

# H5Z\_GET\_FILTER\_INFO

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

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

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

## H5Z\_GET\_FILTER\_INFO

Retrieves information about a filter

### Procedure:

H5Z\_GET\_FILTER\_INFO (filter, filter\_config)

### Signature:

```
herr_t H5Zget_filter_info( H5Z_filter_t filter, unsigned int *filter_config )
```

```
SUBROUTINE h5zget_filter_info_f(filter, config_flags, hdferr)
```

```
IMPLICIT NONE
INTEGER, INTENT(IN)  :: filter           ! Filter, may be one of the
                                           ! following:
                                           !   H5Z_FILTER_DEFLATE_F
                                           !   H5Z_FILTER_SHUFFLE_F
                                           !   H5Z_FILTER_FLETCHER32_F
                                           !   H5Z_FILTER_SZIP_F
INTEGER, INTENT(OUT) :: config_flags    ! Bit field indicating whether
                                           ! a filter's encoder and/or
                                           ! decoder are available
INTEGER, INTENT(OUT) :: hdferr          ! Error code
```

```
END SUBROUTINE h5zfilter_avail_f
```

### Parameters:

---

<code>H5Z_filter_t filter</code>	IN: Identifier of the filter to query. See the introduction to this section of the reference manual for a list of valid filter identifiers
<code>unsigned int *filter_config</code>	OUT: A bit field encoding the returned filter information

### Description:

`H5Z_GET_FILTER_INFO` retrieves information about a filter. At present, this means that the function retrieves a filter's configuration flags, indicating whether the filter is configured to decode data, to encode data, neither, or both.

If `filter_config` is not set to `NULL` prior to the function call, the returned parameter contains a bit field specifying the available filter configuration. The configuration flag values can then be determined through a series of bitwise `AND` operations, as described below.

Valid filter configuration flags include the following:

<code>H5Z_FILTER_CONFIG_ENCODE_ENABLED</code>	Encoding is enabled for this filter In Fortran, <code>H5Z_FILTER_ENCODE_ENABLED_F</code>
<code>H5Z_FILTER_CONFIG_DECODE_ENABLED</code>	Decoding is enabled for this filter In Fortran, <code>H5Z_FILTER_DECODE_ENABLED_F</code>
(These flags are defined for C in the HDF5 library source code file <code>H5Zpublic.h</code> .)	

A bitwise `AND` of the returned `filter_config` and a valid filter configuration flag will reveal whether the related configuration option is available. For example, if the value of

```
H5Z_FILTER_CONFIG_ENCODE_ENABLED & filter_config
```

is true, i.e., greater than 0 (zero), the queried filter is configured to encode data; if the value is `FALSE`, i.e., equal to 0 (zero), the filter is not so configured.

If a filter is not encode-enabled, the corresponding `H5Pset_*` function will return an error if the filter is added to a dataset creation property list (which is required if the filter is to be used to encode that dataset). For example, if the `H5Z_FILTER_CONFIG_ENCODE_ENABLED` flag is not returned for the SZIP filter, `H5Z_FILTER_SZIP`, a call to `H5P_SET_SZIP` will fail.

If a filter is not decode-enabled, the application will not be able to read an existing file encoded with that filter.

This function should be called, and the returned `filter_config` analyzed, before calling any other function, such as `H5P_SET_SZIP`, that might require a particular filter configuration.

### Returns:

Returns a non-negative value on success, a negative value on failure.

### Example:

Coming Soon!

### History:

Release	Change
1.6.3	Function introduced in this release. Fortran subroutine introduced in this release.