Creating an Attribute

Attributes are small datasets that can be used to describe the nature and/or the intended usage of the object they are attached to. In this section, we show how to create, read, and write an attribute.

Creating an attribute

Creating an attribute is similar to creating a dataset. To create an attribute, the application must specify the object which the attribute is attached to, the datatype and dataspace of the attribute data, and the attribute creation property list.

The steps to create an attribute are as follows:

1. Obtain the object identifier that the attribute is to be attached to.
2. Define the characteristics of the attribute and specify the attribute creation property list.
    a. Define the datatype.
    b. Define the dataspace.
    c. Specify the attribute creation property list.
3. Create the attribute.
4. Close the attribute and datatype, dataspace, and attribute creation property list, if necessary.

To create and close an attribute, the calling program must use H5A_CREATE and H5A_CLOSE. For example:

```
C:
  attr_id = H5Acreate (dataset_id, "Units", H5T_STD_I32BE, dataspace_id, H5P_DEFAULT, H5P_DEFAULT)
  status = H5Aclose (attr_id);

FORTRAN:
  CALL h5acreate_f (dset_id, attr_nam, type_id, space_id, attr_id, &
                    hdferr, creation_prp=creat_plist_id)
  or
  CALL h5acreate_f (dset_id, attr_nam, type_id, space_id, attr_id, hdferr)

  CALL h5aclose_f (attr_id, hdferr)
```

Reading/Writing an attribute

Attributes may only be read or written as an entire object; no partial I/O is supported. Therefore, to perform I/O operations on an attribute, the application needs only to specify the attribute and the attribute's memory datatype.

The steps to read or write an attribute are as follows.

1. Obtain the attribute identifier.
2. Specify the attribute's memory datatype.
3. Perform the desired operation.
4. Close the memory datatype if necessary.

To read and/or write an attribute, the calling program must contain the H5A_READ and/or H5A_WRITE routines. For example:

```
C:
  status = H5Aread (attr_id, mem_type_id, buf);
  status = H5Awrite (attr_id, mem_type_id, buf);

FORTRAN:
  CALL h5awrite_f (attr_id, mem_type_id, buf, dims, hdferr)
  CALL h5aread_f (attr_id, mem_type_id, buf, dims, hdferr)
```

High Level APIs

The High Level HDF5 Lite APIs include functions that simplify and condense the steps for creating attributes in HDF5. Please be sure to review them, in addition to this tutorial.

Programming Example

Description

This example shows how to create and write a dataset attribute. It opens an existing file `dset.h5` in C (`dsetf.h5` in FORTRAN), obtains the identifier of the dataset `/dset`, defines the attribute's dataspace, creates the dataset attribute, writes the attribute, and then closes the attribute's dataspace, attribute, dataset, and file.

C

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 * Copyright by The HDF Group.                                               *
 * Copyright by the Board of Trustees of the University of Illinois.         *
 * All rights reserved.                                                      *
 *                                                                           *
 * This file is part of HDF5.  The full HDF5 copyright notice, including     *
 * terms governing use, modification, and redistribution, is contained in    *
 * the files COPYING and Copyright.html.  COPYING can be found at the root   *
 * of the source code distribution tree; Copyright.html can be found at the  *
 * root level of an installed copy of the electronic HDF5 document set and   *
 * is linked from the top-level documents page.  It can also be found at     *
 * http://hdfgroup.org/HDF5/doc/Copyright.html.  If you do not have          *
 * access to either file, you may request a copy from help@hdfgroup.org.     *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

/*
 *  This example illustrates how to create an attribute attached to a
 *  dataset.  It is used in the HDF5 Tutorial.
 */

#include "hdf5.h"
#define FILE "dset.h5"

int main() {

   hid_t       file_id, dataset_id, attribute_id, dataspace_id;  /* identifiers */
   hsize_t     dims;
   int         attr_data[2];
   herr_t      status;

   /* Initialize the attribute data. */
   attr_data[0] = 100;
   attr_data[1] = 200;

   /* Open an existing file. */
   file_id = H5Fopen(FILE, H5F_ACC_RDWR, H5P_DEFAULT);

   /* Open an existing dataset. */
   dataset_id = H5Dopen2(file_id, "/dset", H5P_DEFAULT);

   /* Create the data space for the attribute. */
   dims = 2;
   dataspace_id = H5Screate_simple(1, &dims, NULL);

   /* Create a dataset attribute. */
   attribute_id = H5Acreate2 (dataset_id, "Units", H5T_STD_I32BE, dataspace_id,
                              H5P_DEFAULT, H5P_DEFAULT);
```

