Reading From or Writing To a Subset of a Dataset

There are two ways that you can select a subset in an HDF5 dataset and read or write to it:

> ***Hyperslab Selection*** : The `H5S_SELECT_HYPERSLAB` call selects a logically contiguous collection of points in a dataspace, or a regular pattern of points or blocks in a dataspace.
>
> ***Element Selection****:* The `H5S_SELECT_ELEMENTS` call selects elements in an array.

HDF5 allows you to read from or write to a portion or subset of a dataset by:

- Selecting a Subset of the Dataset's Dataspace,
- Selecting a Memory Dataspace,
- Reading From or Writing to a Dataset Subset.

Selecting a Subset of the Dataset's Dataspace

First you must obtain the dataspace of a dataset in a file by calling H5D_GET_SPACE .

Then select a subset of that dataspace by calling H5S_SELECT_HYPERSLAB. The *offset*, *count*, *stride* and *block* parameters of this API define the shape and size of the selection. They must be arrays with the same number of dimensions as the rank of the dataset's dataspace. These arrays **ALL** work together to define a selection. A change to one of these arrays can affect the others.

> *offset*: An array that specifies the offset of the starting element of the specified hyperslab.
>
> *count*: An array that determines how many blocks to select from the dataspace in each dimension. If the block size for a dimension is one then the *count* is the number of elements along that dimension.
>
> *stride*: An array that allows you to sample elements along a dimension. For example, a stride of one (or NULL) will select every element along a dimension, a stride of two will select every other element, and a stride of three will select an element after every two elements.
>
> *block*: An array that determines the size of the element block selected from a dataspace. If the block size is one or NULL then the block size is a single element in that dimension.

Selecting a Memory Dataspace

You must select a memory dataspace in addition to a file dataspace before you can read a subset from or write a subset to a dataset. A memory dataspace can be specified by calling H5S_CREATE_SIMPLE.

The memory dataspace passed to the read or write call *must* contain the same number of elements as the file dataspace. The number of elements in a dataspace selection can be determined with the H5S_GET_SELECT_NPOINTS API.

Reading From or Writing To a Dataset Subset

To read from or write to a dataset subset, the H5D_READ and H5D_WRITE routines are used. The memory and file dataspace identifiers from the selections that were made are passed into the read or write call. For example (C):

```
status = H5Dwrite (.., .., memspace_id, dataspace_id, .., ..);
```

Programming Example

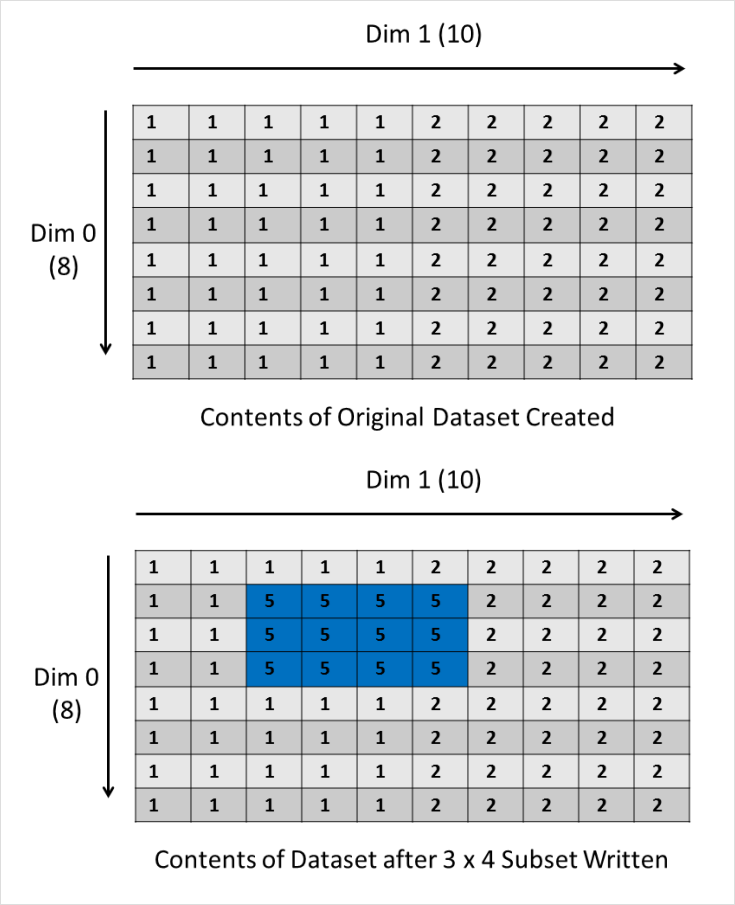
Description

See HDF5 Introductory Examples for the examples used in the Learning the Basics tutorial.

