

HDF5 in Igor Pro

HDF5 User Group Meeting, 2021-10-13

Howard Rodstein, WaveMetrics, Inc.

About Igor Pro

Igor Pro is a commercial scientific graphing and data analysis program with a built-in programming environment.

It runs on Macintosh and Windows. It is similar to Matlab, Origin.

Igor is used in:

- A wide range of scientific fields in academia
- Government research labs all over the world
- High-tech companies

Some large scientific projects are built on top of Igor. For example:

- Multichannel Igor Electrophysiology Suite (Allen Brain Institute)
- Irena package for analysis of small-angle scattering data (Argonne Nat'l Lab)
- SutterPatch (Sutter Instruments)

Igor has supported import and export of HDF5 files since 2005.

The current version, Igor Pro 9, adds the ability to save entire Igor Pro workspaces to HDF5 files and to restore them from HDF5 files.

Igor's HDF5 Support

This presentation will demonstrate the ways through which Igor Pro provides access to HDF5 files:

- Via Igor's HDF5 Browser
- Via Igor's programming environment
- Via Igor HDF5-based workspace files

HDF5 Versus Igor Terminology

HDF5 Term

Igor Term

Dataset

Wave (short for waveform)

Group

Data folder

Experiment (workspace)

File

Experiment File (workspace file)

Igor "Experiment" Files

"Experiment" means "project" or "workspace".

An experiment file stores an entire Igor experiment (workspace) including:

- Data folder hierarchy
- Waves
- Graphs, tables and other types of windows
- Procedures (code written in Igor's programming language)

Main Types of Experiment Files:

- Packed Experiment File (.pxp)
- HDF5 Packed Experiment File (.h5xp)

Igor's HDF5 Browser is something like HDFView

Lists:

- Groups
- Group Attributes
- Datasets
- Dataset Attributes

Preview:

- Graph
- Table
- Dump

Load:

- Dataset or attribute
- Group with or without recursion

Save:

- Dataset
- Group with or without recursion

Igor's HDF5 Browser is implemented using Igor code

- Igor Pro Control Panel
- Independent Module
- Support Data is stored in experiment file

Igor's HDF5 Programming Support

Operations and Functions

- As seen via Help Browser
- As seen via the "Igor HDF5 Guide" help file

The interface is Igor-like, not HDF5-like.

HDF5 Coding Example

Guided Tour "Loading HDF5 Data Programmatically" Section

- Create HDF5Samples Symbolic Path
- Paste function into procedure window
- Execute function via command line
- Generalize function

HDF5 Packed Experiment File Format

Three experiment file formats:

- Unpacked (.uxp, 1987)
- Packed (.pxp, 1989)
- HDF5 Packed (.h5xp, 2021)

Open "Demo Experiment #1.pxp"

Save as "Demo Experiment #1.h5xp"
(Convert to UTF-8 first)

Re-open "Demo Experiment #1.h5xp"

Inspect "Demo Experiment #1.h5xp"

User Comment on Igor HDF5 Packed Experiment File Format

FYI, my favorite new feature is saving experiments as HDF5.

By combining this with Save Graph Copy, it will be very easy to create self-contained HDF5 files to go with any figure one publishes and hence make it easy to publish the data along with the figure which is an increasing trend/requirement for science publications.

Size comparison: .pxp versus .h5xp

- Using dense attribute storage cause .h5xp experiment file sizes to increase by about 4X
- Using compact attribute storage returned .h5xp experiment file sizes to reasonable values
- To store large attributes, you must enable use of HDF5 1.8 object formats

Support for large attributes (>64K) was added in HDF5 1.8.0 but I did not understand that.

I misunderstood the documentation on large attributes ("Large Attributes Stored in Dense Attribute Storage" in the user guide) to say that I had call `H5Pset_attr_phase_change` to enable "dense" attribute storage, as opposed to the default "compact" attribute storage, in order to support large attributes.

With dense attributes turned on, I was surprised to find that the .h5xp experiment file was about 4X larger than the corresponding .pxp experiment file with compression off.

I then determined that the .h5xp experiment file is about the same size as the .pxp experiment file when using compact storage.

With the help of The HDF5 Group, I determined that dense storage is not required to support large attributes with HDF5 1.8.0 or later. Since I am using 1.10.7, I turned dense attributes off and the experiment file size is now quite reasonable.

I also had to set the HDF5 library low bounds to `H5F_LIBVER_V18`:

```
H5Pset_libver_bounds(plist_id, H5F_LIBVER_V18, H5F_LIBVER_LATEST);
```

That tells the library to use HDF5 1.8 features for storing objects.

Speed comparison: .pxp versus .h5xp

I created Igor test code which wrote experiments in .pxp format and .h5xp format to compare size and speed. It write experiments containing 1 and 10 data folders (groups) and 1, 100, 1000 or 10000 waves (datasets).

DFs	Total Waves	Waves	Time Ratio
1	1	1	0.49
10	1	10	0.64
1	100	100	0.86
1	1000	1000	1.65
10	100	1000	1.59
1	10000	10000	5.28
10	1000	10000	3.6
1	100000	100000	26.6
10	10000	100000	5.73

.pxp is much faster than .h5xp for very large numbers of waves, especially when they are saved in one group.