```
/* Write the attribute data. */
status = H5Awrite(attribute_id, H5T_NATIVE_INT, attr_data);

/* Close the attribute. */
status = H5Aclose(attribute_id);

/* Close the dataspace. */
status = H5Sclose(dataspace_id);

/* Close to the dataset. */
status = H5Dclose(dataset_id);

/* Close the file. */
```

```
   status = H5Fclose(file_id);
}
```

Fortran

```
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
!   Copyright by The HDF Group.                                               *
!   Copyright by the Board of Trustees of the University of Illinois.         *
!   All rights reserved.                                                      *
!                                                                             *
!   This file is part of HDF5.  The full HDF5 copyright notice, including     *
!   terms governing use, modification, and redistribution, is contained in    *
!   the files COPYING and Copyright.html.  COPYING can be found at the root   *
!   of the source code distribution tree; Copyright.html can be found at the  *
!   root level of an installed copy of the electronic HDF5 document set and   *
!   is linked from the top-level documents page.  It can also be found at     *
!   http://hdfgroup.org/HDF5/doc/Copyright.html.  If you do not have          *
!   access to either file, you may request a copy from help@hdfgroup.org.     *
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
!
! This example shows how to create and write a dataset attribute.
! It opens the existing file 'dset.h5', obtains the identifier of
! the dataset "/dset", defines attribute's dataspace,
! creates dataset attribute, writes the attribute, and then closes
! the attribute's dataspace, attribute, dataset, and file.
!
! This example is used in the HDF5 Tutorial.

PROGRAM H5_CRTATT

  USE HDF5 ! This module contains all necessary modules

  IMPLICIT NONE

  CHARACTER(LEN=8), PARAMETER :: filename = "dsetf.h5" ! File name
  CHARACTER(LEN=4), PARAMETER :: dsetname = "dset"     ! Dataset name
  CHARACTER(LEN=9), PARAMETER :: aname = "attr_long"   ! Attribute name

  INTEGER(HID_T) :: file_id       ! File identifier
  INTEGER(HID_T) :: dset_id       ! Dataset identifier
  INTEGER(HID_T) :: attr_id       ! Attribute identifier
  INTEGER(HID_T) :: aspace_id     ! Attribute Dataspace identifier
  INTEGER(HID_T) :: atype_id      ! Attribute Dataspace identifier
  INTEGER(HSIZE_T), DIMENSION(1) :: adims = (/2/) ! Attribute dimension
  INTEGER     ::   arank = 1                     ! Attribure rank
  INTEGER(SIZE_T) :: attrlen    ! Length of the attribute string

  CHARACTER(LEN=80), DIMENSION(2) ::  attr_data  ! Attribute data

  INTEGER     ::   error ! Error flag
  INTEGER(HSIZE_T), DIMENSION(1) :: data_dims

  !
  ! Initialize attribute's data
  !
```

```
attr_data(1) = "Dataset character attribute"
attr_data(2) = "Some other string here     "
attrlen = 80
!
! Initialize FORTRAN interface.
!
CALL h5open_f(error)
!
! Open an existing file.
!
CALL h5fopen_f (filename, H5F_ACC_RDWR_F, file_id, error)
!
! Open an existing dataset.
!
CALL h5dopen_f(file_id, dsetname, dset_id, error)
!
! Create scalar data space for the attribute.
!
CALL h5screate_simple_f(arank, adims, aspace_id, error)
!
! Create datatype for the attribute.
!
CALL h5tcopy_f(H5T_NATIVE_CHARACTER, atype_id, error)
CALL h5tset_size_f(atype_id, attrlen, error)
!
! Create dataset attribute.
!
CALL h5acreate_f(dset_id, aname, atype_id, aspace_id, attr_id, error)
!
! Write the attribute data.
!
data_dims(1) = 2
CALL h5awrite_f(attr_id, atype_id, attr_data, data_dims, error)
!
! Close the attribute.
!
CALL h5aclose_f(attr_id, error)
!
! Terminate access to the data space.
!
CALL h5sclose_f(aspace_id, error)
!
! End access to the dataset and release resources used by it.
!
CALL h5dclose_f(dset_id, error)
!
! Close the file.
!
CALL h5fclose_f(file_id, error)
!
! Close FORTRAN interface.
!
CALL h5close_f(error)
```