The example creates an 8 x 10 integer dataset in an HDF5 file. It then selects and writes to a 3 x 4 subset of the dataset created with the dimensions offset by 1 x 2. (If using Fortran, the dimensions will be swapped. The dataset will be 10 x 8, the subset will be 4 x 3, and the offset will be 2 x 1.)

PLEASE NOTE that the examples and images below were created using C.

The following image shows the dataset that gets written originally, and the subset of data that gets modified afterwards. Dimension 0 is vertical and Dimension 1 is horizontal as shown below:



The subset on the right above is created using these values for offset, count stride, and block:

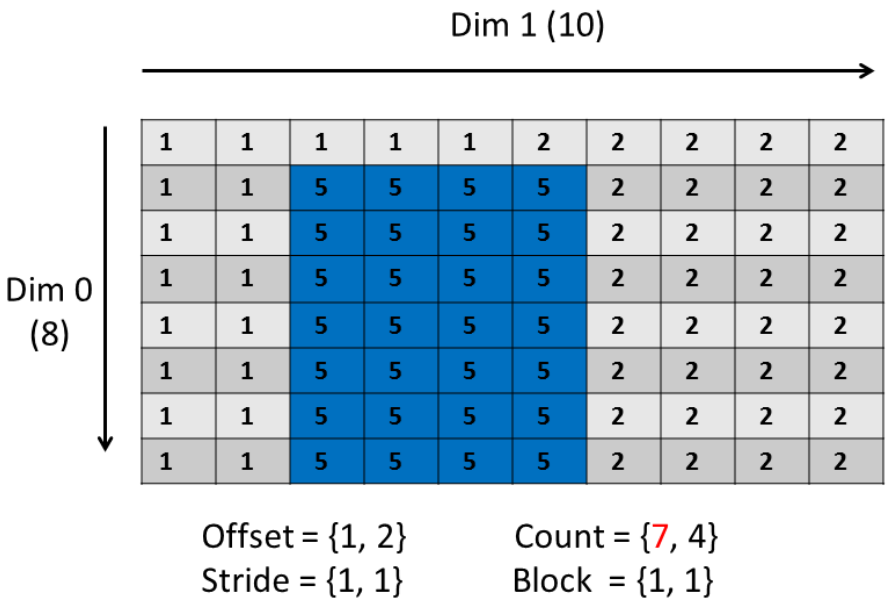
```
offset = {1, 2}
count = {3, 4}
stride = {1, 1}
block = {1, 1}
```

Experiments with Different Selections

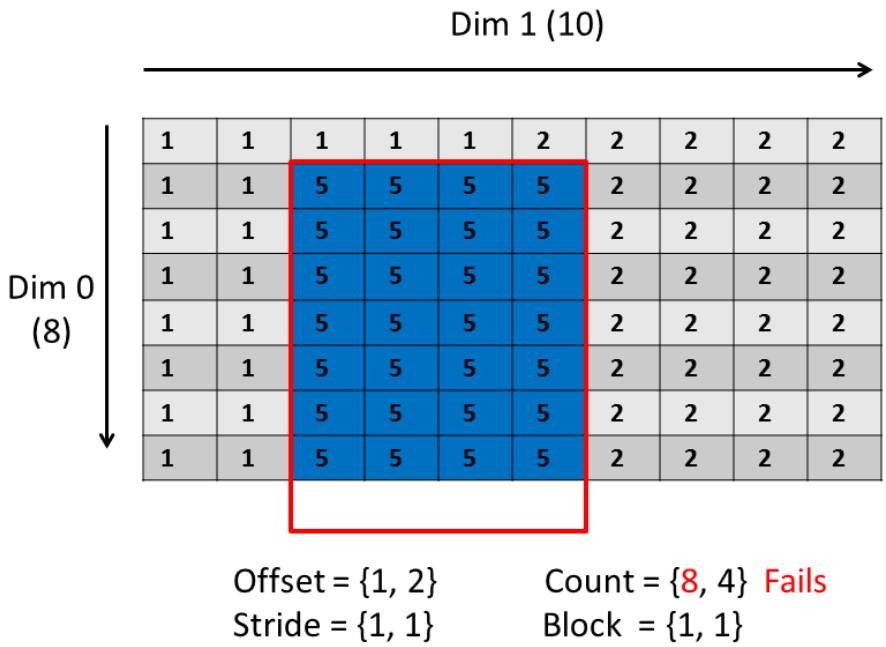
Following are examples of changes that can be made to the example code provided to better understand how to make selections.

