

H5D_WRITE

[Expand all](#) [Collapse all](#)

- [Jump to ...](#)
- [Summary](#)
- [Description](#)
- [Example](#)
- [Switch language ...](#)
- [C](#)
- [C++](#)
- [FORTRAN](#)
- [JAVA](#)

[Summary](#)
[Description](#)
[Example](#)
[JAVA](#)
[FORTRAN](#)
[C++](#)
[C](#)

H5D_WRITE

Writes raw data from a buffer to a dataset

Procedure:

H5D_WRITE(dataset_id, mem_type_id, mem_space_id, file_space_id, xfer_plist_id, buf)

Signature:

```
herr_t H5Dwrite( hid_t dataset_id, hid_t mem_type_id, hid_t mem_space_id, hid_t file_space_id, hid_t xfer_p  
list_id, const void * buf )
```

Fortran2003:

```
SUBROUTINE h5dwrite_f(dset_id, mem_type_id, buf, hdferr, &  
                    mem_space_id, file_space_id, xfer_prp)  
  INTEGER(HID_T), INTENT(IN)           :: dset_id  
  INTEGER(HID_T), INTENT(IN)           :: mem_type_id  
  TYPE(C_PTR)    , INTENT(IN)           :: buf  
  INTEGER        , INTENT(OUT)          :: hdferr  
  INTEGER(HID_T), INTENT(IN)           , OPTIONAL :: mem_space_id  
  INTEGER(HID_T), INTENT(IN)           , OPTIONAL :: file_space_id  
  INTEGER(HID_T), INTENT(IN)           , OPTIONAL :: xfer_prp
```

Fortran90:

There is no direct Fortran90 counterpart for the C function H5Dwrite. Instead, that functionality is provided by two Fortran90 subroutines:

h5dwrite_f	Purpose: Writes data other than variable-length data.
h5dwrite_vl_f	Purpose: Writes variable-length data.

```

SUBROUTINE h5dwrite_f(dset_id, mem_type_id, buf, dims, hdferr, &
                    mem_space_id, file_space_id, xfer_prp)
    IMPLICIT NONE
    INTEGER(HID_T), INTENT(IN) :: dset_id      ! Dataset identifier
    INTEGER(HID_T), INTENT(IN) :: mem_type_id  ! Memory datatype identifier
    TYPE, INTENT(IN) :: buf                   ! Data buffer; may be a scalar
                                              ! or an array
    DIMENSION(*), INTEGER(HSIZE_T), INTENT(IN) :: dims
                                              ! Array to hold corresponding
                                              ! dimension sizes of data
                                              ! buffer buf; dim(k) has value
                                              ! of the k-th dimension of
                                              ! buffer buf; values are
                                              ! ignored if buf is a scalar
    INTEGER, INTENT(OUT) :: hdferr            ! Error code
                                              ! 0 on success and -1 on failure

    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: mem_space_id
                                              ! Memory dataspace identifier
                                              ! Default value is H5S_ALL_F
    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: file_space_id
                                              ! File dataspace identifier
                                              ! Default value is H5S_ALL_F
    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: xfer_prp
                                              ! Transfer property list
                                              ! identifier; default value
                                              ! is H5P_DEFAULT_F
END SUBROUTINE h5dwrite_f

SUBROUTINE h5dwrite_vl_f(dset_id, mem_type_id, buf, dims, len, hdferr, &
                       mem_space_id, file_space_id, xfer_prp)
    IMPLICIT NONE
    INTEGER(HID_T), INTENT(IN) :: dset_id      ! Dataset identifier
    INTEGER(HID_T), INTENT(IN) :: mem_type_id  ! Memory datatype identifier
    TYPE, INTENT(IN), & DIMENSION(dims(1),dims(2)) :: buf
                                              ! Data buffer; may be a scalar
                                              ! or an array
                                              ! TYPE must be one of the following
                                              !   INTEGER
                                              !   REAL
                                              !   CHARACTER
    INTEGER(HSIZE_T), INTENT(IN), DIMENSION(2) :: dims
                                              ! Array to hold corresponding
                                              ! dimension sizes of data
                                              ! buffer buf
                                              ! dim(k) has value of the k-th
                                              ! dimension of buffer buf
                                              ! Values are ignored if buf is
                                              ! a scalar
    INTEGER(SIZE_T), INTENT(IN), DIMENSION(*) :: len
                                              ! Array to store length of
                                              ! each element
    INTEGER, INTENT(OUT) :: hdferr            ! Error code
                                              ! 0 on success and -1 on failure

    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: mem_space_id
                                              ! Memory dataspace identifier
                                              ! Default value is H5S_ALL_F
    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: file_space_id
                                              ! File dataspace identifier
                                              ! Default value is H5S_ALL_F
    INTEGER(HID_T), OPTIONAL, INTENT(IN) :: xfer_prp
                                              ! Transfer property list identifier
                                              ! Default value is H5P_DEFAULT_F
END SUBROUTINE h5dwrite_vl_f

