The Property Lists Chapter for the HDF5 User’s Guide

Getting Started

The purpose of this document is to help you write the “Property Lists” chapter for the *HDF5 User’s Guide*.

There are two chapters in this document. One, “Formatting Notes,” has some notes about how we use Microsoft Word to format a chapter in the *HDF5 User’s Guide*. The other, “HDF5 Property Lists,” is the shell of a User’s Guide chapter. The shell has some sections that commonly appear in chapters in the User’s Guide. You can start with these sections and add more sections as needed. Each section in the shell may have some comments or questions that describe or indicate information that you might want to add to the section.



<http://www.HDFGroup.org>

Contents

1. Formatting Notes 3

1.1. < Heading 2> 3

1.1.1. <Heading 3> 3

1.1.1.1. <Heading 4> 3

1.1.1.1.1. <Heading 5> 3

1.2. Sample Figures, Tables, Examples, Function Listings, and Sidebars 4

1.2.1. Sample Figure 5

1.2.2. Sample Example 6

1.2.3. Sample Table 8

1.2.4. Sample Function Listing 9

1.2.5. Sample Sidebar 10

1.3. Other Useful Notes 10

1.3.1. Cross-references 10

2. HDF5 Property Lists 13

2.1. Introduction 14

2.2. Programming Model 15

2.2.1. Available Property Lists 16

2.3. General Properties of Property Lists 16

2.3.1. Properties, Property Lists, and Property List Classes 16

2.4. Other Resources 17

2.5. Function Summaries 17

# Formatting Notes

This chapter has notes about how or when to use certain Microsoft Word features.

Use styles.

* Use Normal for text
* Use Normal for normal text in a table, example, figure, function listing, or sidebar
* Use Plain Text for function names or code samples
* Use a heading tag for chapter and section headings

Angle brackets mark spots where you should enter your text.

Headings and Contents. To generate or regenerate the table of contents, go through these steps:

* Tag text with the styles called Heading 1 through Heading 5
* Put the cursor in the area shaded gray under the Contents heading on page 3
* Hit the F9 function key, or open the References tab of the menu bar and click the Update Table button
* In the dialog box that opens, click on “Update entire table,” and then click OK

For more information, see the “Other Useful Notes” section on page 14.

In these User’s Guide chapters, there is only one title that is tagged with the Heading 1 style. The other Heading styles are shown below. Use as many headings as appropriate. You may not need Headings 4 and 5.

## < Heading 2>

<text>

### <Heading 3>

<text>

#### <Heading 4>

<text>

##### <Heading 5>

<text>

Sub-section

<text>

## Sample Figures, Tables, Examples, Function Listings, and Sidebars

This section has sample formatting for figures, tables, examples, function listings, and sidebars.

The following are some formatting guidelines:

* Caption formatting. Use the Normal style and bold. We are not treating captions as titles. Captions should only have the first word capitalized. We are also not treating captions as complete sentences: no period at the end of the caption is required. Any words that are usually capitalized such as someone’s name should be capitalized in a caption. For an example, see Figure 1 in the Programming Model section 2.2.5 of the **User’s Guide**.
* Function names and code. Use the Plain Text style. Plain Text uses a fixed space Courier font.
* Figures, tables, examples, and function listings are manually numbered.

### Sample Figure

Use the format below to set off a drawing or picture.

|  |
| --- |
| <Insert the drawing or picture here> |
| **Figure <???>. <Caption text in Normal style and bold>** |

The figure above is centered in the page. Drag the left or right margin to change the width of the example. The picture or drawing is also centered in the table row/cell, and there is only one cell per row. For an example, see Figure 2 in the Datasets chapter section 5.3.1 of the **User’s Guide**.

Here’s an example:

|  |
| --- |
|  |
| **Figure 12. Screenshot of HDFView** |

### Sample Example

Use the format below to set off example code.

|  |
| --- |
| <Example text. Use the Plain Text style. > |
| **Example <???>. <Caption text in Normal style and bold>** |

The example above is centered. Drag the left or right margin to change the width of the example.

We like to identify code (sample code or function names) with the Plain Text style. The Plain Text style uses a fixed space Courier font.

See the Datasets chapter in the **User’s Guide** for an example. Examples 1 and 2 are in section 5.3.2.

