

H5P_SET_MDC_IMAGE_CONFIG

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

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

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

H5P_SET_MDC_IMAGE_CONFIG

Sets the metadata cache image option for a file access property list

Procedure:

H5P_SET_MDC_IMAGE_CONFIG (plist_id, config_ptr)

Signature:

```
herr_t H5Pset_mdc_image_config(  
    hid_t plist_id,  
    H5AC_cache_image_config_t * config_ptr)
```

Parameters:

<code>hid_t plist_id</code>	IN: File access property list identifier
<code>H5AC_cache_image_config_t * config_ptr</code>	IN: Pointer to metadata cache image configuration values

Description:

H5P_SET_MDC_IMAGE_CONFIG sets the metadata cache image option with configuration values specified by `config_ptr` for the file access property list specified in `plist_id`.

H5AC_cache_image_config_t is defined as follows:

```

typedef struct H5AC_cache_image_config_t {
    int32_t      version;
    hbool_t     generate_image;
    hbool_t     save_resize_status;
    int32_t     entry_ageout;
} H5AC_cache_image_config_t;

```

Where the fields of H5AC_cache_image_config_t should be initialized as follows:

version:	Must be set to H5AC__CURR_CACHE_IMAGE_CONFIG_VERSION
generate_image:	Set to either TRUE or FALSE depending on whether a cache image is desired
save_resize_status:	Set to either TRUE or FALSE depending on whether the user wishes the metadata cache image resize configuration to be stored in the cache image and restored on file open Note: This field is ignored at present.
entry_ageout:	This field allows the user to specify the number of times a cache entry can appear in subsequent cache images (created in subsequent file closes) without being accessed. The default value is -1, which indicates that the entry may appear in an indefinitely long sequence of cache images. The maximum value is 100. Note: This field is ignored at present.

Limitations:

While it is an obvious error to request a cache image when opening the file read only, it is not in general possible to test for this error in the H5P_SET_MDC_IMAGE_CONFIG call. Rather than fail the subsequent file open, the library silently ignores the file image request in this case.

It is also an error to request a cache image on a file that does not support superblock extension messages (i.e. a superblock version less than 2). As above, it is not always possible to detect this error in the H5P_SET_MDC_IMAGE_CONFIG call, and thus the request for a cache image will fail silently in this case as well.

Creation of cache images is currently disabled in parallel -- as above, any request for a cache image in this context will fail silently.

Files with cache images may be read in parallel applications, but note that the load of the cache image is a collective operation triggered by the first operation that accesses metadata after file open (or, if persistent free space managers are enabled, on the first allocation or deallocation of file space, or read of file space manager status, whichever comes first). Thus the parallel process may deadlock if any process does not participate in this access.

In long sequences of file closes and opens, infrequently accessed metadata can accumulate in the cache image to the point where the cost of storing and restoring this metadata exceeds the benefit of retaining frequently used metadata in the cache image. When implemented, the entry_ageout should address this problem. In the interim, not requesting a cache image every n file close/open cycles may be an acceptable work around. The choice of n will be driven by application behavior, but n = 10 seems a good starting point.

Returns:

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

Example:

Coming Soon!

History:

Release	Change
1.10.1	C function introduced with this release.

--- Last Modified: July 23, 2019 | 08:33 AM