

Command-Line Tools for Editing HDF5 Files

Contents:

- [Remove Inaccessible Objects and Unused Space in a File](#)
- [Change a Dataset's Storage Layout](#)
- [Apply Compression Filter to a Dataset](#)
- [Copy Objects to Another File](#)
- [Add or Remove User Block from File](#)

Remove Inaccessible Objects and Unused Space in a File

HDF5 files may accumulate unused space when they are read and rewritten to or if objects are deleted within them. With many edits and deletions this unused space can add up to a sizable amount.

The `h5repack` tool can be used to remove unused space in an HDF5 file. If no options other than the input and output HDF5 files are specified on the `h5repack` command line, it will write the file to the new file, getting rid of the unused space:

```
h5repack <input file> <output file>
```

Change a Dataset's Storage Layout

The `h5repack` utility can be used to change a dataset's storage layout. By default, the storage layout of a dataset is defined at creation time and it cannot be changed. However, with `h5repack` you can write an HDF5 file to a new file and change the layout for objects in the new file.

The `-l` option in `h5repack` is used to change the layout for an object. The string following the `-l` option defines the layout type and parameters for specified objects (or all objects):

```
h5repack -l [list of objects:]<layout type>=<layout parameters> <input file> <output file>
```

If no object is specified, then everything in the input file will be written to the output file with the specified layout type and parameters. If objects are specified then everything in the input file will be written to the output file as is, except for those specified objects. They will be written to the output file with the given layout type and parameters.

Following is a description of the dataset layouts and the `h5repack` options to use to change a dataset:

Storage Layout	h5repack Option	Description
Contiguous	CONTI	Data is stored physically together
Chunked	CHUNK=DIM[xDIM...xDIM]	Data is stored in DIM[xDIM...xDIM] chunks
Compact	COMPA	Data is stored in the header of the object (less I/O)

If you type `h5repack -h` on the command line, you will see a detailed usage statement with examples of modifying the layout.

In the following example, the dataset `/dset` in the file `dset.h5` is contiguous, as shown by the `h5dump -pH` command. The `h5repack` utility writes `dset.h5` to a new file, `dsetrp.h5`, where the dataset `dset` is chunked. This can be seen by examining the resulting `dsetrp.h5` file with `h5dump`, as shown:

```

$ h5dump -pH dset.h5
HDF5 "dset.h5" {
GROUP "/" {
  DATASET "dset" {
    DATATYPE  H5T_STD_I32BE
    DATASPACE SIMPLE { ( 4, 6 ) / ( 4, 6 ) }
    STORAGE_LAYOUT {
      CONTIGUOUS
      SIZE 96
      OFFSET 1400
    }
    FILTERS {
      NONE
    }
    FILLVALUE {
      FILL_TIME H5D_FILL_TIME_IFSET
      VALUE 0
    }
    ALLOCATION_TIME {
      H5D_ALLOC_TIME_LATE
    }
  }
}
}

$ h5repack -l dset:CHUNK=4x6 dset.h5 dsetrp.h5

$ h5dump -pH dsetrp.h5
HDF5 "dsetrp.h5" {
GROUP "/" {
  DATASET "dset" {
    DATATYPE  H5T_STD_I32BE
    DATASPACE SIMPLE { ( 4, 6 ) / ( 4, 6 ) }
    STORAGE_LAYOUT {
      CHUNKED ( 4, 6 )
      SIZE 96
    }
    FILTERS {
      NONE
    }
    FILLVALUE {
      FILL_TIME H5D_FILL_TIME_IFSET
      VALUE 0
    }
    ALLOCATION_TIME {
      H5D_ALLOC_TIME_INCR
    }
  }
}
}

```

There can be many reasons that the storage layout needs to be changed for a dataset. For example, there may be a performance issue with a dataset due to a small chunk size.

Apply Compression Filter to a Dataset

The `h5repack` utility can be used to compress or remove compression from a dataset in a file. By default, compression cannot be added to or removed from a dataset once it has been created. However, with `h5repack` you can write a file to a new file and specify a compression filter to apply to a dataset or datasets in the new file.

To apply a filter to an object in an HDF5 file, specify the `-f` option, where the string following the `-f` option defines the filter and its parameters (if there are any) to apply to a given object or objects:

```
h5repack -f [list of objects:]<name of filter>=<filter parameters> <input file> <output file>
```

If no objects are specified then everything in the input file will be written to the output file with the filter and parameters specified. If objects are specified, then everything in the input file will be written to the output file as is, except for the specified objects. They will be written to the output file with the filter and parameters specified.

If you type `h5repack --help` on the command line, you will see a detailed usage statement with examples of modifying a filter. There are actually numerous filters that you can apply to a dataset:

Filter	Options
GZIP compression (levels 1-9)	GZIP=<deflation level>
SZIP compression	SZIP=<pixels per block,coding>
Shuffle filter	SHUF
Checksum filter	FLET
NBIT compression	NBIT
HDF5 Scale/Offset filter	SOFF=<scale_factor,scale_type>
User defined filter	UD=<filter_number,cd_value_count,value_1[,value_2,...,value_N]>
Remove ALL filters	NONE

Be aware that a dataset must be chunked to apply compression to it. If the dataset is not already chunked, then `h5repack` will apply chunking to it. Both chunking and compression cannot be applied to a dataset at the same time with `h5repack`.

In the following example,

- `h5dump` lists the properties for the objects in `dset.h5`. Note that the dataset `dset` is contiguous.
- `h5repack` writes `dset.h5` into a new file `dsetrp.h5`, applying GZIP Level 5 compression to the dataset `/dset` in `dsetrp.h5`.
- `h5dump` lists the properties for the new `dsetrp.h5` file. Note that `/dset` is both compressed and chunked.

