Creating Groups using Absolute and Relative Names

Recall that to create an HDF5 object, we have to specify the location where the object is to be created. This location is determined by the identifier of an HDF5 object and the name of the object to be created. The name of the created object can be either an absolute name or a name relative to the specified identifier. In the previous example, we used the file identifier and the absolute name `/MyGroup` to create a group.

In this section, we discuss HDF5 names and show how to use absolute and relative names.

Names

HDF5 object names are a slash-separated list of components. There are few restrictions on names: component names may be any length except zero and may contain any character except slash (`/`) and the null terminator. A full name may be composed of any number of component names separated by slashes, with any of the component names being the special name `.` (a dot or period). A name which begins with a slash is an *absol ute name* which is accessed beginning with the root group of the file; all other names are *relative names* and and the named object is accessed beginning with the specified group. A special case is the name `/` (or equivalent) which refers to the root group.

Functions which operate on names generally take a location identifier, which can be either a file identifier or a group identifier, and perform the lookup with respect to that location. Several possibilities are described in the following table:

Location Type	Object Name	Description
File identifier	`/foo/bar`	The object `bar` in group `foo` in the root group.
Group identifier	`/foo/bar`	The object `bar` in group `foo` in the root group of the file containing the specified group. In other words, the group identifier's only purpose is to specify a file.
File identifier	`/`	The root group of the specified file.
Group identifier	`/`	The root group of the file containing the specified group.
Group identifier	`foo/bar`	The object `bar` in group `foo` in the specified group.
File identifier	`.`	The root group of the file.
Group identifier	`.`	The specified group.
Other identifier	`.`	The specified object.

Programming Example

Description

The following example code shows how to create groups using absolute and relative names. It creates three groups: the first two groups are created using the file identifier and the group absolute names while the third group is created using a group identifier and a name relative to the specified group.

C

```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 the creation of groups using absolute and
 *  relative names.  It is used in the HDF5 Tutorial.
 */

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

int main() {

   hid_t       file_id, group1_id, group2_id, group3_id;  /* identifiers */
   herr_t      status;

   /* Create a new file using default properties. */
   file_id = H5Fcreate(FILE, H5F_ACC_TRUNC, H5P_DEFAULT, H5P_DEFAULT);

   /* Create group "MyGroup" in the root group using absolute name. */
   group1_id = H5Gcreate2(file_id, "/MyGroup", H5P_DEFAULT, H5P_DEFAULT,
H5P_DEFAULT);

   /* Create group "Group_A" in group "MyGroup" using absolute name. */
   group2_id = H5Gcreate2(file_id, "/MyGroup/Group_A", H5P_DEFAULT, H5P_DEFAULT,
H5P_DEFAULT);

   /* Create group "Group_B" in group "MyGroup" using relative name. */
   group3_id = H5Gcreate2(group1_id, "Group_B", H5P_DEFAULT, H5P_DEFAULT,
H5P_DEFAULT);

   /* Close groups. */
   status = H5Gclose(group1_id);
   status = H5Gclose(group2_id);
   status = H5Gclose(group3_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.     *
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
!
!
! The following example code shows how to create groups
! using absolute and relative names. It creates three groups:
! the first two groups are created using the file identifier and
! the group absolute names, and the third group is created using
! a group identifier and the name relative to the specified group.
!
! This example is used in the HDF5 Tutorial.

PROGRAM H5_CRTGRPAR

  USE HDF5 ! This module contains all necessary modules

  IMPLICIT NONE

  CHARACTER(LEN=10), PARAMETER :: filename = "groupsf.h5" ! File name
  CHARACTER(LEN=8),  PARAMETER :: groupname1 = "/MyGroup" ! Group name
  CHARACTER(LEN=16), PARAMETER :: groupname2 = "/MyGroup/Group_A"
                                                          ! Group name
  CHARACTER(LEN=7),  PARAMETER :: groupname3 = "Group_B"  ! Group name

  INTEGER(HID_T) :: file_id       ! File identifier
  INTEGER(HID_T) :: group1_id, group2_id, group3_id ! Group identifiers

  INTEGER     ::   error ! Error flag
  !
  ! Initialize FORTRAN interface.
  !
  CALL h5open_f(error)

  !
  ! Create a new file using default properties.
  !
  CALL h5fcreate_f(filename, H5F_ACC_TRUNC_F, file_id, error)

  !
  ! Create group "MyGroup" in the root group using absolute name.
  !
  CALL h5gcreate_f(file_id, groupname1, group1_id, error)

  !
  ! Create group "Group_A" in group "MyGroup" using absolute name.
  !
  CALL h5gcreate_f(file_id, groupname2, group2_id, error)
```

```
!
! Create group "Group_B" in group "MyGroup" using relative name.
!
CALL h5gcreate_f(group1_id, groupname3, group3_id, error)

!
! Close the groups.
!
CALL h5gclose_f(group1_id, error)
CALL h5gclose_f(group2_id, error)
CALL h5gclose_f(group3_id, error)

!
! Terminate access to the file.
!
CALL h5fclose_f(file_id, error)

!
! Close FORTRAN interface.
!
CALL h5close_f(error)
```

```
END PROGRAM H5_CRTGRPAR
```

