H5_RESIZE_MEMORY

Expand all Collapse all

- Jump to ...
- Summary
- Description
- Example
- Switch language ...
- C
- C++
- FORTRAN
- JAVA

Summary
Description
Example
JAVA
FORTRAN
C++
C

H5_RESIZE_MEMORY

Resizes and, if required, re-allocates memory that will later be freed internally by the HDF5 library

Procedure:

H5_RESIZE_MEMORY(mem, size)

Signature:

```
void * H5resize_memory
(
    void   *mem,
    size_t  size
)
```

Parameters:

void *mem	IN: Pointer to a buffer to be resized May be NULL
size_t size	IN: New size of the buffer, in bytes

Description:

H5_RESIZE_MEMORY takes a pointer to an existing buffer and resizes the buffer to match the value in `size`. If necessary, the buffer is reallocated. If `size` is `0`, the buffer is released.

The input buffer must either be `NULL` or have been allocated by H5_ALLOCATE_MEMORY since the input buffer may be freed by the library.

For certain behaviors, the pointer `mem` may be passed in as `NULL`.

This function is intended to have the semantics of `realloc()`:

`H5resize_memory(buffer, size)`	Resizes `buffer`. Returns pointer to resized buffer.
`H5resize_memory(NULL, size)`	Allocates memory using HDF5 Library allocator. Returns pointer to new buffer
`H5resize_memory(buffer, 0)`	Frees memory using HDF5 Library allocator. Returns `NULL`.
`H5resize_memory(NULL, 0)`	Returns `NULL` (undefined in C standard).

Unlike `realloc()`, which allows for a “special” pointer to be returned instead of `NULL`, this function always returns `NULL` on failure or when size is `0` (zero).

At this time, the only intended use for this function is to resize or reallocate memory that will be returned to the library (and eventually to the user) as a data buffer from a third-party HDF5 filter.

> To avoid heap corruption, allocated memory should be freed using the same library that initially allocated it. In most cases, the HDF5 API uses resources that are allocated and freed either entirely by the user or entirely by the library, so this is not a problem. In rare cases, however, HDF5 API calls will free memory that the user allocated. This function allows the user to safely allocate this memory.
>
> It is particularly important to use this function to resize memory on Microsoft Windows systems. In Windows, the C standard library is implemented in dynamic link libraries (DLLs) known as the C run-time (CRT). Each version of Visual Studio comes with multiple versions of the CRT DLLs (debug, release, et cetera) and allocating and freeing memory across DLL boundaries can cause resource leaks and subtle bugs due to heap corruption.
>
> Even when using this function, it is still best to ensure that all components of a C application are built with the same version of Visual Studio and the same configuration (Debug or Release), and thus linked against the same CRT.
>
> Only use this function to resize memory inside third-party HDF5 filters. It will generally not be safe to use this function to resize memory for any other purpose.

Returns:

On success, returns pointer to resized or reallocated buffer or returns `NULL` if size is `0` (zero).

Returns `NULL` on failure.

Example:

Coming soon!

See Also:

- H5_ALLOCATE_MEMORY — Allocates memory that will later be freed internally by the HDF5 library
- H5_FREE_MEMORY — Frees memory allocated by the HDF5 library

History:

Release	Change
1.8.15	C function introduced with this release.

--- Last Modified: April 10, 2018 | 01:46 PM