

# H5G\_LINK

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

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

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

# H5G\_LINK

Creates a link of the specified type from `new_name` to `current_name`

*This function is deprecated in favor of the functions [H5L\\_CREATE\\_HARD](#) and [H5L\\_CREATE\\_SOFT](#).*

## Procedure:

H5G\_LINK(`loc_id`, `link_type`, `current_name`, `new_name`)

## Signature:

```
herr_t H5glink(hid_t loc_id, H5G_link_t link_type, const char *current_name, const char *new_name )
```

```
SUBROUTINE h5glink_f(loc_id, link_type, current_name, new_name, hdferr)
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: loc_id      ! File or group location identifier
  INTEGER, INTENT(IN)      :: link_type    ! Link type, possible values are:
                                          !     H5G_LINK_HARD_F
                                          !     H5G_LINK_SOFT_F
  CHARACTER(LEN=*), INTENT(IN) :: current_name
                                          ! Current object name relative
                                          ! to loc_id
  CHARACTER(LEN=*), INTENT(IN) :: new_name ! New object name
  INTEGER, INTENT(OUT) :: hdferr          ! Error code
                                          ! 0 on success and -1 on failure

END SUBROUTINE h5glink_f
```

### Parameters:

<i>hid_t</i> loc_id	IN: File or group identifier
<i>H5G_link_t</i> link_type	IN: Link type. Possible values are H5G_LINK_HARD and H5G_LINK_SOFT
<i>const char *</i> current_name	IN: Name of the existing object if link is a hard link. Can be anything for the soft link
<i>const char *</i> new_name	IN: New name for the object

### Description:

H5G\_LINK creates a new name for an object that has some current name, possibly one of many names it currently has.

If *link\_type* is H5G\_LINK\_HARD, then *current\_name* must specify the name of an existing object and both names are interpreted relative to *loc\_id*, which is either a file identifier or a group identifier.

If *link\_type* is H5G\_LINK\_SOFT, then *current\_name* can be anything and is interpreted at lookup time relative to the group which contains the final component of *new\_name*. For instance, if *current\_name* is *./foo*, *new\_name* is *./x/y/bar*, and a request is made for *./x/y/bar*, then the actual object looked up is *./x/y/./foo*.

### Returns:

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

### Example:

Coming soon!

### History:

Release	Change
1.8.0	Function deprecated in this release.

--- Last Modified: April 25, 2019 | 11:35 AM