H5A_CREATE

Expand all Collapse all

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

Summary
Description
Example
JAVA
FORTRAN
C++
C

H5A_CREATE

Creates an attribute attached to a specified object

Signature:

```
hid_t H5Acreate( hid_t loc_id, const char *attr_name, hid_t type_id, hid_t space_id, hid_t acpl_id )
```

```
hid_t H5Acreate( hid_t loc_id, const char *attr_name, hid_t type_id, hid_t space_id, hid_t acpl_id, hid_t a
apl_id )
```

```
SUBROUTINE h5acreate_f(loc_id, name, type_id, space_id, attr_id, hdferr, &
                       acpl_id, aapl_id )
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: loc_id    ! Object identifier
  CHARACTER(LEN=*), INTENT(IN) :: name    ! Attribute name
  INTEGER(HID_T), INTENT(IN) :: type_id   ! Attribute datatype identifier
  INTEGER(HID_T), INTENT(IN) :: space_id  ! Attribute dataspace identifier
  INTEGER(HID_T), INTENT(OUT) :: attr_id  ! Attribute identifier
  INTEGER, INTENT(OUT) :: hdferr          ! Error code:
                                          ! 0 on success and -1 on failure
  INTEGER(HID_T), OPTIONAL, INTENT(IN) :: acpl_id
                                          ! Attribute creation property
                                          ! list identifier
  INTEGER(HID_T), OPTIONAL, INTENT(IN) :: aapl_id
                                          ! Attribute access property
                                          ! list identifier
END SUBROUTINE h5acreate_f
```

Description:

H5A_CREATE is a macro that is mapped to either H5A_CREATE1 or H5A_CREATE2, depending on the needs of the application.

Such macros are provided to facilitate application compatibility. For example:

- The H5A_CREATE macro will be mapped to H5A_CREATE1 and will use the H5A_CREATE1 syntax (first signature above) if an application is coded for HDF5 Release 1.6.x.
- The H5A_CREATE macro will be mapped to H5A_CREATE2 and will use the H5A_CREATE2 syntax (second signature above) if an application is coded for HDF5 Release 1.8.x.

See API Compatibility Macros in HDF5 for information on these macros and mappings.

When both the HDF5 library and the application are built and installed with no specific compatibility flags, H5A_CREATE is mapped to the most recent version of the function, currently H5A_CREATE2. If the library and/or application is compiled for Release 1.6 emulation, H5A_CREATE will be mapped to H5A_CREATE1. Function-specific flags are available to override these settings on a function-by-function basis when the application is compiled.

Specific compile-time compatibility flags and the resulting mappings are as follows:

Compatibility setting	H5A_CREATE mapping
Global settings	
No compatibility flag	H5A_CREATE2
Enable deprecated symbols	H5A_CREATE2
Disable deprecated symbols	H5A_CREATE2
Emulate Release 1.6 interface	H5A_CREATE1
Function-level macros	
`H5Acreate_vers = 2`	H5A_CREATE2
`H5Acreate_vers = 1`	H5A_CREATE1

The attribute identifier returned by this macro must be released with H5A_CLOSE or resource leaks will develop.

Interface history: Signature [1] above is the original H5A_CREATE interface and the only interface available prior to HDF5 Release 1.8.0. This signature and the corresponding function are now deprecated but will remain directly callable as H5A_CREATE1.

Signature [2] above was introduced with HDF5 Release 1.8.0 and is the recommended and default interface. It is directly callable as H5A_CREATE2.

See API Compatibility Macros in HDF5 for circumstances under which either of these functions might not be available in an installed instance of the HDF5 library.

History:

Release	Create
1.8.0	The function `H5Acreate` renamed to `H5Acreate1` and deprecated in this release. The macro `H5Acreate` and the functions `H5Acreate2` and `H5Acreate_by_name` introduced in this release.

--- Last Modified: December 20, 2018 | 01:55 PM