Example:

```

$ h5dump -pH dset.h5
HDF5 "dset.h5" {
GROUP "/" {
  DATASET "dset" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 12, 18 ) / ( 12, 18 ) }
    STORAGE_LAYOUT {
      CONTIGUOUS
      SIZE 864
      OFFSET 1400
    }
    FILTERS {
      NONE
    }
    FILLVALUE {
      FILL_TIME H5D_FILL_TIME_IFSET
      VALUE 0
    }
    ALLOCATION_TIME {
      H5D_ALLOC_TIME_LATE
    }
  }
}
}

$ h5repack -f dset:GZIP=5 dset.h5 dsetrp.h5

$ h5dump -pH dsetrp.h5
HDF5 "dsetrp.h5" {
GROUP "/" {
  DATASET "dset" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 12, 18 ) / ( 12, 18 ) }
    STORAGE_LAYOUT {
      CHUNKED ( 12, 18 )
      SIZE 160 (5.400:1 COMPRESSION)
    }
    FILTERS {
      COMPRESSION DEFLATE { LEVEL 5 }
    }
    FILLVALUE {
      FILL_TIME H5D_FILL_TIME_IFSET
      VALUE 0
    }
    ALLOCATION_TIME {
      H5D_ALLOC_TIME_INCR
    }
  }
}
}

```

Copy Objects to Another File

The `h5copy` utility can be used to copy an object or objects from one HDF5 file to another or to a different location in the same file. It uses the [H5O_COPY](#) and [H5L_COPY](#) APIs in HDF5.

Following are some of the options that can be used with `h5copy`.

h5copy Options	Description
<code>-i, --input</code>	Input file name
<code>-o, --output</code>	Output file name

-s, --source	Source object name
-d, --destination	Destination object name
-p, --parents	Make parent groups as needed
-v, --verbose	Verbose mode
-f, --flag	Flag type

For a complete list of options and information on using `h5copy`, type:

```
h5copy -help
```

In the example below, the dataset `/MyGroup/Group_A/dset2` in `groups.h5` gets copied to the root ("/") group of a new file, `newgroup.h5`, with the name `dset3`:

```
$h5dump -H groups.h5
HDF5 "groups.h5" {
GROUP "/" {
  GROUP "MyGroup" {
    GROUP "Group_A" {
      DATASET "dset2" {
        DATATYPE H5T_STD_I32BE
        DATASPACE SIMPLE { ( 2, 10 ) / ( 2, 10 ) }
      }
    }
    GROUP "Group_B" {
    }
  }
  DATASET "dset1" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 3, 3 ) / ( 3, 3 ) }
  }
}
}

$ h5copy -i groups.h5 -o newgroup.h5 -s /MyGroup/Group_A/dset2 -d /dset3

$ h5dump -H newgroup.h5
HDF5 "newgroup.h5" {
GROUP "/" {
  DATASET "dset3" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 2, 10 ) / ( 2, 10 ) }
  }
}
}
```

There are also `h5copy` flags that can be specified with the `-f` option. In the example below, the `-f shallow` option specifies to copy only the immediate members of the group `/MyGroup` from the `groups.h5` file mentioned above to a new file `mygrouponly.h5`:

```
h5copy -v -i groups.h5 -o mygrouponly.h5 -s /MyGroup -d /MyGroup -f shallow
```

The output of the above command is shown below. The verbose option `-v` describes the action that was taken, as shown in the highlighted text.

```
Copying file <groups.h5> and object </MyGroup> to file <mygrouponly.h5> and object </MyGroup>  
Using shallow flag
```

```
$ h5dump -H mygrouponly.h5  
HDF5 "mygrouponly.h5" {  
  GROUP "/" {  
    GROUP "MyGroup" {  
      GROUP "Group_A" {  
      }  
      GROUP "Group_B" {  
      }  
      DATASET "dset1" {  
        DATATYPE  H5T_STD_I32BE  
        DATASPACE  SIMPLE { ( 3, 3 ) / ( 3, 3 ) }  
      }  
    }  
  }  
}
```

Add or Remove User Block from File

The user block is a space in an HDF5 file that is not interpreted by the HDF5 library. It is a property list that can be added when creating a file. See the [H5P_SET_USERBLOCK](#) API in the HDF5 Reference Manual for more information regarding this property.

Once created in a file, the user block cannot be removed. However, you can use the `h5jam` and `h5unjam` utilities to add or remove a user block from a file into a new file.

These two utilities work similarly, except that `h5jam` adds a user block to a file and `h5unjam` removes the user block. You can also overwrite or delete a user block in a file.

Specify the `-h` option to see a complete list of options that can be used with `h5jam` and `h5unjam`. For example:

```
h5jam -h  
h5unjam -h
```

Below are the basic options for adding or removing a user block with `h5jam` and `h5unjam`:

Options	Description
<code>-i</code>	Input File
<code>-o</code>	Output File
<code>-u</code>	File to add or remove from user block

Let's say you wanted to add the program that creates an HDF5 file to its user block. As an example, you can take the `h5_crtgrpar.c` program from the [HDF5 Introductory Tutorial Examples](#) and add it to the file it creates, `groups.h5`. This can be done with `h5jam`, as follows:

```
h5jam -i groups.h5 -u h5_crtgrpar.c -o groupsub.h5
```

You can actually view the file with `more groupsub.h5` to see that the `h5_crtgrpar.c` file is indeed included.

To remove the user block that was just added, type:

```
h5unjam -i groupsub.h5 -u h5_crtgrparNEW.c -o groups-noub.h5
```

This writes the user block in the file `groupsub.h5` into `h5_crtgrparNEW.c`. The new HDF5 file, `groups-noub.h5`, will not contain a user block.