would a developer want to use a feature such as a property list? For example, a benefit of links is that they can be used to provide different pathways to the data. These pathways can emphasize different views of the raw data. One view might be time oriented. Another view of the same data might be location oriented.

See the [UG Template: Feature Benefits](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Feature+Benefits) page for more information.

### How is the feature connected to other features?

The purpose of this section is to describe how the feature relates to, depends on, or connects with other features. Some examples are the following:

* A dataspace is a component of a dataset.
* Groups hold datasets and other groups.
  + See section 4.2.2. The Hierarchy of Data Objects in the Groups chapter
  + See section 4.6. Examples of File Structures in the Groups chapter for some files with groups, links, and datasets
* Attributes can be attached to datasets and groups.

See the [UG Template: Bigger Picture](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Bigger+Picture) page for more information.

### Examples

Provide examples to help illustrate how a feature might be used.

The following are some examples from the UG:

* Section 4.1. “Introduction” and section 4.2.1. “The Group Object” in the Groups chapter
* Section 6.1. “Introduction and Definitions” in the Datatypes chapter
* Section 10.1. “Introduction” in the Property Lists chapter

## Programming

This section holds programming related information. Sub-sections include the programming model, descriptions of how common operations are done, any special issues, and function listings.

### Programming Model

The purpose of this section is to describe how to write a programming model.

What you are aiming for, the end result, when you write a programming model is a set of briefly stated steps followed by explanations, notes, or cross-references that clarify each step. The clarifying comments may include notes on how to do common operations such as read, write, or create and may include examples or code samples. As part of these comments, the APIs that are needed for the operations should be mentioned or discussed. For example, H5Dcreate is used to create a dataset. To clarify more complicated steps in the model, feel free to refer readers to other sections or chapters. For example, one of the main points in the dataset model mentions datatypes and dataspaces. See section 5.3.1. in the UG. Add a sentence something like see the "Datatypes" chapter for more information.

Examples help illustrate how an operation might be accomplished with a particular API or how the model as a whole might be implemented. See the “Programming Model for Groups” section, section 4.5., in the Groups chapter in the user’s guide. The model is listed at the beginning of the section. The sub-sections that follow comment on common group operations and include sample code.

“Creating and Initializing a Dataset,” section 2.2.2., is a good example of a programming model and code snippet. The section has three steps that users must go through to create a dataset. The steps are simple and clear. The steps are followed by a code example that includes the three steps.

If examples are included with the description of the model, put the model first in the section and the examples later.

Here is a model that Dana wrote on 11.4.2015 in an email in response to a forum question:

It's not something that we've ever tested or would be likely to support, but essentially:

1. Convince CMake to build the unsupported thread-safe + static configuration. There should be an "enable unsupported" option for this.
2. Possibly modify the HDF5 source code since our #ifdefs might assume that thread-safety and shared builds are standard on Windows.
3. Call H5TS\_win32\_thread\_enter/exit() in your own DllMain() code like we do in H5.c. Note that those are private functions, so you'll have to pull in our private headers.
4. If you plan on ever updating the version of HDF5 that we use, you'll need to hope we don't change the way we handle TLS in the future since you are using non-public-API functions.

I would definitely recommend against this, even if it can be made to work.

In this model, Dana is walking the user through a series of steps. In step 1, he uses the phrase “convince CMake to build….” He tells the user that the first step is to run CMake in a certain way without being too detailed. In step 3, he mentions a function call, and he also mentions private functions and that the user will have to do some extra work to make this step possible. In this model, Dana has made some general statements that will guide the user, and he has included some warnings about specific difficulties.

### Commonly Used APIs

If some commonly used APIs are mentioned in the programming model, comments describing how the APIs are used could be included in the programming model section. Alternatively, you could put the APIs and comments describing their use in a separate section. Separating the APIs from the programming model might make sense if you have long descriptions of each step or if you have a number of code examples.

### Best Practices, Special Issues

The purpose of this section is to collect any other information that might be helpful to those who use the feature. The following are some questions you might consider:

* Are there any best practices users should be made aware of?
* Are there any special situations, gotchas, or other issues that users should know about?
* Are there any issues that users should be cautious about?
* Can any programmer use the software, or should only developers with advanced skills use the software?
* Can you define a certain kind of experience that a developer needs in order to be able to work with the software?

Some examples can be found in the following sections:

* “Special Issues” section in the Attributes chapter (section 8.4.) where large attribute storage options are discussed
* “Working with Multiple HDF5 Files” in the File chapter (section 3.12.)
* “Notes” in the Property List chapter (section 10.6)

See the [UG Template: Best Practices](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Best+Practices) page for more information.

### Function Listings

The purpose of the function listings section is to list the functions and brief descriptions that will help users with the feature.

For some UG chapters, every function used by the interface could be listed in one of the Function Listing sections. Every function for some interfaces could be listed because there are not many functions for some interfaces. For example, H5R has seven functions, and H5A has 33 functions. However, some interfaces may have too many to handle easily. The property list interface has about 200 functions. The datatypes interface has about 80. We currently seem to take an approach where all of the functions are listed. What if we took a more illustrative or indicative approach? With an indicative approach, we would want to indicate the breadth or depth of the available functions; we would not want to list every function. We can refer readers to the RM for a complete list. Why? One reason is there are a lot of functions in some of the interfaces (for example, property lists and datatypes). Another is we don't want readers thinking the doc is out of date if we are late reminding them of a new feature: the RM should list every function. The UG does not have to list every function; the UG can say something like the following: here are the main functions, or here are some of the commonly used functions, and go to the RM for more information.

9.29.2015. What are some good ways to group the function listings? Some function listings are already grouped. Are these the best ways to group them?

See the [UG Template: Function Listings](https://confluence.hdfgroup.org/display/TECHDOC/UG+Template%3A+Function+Listings) page for more information.

## Tools

The purpose of this section is to include information on how to use HDF5 tools with the feature. For example, the Groups chapter has a section called “Using h5dump.” In the File chapter, see “h5dump” (3.5) and “h5repart” (3.10.8.1).

## Related Topics

The purpose of this section is to provide information on other topics related to the feature.