



```
/* Create the data space for the first dataset. */
dims[0] = 3;
dims[1] = 3;
dataspace_id = H5Screate_simple(2, dims, NULL);

/* Create a dataset in group "MyGroup". */
dataset_id = H5Dcreate2(file_id, "/MyGroup/dset1", H5T_STD_I32BE, dataspace_id,
                        H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);

/* Write the first dataset. */
status = H5Dwrite(dataset_id, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_DEFAULT,
                  dset1_data);

/* Close the data space for the first dataset. */
status = H5Sclose(dataspace_id);

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

/* Open an existing group of the specified file. */
group_id = H5Gopen2(file_id, "/MyGroup/Group_A", H5P_DEFAULT);

/* Create the data space for the second dataset. */
dims[0] = 2;
dims[1] = 10;
dataspace_id = H5Screate_simple(2, dims, NULL);

/* Create the second dataset in group "Group_A". */
dataset_id = H5Dcreate2(group_id, "dset2", H5T_STD_I32BE, dataspace_id,
                        H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);

/* Write the second dataset. */
status = H5Dwrite(dataset_id, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_DEFAULT,
                  dset2_data);

/* Close the data space for the second dataset. */
status = H5Sclose(dataspace_id);

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

/* Close the group. */
status = H5Gclose(group_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 a dataset in a particular group.
! It opens the file created in the previous example and creates two datasets.
! Absolute and relative dataset names are used.
!
! This example is used in the HDF5 Tutorial.

PROGRAM H5_CRTGRPD

  USE HDF5 ! This module contains all necessary modules

  IMPLICIT NONE

  CHARACTER(LEN=10), PARAMETER :: filename = "groupsf.h5" ! File name
  CHARACTER(LEN=15), PARAMETER :: groupname = "MyGroup/Group_A" ! Group name
  CHARACTER(LEN=13), PARAMETER :: dsetname1 = "MyGroup/dset1" ! Dataset name
  CHARACTER(LEN=5), PARAMETER :: dsetname2 = "dset2" ! dataset name

  INTEGER(HID_T) :: file_id ! File identifier
  INTEGER(HID_T) :: group_id ! Group identifier
  INTEGER(HID_T) :: dataset_id ! Dataset identifier
  INTEGER(HID_T) :: dataspace_id ! Data space identifier

  INTEGER :: i, j
  INTEGER :: error ! Error flag

  INTEGER, DIMENSION(3,3) :: dset1_data ! Data arrays
  INTEGER, DIMENSION(2,10) :: dset2_data !

  INTEGER(HSIZE_T), DIMENSION(2) :: dims1 = (/3,3/) ! Datasets dimensions
  INTEGER(HSIZE_T), DIMENSION(2) :: dims2 = (/2,10/)
  INTEGER(HSIZE_T), DIMENSION(2) :: data_dims

  INTEGER :: rank = 2 ! Datasets rank
```

```

!
!Initialize dset1_data array
!
DO i = 1, 3
  DO j = 1, 3
    dset1_data(i,j) = j;
  END DO
END DO

!
!Initialize dset2_data array
!
DO i = 1, 2
  DO j = 1, 10
    dset2_data(i,j) = j;
  END DO
END DO

!
! Initialize FORTRAN interface.
!
CALL h5open_f(error)

!
! Open an existing file.
!
CALL h5fopen_f (filename, H5F_ACC_RDWR_F, file_id, error)

!
! Create the data space for the first dataset.
!
CALL h5screate_simple_f(rank, dims1, dataspace_id, error)

!
! Create a dataset in group "MyGroup" with default properties.
!
CALL h5dcreate_f(file_id, dsetname1, H5T_NATIVE_INTEGER, dataspace_id, &
  dataset_id, error)

!
! Write the first dataset.
!
data_dims(1) = 3
data_dims(2) = 3
CALL h5dwrite_f(dataset_id, H5T_NATIVE_INTEGER, dset1_data, data_dims, error)

!
! Close the dataspace for the first dataset.
!
CALL h5sclose_f(dataspace_id, error)

!
! Close the first dataset.
!
CALL h5dclose_f(dataset_id, error)

!
! Open an existing group in the specified file.
!

