

H5VL_REGISTER_CONNECTOR

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

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

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

H5VL_REGISTER_CONNECTOR

Registers a new VOL connector

Procedure:

H5VL_REGISTER_CONNECTOR (cls, vipl_id)

Signature:

```
hid_t H5VLregister_connector(const H5VL_class_t *cls, hid_t vipl_id)
```

Parameters:

<code>const H5VL_class_t *cls</code>	IN: A pointer to the plugin structure to register
<code>hid_t vipl_id</code>	IN: Property list identifier

Description

H5VL_REGISTER_CONNECTOR registers a new VOL connector as a member of the virtual object layer class. This VOL connector identifier is good until the library is closed or the connector is unregistered.

`vipl_id` is either H5P_DEFAULT or the identifier of a VOL initialization property list created with [H5P_CREATE](#) (`H5Pcreate(H5P_VOL_INITIALIZE)`). When created, this property list contains no library properties. If a VOL connector author decides that initialization-specific data are needed, they can be added to the empty list and retrieved by the connector in the VOL connector's initialize callback. Use of the VOL initialization property list is uncommon, as most VOL-specific properties are added to the file access property list via the connector's API calls which set the VOL connector for the file open/create. For more information, see the VOL documentation.

`H5VL_class_t` is defined in `H5VLconnector.h` in the source code. It contains class information for each VOL connector:

src / H5VLconnector.h [463:494]

hdf5_1_12 HDF5V/hdf5

```
/* Infrastructure / Services */
H5VL_introspect_class_t introspect_cls; /* Container/connector introspection class
callbacks */
H5VL_request_class_t request_cls; /* Asynchronous request class callbacks */
H5VL_blob_class_t blob_cls; /* 'Blob' class callbacks */
H5VL_token_class_t token_cls; /* VOL connector object token class
callbacks */

/* Catch-all */
herr_t (*optional)(void *obj, int op_type, hid_t dxpl_id, void **req,
va_list arguments); /* Optional callback */
} H5VL_class_t;

/*****/
/* Public Variables */
/*****/

/*****/
/* Public Prototypes */
/*****/

#ifdef __cplusplus
extern "C" {
#endif

/* Helper routines for VOL connector authors */
H5_DLL hid_t H5VLregister_connector(const H5VL_class_t *cls, hid_t vipl_id);
H5_DLL void *H5VLobject(hid_t obj_id);
H5_DLL hid_t H5VLget_file_type(void *file_obj, hid_t connector_id, hid_t dtype_id);
H5_DLL hid_t H5VLpeek_connector_id_by_name(const char *name);
H5_DLL hid_t H5VLpeek_connector_id_by_value(H5VL_class_value_t value);

#ifdef __cplusplus
```

Returns:

Returns a VOL connector identifier on success and `H5I_INVALID_HID` on failure.

Example:

None

History:

Release	Change
1.12.0	Function introduced

--- Last Modified: February 18, 2020 | 09:09 AM