

H5G_CREATE_ANON

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

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

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

H5G_CREATE_ANON

Creates a new empty group without linking it into the file structure

Procedure:

H5G_CREATE_ANON(*loc_id*, *gcpl_id*, *gapl_id*)

Signature:

```
hid_t H5gcreate_anon( hid_t loc_id, hid_t gcpl_id, hid_t gapl_id )
```

```
SUBROUTINE h5gcreate_anon_f(loc_id, grp_id, hdferr, gcpl_id, gapl_id)
```

```
IMPLICIT NONE
INTEGER(HID_T), INTENT(IN) :: loc_id    ! Location identifier
INTEGER(HID_T), INTENT(OUT) :: grp_id   ! Group identifier
INTEGER, INTENT(OUT) :: hdferr         ! Error code
                                       ! 0 on success and -1 on failure
INTEGER(HID_T), OPTIONAL, INTENT(IN) :: gcpl_id
                                       ! Property list for group creation
INTEGER(HID_T), OPTIONAL, INTENT(IN) :: gapl_id
                                       ! Property list for group access
END SUBROUTINE h5gcreate_anon_f
```

Parameters:

<i>hid_t</i> <i>loc_id</i>	IN: Object identifier specifying the file in which the new group is to be created; may be a file, group, dataset, named datatype or attribute
----------------------------	---

<code>hid_t gcpl_id</code>	IN: Group creation property list identifier (<code>H5P_DEFAULT</code> for the default property list)
<code>hid_t gapl_id</code>	IN: Group access property list identifier (No group access properties have been implemented at this time; use <code>H5P_DEFAULT</code> .)

Description:

`H5G_CREATE_ANON` creates a new empty group in the file specified by `loc_id`. With default settings, `H5G_CREATE_ANON` provides similar functionality to that provided by `H5G_CREATE`, with the differences described below.

The new group's creation and access properties are specified in `gcpl_id` and `gapl_id`, respectively.

`H5G_CREATE_ANON` returns a new group identifier. This identifier *must* be linked into the HDF5 file structure with `H5O_LINK` or it will be deleted from the file when the file is closed.

The differences between this function and `H5G_CREATE1` are as follows:

- `H5G_CREATE1` does not provide for the use of custom property lists; `H5G_CREATE1` always uses default properties.
- `H5G_CREATE_ANON` neither provides the new group's name nor links it into the HDF5 file structure; those actions must be performed separately through a call to `H5O_LINK`, which offers greater control over linking.
- `H5G_CREATE_ANON` does not directly provide a *hint* mechanism for the group's heap size. Comparable information can be included in the group creation property list `gcpl_id` through a `H5P_SET_LOCAL_HEAP_SIZE_HINT` call.

A group created with this function should be closed with `H5G_CLOSE` when the group is no longer needed so that resource leaks will not develop.

See Also:

- [H5G_CREATE](#)
- [H5O_LINK](#)
- [Using Identifiers](#)

Returns:

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

Example:

```
gcpl = H5Pcreate(H5P_GROUP_CREATE);
...
/* Create the group anonymously and link it in */
grp = H5Gcreate_anon(fid, gcpl, H5P_DEFAULT);

! Create a group with no name
CALL H5Gcreate_anon_f(file_id, group_id, error)
```

History:

Release	Change
1.8.0	Function introduced in this release.