

H5F_GET_MDC_CONFIG

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

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

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

H5F_GET_MDC_CONFIG

Obtains current metadata cache configuration for target file

Procedure:

H5F_GET_MDC_CONFIG (file_id, config_ptr)

Signature:

```
herr_t H5Fget_mdc_config(hid_t file_id, H5AC_cache_config_t *config_ptr)
```

Parameters:

<i>hid_t</i> file_id	IN: Identifier of the target file
<i>H5AC_cache_config_t</i> *config_ptr	IN/OUT: Pointer to the instance of <i>H5AC_cache_config_t</i> in which the current metadata cache configuration is to be reported. The fields of this structure are discussed below:
General configuration section:	
<i>int</i> version	IN: Integer field indicating the the version of the <i>H5AC_cache_config_t</i> in use. This field should be set to <i>H5AC_CURR_CACHE_CONFIG_VERSION</i> (defined in <i>H5ACpublic.h</i>).

<code>hbool_t rpt_fcn_enabled</code>	<p>OUT: Boolean flag indicating whether the adaptive cache resize report function is enabled. This field should almost always be set to disabled (0). Since resize algorithm activity is reported via stdout, it MUST be set to disabled (0) on Windows machines.</p> <p>The report function is not supported code, and can be expected to change between versions of the library. Use it at your own risk.</p>
<code>hbool_t open_trace_file</code>	<p>OUT: Boolean field indicating whether the <code>trace_file_name</code> field should be used to open a trace file for the cache. This field will always be set to 0 in this context.</p>
<code>hbool_t close_trace_file</code>	<p>OUT: Boolean field indicating whether the current trace file (if any) should be closed. This field will always be set to 0 in this context.</p>
<code>char *trace_file_name</code>	<p>OUT: Full path name of the trace file to be opened if the <code>open_trace_file</code> field is set to 1. This field will always be set to the empty string in this context.</p>
<code>hbool_t evictions_enabled</code>	<p>OUT: Boolean flag indicating whether metadata cache entry evictions are enabled.</p>
<code>hbool_t set_initial_size</code>	<p>OUT: Boolean flag indicating whether the cache should be created with a user specified initial maximum size. If the configuration is loaded from the cache, this flag will always be set to 0.</p>
<code>size_t initial_size</code>	<p>OUT: Initial maximum size of the cache in bytes, if applicable. If the configuration is loaded from the cache, this field will contain the cache maximum size as of the time of the call.</p>
<code>double min_clean_fraction</code>	<p>OUT: Float value specifying the minimum fraction of the cache that must be kept either clean or empty when possible.</p>
<code>size_t max_size</code>	<p>OUT: Upper bound (in bytes) on the range of values that the adaptive cache resize code can select as the maximum cache size.</p>
<code>size_t min_size</code>	<p>OUT: Lower bound (in bytes) on the range of values that the adaptive cache resize code can select as the maximum cache size.</p>
<code>long int epoch_length</code>	<p>OUT: Number of cache accesses between runs of the adaptive cache resize code.</p>
Increment configuration section:	
<code>enum H5C_cache_incr_mode incr_mode</code>	<p>OUT: Enumerated value indicating the operational mode of the automatic cache size increase code. At present, only the following values are legal:</p> <p>H5C_incr__off: Automatic cache size increase is disabled.</p> <p>H5C_incr__threshold: Automatic cache size increase is enabled using the hit rate threshold algorithm.</p>
<code>double lower_hr_threshold</code>	<p>OUT: Hit rate threshold used in the hit rate threshold cache size increase algorithm.</p>
<code>double increment</code>	<p>OUT: The factor by which the current maximum cache size is multiplied to obtain an initial new maximum cache size if a size increase is triggered in the hit rate threshold cache size increase algorithm.</p>
<code>hbool_t apply_max_increment</code>	<p>OUT: Boolean flag indicating whether an upper limit will be applied to the size of cache size increases.</p>
<code>size_t max_increment</code>	<p>OUT: The maximum number of bytes by which the maximum cache size can be increased in a single step -- if applicable.</p>

<i>enum</i> <i>H5C_cache_flash_incr_mode</i> <i>flash_incr_mode</i>	OUT: Enumerated value indicating the operational mode of the flash cache size increase code. At present, only the following values are legal: H5C_flash_incr__off: Flash cache size increase is disabled. H5C_flash_incr__add_space: Flash cache size increase is enabled using the add space algorithm.
<i>double</i> <i>flash_threshold</i>	OUT: The factor by which the current maximum cache size is multiplied to obtain the minimum size entry / entry size increase which may trigger a flash cache size increase.
<i>double</i> <i>flash_multiple</i>	OUT: The factor by which the size of the triggering entry / entry size increase is multiplied to obtain the initial cache size increment. This increment may be reduced to reflect existing free space in the cache and the <i>max_size</i> field above.
Decrement configuration section:	
<i>enum</i> <i>H5C_cache_decr_mode</i> <i>decr_mode</i>	OUT: Enumerated value indicating the operational mode of the automatic cache size decrease code. At present, the following values are legal: H5C_decr__off: Automatic cache size decrease is disabled, and the remaining decrement fields are ignored. H5C_decr__threshold: Automatic cache size decrease is enabled using the hit rate threshold algorithm. H5C_decr__age_out: Automatic cache size decrease is enabled using the ageout algorithm. H5C_decr__age_out_with_threshold: Automatic cache size decrease is enabled using the ageout with hit rate threshold algorithm
<i>double</i> <i>upper_hr_threshold</i>	OUT: Upper hit rate threshold. This value is only used if the <i>decr_mode</i> is either <i>H5C_decr__threshold</i> or <i>H5C_decr__age_out_with_threshold</i> .
<i>double</i> <i>decrement</i>	OUT: Factor by which the current max cache size is multiplied to obtain an initial value for the new cache size when cache size reduction is triggered in the hit rate threshold cache size reduction algorithm.
<i>hbool_t</i> <i>apply_max_decrement</i>	OUT: Boolean flag indicating whether an upper limit should be applied to the size of cache size decreases.
<i>size_t</i> <i>max_decrement</i>	OUT: The maximum number of bytes by which cache size can be decreased if any single step, if applicable.
<i>int</i> <i>epochs_before_eviction</i>	OUT: The minimum number of epochs that an entry must reside unaccessed in cache before being evicted under either of the ageout cache size reduction algorithms.
<i>hbool_t</i> <i>apply_empty_reserve</i>	OUT: Boolean flag indicating whether an empty reserve should be maintained under either of the ageout cache size reduction algorithms.
<i>double</i> <i>empty_reserve</i>	OUT: Empty reserve for use with the ageout cache size reduction algorithms, if applicable.
Parallel configuration section:	
<i>int</i> <i>dirty_bytes_threshold</i>	OUT: Threshold number of bytes of dirty metadata generation for triggering synchronizations of the metadata caches serving the target file in the parallel case. Synchronization occurs whenever the number of bytes of dirty metadata created since the last synchronization exceeds this limit.

Description:

H5F_GET_MDC_CONFIG loads the current metadata cache configuration into the instance of H5AC_cache_config_t pointed to by the config_ptr parameter.

Note that the version field of *config_ptr must be initialized --this allows the library to support old versions of the H5AC_cache_config_t structure.

See the overview of the metadata cache in the special topics section of the user manual for details on metadata cache configuration. If you haven't read and understood that documentation, the results of this call will not make much sense.

Returns:

Returns a non-negative value if successful; otherwise returns a negative value.

Example:

History:

Release	Change
1.8.0	Function added in this release.

--- Last Modified: July 22, 2020 | 02:37 PM