H5F_CLOSE

H5F_CLOSE

Terminates access to an HDF5 file

Procedure:

H5F_CLOSE(file_id)

Signature:

```
herr_t H5Fclose( hid_t file_id )
```

```
SUBROUTINE h5fclose_f(file_id, hdferr)
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: file_id ! File identifier
  INTEGER, INTENT(OUT) :: hdferr        ! Error code
                                        ! 0 on success and -1 on failure
END SUBROUTINE h5fclose_f
```

Parameters:

hid_t file_id	IN: Identifier of a file to terminate access to

Description:

H5F_CLOSE terminates access to an HDF5 file (specified by `file_id`) by flushing all data to storage.

If this is the last file identifier open for the file and no other access identifier is open (e.g., a dataset identifier, group identifier, or shared datatype

identifier), the file will be fully closed and access will end.

Delayed close:
Note the following deviation from the above-described behavior. If H5F_CLOSE is called for a file but one or more objects within the file remain open, those objects will remain accessible until they are individually closed. Thus, if the dataset data_sample is open when H5F_CLOSE is called for the file containing it, data_sample will remain open and accessible (including writable) until it is explicitly closed. The file will be automatically closed once all objects in the file have been closed.

Be warned, however, that there are circumstances where it is not possible to delay closing a file. For example, an MPI-IO file close is a collective call; all of the processes that opened the file must close it collectively. The file cannot be closed at some time in the future by each process in an independent fashion. Another example is that an application using an AFS token-based file access privilege may destroy its AFS token after H5F_CLOSE has returned successfully. This would make any future access to the file, or any object within it, illegal.

In such situations, applications must close all open objects in a file before calling H5F_CLOSE. It is generally recommended to do so in all cases.

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

Example:

examples / h5_crtdat.c [46:47]	1.10/master	HDFFV/hdf5

```
/* Close the file. */
   status = H5Fclose(file_id);
```

fortran / examples / h5_crtdat.f90 [73:75]	1.10/master	HDFFV/hdf5

```
! Close the file.
  !
  CALL h5fclose_f(file_id, error)
```

--- Last Modified: May 10, 2018 | 02:14 PM