Example 1:

By default the example code will select and write to a 3 x 4 subset. You can modify the *count* parameter in the example code to select a different subset, by changing the value of DIM0_SUB (C, C++) / dim0_sub (Fortran) near the top. Change its value to 7 to create a 7 x 4 subset:



If you were to change the subset to 8 x 4, the selection would be beyond the extent of the dimension:



The write will fail with the error: **"file selection+offset not within extent"**

Example 2:

In the example code provided, the memory and file dataspaces passed to the H5Dwrite call have the same size, 3 x 4 (DIM0_SUB x DIM1_SUB). Change the size of the memory dataspace to be 4 x 4 so that they do not match, and then compile:

```
dimsm[0] = DIM0_SUB + 1;
    dimsm[1] = DIM1_SUB;
    memspace_id = H5Screate_simple (RANK, dimsm, NULL);
```

The code will fail with the error: **"src and dest data spaces have different sizes"**

How many elements are in the memory and file dataspaces that were specified above? Add these lines:

```
hssize_t    size;

    /* Just before H5Dwrite call the following */
    size = H5Sget_select_npoints (memspace_id);
    printf ("\nmemspace_id size: %i\n", size);
    size = H5Sget_select_npoints (dataspace_id);
    printf ("dataspace_id size: %i\n", size);
```

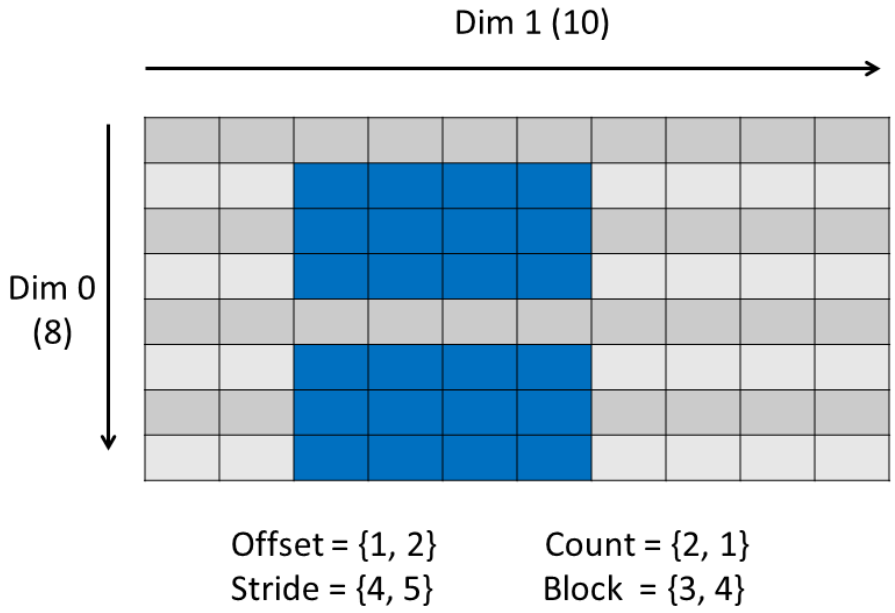
You should see these lines followed by the error:

```
memspace_id size: 16
    dataspace_id size: 12
```

