

# h5repack

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

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

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

# h5repack

Copies an HDF5 file to a new file with or without compression and/or chunking

## Syntax:

```
h5repack [OPTIONS] in_file out_file
```

```
h5repack -i in_file -o out_file [OPTIONS]
```

## Description:

`h5repack` is a command line tool that applies HDF5 filters to an input file *in\_file*, saving the output in a new output file, *out\_file*.

### Tip if encountering poor performance

If encountering poor performance using `h5repack` with large datasets, please note that the `H5TOOLS_BUFSIZE` environment variable can be used to improve performance. This environment variable specifies the hyperslab (selection) buffer size (in bytes) that is used by `h5repack`. Its default value is 32 MB ( $32 \times 1024 \times 1024 = 33554432$  bytes), which may be very small for large datasets. *The dataset does not have to be chunked to use this environment variable.*

For example, if encountering a performance issue when using `h5repack` with a large 3D dataset with a chunk size of  $512 \times 512 \times 512$  and a datatype of 32-bit float (4 bytes in size), then setting `H5TOOLS_BUFSIZE` to the size of (at least) one chunk times the size of the datatype ( $512 \times 512 \times 512 \times 4 = 536870912$ ) should improve performance. On Unix the `H5TOOLS_BUFSIZE` environment variable can be set as follows:

```
setenv H5TOOLS_BUFSIZE 536870912
```

Please be aware that if `H5TOOLS_BUFSIZE` is *too large* it can also affect performance. If you are encountering a performance issue and `H5TOOLS_BUFSIZE` already has a large value, then try specifying a smaller value. If the dataset is chunked, try setting `H5TOOLS_BUFSIZE` to a value closer to the size of one chunk times the size of the datatype.

## Options and Parameters:

```
h5repack [OPTIONS] file1 file2
    file1                Input HDF5 File
    file2                Output HDF5 File
OPTIONS
-h, --help              Print a usage message and exit
-v, --verbose           Verbose mode, print object information
-V, --version           Print version number and exit
-n, --native            Use a native HDF5 type when repacking
--enable-error-stack    Prints messages from the HDF5 error stack as they occur
-L, --latest            Use latest version of file format
                        This option will take precedence over the -j and -k options
--low=BOUND             The low bound for library release versions to use when creating
                        objects in the file (default is H5F_LIBVER_EARLIEST)
--high=BOUND            The high bound for library release versions to use when creating
                        objects in the file (default is H5F_LIBVER_LATEST)
-c L1, --compact=L1    Maximum number of links in header messages
-d L2, --indexed=L2    Minimum number of links in the indexed format
-s S[:F], --ssize=S[:F] Shared object header message minimum size
-m M, --minimum=M      Do not apply the filter to datasets smaller than M
-e E, --file=E          Name of file E with the -f and -l options
-u U, --ublock=U        Name of file U with user block data to be added
-b B, --block=B         Size of user block to be added
-M A, --metadata_block_size=A Metadata block size for H5Pset_meta_block_size
-t T, --threshold=T     Threshold value for H5Pset_alignment
-a A, --alignment=A     Alignment value for H5Pset_alignment
-q Q, --sort_by=Q       Sort groups and attributes by index Q
-z Z, --sort_order=Z    Sort groups and attributes by order Z
-f FILT, --filter=FILT Filter type
-l LAYT, --layout=LAYT Layout type
-S FS_STRATEGY, --fs_strategy=FS_STRATEGY File space management strategy for
H5Pset_file_space_strategy
-P FS_PERSIST, --fs_persist=FS_PERSIST Persisting or not persisting free-space for
H5Pset_file_space_strategy
-T FS_THRESHOLD, --fs_threshold=FS_THRESHOLD Free-space section threshold for
H5Pset_file_space_strategy
-G FS_PAGESIZE, --fs_pagesize=FS_PAGESIZE File space page size for
H5Pset_file_space_page_size

M - is an integer greater than 1, size of dataset in bytes (default is 0)
E - is a filename.
S - is an integer
U - is a filename.
T - is an integer
A - is an integer greater than zero
Q - is the sort index type for the input file. It can be "name" or "creation_order"
(default)
Z - is the sort order type for the input file. It can be "descending" or "ascending"
(default)
B - is the user block size, any value that is 512 or greater and is
a power of 2 (1024 default)
F - is the shared object header message type, any of <dspace|dtype|fill|
pline|attr>. If F is not specified, S applies to all messages

BOUND is an integer indicating the library release versions to use when creating
objects in the file (see H5Pset_libver_bounds()):
0: This is H5F_LIBVER_EARLIEST in H5F_libver_t struct
1: This is H5F_LIBVER_V18 in H5F_libver_t struct
2: This is H5F_LIBVER_V110 in H5F_libver_t struct
(H5F_LIBVER_LATEST is aliased to H5F_LIBVER_V110 for this release

FS_STRATEGY is a string indicating the file space strategy used:
FSM_AGGR:
    The mechanisms used in managing file space are free-space managers, aggregators
and virtual file driver.
PAGE:
    The mechanisms used in managing file space are free-space managers with embedded
paged aggregation and virtual file driver.
AGGR:
    The mechanisms used in managing file space are aggregators and virtual file
```

driver.

