H5D_VLEN_RECLAIM

Expand all Collapse all

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

H5D_VLEN_RECLAIM

Reclaims variable-length (VL) datatype memory buffers (DEPRECATED)

> *This function has been deprecated in HDF5-1.12 in favor of the function H5T_RECLAIM.*

Procedure:

H5D_VLEN_RECLAIM(type_id, space_id, plist_id, buf)

Signature:

```
herr_t H5Dvlen_reclaim( hid_t type_id, hid_t space_id, hid_t plist_id, void *buf )
```

```
Fortran2003: SUBROUTINE h5dvlen_reclaim_f(type_id, space_id, plist_id, buf, hdferr)
    INTEGER(HID_T), INTENT(IN)    :: type_id
    INTEGER(HID_T), INTENT(IN)    :: space_id
    INTEGER(HID_T), INTENT(IN)    :: plist_id
    TYPE(C_PTR)   , INTENT(INOUT) :: buf
    INTEGER       , INTENT(OUT)   :: hdferr
```

Parameters:

hid_t `type_id`	IN: Identifier of the datatype
hid_t `space_id`	IN: Identifier of the dataspace
hid_t `plist_id`	IN: Identifier of the property list used to create the buffer
void *`buf`	IN:Pointer to the buffer to be reclaimed

Description:

H5D_VLEN_RECLAIM reclaims memory buffers created to store VL datatypes.

The `type_id` must be the datatype stored in the buffer. The `space_id` describes the selection for the memory buffer to free the VL datatypes within. The `plist_id` is the dataset transfer property list which was used for the I/O transfer to create the buffer. And `buf` is the pointer to the buffer to be reclaimed.

The VL structures (`hvl_t`) in the user's buffer are modified to zero out the VL information after the memory has been reclaimed.

If nested VL datatypes were used to create the buffer, this routine frees them *from the bottom up*, releasing all the memory without creating memory leaks.

Returns:

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

Example:

Coming Soon!

History:

Release	Change
1.12.0	Routine was deprecated
1.8.8	Fortran interface was added

--- Last Modified: December 17, 2019 | 12:00 PM