

H5F_DELETE

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

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

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

H5F_DELETE

Deletes an HDF5 file

Procedure:

H5F_DELETE (filename, fapl_id)

Signature:

```
herr_t H5Fdelete ( const char *filename, hid_t fapl_id )
```

Parameters:

<i>const char</i> *filename	Name of the file to delete
<i>hid_t</i> fapl_id	File access property list

Description:

H5F_DELETE deletes an HDF5 file *filename* with a file access property list *fapl_id*. The *fapl_id* should be configured with the same VOL connector or VFD that was used to open the file.

This API was introduced for use with the Virtual Object Layer (VOL). With the VOL, HDF5 "files" can map to arbitrary storage schemes such as object stores and relational database tables. The data created by these implementations may be inconvenient for a user to remove without a detailed knowledge of the storage scheme. H5F_DELETE gives VOL connector authors the ability to add connector-specific delete code to their connectors so that users can remove these "files" without detailed knowledge of the storage scheme.

For a VOL connector, H5F_DELETE deletes the file in a way that makes sense for the specified VOL connector.

For the native HDF5 connector, HDF5 files will be deleted via the VFDs, each of which will have to be modified to delete the files it creates.

For all implementations, H5F_DELETE will first check if the file is an HDF5 file via H5F_IS_ACCESSIBLE. This is done to ensure that H5F_DELETE cannot be used as an arbitrary file deletion call.

Returns:

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

Example:

```
test / tfile.c [1898:1913]          hdf5_1_12   HDF5V/hdf5
/*****/

/* Get fapl-dependent filename */
h5_fixname(FILE_DELETE_NOT_HDF5, fapl_id, filename, sizeof(filename));

/* Create a non-HDF5 file */
fd = HDopen(filename, O_RDWR | O_CREAT | O_TRUNC, H5_POSIX_CREATE_MODE_RW);
CHECK_I(fd, "HDopen");

/* Close the file */
ret = HDclose(fd);
VERIFY(ret, 0, "HDclose");

/* Verify that the file is not an HDF5 file */
/* Note that you can get a FAIL result when h5_fixname()
 * perturbs the filename as a file with that exact name
```

History:

Release	Change
1.12.0	Function was introduced