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 the creation of groups using absolute and
 *  relative names. It is used in the HDF5 Tutorial.
 */

#include <iostream>
#include <string>

#include "H5Cpp.h"

#ifndef H5_NO_NAMESPACE
    using namespace H5;
#ifndef H5_NO_STD
    using std::cout;
    using std::endl;
#endif  // H5_NO_STD
#endif

const H5std_string FILE_NAME("h5tutr_groups.h5");

int main(void)
{

    // 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();

 // Create a new file using default properties.

 H5File file(FILE_NAME, H5F_ACC_TRUNC);
```

```
 // Create group "MyGroup" in the root group using an absolute name.

 Group group1(file.createGroup( "/MyGroup"));

 // Create group "Group_A" in group "MyGroup" using an
 // absolute name.

 Group group2(file.createGroup("/MyGroup/Group_A"));

 // Create group "Group_B" in group "MyGroup" using a
 // relative name.

 Group group3(group1.createGroup ("Group_B"));

 // Close the groups and file.

 group1.close();
 group2.close();
 group3.close();
 file.close();

    } // end of try block

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

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

    return 0;
}
```

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: [Compile Information]

Remarks

H5G_CREATE creates a group at the location specified by a location identifier and a name. The location identifier can be a file identifier or a group identifier and the name can be relative or absolute.

The first `H5Gcreate/h5gcreate_f` creates the group `MyGroup` in the root group of the specified file.

The second `H5Gcreate/h5gcreate_f` creates the group `Group_A` in the group `MyGroup` in the root group of the specified file. Note that the parent group (`MyGroup`) already exists.

The third `H5Gcreate/h5gcreate_f` creates the group `Group_B` in the specified group.

File Contents

The file contents are shown below:

Fig. 9.1 *The Contents of `groups.h5` (`groupsf.h5` for FORTRAN)*

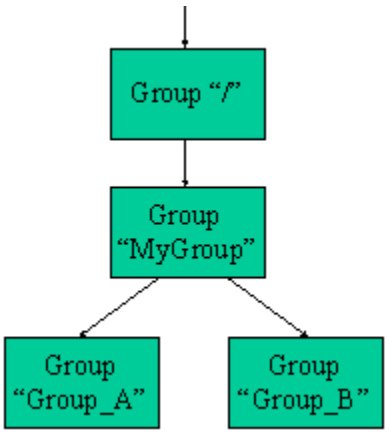

Fig. 9.2 *`groups.h5` in DDL (for FORTRAN, the name in the first line is `groupsf.h5`)*

```
HDF5 "groups.h5" {
GROUP "/" {
 GROUP "MyGroup" {
    GROUP "Group_A" {
    }
    GROUP "Group_B" {
    }
 }
}
}
```