

H5_CHECK_VERSION

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

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

H5_CHECK_VERSION

Verifies that HDF5 library versions are consistent

Procedure:

H5_CHECK_VERSION(majnum, minnum, relnum)

Signature:

```
herr_t H5check_version  
(  
    unsigned majnum,  
    unsigned minnum,  
    unsigned relnum  
)
```

```
SUBROUTINE h5check_version_f(majnum, minnum, relnum, error)  
    INTEGER, INTENT(IN)  :: majnum, minnum, relnum  
    INTEGER, INTENT(OUT) :: error
```

Parameters:

<i>unsigned</i> majnum	IN: HDF5 library major version number
<i>unsigned</i> minnum	IN: HDF5 library minor version number
<i>unsigned</i> relnum	IN: HDF5 library release number

Description:

H5_CHECK_VERSION verifies that the version of the HDF5 library with which an application was compiled, as indicated by the passed parameters, matches the version of the HDF5 library against which the application is currently linked.

`majnum` is the major version number of the HDF5 library with which the application was compiled, `minnum` is the minor version number, and `relnum` is the release number. Consider the following illustration:

An official HDF5 release is labelled as follows:

HDF5 Release <`majnum`>.<`minnum`>.<`relnum`>

For example, in HDF5 Release 1.8.5:

- 1 is the major version number, `majnum`.
- 8 is the minor version number, `minnum`.
- 5 is the release number, `relnum`.

As stated above, `H5_CHECK_VERSION` first verifies that the version of the HDF5 library with which an application was compiled matches the version of the HDF5 library against which the application is currently linked. If this check fails, `H5_CHECK_VERSION` causes the application to abort (by means of a standard C `abort()` call) and prints information that is usually useful for debugging. This precaution is taken to avoid the risks of data corruption or segmentation faults.

The most common cause of this failure is that an application was compiled with one version of HDF5 and is dynamically linked with a different version different version.

If the above test passes, `H5_CHECK_VERSION` proceeds to verify the consistency of additional library version information. This is designed to catch source code inconsistencies that do not normally cause failures; if this check reveals an inconsistency, an informational warning is printed but the application is allowed to run.

Returns:

Returns a non-negative value if successful. Upon failure, this function causes the application to abort.

Example:

Coming soon!

History:

Release	Change
1.4.5	Fortran subroutine introduced in this release.
1.8.8	Fortran updated to Fortran 2003.