Here’s another example:

|  |
| --- |
| // Creating a file and dataset.  import ncsa.hdf.hdf5lib.H5;  import ncsa.hdf.hdf5lib.HDF5Constants;  public class H5\_CreateDataset  {  private static void CreateDataset() throws Exception  {  int fid=-1, sid=-1, did=-1;  long[] dims = { 4, 6 };  // Create a new file using default properties.  fid = H5.H5Fcreate("dset.h5", HDF5Constants.H5F\_ACC\_TRUNC,  HDF5Constants.H5P\_DEFAULT, HDF5Constants.H5P\_DEFAULT);  // Create the data space for the dataset.  sid = H5.H5Screate\_simple(2, dims, null);  // Create the dataset.  if ((fid >= 0) && (sid >= 0))  did = H5.H5Dcreate(fid, "/dset", HDF5Constants.H5T\_STD\_I32BE, sid,  HDF5Constants.H5P\_DEFAULT);  // End access to the dataset and release resources used by it.  if (did >= 0)  H5.H5Dclose(did);  // Terminate access to the data space.  if (sid >= 0)  H5.H5Sclose(sid);  // Close the file.  if (fid >= 0)  H5.H5Fclose(fid);  }  public static void main(String[] args)  {  try {  H5\_CreateDataset.CreateDataset();  } catch (Exception e) {  e.printStackTrace();  }  }  } |
| **Example 6. Example showing use of H5 methods** |

### Sample Table

Use the format below to show information in table form.

| **Table <?>. <Text of table caption in Normal style and bold>** | | |
| --- | --- | --- |
| **<Heading cell text in Normal style and Bold>** | **<Heading cell text in Normal style and Bold>** | **<Heading cell text in Normal style and Bold>** |
| <Text in Normal style> | <Text in Normal style> | <Text in Normal style> |
| <Text in Normal style> | <Text in Normal style> | <Text in Normal style> |
| <Text in Normal style> | <Text in Normal style> | <Text in Normal style> |
| <Text in Normal style> | <Text in Normal style> | <Text in Normal style> |

I tried using green for color of table row borders, but Word doesn’t seem to like that. Black is ok. The PDF borders are also black.

The table above is centered. Drag cell borders to the left or right to increase the width of a column. Drag the left or right table border to the left or right to change the width of the table.

One difference between the table above and tables in the **User’s Guide** is that the top row with the table number and caption will also appear on the next page if a table is on more than one page. This does not happen in the **User’s Guide**. I tagged the top two rows in the table above as Header Rows Repeat.

See Table 1 in section 5.3.2 of the Datasets chapter in the User’s Guide for an example.

Here’s an example:

| **Table 1. Common group, link, and object function calls** | |
| --- | --- |
| **To do this operation:** | **Use this function call:** |
| Create a group | H5Gcreate |
| Open a group | H5Gopen |
| Close a group | H5Gclose |
| Add an object to a group | H5Lcreate\_hard  H5Lcreate\_soft  H5Lcreate\_external |
| Remove an object from a group | H5Ldelete |
| Get the identifier for an object | H5Ovisit  H5Oget\_info |
| Get the identifiers for all of the objects in a group | H5Ovisit |
| Get the name of a link | H5Lget\_info |
| Get the names of all the links in a group | H5Literate  H5Lvisit |

### Sample Function Listing

Use the format below for function listings.

| **Function Listing <?>. <Text of function listing caption>** | |
| --- | --- |
| **C Function**  **Fortran Function** | **Purpose** |
| <Function name in Plain Text style> | <Text in Normal style> |
| <Function name in Plain Text style> | <Text in Normal style> |
| <Function name in Plain Text style> | <Text in Normal style> |

The function listing is a specialized table. The heading row can change in a table. In a function listing, the heading row is always the same. The left column holds C and/or Fortran function names. The Plain Text style should be used to format the function names. We like to identify code (sample code or function names) with the Plain Text style. The Plain Text style uses a fixed space Courier font. Use “(none)” if there is no C or Fortran function. See Function Listing 1 in section 5.2 of the Datasets chapter in the **User’s Guide** for an example.

The base text style of tables and function listings is Normal.

To get a blank space after a paragraph in a table or function listing, hit the Enter key.

This function listing is a useful way to list and describe functions in the **User’s Guide**. It might not be useful in other user guides.

### Sample Sidebar

Use the format below to highlight explanatory text that needs special emphasis.

|  |
| --- |
| **<Sidebar caption in Normal style and bold>** |
| <Sidebar text in Normal style> |

The sidebar text can be any number of paragraphs.

So far, I have not been numbering sidebars.

Sidebars are centered just like tables. Drag a side of the table to the left or right to change the width of the sidebar.

See the Groups chapter of the **User’s Guide** in the trunk of the doc repository for examples. Sections 4.2.3 and 4.2.4 have examples.

