

# H5P\_GET\_FILTER2

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

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

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

# H5P\_GET\_FILTER2

Returns information about a filter in a pipeline

## Procedure:

H5P\_GET\_FILTER2 ( plist\_id, idx, flags, cd\_nelmts, cd\_values, namelen, name, filter\_config )

## Signature:

```
H5Z_filter_t H5Pget_filter2(  
    hid_t plist_id,  
    unsigned idx,  
    unsigned int *flags,  
    size_t *cd_nelmts,  
    unsigned cd_values[],  
    size_t namelen,  
    char name[],  
    unsigned *filter_config  
)
```

Fortran90 Interface: h5pget\_filter\_f

```
SUBROUTINE h5pget_filter_f(prp_id, filter_number, flags, cd_nelmts,
                          cd_values, namelen, name, filter_id, hdferr)
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: prp_id      ! Property list identifier
  INTEGER, INTENT(IN) :: filter_number     ! Sequence number within the filter
                                          ! pipeline of the filter for which
                                          ! information is sought
  INTEGER, DIMENSION(*), INTENT(OUT) :: cd_values
                                          ! Auxiliary data for the filter
  INTEGER, INTENT(OUT) :: flags           ! Bit vector specifying certain
                                          ! general properties of the filter
  INTEGER(SIZE_T), INTENT(INOUT) :: cd_nelmts
                                          ! Number of elements in cd_values
  INTEGER(SIZE_T), INTENT(IN) :: namelen  ! Anticipated number of characters
                                          ! in name
  CHARACTER(LEN=*), INTENT(OUT) :: name   ! Name of the filter
  INTEGER, INTENT(OUT) :: filter_id      ! Filter identification number
  INTEGER, INTENT(OUT) :: hdferr        ! Error code
                                          ! 0 on success and -1 on failure
END SUBROUTINE h5pget_filter_f
```

### Parameters:

<i>hid_t</i> plist_id	IN: Dataset or group creation property list identifier
<i>int</i> idx	IN: Sequence number within the filter pipeline of the filter for which information is sought
<i>unsigned int</i> * flags	OUT: Bit vector specifying certain general properties of the filter
<i>size_t</i> * cd_nelmts	IN/OUT: Number of elements in cd_values
<i>unsigned int</i> * cd_values	OUT: Auxiliary data for the filter
<i>size_t</i> namelen	IN: Anticipated number of characters in name
<i>char</i> name[ ]	OUT: Name of the filter
<i>unsigned int</i> * filter_config	OUT: Bit field, as described in <a href="#">H5Z_GET_FILTER_INFO</a>

### Description:

H5P\_GET\_FILTER2 returns information about a filter, specified by its filter number, in a filter pipeline, specified by the property list with which it is associated.

plist\_id must be a dataset or group creation property list.

idx is a value between zero and  $N-1$ , as described in [H5P\\_GET\\_NFILTERS](#). The function will return a negative value if the filter number is out of range.

The structure of the flags argument is discussed in [H5P\\_SET\\_FILTER](#).

On input, cd\_nelmts indicates the number of entries in the cd\_values array, as allocated by the caller; on return, cd\_nelmts contains the number of values defined by the filter.

If name is a pointer to an array of at least namelen bytes, the filter name will be copied into that array. The name will be null terminated if namelen is large enough. The filter name returned will be the name appearing in the file, the name registered for the filter, or an empty string.

filter\_config is the bit field described in [H5Z\\_GET\\_FILTER\\_INFO](#).

### Returns:

Returns the filter identifier if successful:

H5Z_FILTER_DEFLATE	Data compression filter, employing the gzip algorithm
H5Z_FILTER_SHUFFLE	Data shuffling filter
H5Z_FILTER_FLETCHER32	Error detection filter, employing the Fletcher32 checksum algorithm
H5Z_FILTER_SZIP	Data compression filter, employing the SZIP algorithm
H5Z_FILTER_NBIT	Data compression filter, employing the N-bit algorithm
H5Z_FILTER_SCALEOFFSET	Data compression filter, employing the scale-offset algorithm

Otherwise returns a negative value.

#### Example:

examples / h5\_cmrss.c [93:115]

1.10/master

HDF5V/hdf5

```
/* Retrieve filter information. */
plist_id = H5Dget_create_plist (dataset_id);

numfilt = H5Pget_nfilters (plist_id);
printf ("Number of filters associated with dataset: %i\n", numfilt);

for (i=0; i<numfilt; i++) {
    nelmts = 0;
    filter_type = H5Pget_filter2 (plist_id, 0, &flags, &nelmts, NULL, 0, NULL,
                                &filter_info);
    printf ("Filter Type: ");
    switch (filter_type) {
        case H5Z_FILTER_DEFLATE:
            printf ("H5Z_FILTER_DEFLATE\n");
            break;
        case H5Z_FILTER_SZIP:
            printf ("H5Z_FILTER_SZIP\n");
            break;
        default:
            printf ("Other filter type included.\n");
    }
}
```

/hdf5

```
! Retrieve filter information.
CALL h5dget_create_plist_f(dataset_id, plist_id, error)

CALL h5pget_nfilters_f(plist_id, numfilt, error)
WRITE(*,'(A, I0)') "Number of filters associated with dataset: ", numfilt

DO i = 1, numfilt
  nelmts = 1
  CALL h5pget_filter_f(plist_id, 0, flags, nelmts, cd_values, &
    namelen, name, filter_id, error)

  WRITE(*,'(30X,A)', ADVANCE='NO')"Filter Type: "
  IF(filter_id.EQ.H5Z_FILTER_DEFLATE_F)THEN
    WRITE(*,'(A)') "H5Z_FILTER_DEFLATE"
  ELSEIF (filter_id.EQ.H5Z_FILTER_SZIP_F)THEN
    WRITE(*,'(A)') "H5Z_FILTER_SZIP"
  ELSE
    WRITE(*,'(A)') "Other filter type included"
  ENDIF
ENDDO
```

**History:**

Release	Change
1.8.5	Function extended to work with group creation property lists.
1.8.0	Function introduced in this release.

--- Last Modified: August 07, 2019 | 10:53 AM