

H5T_VLEN_CREATE

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

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

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

H5T_VLEN_CREATE

Creates a new variable-length array datatype.

Procedure:

H5T_VLEN_CREATE(base_type_id)

Signature:

```
hid_t H5Tvlen_create( hid_t base_type_id )
```

```
SUBROUTINE h5tvlen_create_f(type_id, vltype_id, hdferr)
  IMPLICIT NONE
  INTEGER(HID_T), INTENT(IN) :: type_id      ! Datatype identifier of base type
                                           ! Base type can only be atomic
  INTEGER(HID_T), INTENT(OUT) :: vltype_id ! VL datatype identifier
  INTEGER, INTENT(OUT) :: hdferr           ! Error code
END SUBROUTINE h5tvlen_create_f
```

Parameters:

<i>hid_t</i> base_type_id	IN: Base type of datatype to create.
---------------------------	--------------------------------------

Description:

H5Tvlen_create creates a new one-dimensional array datatype of variable-length (VL) with the base datatype base_type_id.

This one-dimensional array often represents a data sequence of the base datatype, such as characters for character sequences or vertex coordinates for polygon lists. The base type specified for the VL datatype can be any HDF5 datatype, including another VL datatype, a compound datatype, or an atomic datatype.

When necessary, use `H5Tget_super` to determine the base type of the VL datatype.

The datatype identifier returned from this function should be released with `H5Tclose` or resource leaks will result. Under certain circumstances, `H5Dvlen_reclaim` must also be used.

`H5Tvlen_create` cannot be used to create a variable-length string datatype. `H5Tvlen_create` called with a string or character base type creates a variable-length sequence of strings (a variable-length, 1-dimensional array), with each element of the array being of the string or character base type.

To create a variable-length string datatype, see [“Creating variable-length string datatypes.”](#)

Returns:

Returns datatype identifier if successful; otherwise returns a negative value.

Example:

Coming Soon!

History:

Release	Change
1.4.5	Fortran subroutine introduced in this release.

--- Last Modified: January 24, 2018 | 11:04 AM