

Please add questions or comments for Elena to address regarding the HDF5 Roadmap (Please add your name for any follow up discussion):

- Would like to see an HDF5 Recipes/Examples project where there are a bunch of living, breathing, working, documented and kept up to date examples of HDF5 “mini-apps” (C, C++ and/or Fortran) demonstrating how to do various common things, scalably, in HDF5.
- When the new reference docs arrive, please do redirect from the old locations to the new (if confluence allows this). We quite frequently link to the reference information for specific functions.
- A simple to follow tutorial how to create a new VOL plugin.
- A filter “template” C code for new filter writers to develop filter code that integrates “naturally” with the rest of the eco-system and does things like error handling, ensuring works also on Windows, etc. in a compatible way.
- A tutorial on ‘ipyparallel’ and common workers output to the H5. (python)
- Any interest in reviving (taking over) H5PyVol? (<https://github.com/pierlauro/H5PyVOL>) (EP: I tried to recruit for it the author :-). A this time, no plans to do it).
- An API oriented way for apps to tell where to load dynamic plugins from instead of just the env. Variable.
- What is the difference between the multi-threaded hdf5 library and the current thread-safe option for compiling the hdf5 library (Ans: Its actually allowing multiple threads to issue for example H5Dread calls and having those execute concurrently)
- Pdfs of the documentation especially guides and tutorials, these used to be available (EP: noted)
- Dynamic delete of Dataset/Group/Attribute - without closing file. (EP: you mean freeing space and making HDF5 file smaller?) yes
- Is there a possibility of supporting concurrency at just the file-level, as an earlier roadmap milestone? So one thread could write file A while another thread writes file B? This was one of our biggest surprises when using HDF - that the thread safety prevented us from writing (or reading) two separate files at the same time.