

# H5T\_COMPILER\_CONV

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

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

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

# H5T\_COMPILER\_CONV

Check whether the library's default conversion is hard conversion

## Procedure:

H5T\_COMPILER\_CONV(src\_id, dst\_id)

## Signature:

```
htri_t H5tcompiler_conv ( hid_t src_id, hid_t dst_id )
```

```
SUBROUTINE h5tcompiler_conv_f( src_id, dst_id, flag, hdferr)  
  IMPLICIT NONE  
  INTEGER(HID_T), INTENT(IN) :: src_id ! Id for the source datatype.  
  INTEGER(HID_T), INTENT(IN) :: dst_id ! Id for the destination datatype.  
  LOGICAL, INTENT(OUT) :: flag        ! Positive for compiler conversion,  
                                       ! Zero for library conversion  
  INTEGER, INTENT(OUT) :: hdferr      ! Error code:  
                                       ! 0 on success and -1 on failure  
END SUBROUTINE h5tcompiler_conv_f
```

## Parameters:

<i>hid_t</i> src_id	IN: Identifier for the source datatype
<i>hid_t</i> dst_id	IN: Identifier for the destination datatype

**Description:**

H5T\_COMPILER\_CONV determines whether the library's conversion function from type `src_id` to type `dst_id` is a compiler (hard) conversion or not. A compiler conversion uses compiler's casting; a library (soft) conversion uses the library's own conversion function.

**Returns:**

Returns a positive value if a compiler conversion was used.  
Returns 0 if a library conversion was not used.  
Returns a negative value when the function fails.

**Example:****History:**

Release	Change
1.8.0	Call introduced in this release.

--- Last Modified: July 22, 2020 | 03:54 PM