

H5LT_READ_DATASET

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

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

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

H5LT_READ_DATASET

Reads a dataset from disk.

Procedure:

H5LT_READ_DATASET(loc_id, dset_name, type_id, buffer)

Signature:

```
herr_t H5LTread_dataset ( hid_t loc_id, const char *dset_name, hid_t type_id, void *buffer )
```

Fortran90:

```
subroutine h5ltread_dataset_f(loc_id, dset_name, type_id, buf, dims, errcode)
  implicit none
  integer(HID_T), intent(IN) :: loc_id           ! file or group identifier
  character(LEN=*), intent(IN) :: dset_name     ! name of the dataset
  integer(HID_T), intent(IN) :: type_id        ! datatype identifier
  integer(HSIZE_T), dimension(*), intent(IN) :: dims ! size of the buffer buf
  , intent(INOUT), dimension(*) :: buf        ! data buffer
  integer :: errcode                           ! error code
end subroutine h5ltread_dataset_f
```

Fortran2003:

```

subroutine h5ltread_dataset_f(loc_id, dset_name, type_id, buf, errcode)
  implicit none
  integer(HID_T), intent(IN) :: loc_id           ! file or group identifier
  character(LEN=*), intent(IN) :: dset_name      ! name of the dataset
  integer(HID_T), intent(IN) :: type_id        ! datatype identifier
  type(C_PTR) :: buf                          ! data buffer
  integer :: errcode                          ! error code
end subroutine h5ltread_dataset_f

```

Parameters:

<i>hid_t</i> loc_id	IN: Identifier of the file or group to read the dataset within.
<i>const char</i> *dset_name	IN: The name of the dataset to read.
<i>hid_t</i> type_id	IN: Identifier of the datatype to use when reading the dataset.
<i>void</i> *buffer	OUT: Buffer with data.

Description:

H5LTread_dataset reads a dataset named dset_name attached to the object specified by the identifier loc_id.

Returns:

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

Example:

Coming Soon!

History:

Release	Change
1.8.7	Fortran subroutine modified in this release to accomodate arrays with more than three dimensions.
1.10.0	Fortran 2003 subroutine added to accept a C address of the data buffer.