Here’s an example:

|  |
| --- |
| **Programming reminder** |
| Applications should keep track of every identifier that is returned to the application. After the application is finished using an object or file, the appropriate close function call should be executed with the identifier of the object or file that will be closed. See “Closing a Group” for more information. |

## Other Useful Notes

### Cross-references

There are different kinds of cross-references in Word. One kind refers to a bookmark much like the href tag in HTML. Go through the following steps to set up a bookmark and a cross-reference to that bookmark.

* Put the cursor at the point in the file where you want the bookmark.
* In the menu bar, click on the Insert tab, and then click on the Bookmark button. This opens the Bookmark dialog box. See the screenshot below.

|  |
| --- |
|  |
| **Figure 1. The Bookmark dialog box** |

* In the “Bookmark name” field, type in a new bookmark, and then click on the Add button.
  + The point at which a bookmark is placed can be seen or hidden. For example, a bookmark has been added to the word Add in the sentence above. This marker is not printed. To toggle the marker off and on, click File > Options > Advanced > Show bookmarks. The “Show bookmarks” field is in a set of fields under the “Show document content” heading.
  + Note that you can also use this dialog box to delete, move, or go to a bookmark.
* To position the cross-reference, move the cursor to where you want the page number to appear.
* In the menu bar, click on the Insert tab, and then click on the Cross-reference button. This opens the Cross-reference dialog box. See the screenshot below.

|  |
| --- |
|  |
| **Figure 2. The Cross-reference dialog box** |

* In the Cross-reference dialog box, select **Bookmark** in the “Reference type” field, select **Page number** in the “Insert reference to” field, select the bookmark in the “For which bookmark” field, and then click the Insert button. A page number appears.
* To update a page number, move the cursor to the page number, and then hit the F9 function key.
  + You can also regenerate a table of contents with the F9 function key. Regenerating the table of contents will also regenerate all the cross-references in the file.

Here is an example.

For more information, see the “Formatting Notes” section on page 4.

The page number in the sentence above is shaded in gray on a computer monitor. To change the shading, click File > Options > Advanced. Under the “Show document content” heading is an item called “Field shading.” Select a setting from the list box to change how cross-reference fields appear in the document. Fields that appear shaded on computer monitors are not printed with the shading.

# HDF5 Property Lists

The goal of this document is to help you write a chapter on property lists for the User’s Guide. The purpose of this chapter is to help you write content for the UG chapter and to help you organize the content.

General Document Content

The purpose of documentation is to help users work with the software we make. The information that is included in a document might vary quite a bit depending on the software and the users. Use the following questions to guide your writing:

* What is it?
  + What is the name of the software?
  + Is this piece of software part of a set?
  + Does it fit with or is it related to other software?
* How does it work?
  + What does the software do when it runs or when it is turned on?
  + What happens inside the software when it is working? Describe what happens under the hood.
  + How is the software started?
  + How is the software stopped?
* How can the software be configured? How is the software controlled?
  + What parameters are there?
  + What are the possible values for the parameters?
  + What are the default values for the parameters?
  + What arguments does the function expect? Possible and default values?
* What hardware or software is required?
  + What operating systems does this software work with?
    - We assume the software will work with all operating systems. Make a comment if this assumption is not true.
  + What versions of HDF software does the software work with?
  + What computers systems does the software work on?
    - We assume the software will work with all computer systems. Make a comment if this assumption is not true.
* What skills does the user need?
  + Can any programmer use the software, or should only developers with advanced skills use the software?
  + Is experience with callback functions required?
  + Can you define a certain kind of experience that a developer needs in order to be able to work with the software?
* Are there any issues that users should be cautious about?
* Why use it? What does the user get?
  + What are the benefits that the user might get from using the software?

Property List Chapter Content

The questions in the section above relate to documentation in general. The following are a few general questions for this chapter on property lists:

* What should new users know about property lists?
* Is there a key idea?
* What are the principles of property lists so to speak?
* What would help new users get started?
* What might help them use property lists fully?

Most property list functions are already listed in the chapter they are related to most. For example, the Datasets chapter has two function listings for property lists: one is for dataset creation property list functions, and the other is for dataset access property list functions. The purpose of this chapter is not to describe how every property list and every property list function works.

The sections in this chapter are sections that most chapters in the User’s Guide have. Each section has some questions that will help you think about what information should be included.

## Introduction

Tell us about property lists.

Questions:

* What is a property list?
* What are the benefits of property lists? In other words, why would a developer want to use a property list?
* What can users get when they use property lists?

The following is a paragraph from the file image operations doc:

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.

## Programming Model

In this section, tell us about how to develop software using property lists.

Questions:

* How do property lists work?
* How are property lists used? Include sample code. Diagrams can also be included.
* How are property lists configured? How are they set up? How are they changed?
* When can a property list be changed and when is it immutable?
* Which property lists are recorded in the file and which are transient?
* What property lists are available, and what does each do? See the “Available Property Lists” section below for more information.
* What are the most used property lists?
* What property lists seem to make the most difference when they are used?