```
END PROGRAM H5_CRTATT
```

C++

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 * Copyright by The HDF Group.           *
 * Copyright by the Board of Trustees of the University of Illinois.      *
 * All rights reserved.            *
 *                                                                         *
 * This file is part of HDF5.  The full HDF5 copyright notice, including     *
 * terms governing use, modification, and redistribution, is contained in    *
 * the files COPYING and Copyright.html.  COPYING can be found at the root   *
 * of the source code distribution tree; Copyright.html can be found at the  *
 * root level of an installed copy of the electronic HDF5 document set and   *
 * is linked from the top-level documents page.  It can also be found at     *
 * http://hdfgroup.org/HDF5/doc/Copyright.html.  If you do not have      *
 * access to either file, you may request a copy from help@hdfgroup.org.     *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

/*
 *  This example illustrates how to create an attribute attached to a
 *  dataset. It is used in the HDF5 Tutorial.
 */

#include <iostream>
#include <string>

#include "H5Cpp.h"

#ifndef H5_NO_NAMESPACE
    using namespace H5;
#endif

const H5std_string FILE_NAME( "h5tutr_dset.h5" );
const H5std_string DATASET_NAME( "dset" );
const H5std_string ATTR_NAME( "Units" );

const int DIM1 = 2;

int main (void)
{
   int attr_data[2] = { 100, 200};
   hsize_t dims[1] = { DIM1 };


   // Try block to detect exceptions raised by any of the calls inside it
   try
   {
 // Turn off the auto-printing when failure occurs so that we can
 // handle the errors appropriately
 Exception::dontPrint();

 // Open an existing file and dataset.
 H5File file( FILE_NAME, H5F_ACC_RDWR );
```

```
DataSet dataset = file.openDataSet( DATASET_NAME );

// Create the data space for the attribute.
DataSpace attr_dataspace = DataSpace (1, dims );

// Create a dataset attribute.
Attribute attribute = dataset.createAttribute( ATTR_NAME, PredType::STD_I32BE,
                                               attr_dataspace);

// Write the attribute data.
attribute.write( PredType::NATIVE_INT, attr_data);

  }  // end of try block

  // catch failure caused by the H5File operations
  catch( DataSpaceIException error )
  {
error.printError();
return -1;
  }

  // catch failure caused by the H5File operations
  catch( AttributeIException error )
  {
error.printError();
return -1;
  }

  // catch failure caused by the H5File operations
  catch( FileIException error )
  {
error.printError();
return -1;
  }

  // catch failure caused by the DataSet operations
  catch( DataSetIException error )
  {
error.printError();
return -1;
  }

  return 0;  // successfully terminated
```

```
}
```

Python

See HDF5 Introductory Examples for the examples used in the Learning the Basics tutorial. There are examples for several other languages, including Java.

For details on compiling an HDF5 application: [Compiling HDF5 Applications]

Remarks

H5A_CREATE creates an attribute which is attached to the object specified by the first parameter, and returns an identifier.

H5A_WRITE writes the entire attribute, and returns the status of the write.

When an attribute is no longer accessed by a program, H5A_CLOSE must be called to release the attribute from use. An `H5Aclose`/`h5aclose_f` call is mandatory.

File Contents

The contents of `dset.h5` (`dsetf.h5` for FORTRAN) and the attribute definition are shown below:

Fig. 7.1a *`dset.h5` in DDL*

```
HDF5 "dset.h5" {
GROUP "/" {
DATASET "dset" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 4, 6 ) / ( 4, 6 ) }
DATA {
 1, 2, 3, 4, 5, 6,
 7, 8, 9, 10, 11, 12,
 13, 14, 15, 16, 17, 18,
 19, 20, 21, 22, 23, 24
}
ATTRIBUTE "attr" {
 DATATYPE { H5T_STD_I32BE }
 DATASPACE { SIMPLE ( 2 ) / ( 2 ) }
 DATA {
    100, 200
 }
}
}
}
}
```

Fig. 7.1b *`dsetf.h5` in DDL*

```
HDF5 "dsetf.h5" {
GROUP "/" {
DATASET "dset" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 6, 4 ) / ( 6, 4 ) }
DATA {
 1, 7, 13, 19,
 2, 8, 14, 20,
 3, 9, 15, 21,
 4, 10, 16, 22,
 5, 11, 17, 23,
 6, 12, 18, 24
}
ATTRIBUTE "attr" {
 DATATYPE { H5T_STD_I32BE }
 DATASPACE { SIMPLE ( 2 ) / ( 2 ) }
 DATA {
    100, 200
 }
}
}
}
}
```

Attribute Definition in DDL

Fig. 7.2 *HDF5 Attribute Definition*

```
<attribute> ::= ATTRIBUTE "<attr_name>" { <datatype>
           <dataspace>
           <data>  }
```