H5F_OPEN

Expand all Collapse all

- Jump to ...
- Summary
- Description
- Example
- Switch language ...
- C
- C++
- FORTRAN
- JAVA

H5F_OPEN

Opens an existing HDF5 file

Procedure:

H5F_OPEN (name, flags, fapl_id)

Signature:

```
hid_t H5Fopen( const char *name, unsigned flags, hid_t fapl_id )
```

```
SUBROUTINE h5fopen_f(name, access_flags, file_id, hdferr, &
                     access_prp)
  IMPLICIT NONE
  CHARACTER(LEN=*), INTENT(IN) :: name   ! Name of the file
  INTEGER, INTENT(IN) :: access_flag     ! File access flags
                                         ! Possible values are:
                                         !     H5F_ACC_RDWR_F
                                         !     H5F_ACC_RDONLY_F
  INTEGER(HID_T), INTENT(OUT) :: file_id ! File identifier
  INTEGER, INTENT(OUT) :: hdferr         ! Error code
                                         ! 0 on success and -1 on failure
  INTEGER(HID_T), OPTIONAL, INTENT(IN) :: access_prp
                                         ! File access property list
                                         ! identifier
END SUBROUTINE h5fopen_f
```

Parameters:

const char * `name`	IN: Name of the file to be opened
unsigned `flags`	IN: File access flags. Allowable values are: `H5F_ACC_RDWR`: Allows read and write access to file `H5F_ACC_RDONLY`: Allows read-only access to file `H5F_ACC_RDWR` | `H5F_ACC_SWMR_WRITE`: Indicates that the file is open for writing in a single-writer/multi-writer (SWMR) scenario. `H5F_ACC_RDONLY` | `H5F_ACC_SWMR_READ`: Indicates that the file is open for reading in a single-writer/multi-reader (SWMR) scenario. `H5F_ACC_RDWR` and `H5F_ACC_RDONLY` are mutually exclusive; use exactly one. An additional flag, `H5F_ACC_DEBUG`, prints debug information. This flag can be combined with one of the above values using the bit-wise OR operator (`|`), but it is used only by HDF5 Library developers; *it is neither tested nor supported* for use in applications.
hid_t `fapl_id`	IN: Identifier for the file access properties list. If parallel file access is desired, this is a collective call according to the communicator stored in the `fapl_id`. Use `H5P_DEFAULT` for default file access properties.

Description:

H5F_OPEN is the primary function for accessing existing HDF5 files. This function opens the named file in the specified access mode and with the specified access property list.

Note that H5F_OPEN does not create a file if it does not already exist; see H5F_CREATE.

The `name` parameter specifies the name of the file to be opened.

The `fapl_id` parameter specifies the file access property list. Use of `H5P_DEFAULT` specifies that default I/O access properties are to be used

The `flags` parameter specifies whether the file will be opened in read-write or read-only mode, `H5F_ACC_RDWR` or `H5F_ACC_RDONLY`, respectively. More complex behaviors of file access are controlled through the file-access property list.

The return value is a file identifier for the open file; this file identifier should be closed by calling H5F_CLOSE when it is no longer needed.

Special cases — Multiple opens:
A file can often be opened with a new H5F_OPEN call without closing an already-open identifier established in a previous H5F_OPEN or H5F_CREATE call. Each such H5F_OPEN call will return a unique identifier and the file can be accessed through any of these identifiers as long as the identifier remains valid. In such multiply-opened cases, the open calls must use the same `flags` argument and the file access property lists must use the same file close degree property setting (see the external link discussion below and H5P_SET_FCLOSE_DEGREE).

In some cases, such as files on a local Unix file system, the HDF5 library can detect that a file is multiply opened and will maintain coherent access among the file identifiers.

But in many other cases, such as parallel file systems or networked file systems, it is not always possible to detect multiple opens of the same physical file. In such cases, HDF5 will treat the file identifiers as though they are accessing different files and will be unable to maintain coherent access. Errors are likely to result in these cases. While unlikely, the HDF5 library may not be able to detect, and thus report, such errors.

It is generally recommended that applications avoid multiple opens of the same file.

Special restriction on multiple opens of a file first opened by means of an external link: When an external link is followed, the external file is always opened with the weak file close degree property setting, `H5F_CLOSE_WEAK` (see H5L_CREATE_EXTERNAL and H5P_SET_FCLOSE_DEGREE). If the file is reopened with H5F_OPEN while it remains held open from such an external link call, the file access property list used in the open call must include the file close degree setting `H5F_CLOSE_WEAK` or the open will fail.

See Also:

Using Identifiers

Returns:

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

Example:

Coming soon!

History:

Release	Change
1.10.0	The `H5F_ACC_SWMR_WRITE` and `H5F_ACC_SWMR_READ` flags were added.

--- Last Modified: May 15, 2019 | 03:07 PM