```

Parameters:

<i>hid_t</i> dataset_id	IN: Identifier of the dataset to write to
<i>hid_t</i> mem_type_id	IN: Identifier of the memory datatype
<i>hid_t</i> mem_space_id	IN: Identifier of the memory dataspace
<i>hid_t</i> file_space_id	IN: Identifier of the dataset's dataspace in the file
<i>hid_t</i> xfer_plist_id	IN: Identifier of a transfer property list for this I/O operation
<i>const void *</i> buf	IN: Buffer with data to be written to the file

Description:

H5D_WRITE writes a (partial) dataset, specified by its identifier *dataset_id*, from the application memory buffer *buf* into the file. Data transfer properties are defined by the argument *xfer_plist_id*. The memory datatype of the (partial) dataset is identified by the identifier *mem_type_id*. The part of the dataset to write is defined by *mem_space_id* and *file_space_id*.

If *mem_type_id* is either a fixed-length or variable-length string, it is important to set the string length when defining the datatype. String datatypes are derived from H5T_C_S1 (or H5T_FORTRAN_S1 for Fortran codes), which defaults to 1 character in size. See H5T_SET_SIZE and [Creating variable-length string datatypes](#).

file_space_id is used to specify only the selection within the file dataset's dataspace. Any dataspace specified in *file_space_id* is ignored by the library and the dataset's dataspace is always used. *file_space_id* can be the constant H5S_ALL, which indicates that the entire file dataspace, as defined by the current dimensions of the dataset, is to be selected.

mem_space_id is used to specify both the memory dataspace and the selection within that dataspace. *mem_space_id* can be the constant H5S_ALL, in which case the file dataspace is used for the memory dataspace and the selection defined with *file_space_id* is used for the selection within that dataspace.

The behavior of the library for the various combinations of valid dataspace IDs and H5S_ALL for the *mem_space_id* and the *file_space_id* parameters is described below:

<i>mem_space_id</i>	<i>file_space_id</i>	Behavior
valid dataspace identifier	valid dataspace identifier	<i>mem_space_id</i> specifies the memory dataspace and the selection within it. <i>file_space_id</i> specifies the selection within the file dataset's dataspace.
H5S_ALL	valid dataspace identifier	The file dataset's dataspace is used for the memory dataspace and the selection specified with <i>file_space_id</i> specifies the selection within it. The combination of the file dataset's dataspace and the selection from <i>file_space_id</i> is used for memory also.
valid dataspace identifier	H5S_ALL	<i>mem_space_id</i> specifies the memory dataspace and the selection within it. The selection within the file dataset's dataspace is set to the "all" selection.
H5S_ALL	H5S_ALL	The file dataset's dataspace is used for the memory dataspace and the selection within the memory dataspace is set to the "all" selection. The selection within the file dataset's dataspace is set to the "all" selection.

Setting an "all" selection indicates that the entire dataspace, as defined by the current dimensions of a dataspace, will be selected. The number of elements selected in the memory dataspace must match the number of elements selected in the file dataspace.

`xfer_plist_id` can be the constant `H5P_DEFAULT`. in which case the default data transfer properties are used.

Writing to a dataset will fail if the HDF5 file was not opened with write access permissions.

Datatype conversion takes place at the time of a read or write and is automatic. See the [Data Transfer: Datatype Conversion and Selection](#) section in the "HDF5 Datatypes" chapter of the *HDF5 User's Guide* for a discussion of data conversion.

If the dataset's space allocation time is set to `H5D_ALLOC_TIME_LATE` or `H5D_ALLOC_TIME_INCR` and the space for the dataset has not yet been allocated, that space is allocated when the first raw data is written to the dataset. Unused space in the dataset will be written with fill values at the same time if the dataset's fill time is set to `H5D_FILL_TIME_IFSET` or `H5D_FILL_TIME_ALLOC`. (Also see `H5P_SET_FILL_TIME` and `H5P_SET_ALLOC_TIME`.)

If a dataset's storage layout is 'compact', care must be taken when writing data to the dataset in parallel. A compact dataset's raw data is cached in memory and may be flushed to the file from any of the parallel processes, so parallel applications should always attempt to write identical data to the dataset from all processes.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

Example:

Coming Soon!

History:

Release	Change
1.8.8	Fortran updated to Fortran2003
1.4.2	<code>dims</code> parameter added in Fortran interface

--- Last Modified: December 18, 2018 | 01:39 PM