

H5G_GET_OBJINFO

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

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

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

H5G_GET_OBJINFO

Returns information about an object.

This function is deprecated in favor of the function [H5O_GET_INFO](#) and [H5L_GET_INFO1](#).

Procedure:

H5G_GET_OBJINFO(loc_id, name, follow_link, statbuf)

Signature:

```
herr_t H5Gget_objinfo(hid_t loc_id, const char *name, hbool_t follow_link, H5G_stat_t *statbuf )
```

Parameters:

<i>hid_t</i> loc_id	IN: File or group identifier <i>Alternative:</i> An object identifier, obj_id
<i>const char</i> *name	IN: Name of the object for which status is being sought <i>Alternative:</i> If the preceding parameter is the object's direct identifier, i.e., the obj_id, this parameter should be a dot (.).
<i>hbool_t</i> follow_link	IN: Link flag
<i>H5G_stat_t</i> *statbuf	OUT: Buffer in which to return information about the object

Description:

H5G_GET_OBJINFO returns information about the specified object through the `statbuf` argument.

A file or group identifier, `loc_id`, and an object name, `name`, relative to `loc_id`, are commonly used to specify the object. However, if the object identifier is already known to the application, an alternative approach is to use that identifier, `obj_id`, in place of `loc_id`, and a dot (.) in place of `name`. Thus, the alternative versions of the first portion of an H5G_GET_OBJINFO call would be as follows:

```
H5Gget_objinfo (loc_id name ...)
H5Gget_objinfo (obj_id .    ...)
```

If the object is a symbolic link and `follow_link` is zero (0), then the information returned describes the link itself; otherwise the link is followed and the information returned describes the object to which the link points. If `follow_link` is non-zero but the final symbolic link is dangling (does not point to anything), then an error is returned. The `statbuf` fields are undefined for an error. The existence of an object can be tested by calling this function with a null `statbuf`.

H5Gget_objinfo fills in the following data structure (defined in H5Gpublic.h):

```
typedef struct H5G_stat_t {
    unsigned long fileno[2];
    unsigned long objno[2];
    unsigned nlink;
    H5G_obj_t type;
    time_t mtime;
    size_t linklen;
    H5O_stat_t ohdr;
} H5G_stat_t
```

where H5O_stat_t (defined in H5Opublic.h) is:

```
typedef struct H5O_stat_t {
    hsize_t size;
    hsize_t free;
    unsigned nmsgs;
    unsigned nchunks;
} H5O_stat_t
```

The `fileno` and `objno` fields contain four values which uniquely identify an object among those HDF5 files which are open: if all four values are the same between two objects, then the two objects are the same (provided both files are still open).

- Note that if a file is closed and re-opened, the value in `fileno` will change.
- If a VFL driver either does not or cannot detect that two `H5Fopen` calls referencing the same file actually open the same file, each will get a different `fileno`.

The `nlink` field is the number of hard links to the object or zero when information is being returned about a symbolic link (symbolic links do not have hard links but all other objects always have at least one).

The `type` field contains the type of the object, one of H5G_GROUP, H5G_DATASET, H5G_LINK, or H5G_TYPE.

The `mtime` field contains the modification time.

If information is being returned about a symbolic link then `linklen` will be the length of the link value (the name of the pointed-to object with the null terminator); otherwise `linklen` will be zero.

The fields in the `H5O_stat_t` struct contain information about the object header for the object queried:

<code>size</code>	The total size of all the object header information in the file (for all chunks).
<code>free</code>	The size of unused space in the object header.
<code>nmsgs</code>	The number of object header messages.
<code>nchunks</code>	The number of chunks the object header is broken up into.

Other fields may be added to this structure in the future.

Some systems will be able to record the time accurately but unable to retrieve the correct time; such systems (e.g., Irix64) will report an `mtime` value of 0 (zero).

Returns:

Returns a non-negative value if successful, with the fields of `statbuf` (if non-null) initialized. Otherwise returns a negative value.

Example:

Coming soon!

History:

Release	Change
1.6.1	Two new fields were added to the <code>H5G_stat_t</code> struct in this release.
1.8.0	Function deprecated in this release.

--- Last Modified: April 25, 2019 | 11:22 AM