

HDF5

Hierarchical Data Format library. Serial and MPI parallel version.

HDF5 (Hierarchical Data Format) is a general purpose library and file format for storing scientific data. HDF5 can store two primary objects: datasets and groups. A dataset is essentially a multidimensional array of data elements, and a group is a structure for organizing objects in an HDF5 file. Using these two basic objects, one can create and store almost any kind of scientific data structure, such as images, arrays of vectors, and structured and unstructured grids. You can also mix and match them in HDF5 files according to your needs.

Versions **1.8.11** and **1.8.13** of HDF5 library are available on Anselm, compiled for **Intel MPI** and **OpenMPI** using **intel** and **gnu**** compilers. These are available via modules:

|Version |Parallelization |module

C linker options

C++ linker options

Fortran linker options | | — | — | |HDF5 icc serial |pthread |hdf5/1.8.11
|\$HDF5_INC HDF5_SHLIB|HDF5_INC HDF5_CPP_LIB|HDF5_INC
HDF5_F90_LIB||HDF5_iccparallelMPI |pthread, IntelMPI|hdf5-parallel/1.8.11|HDF5_INC
HDF5_SHLIB|Notsupported|HDF5_INC HDF5_F90_LIB||HDF5_iccserial|pthread|hdf5/1.8.13|HDF5_INC
HDF5_SHLIB|HDF5_INC HDF5_CPP_LIB|HDF5_INC HDF5_F90_LIB||HDF5_iccparallelMPI |pthread, IntelMPI|hdf5-parallel/1.8.13|HDF5_INC
HDF5_SHLIB|Notsupported|HDF5_INC
HDF5_F90_LIB||HDF5_gccparallelMPI |pthread, OpenMPI1.6.5, gcc4.8.1|hdf5-parallel/1.8.11 - gcc|HDF5_INC HDF5_SHLIB|Notsupported|HDF5_INC
HDF5_F90_LIB||HDF5_gccparallelMPI |pthread, OpenMPI1.6.5, gcc4.8.1|hdf5-parallel/1.8.13 - gcc|HDF5_INC HDF5_SHLIB|Notsupported|HDF5_INC
HDF5_F90_LIB||HDF5_gccparallelMPI |pthread, OpenMPI1.8.1, gcc4.9.0|hdf5-parallel/1.8.13 - gcc49|HDF5_INC HDF5_SHLIB|Notsupported|HDF5_INC
\$HDF5_F90_LIB |

\$ module load hdf5-parallel

The module sets up environment variables, required for linking and running HDF5 enabled applications. Make sure that the choice of HDF5 module is consistent with your choice of MPI library. Mixing MPI of different implementations may have unpredictable results.

Be aware, that GCC version of **HDF5 1.8.11** has serious performance issues, since it's compiled with -O0 optimization flag. This version is provided only for testing of code compiled only by GCC and IS NOT recommended for production computations. For more informations, please see: <http://www.hdf5.org/>

hdfgroup.org/ftp/HDF5/prev-releases/ReleaseFiles/release5-1811 All GCC versions of **HDF5 1.8.13** are not affected by the bug, are compiled with -O3 optimizations and are recommended for production computations.

Example

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

int main() {

    hid_t      file_id, dataset_id, dataspace_id; /* identifiers */
    hsize_t    dims[2];
    herr_t     status;
    int        i, j, dset_data[4][6];

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

    /* Create the data space for the dataset. */
    dims[0] = 4;
    dims[1] = 6;
    dataspace_id = H5Screate_simple(2, dims, NULL);

    /* Initialize the dataset. */
    for (i = 0; i < 4; i++)
        for (j = 0; j < 6; j++)
            dset_data[i][j] = i * 6 + j + 1;

    /* Create the dataset. */
    dataset_id = H5Dcreate2(file_id, "/dset", H5T_STD_I32BE, dataspace_id,
                           H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);

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

    status = H5Dread(dataset_id, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_DEFAULT,
                     dset_data);

    /* End access to the dataset and release resources used by it. */
    status = H5Dclose(dataset_id);

    /* Terminate access to the data space. */
    status = H5Sclose(dataspace_id);
```

```

    /* Close the file. */
    status = H5Fclose(file_id);
}

```

Load modules and compile:

```

$ module load intel impi
$ module load hdf5-parallel

```

```

$ mpicc hdf5test.c -o hdf5test.x -Wl,-rpath=$LIBRARY_PATH $HDF5_INC $HDF5_SHLIB

```

Run the example as Intel MPI program.

For further informations, please see the website: <http://www.hdfgroup.org/HDF5/>

```

class="smarterwiki-popup-bubble-tip">

```

```

btnI=I'm+Feeling+Lucky&btnI=I'm+Feeling+Lucky&q=HDF5%20icc%20serial%09pthread%09hdf5%2F1.8.
parallel%2F1.8.13%09%24HDF5__INC%20%24HDF5__SHLIB%09Not%20supported%09%24HDF5__INC%20%
"Search Wikipedia"){.smarterwiki-popup-bubble

```