NONE:

The mechanisms used in managing file space are virtual file driver.  
The default strategy when not set is FSM\_AGGR without persisting free-space.

FS\_PERSIST is 1 to persisting free-space or 0 to not persisting free-space.  
The default when not set is not persisting free-space.  
The value is ignored for AGGR and NONE strategies.

library.

FS\_THRESHOLD is the minimum size (in bytes) of free-space sections to be tracked by the

The default when not set is 1.  
The value is ignored for AGGR and NONE strategies.

strategy PAGE is used.

FS\_PAGESIZE is the size (in bytes)  $\geq 512$  that is used by the library when the file space  
The default when not set is 4096.

FILT - is a string with the format:

<list of objects>:<name of filter>=<filter parameters>

<list of objects> is a comma separated list of object names, meaning apply  
compression only to those objects. If no names are specified, the filter  
is applied to all objects

<name of filter> can be:

GZIP, to apply the HDF5 GZIP filter (GZIP compression)  
SZIP, to apply the HDF5 SZIP filter (SZIP compression)  
SHUF, to apply the HDF5 shuffle filter  
FLET, to apply the HDF5 checksum filter  
NBIT, to apply the HDF5 NBIT filter (NBIT compression)  
SOFF, to apply the HDF5 Scale/Offset filter  
UD, to apply a user defined filter  
NONE, to remove all filters

<filter parameters> is optional filter parameter information

GZIP=<deflation level> from 1-9  
SZIP=<pixels per block,coding> pixels per block is a even number in  
2-32 and coding method is either EC or NN  
SHUF (no parameter)  
FLET (no parameter)  
NBIT (no parameter)  
SOFF=<scale\_factor,scale\_type> scale\_factor is an integer and scale\_type  
is either IN or DS  
UD=<filter\_number,filter\_flag,cd\_value\_count,value\_1[,value\_2,...,value\_N]>  
required values for filter\_number,filter\_flag,cd\_value\_count,value\_1  
optional values for value\_2 to value\_N  
NONE (no parameter)

LAYT - is a string with the format:

<list of objects>:<layout type>=<layout parameters>

<list of objects> is a comma separated list of object names, meaning that  
layout information is supplied for those objects. If no names are  
specified, the layout type is applied to all objects

<layout type> can be:

CHUNK, to apply chunking layout  
COMPA, to apply compact layout  
CONTI, to apply contiguous layout

<layout parameters> is optional layout information

CHUNK=DIM[xDIM...xDIM], the chunk size of each dimension

COMPA (no parameter)  
CONTI (no parameter)

### Exit Status:

0	Succeeded.
> 0	An error occurred.

### Examples:

1. `h5repack -f GZIP=1 -v file1 file2`  
Applies GZIP compression with level 1 to all objects in `file1` and saves the output in `file2`. Prints verbose output.
2. `h5repack -f dset1:SZIP=8,NN file1 file2`  
Applies SZIP compression with 8 pixels per block and NN coding method only to object `dset1`.
3. `h5repack -l dset1,dset2:CHUNK=20x10 file1 file2`  
Applies chunked layout with size 20x10 to objects `dset1` and `dset2`.
4. `h5repack -L -c 10 -s 20:dtype file1 file2`  
Applies the latest file format with a maximum compact group size of 10 and minimum shared datatype size of 20.
5. `h5repack --low=0 --high=1 file1 file2`  
Sets `low=H5F_LIBVER_EARLIEST` and `high=H5F_LIBVER_V18` via `H5Pset_libver_bounds()` when creating the h5repacked file: `file2`
6. `h5repack -f SHUF -f GZIP=1 file1 file2`  
Applies both filters SHUF and GZIP in this order to all datasets
7. `h5repack -f UD=307,0,1,9 file1 file2`  
Applies `bzip2` filter to all datasets.

### History:

Release	Change
1.10.1	Options added or modified in this release for file space management and page buffering: -G, --fs_page_size -P, --fs_persist -S, --fs_strategy (modified)
1.10.0	Options added in this release for file space management: -S, --fs_strategy -T, --fs_threshold
1.8.12	Added user-defined filter parameter (UD) to <code>-f filter</code> , <code>--filter =filter</code> option for use in read and write operations.
1.8.9	<code>-M number</code> , <code>--medata_block_size=number</code> option introduced in this release.
1.8.1	Original syntax restored; both the new and the original syntax are now supported.
1.8.0	<code>h5repack</code> command line syntax changed in this release.
1.6.2	<code>h5repack</code> introduced in this release.

--- Last Modified: September 30, 2019 | 04:00 PM