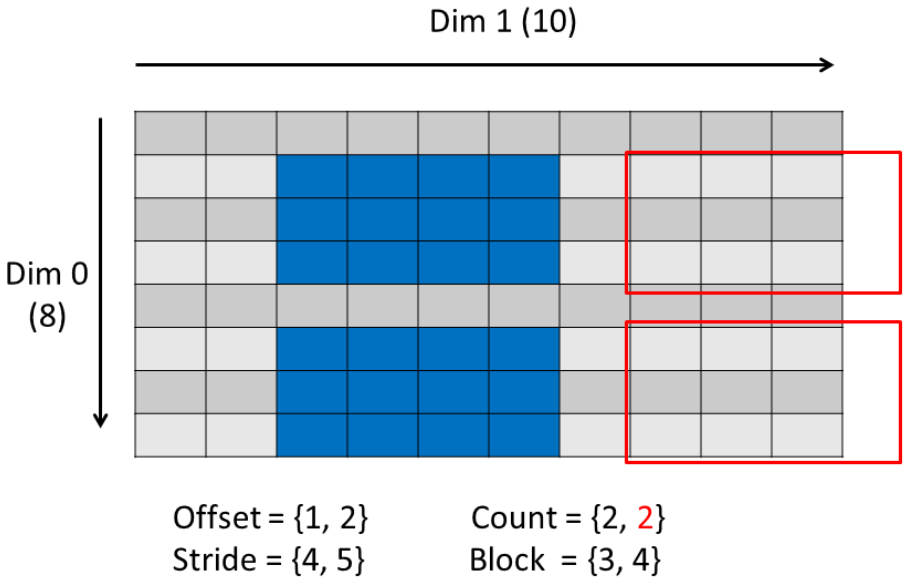
Example 3:

This example shows the selection that occurs if changing the values of the *offset*, *count*, *stride* and *block* parameters in the example code.

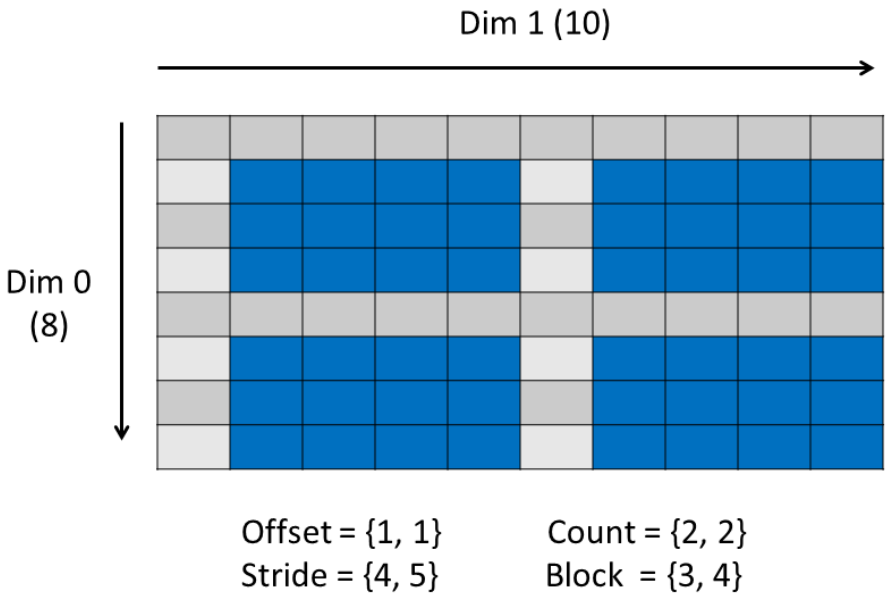
This will select two blocks. The *count* array specifies the number of blocks. The *block* array specifies the size of a block. The *stride* must be modified to accommodate the block size.



Now try modifying the *count* as shown below. The write will fail because the selection goes beyond the extent of the dimension:



If the offset were 1x1 (instead of 1x2), then the selection can be made:



The selections above were tested with the h5_subsetbk.c example code. The memory dataspace was defined as one-dimensional.

Remarks

- In addition to H5S_SELECT_HYPERSLAB, this example introduces the H5D_GET_SPACE call to obtain the dataspace of a dataset.
- If using the default values for the *stride* and *block* parameters of H5S_SELECT_HYPERSLAB, then, for C you can specify NULL for these parameters, rather than passing in an array for each, and for Fortran 90 you can omit these parameters.