```

```
CALL h5gopen_f(file_id, groupname, group_id, error)

!
!Create the data space for the second dataset.
!
CALL h5screate_simple_f(rank, dims2, dataspace_id, error)

!
! Create the second dataset in group "Group_A" with default properties.
!
CALL h5dcreate_f(group_id, dsetname2, H5T_NATIVE_INTEGER, dataspace_id, &
    dataset_id, error)

!
! Write the second dataset.
!
data_dims(1) = 2
data_dims(2) = 10
CALL h5dwrite_f(dataset_id, H5T_NATIVE_INTEGER, dset2_data, data_dims, error)

!
! Close the dataspace for the second dataset.
!
CALL h5sclose_f(dataspace_id, error)

!
! Close the second dataset.
!
CALL h5dclose_f(dataset_id, error)

!
! Close the group.
!
CALL h5gclose_f(group_id, error)

!
! Close the file.
!
CALL h5fclose_f(file_id, error)

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



```

// Turn off the auto-printing when failure occurs so that we can
// handle the errors appropriately
Exception::dontPrint();

// Initialize the first dataset.
for (i = 0; i < D1DIM1; i++)
    for (j = 0; j < D1DIM2; j++)
        dset1_data[i][j] = j + 1;

// Initialize the second dataset.
for (i = 0; i < D2DIM1; i++)
    for (j = 0; j < D2DIM2; j++)
        dset2_data[i][j] = j + 1;

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

        // Create the data space for the first dataset. Note the use of
        // pointer for the instance 'dataspace'. It can be deleted and
        // used again later for another data space. An HDF5 identifier is
        // closed by the destructor or the method 'close()'.
hsize_t dims[RANK]; // dataset dimensions
dims[0] = D1DIM1;
dims[1] = D1DIM2;
DataSpace *dataspace = new DataSpace (RANK, dims);

// Create the dataset in group "MyGroup". Same note as for the
// dataspace above.
DataSet *dataset = new DataSet (file.createDataSet(DATASET_NAME1,
        PredType::STD_I32BE, *dataspace));

// Write the data to the dataset using default memory space, file
// space, and transfer properties.
dataset->write(dset1_data, PredType::NATIVE_INT);

// Close the current dataset and data space.
delete dataset;
delete dataspace;

// Create the data space for the second dataset.
dims[0] = D2DIM1;
dims[1] = D2DIM2;
dataspace = new DataSpace (RANK, dims);

// Create group "Group_A" in group "MyGroup".
Group group(file.openGroup("/MyGroup/Group_A"));

// Create the second dataset in group "Group_A".
dataset = new DataSet (group.createDataSet(DATASET_NAME2,
        PredType::STD_I32BE, *dataspace));

// Write the data to the dataset using default memory space, file
// space, and transfer properties.
dataset->write(dset2_data, PredType::NATIVE_INT);

// Close all objects.
delete dataspace;
delete dataset;
group.close();

```

```
    } // end of try block

    // catch failure caused by the H5File operations
    catch(FileIOException error)
    {
    error.printStackTrace();
    return -1;
    }
    // catch failure caused by the DataSet operations
    catch(DataSetIOException error)
    {
    error.printStackTrace();
    return -1;
    }

    // catch failure caused by the DataSpace operations
    catch(DataSpaceIOException error)
    {
    error.printStackTrace();
    return -1;
    }

    // catch failure caused by the Group operations
    catch(GroupIOException error)
    {
    error.printStackTrace();
    return -1;
    }
```



```

    return 0;
}

```

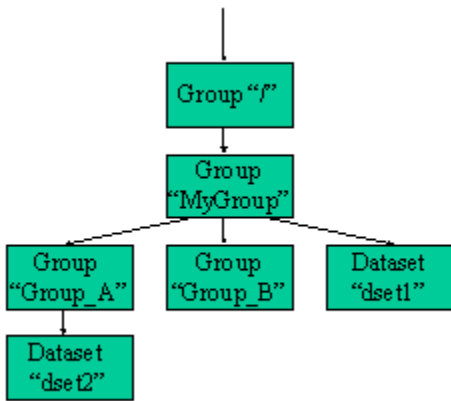
Python

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: [ [Compiling HDF5 Applications](#) ]

## File Contents

**Fig. 10.1** The Contents of *groups.h5* (*groupsf.h5* for FORTRAN)



**Fig. 10.2a** *groups.h5* in DDL

```

HDF5 "groups.h5" {
GROUP "/" {
GROUP "MyGroup" {
GROUP "Group_A" {
DATASET "dset2" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 2, 10 ) / ( 2, 10 ) }
DATA {
1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
}
}
}
}
GROUP "Group_B" {
}
DATASET "dset1" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 3, 3 ) / ( 3, 3 ) }
DATA {
1, 2, 3,
1, 2, 3,
1, 2, 3
}
}
}
}
}
}
}

```

**Fig. 10.2b** *groupsf.h5* in DDL

```
HDF5 "groupsf.h5" {
GROUP "/" {
GROUP "MyGroup" {
GROUP "Group_A" {
DATASET "dset2" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 10, 2 ) / ( 10, 2 ) }
DATA {
1, 1,
2, 2,
3, 3,
4, 4,
5, 5,
6, 6,
7, 7,
8, 8,
9, 9,
10, 10
}
}
}
GROUP "Group_B" {
}
DATASET "dset1" {
DATATYPE { H5T_STD_I32BE }
DATASPACE { SIMPLE ( 3, 3 ) / ( 3, 3 ) }
DATA {
1, 1, 1,
2, 2, 2,
3, 3, 3
}
}
}
}
}
```