

# H5O\_OPEN\_BY\_ADDR

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

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

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

# H5O\_OPEN\_BY\_ADDR

Opens an object using its address within an HDF5 file

## Procedure:

H5O\_OPEN\_BY\_ADDR(*loc\_id*, *addr*)

## Signature:

```
hid_t H5oopen_by_addr( hid_t loc_id, haddr_t addr )
```

```
SUBROUTINE h5oopen_by_addr_f( loc_id, addr, obj_id, hdferr )  
  IMPLICIT NONE  
  INTEGER(HID_T) , INTENT(IN) :: loc_id  
  INTEGER(HADDR_T), INTENT(IN) :: addr  
  INTEGER(HID_T) , INTENT(OUT) :: obj_id  
  INTEGER          , INTENT(OUT) :: hdferr
```

## Parameters:

<i>hid_t</i> <i>loc_id</i>	IN: Location identifier; may be a file, group, dataset, named datatype, or attribute identifier
<i>haddr_t</i> <i>addr</i>	IN: Object's address in the file

## Description:

H5O\_OPEN\_BY\_ADDR opens a group, dataset, or committed (named) datatype using its address within an HDF5 file, `addr`. The resulting opened object is identical to an object opened with H5O\_OPEN and should be closed with H5O\_CLOSE or an object-type-specific closing function (such as H5G\_CLOSE) when no longer needed.

`loc_id` is a location identifier in the file.

The object's address within the file, `addr`, is the byte offset of the first byte of the object header from the beginning of the HDF5 file space, i.e., from the beginning of the super block (see the "HDF5 Storage Model" section of the [The HDF5 Data Model and File Structure](#) chapter of the *HDF 5 User's Guide*).

`addr` can be obtained via either of two function calls. H5G\_GET\_OBJINFO returns the object's address in the `objno` field of the `H5G_stat_t` struct; H5L\_GET\_INFO returns the address in the `address` field of the `H5L_linkinfo_t` struct.

Warning: This function must be used with care!  
Improper use can lead to inaccessible data, wasted space in the file, or **file corruption**.

This function is dangerous if called on an invalid address. The risk can be safely overcome by retrieving the object address with H5G\_GET\_OBJINFO or H5L\_GET\_LINKINFO immediately before calling H5O\_OPEN\_BY\_ADDR. The immediacy of the operation can be important; if time has elapsed and the object has been deleted from the file, the address will be invalid and file corruption can result.

The address of the HDF5 file on a physical device has no effect on H5O\_OPEN\_BY\_ADDR, nor does the use of any file driver. As stated above, the object address is its offset within the HDF5 file; HDF5's file drivers will transparently map this to an address on a storage device.

#### Returns:

Returns an object identifier for the opened object if successful; otherwise returns a negative value.

#### Example:

Coming Soon!

#### History:

Release	Change
1.8.0	Function introduced in this release.
1.8.4	Fortran subroutine added in this release.

--- Last Modified: May 08, 2018 | 10:03 AM