

H5S_GET_REGULAR_HYPERSLAB

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

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

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

H5S_GET_REGULAR_HYPERSLAB

Retrieves a regular hyperslab selection

Procedure:

H5S_GET_REGULAR_HYPERSLAB (space_id, start, stride, count, block)

Signature:

```
herr_t H5Sget_regular_hyperslab(  
    hid_t space_id,  
    hsize_t start[],  
    hsize_t stride[],  
    hsize_t count[],  
    hsize_t block[]  
)
```

Fortran Interface: `h5sget_regular_hyperslab_f`

Signature:

```
SUBROUTINE h5sget_regular_hyperslab_f(space_id, start, stride, count,
                                     block, hdferr)
    INTEGER(HID_T), INTENT(IN) :: space_id
    INTEGER(HSIZE_T), INTENT(OUT), DIMENSION(*), TARGET :: start
    INTEGER(HSIZE_T), INTENT(OUT), DIMENSION(*), TARGET :: stride
    INTEGER(HSIZE_T), INTENT(OUT), DIMENSION(*), TARGET :: count
    INTEGER(HSIZE_T), INTENT(OUT), DIMENSION(*), TARGET :: block
    INTEGER, INTENT(OUT) :: hdferr
```

Inputs:

`space_id` - The identifier of the dataspace.

Outputs:

`start` - Offset of the start of the regular hyperslab.
`stride` - Stride of the regular hyperslab.
`count` - Number of blocks in the regular hyperslab.
`block` - Size of a block in the regular hyperslab.
`hdferr` - Returns 0 if successful and -1 if fails.

Parameters:

<code>hid_t space_id</code>	IN: The identifier of the dataspace
<code>hsize_t start[]</code>	OUT: Offset of the start of the regular hyperslab
<code>hsize_t stride[]</code>	OUT: Stride of the regular hyperslab
<code>hsize_t count[]</code>	OUT: Number of blocks in the regular hyperslab
<code>hsize_t block[]</code>	OUT: Size of a block in the regular hyperslab

Description:

`H5S_GET_REGULAR_HYPERSLAB` takes the dataspace identifier, `space_id`, and retrieves the values of `start`, `stride`, `count`, and `block` for the regular hyperslab selection.

A regular hyperslab selection is a hyperslab selection described by setting the `offset`, `stride`, `count`, and `block` parameters to the `H5S_SELECT_HYPERSLAB` call. If several calls to `H5S_SELECT_HYPERSLAB` are needed, the hyperslab selection is irregular.

See `H5S_SELECT_HYPERSLAB` for descriptions of `offset`, `stride`, `count`, and `block`.

Note: If a hyperslab selection is originally regular, then becomes irregular through selection operations, and then becomes regular again, the final regular selection may be equivalent but not identical to the original regular selection.

Returns:

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

Example:

```

if (H5Sis_regular_hyperslab(vspace)) {
    status = H5Sget_regular_hyperslab (vspace, start_out, stride_out,
count_out, block_out);
    printf("        start = [%llu, %llu] \n", (unsigned long
long)start_out[0], (unsigned long long)start_out[1]);
    printf("        stride = [%llu, %llu] \n", (unsigned long
long)stride_out[0], (unsigned long long)stride_out[1]);
    printf("        count = [%llu, %llu] \n", (unsigned long
long)count_out[0], (unsigned long long)count_out[1]);
    printf("        block = [%llu, %llu] \n", (unsigned long
long)block_out[0], (unsigned long long)block_out[1]);
}

```

```
PROGRAM COMPOUNDEXAMPLE
```

```
USE HDF5 ! This module contains all necessary modules
```

```
IMPLICIT NONE
```

```
CHARACTER(LEN=11), PARAMETER :: filename = "compound.h5" ! File name
```

```
CHARACTER(LEN=8), PARAMETER :: dsetname = "Compound" ! Dataset name
```

```
INTEGER, PARAMETER :: dimsize = 6 ! Size of the dataset
```

```
INTEGER(HID_T) :: file_id ! File identifier
```

History:

Release	Change
1.10.0	C function introduced with this release.