

HDF5 Layout Chunk Size

Layout refers to how data for a given dataset is arranged on disk. The HDF5 library supports three types of layout: contiguous, compact, chunked. The HDF5SaveData and HDF5SaveGroup operations default to contiguous layout which is appropriate for uncompressed datasets.

The HDF5 library requires that compressed datasets use the chunked layout. This means that the data for the dataset is written in some number of discrete chunks rather than in one contiguous block. In the context of compression, chunking is mainly useful for speeding up accessing subsets of a dataset. Without chunking, the entire dataset has to be read and decompressed to read any subset. If the dataset is always read in its entirety, such as when Igor loads an HDF5 packed experiment file, chunking does not enhance speed.

Chunked storage requires telling the HDF5 library the size of a chunk for each dimension of the dataset. There is no general way to choose appropriate chunk sizes because it depends on the dimensions and nature of the data as well as tradeoffs that require the judgement of the user. In Igor chunk sizes can be specified

- Using the HDF5SaveData operation /LAYO flag (see [HDF5SaveData](#) for details)
- Using HDF5SaveDataHook (see [Using HDF5SaveDataHook](#) on page II-215 for details)

The HDF5SaveGroup, SaveExperiment, SaveData, SaveGraphCopy, SaveTableCopy, and SaveGizmoCopy operations and HDF5 default compression always save a compressed dataset in one chunk, as described in the following sections.

Compression parameters for a given dataset are set when the dataset is created and can not be changed when appending to an existing dataset.

HDF5SaveGroup Compression

In Igor Pro 9.00 and later, you can tell the HDF5SaveGroup operation to compress numeric datasets using the /COMP flag.

HDF5SaveGroup applies compression only to numeric waves, not to text waves or other non-numeric waves nor to numeric waves with fewer than the number of elements specified by the /COMP flag.

HDF5SaveGroup compression uses chunk sizes equal to the size of each dimension of the wave. Such chunk sizes mean that the entire wave is written using chunked layout as one chunk. This is fine for datasets that are to be read all at once. For finer control you can use the HDF5SaveDataHook function to override compression specified by /COMP but this is usually not necessary.

SaveExperiment Compression

In Igor Pro 9.00 and later, you can tell the SaveExperiment operation to compress numeric datasets using the /COMP flag. This works the same as HDF5SaveGroup compression discussed in the preceding section.

The same is true for the SaveData, SaveGraphCopy, SaveTableCopy, and SaveGizmoCopy operations.

See [HDF5 Compression for Saving Experiment Files](#) on page II-16 for step-by-step instructions.

HDF5 Default Compression

Igor can perform default dataset compression when you save HDF5 files via the user interface. Default compression does not apply when saving HDF5 files via the HDF5SaveData, HDF5SaveGroup, SaveExperiment, SaveData, SaveGraphCopy, SaveTableCopy, or SaveGizmoCopy operations. HDF5 default compression was added in Igor Pro 9.00.

Default compression is disabled by default because compression can significantly increase the time required to save. Whether compression is worthwhile depends on the size and nature of your data and on how you trade off time versus disk space.

You can enable default compression by choosing Misc→Miscellaneous Settings and clicking the Experiment icon. This allows you to set the following settings: