

# H5L\_MOVE

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

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

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

# H5L\_MOVE

Moves a link within an HDF5 file

## Procedure:

H5L\_MOVE(src\_loc\_id, src\_name, dest\_loc\_id, dest\_name, lcpl\_id, lapl\_id)

## Signature:

```
herr_t H5lmove( hid_t src_loc_id, const char *src_name, hid_t dest_loc_id, const char *dest_name, hid_t
lcpl_id, hid_t lapl_id )
```

```
SUBROUTINE h5lmove_f(src_loc_id, src_name, dest_loc_id, dest_name, hdferr, &
lcpl_id, lapl_id)
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: src_loc_id
                                ! Original file or group identifier.
  CHARACTER(LEN=*), INTENT(IN) :: src_name
                                ! Original link name.
  INTEGER(HID_T), INTENT(IN) :: dest_loc_id
                                ! Destination file or group identifier.
  CHARACTER(LEN=*), INTENT(IN) :: dest_name
                                ! New link name.
  INTEGER(HID_T), INTENT(OUT) :: hdferr ! Error code:
                                ! 0 on success and -1 on failure
  INTEGER(HID_T), OPTIONAL, INTENT(IN) :: lcpl_id
                                ! Link creation property list identifier
                                ! to be associated with the new link.
  INTEGER(HID_T), OPTIONAL, INTENT(IN) :: lapl_id
                                ! Link access property list identifier
                                ! to be associated with the new link.
END SUBROUTINE h5lmove_f
```

## Parameters:

<i>hid_t</i> src_loc_id	IN: Original location identifier; may be a file, group, dataset, named datatype or attribute identifier
<i>const char *</i> src_name	IN: Original link name
<i>hid_t</i> dest_loc_id	IN: Destination location identifier; may be a file, group, dataset, named datatype or attribute identifier
<i>const char *</i> dest_name	IN: New link name
<i>hid_t</i> lcpl_id	IN: Link creation property list identifier to be associated with the new link
<i>hid_t</i> lapl_id	IN: Link access property list identifier to be associated with the new link

## Description:

H5L\_MOVE moves a link within an HDF5 file. The original link, *src\_name*, is removed from *src\_loc\_id* and the new link, *dest\_name*, is inserted at *dest\_loc\_id*. This change is accomplished as an atomic operation.

*src\_loc\_id* and *src\_name* identify the original link. *src\_loc\_id* is the original location identifier; *src\_name* is the path to the link and is interpreted relative to *src\_loc\_id*.

*dest\_loc\_id* and *dest\_name* identify the new link. *dest\_loc\_id* is either a file or group identifier; *dest\_name* is the path to the link and is interpreted relative to *dest\_loc\_id*.

Note that H5L\_MOVE does not modify the value of the link; the new link points to the same object as the original link pointed to. Furthermore, if the object pointed to by the original link was already open with a valid object identifier, that identifier will remain valid after the call to H5L\_MOVE.

*lcpl\_id* and *lapl\_id* are the link creation and link access property lists, respectively, associated with the new link, *dest\_name*.

Through these property lists, several properties are available to govern the behavior of H5L\_MOVE. The property controlling creation of missing intermediate groups is set in the link creation property list with `H5P_SET_CREATE_INTERMEDIATE_GROUP`; H5L\_MOVE ignores any other properties in the link creation property list. Properties controlling character encoding, link traversals, and external link prefixes are set in the link access property list with `H5P_SET_CHAR_ENCODING`, `H5P_SET_NLINKS`, and `H5P_SET_ELINK_PREFIX`, respectively.

Exercise care in moving links as it is possible to render data in a file inaccessible with H5L\_MOVE. If the link being moved is on the only path leading to an HDF5 object, that object may become permanently inaccessible in the file.

## Returns:

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

## Example:

Coming Soon!

## History:

Release	Change
1.8.0	Function introduced in this release.