

# H5R\_GET\_OBJ\_TYPE

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

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

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

# H5R\_GET\_OBJ\_TYPE

Retrieves the type of object that an object reference points to

## Signature:

```
H5G_obj_t H5Rget_obj_type ( hid_t loc_id, H5R_type_t ref_type, void *ref )
```

```
err_t H5Rget_obj_type ( hid_t loc_id, H5R_type_t ref_type, void *ref, H5O_type_t *obj_type )
```

Fortran90 Interface: h5rget\_object\_type\_f

Signature:

```
SUBROUTINE h5rget_object_type_f(loc_id, ref, obj_type, hdferr)
  INTEGER(HID_T)    , INTENT(IN)  :: loc_id
  TYPE(hobj_ref_t_f), INTENT(IN)  :: ref
  INTEGER          , INTENT(OUT) :: obj_type
  INTEGER          , INTENT(OUT) :: hdferr
```

Inputs:

loc\_id - Identifier for the dataset containing the reference or for the group that dataset is in.  
ref\_type - Type of reference to query.  
ref - Reference to query.

Outputs:

obj\_type - Type of referenced object.  
H5G\_UNKNOWN\_F  
H5G\_LINK\_F  
H5G\_GROUP\_F  
H5G\_DATASET\_F  
H5G\_TYPE\_F

hdferr - Returns 0 if successful and -1 if fails

Fortran2003 Interface: h5rget\_object\_type\_f

Signature:

```
SUBROUTINE h5rget_object_type_f(loc_id, ref_type, ref, obj_type, hdferr)
  INTEGER(HID_T), INTENT(IN)  :: loc_id
  INTEGER      , INTENT(IN)  :: ref_type
  TYPE(C_PTR)  , INTENT(IN)  :: ref
  INTEGER      , INTENT(OUT) :: obj_type
  INTEGER      , INTENT(OUT) :: hdferr
```

Inputs:

loc\_id - Identifier for the dataset containing the reference or for the group that dataset is in.  
ref\_type - Type of reference to query.  
ref - Reference to query.

Outputs:

obj\_type - Type of referenced object.  
H5G\_UNKNOWN\_F  
H5G\_LINK\_F  
H5G\_GROUP\_F  
H5G\_DATASET\_F  
H5G\_TYPE\_F

hdferr - Returns 0 if successful and -1 if fails

**Description:**

H5R\_GET\_OBJ\_TYPE is a macro that is mapped to either [H5R\\_GET\\_OBJ\\_TYPE1](#) or [H5R\\_GET\\_OBJ\\_TYPE2](#), depending on the needs of the application.

Such macros are provided to facilitate application compatibility. Their use and mappings are fully described in [API Compatibility Macros in HDF5](#); we urge you to read that document closely.

When both the HDF5 library and the application are built and installed with no specific compatibility flags, H5R\_GET\_OBJ\_TYPE is mapped to the most recent version of the function, currently [H5R\\_GET\\_OBJ\\_TYPE2](#). If the library and/or application is compiled for Release 1.6 emulation, H5R\_GET\_OBJ\_TYPE will be mapped to [H5R\\_GET\\_OBJ\\_TYPE1](#). 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	H5Rget_obj_type mapping
-----------------------	-------------------------

---

Global settings

No compatibility flag	H5Rget_obj_type2
Enable deprecated symbols	H5Rget_obj_type2
Disable deprecated symbols	H5Rget_obj_type2
Emulate Release 1.6 interface	H5Rget_obj_type1

---

Function-level macros

H5Rget_obj_type_vers = 2	H5Rget_obj_type2
H5Rget_obj_type_vers = 1	H5Rget_obj_type1

**History:**

Release	Change
1.8.8	Fortran updated to Fortran2003.
1.8.0	The C function H5Rget_obj_type renamed to H5Rget_obj_type1 and deprecated in this release. The C macro H5Rget_obj_type and the C function H5Rget_obj_type2 introduced in this release.

--- Last Modified: May 03, 2019 | 01:23 PM