

# H5F\_REOPEN

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

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

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

# H5F\_REOPEN

Returns a new identifier for a previously-opened HDF5 file

## Procedure:

H5F\_REOPEN (*file\_id*)

## Signature:

```
hid_t H5Freopen(hid_t file_id )
```

```
SUBROUTINE h5freopen_f(file_id, new_file_id, hdferr)  
  IMPLICIT NONE  
  INTEGER(HID_T), INTENT(IN)  :: file_id      ! File identifier  
  INTEGER(HID_T), INTENT(OUT) :: new_file_id ! New file identifier  
  INTEGER, INTENT(OUT)       :: hdferr       ! Error code  
                                     ! 0 on success and -1 on failure  
END SUBROUTINE h5freopen_f
```

## Parameters:

<i>hid_t file_id</i>	IN: Identifier of a file for which an additional identifier is required
----------------------	---

## Description:

H5F\_REOPEN returns a new file identifier for an already-open HDF5 file, as specified by *file\_id*. Both identifiers share caches and other information. The only difference between the identifiers is that the new identifier is not mounted anywhere and no files are mounted on it.

Note that there is no circumstance under which `H5F_REOPEN` can actually open a closed file; the file must already be open and have an active `file_id`. E.g., one cannot close a file with `H5F_CLOSE(file_id)` then use `H5F_REOPEN(file_id)` to reopen it.

The new file identifier should be closed by calling `H5F_CLOSE` when it is no longer needed.

**Returns:**

Returns a new file identifier if successful; otherwise returns a negative value.

**Example:**

Coming soon!

--- Last Modified: December 20, 2018 | 12:03 PM