Reading From and Writing To a Dataset

During a dataset I/O operation, the library transfers raw data between memory and the file. The data in memory can have a datatype different from that of the file and can also be of a different size (i.e., the data in memory is a subset of the dataset elements, or vice versa). Therefore, to perform read or write operations, the application program must specify:

» The dataset

» The dataset's datatype in memory

» The dataset's dataspace in memory

» The dataset's dataspace in the file

» The dataset transfer property list

> (The dataset transfer property list controls various aspects of the I/O operations, such as the number of processes participating in a collective I/O request or hints to the library to control caching of raw data. In this tutorial, we use the default dataset transfer property list.)

» The data buffer

The steps to read from or write to a dataset are as follows:

1. Obtain the dataset identifier.
2. Specify the memory datatype.
3. Specify the memory dataspace.
4. Specify the file dataspace.
5. Specify the transfer properties.
6. Perform the desired operation on the dataset.
7. Close the dataset.
8. Close the dataspace, datatype, and property list if necessary.

To read from or write to a dataset, the H5D_READ and H5D_WRITE routines are used.

```
C:
   status = H5Dread (set_id, mem_type_id, mem_space_id, file_space_id,
                     xfer_prp, buf );
   status = H5Dwrite (set_id, mem_type_id, mem_space_id, file_space_id,
                     xfer_prp, buf);

FORTRAN:
   CALL h5dread_f(dset_id, mem_type_id, buf, dims, error, &
                     mem_space_id=mspace_id, file_space_id=fspace_id, &
                     xfer_prp=xfer_plist_id)
        or
   CALL h5dread_f(dset_id, mem_type_id, buf, dims,  error)


   CALL h5dwrite_f(dset_id, mem_type_id, buf, dims, error, &
                     mem_space_id=mspace_id, file_space_id=fspace_id, &
                     xfer_prp=xfer_plist_id)
        or
   CALL h5dwrite_f(dset_id, mem_type_id, buf, dims, error)
```

High Level APIs

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

Programming Example

Description

See HDF5 Introductory Examples for the examples used in the Learning the Basics tutorial.

The example shows how to read and write an existing dataset. It opens the file created in the previous example, obtains the dataset identifier for the dataset `/dset`, writes the dataset to the file, then reads the dataset back. It then closes the dataset and file.

Note that `H5S_ALL` is passed in for both the memory and file dataspace parameters in the read and write calls. This indicates that the entire dataspace of the dataset will be read or written to. `H5S_ALL` by itself does not necessarily have this meaning. See the Reference Manual entry for H5Dread or H5Dwrite for more information on using `H5S_ALL`.

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

Remarks

H5F_OPEN opens an existing file and returns a file identifier.

H5D_OPEN opens an existing dataset with the specified name and location.

H5D_WRITE writes raw data from an application buffer to the specified dataset, converting from the datatype and dataspace of the dataset in memory to the datatype and dataspace of the dataset in the file. Specifying `H5S_ALL` for both the memory and file dataspaces indicates that the entire dataspace of the dataset is to be written to. `H5S_ALL` by itself does not necessarily have this meaning. See the Reference Manual entry for H5Dwrite for more information on using `H5S_ALL`.

H5D_READ reads raw data from the specified dataset to an application buffer, converting from the file datatype and dataspace to the memory datatype and dataspace. Specifying `H5S_ALL` for both the memory and file dataspaces indicates that the entire dataspace of the dataset is to be read. `H5S_ALL` by itself does not necessarily have this meaning. See the Reference Manual entry for H5Dread for more information on using `H5S_ALL`.

File Contents

Figure 6.1a shows the contents of `dset.h5` (created by the C program).
Figure 6.1b shows the contents of `dsetf.h5` (created by the FORTRAN program).

Fig. 6.1a *`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
            }
         }
      }
      }
```

Fig. 6.1b *`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
      }
   }
}
}
```