Sample code or code snippets are usually set off from body text in a table that is formatted for the code. The following are examples from the File Image Operations doc:

|  |
| --- |
| typedef struct udata {  void \*init\_ptr;  size\_t init\_size;  int init\_ref\_count;  void \*mod\_ptr;  size\_t mod\_size;  int mod\_ref\_count;  } |
| **Example 1. Using a user data structure to communicate with an application** |

|  |
| --- |
| <allocate and initialize buf\_len and buf>  <allocate fapl\_id>  <set fapl to use Core file driver>  H5Pset\_file\_image(fapl\_id, buf, buf\_len);  <discard buf any time after this point>  <open file>  <discard fapl any time after this point>  <read and/or write file as desired, close> |
| **Example 2. Using H5Pset\_file\_image to initialize the Core file driver** |

### Available Property Lists

In this section, list and describe the available property lists. If possible, include a link to a section in the *HDF5 User’s Guide* for more information.

The following are the property lists currently mentioned in the *HDF5 Reference Manual* and in the *HDF5 User’s Guide*.

* Link Creation
* Link Access
* Object Copy and Object Creation
* Group Creation
* File Creation
* File Access
* Dataset Creation
* Dataset Access
* Dataset Memory
* Dataset Transfer
* Datatype Creation
* Datatype Access
* Attribute Creation

## General Properties of Property Lists

In this section, describe or define any general properties of property lists.

Questions:

* What are the general properties of property lists?
* Are there any generally applicable properties that are not in the set of general properties? Describe how they work and what they do.
* Are there any properties that can apply to many types of objects or can be used in a wide variety of circumstances? If there are, they could be discussed in this general chapter rather than in specific object contexts such as might be found in the Datasets chapter. Describe how these properties work and what they do.

### Properties, Property Lists, and Property List Classes

Describe the hierarchy of the property list universe.

## Other Resources

Are there other docs that might help users? One of the Advanced Topics describes File Image Operations. File Image Operations use property lists.

See these sections in the *HDF5 User’s Guide*:

* 1.2.7. Property List
* 3.1.2. File Creation and File Access Properties
* 3.7. File Property Lists and sub-sections
* 3.9.2. Example with the File Creation Property List
* 3.9.3. Example with File Access Property List

See the “H5P: Property List Interface” section in the *HDF5 Reference Manual* for these sets of functions:

* General Property List Operations
* Generic Property List Functions
* Link Creation Properties
* Link Access Properties
* Object Copy and Object Creation Properties
* Group Creation Properties
* File Creation Properties
* File Access Properties
* Dataset Creation Properties
* Dataset Access, Memory, and Transfer Properties

## Function Summaries

List and briefly describe property list functions in this section.

Any general functions could be listed in this section. A general function might be used with datasets and groups for example and might not be listed in the datasets or groups chapters. It’s possible that all the property list functions are related to a specific area such as groups or datasets. If that is the case, then there may not be any general property list functions that need to be listed in this section.

Are there any commonly used APIs? Putting the commonly used functions in this section might give users an idea of the functions they might use early on and regularly.

Here is a sample table formatted for function listings:

| **Function Listing <?>. <Text of function listing caption>** | |
| --- | --- |
| **C Function**  **Fortran Function** | **Purpose** |
| <Function name in Plain Text style> | <Text in Normal style> |
| <Function name in Plain Text style> | <Text in Normal style> |
| <Function name in Plain Text style> | <Text in Normal style> |

Here is a sample function listing from the File Image Operations doc. Functions used in file image operations are listed below. There are no Fortran functions.

| Function Listing 1. File image operations functions | |
| --- | --- |
| C Function  Fortran Function | Purpose |
| H5Pset\_file\_image  (none) | Allows an application to specify an initial file image. For more information, see page 12. |
| H5Pget\_file\_image  (none) | Allows an application to retrieve a copy of the file image designated for a VFD to use as the initial contents of a file. For more information, see page 12. |
| H5Pset\_file\_image\_callbacks  (none) | Allows an application to manage file image buffer allocation, copying, reallocation, and release. For more information, see page 13. |
| H5Pget\_file\_image\_callbacks  (none) | Allows an application to obtain the current file image callbacks from a file access property list. For more information, see page 16. |
| H5Fget\_file\_image  (none) | Provides a simple way to retrieve a copy of the image of an existing, open file. For more information, see page 18. |
| H5LTopen\_file\_image  (none) | Provides a convenient way to open an initial file image with the Core VFD. For more information, see page 19. |