Data Formats Software Discussion

Tom Junk

DUNE Software Architecture Meeting

October 29, 2021

Repositories in GitHub

- Original proposal: https://github.com/DUNE-DAQ/dataformats
  - Contains headers, documentation, tests
  - had depended on daq build environment, ers, BOOST
- Now:
  - https://github.com/DUNE-DAQ/daqdataformats
  - https://github.com/DUNE-DAQ/detdataformats
  - https://github.com/DUNE-DAQ/detchannelmaps
  - Still has headers and documentation. Depends on standard C++ as far as I can tell.
- Replaces functionality in dune_raw_data, which must still be maintained in order to read ProtoDUNE-SP-1 data
- artdaq::Fragment is "definitely gone" (though we must retain it offline to support ProtoDUNE-SP-1 data

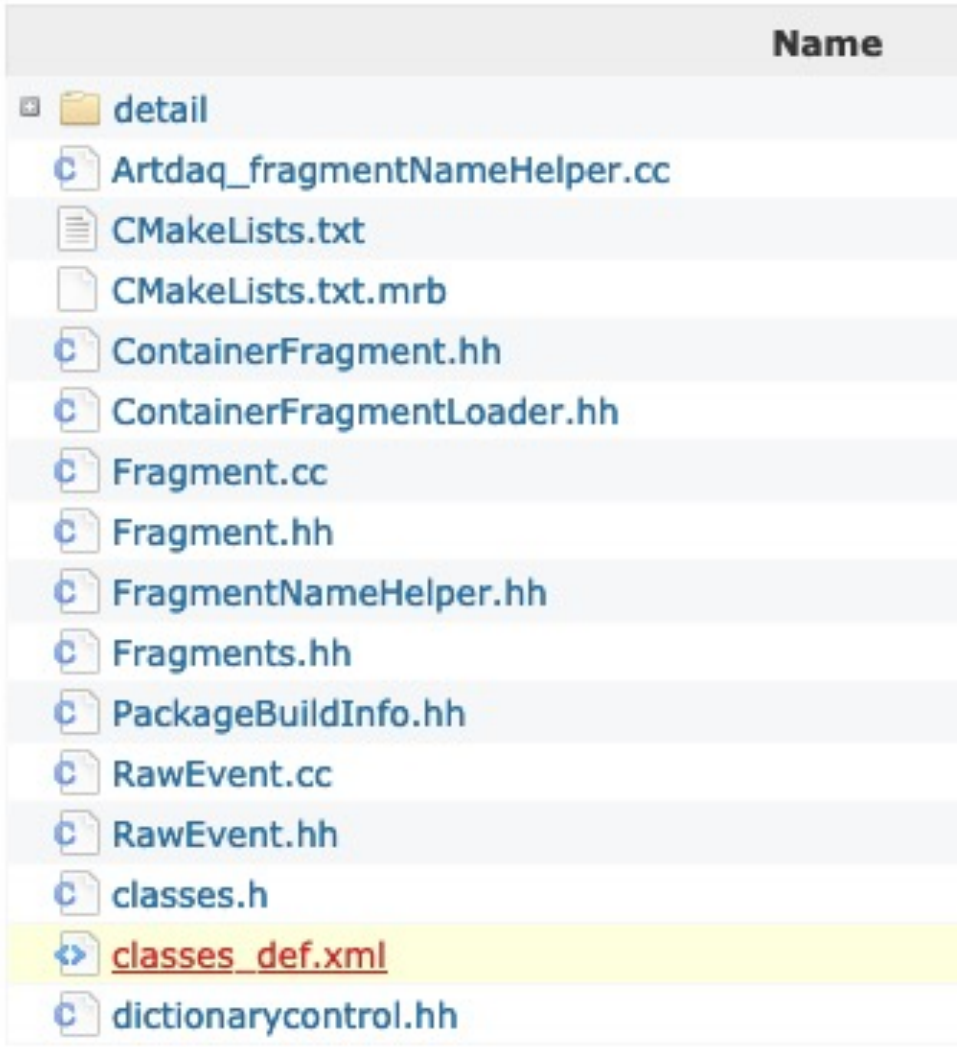
UPS Products

- Names have "dune" prepended to them
  - avoid potential name clashes on the scisoft server where we share a namespace with all experiments at Fermilab.
  - Internal directories are the same.
- dunedaqdataformats
  - has v3_0_0 and v3_1_0 versions available
  - headers and documentation (no tests)
  - No build necessary – unflavored, no qualifiers
- dunedetdataformats
  - v3_0_0, v3_0_1 and v3_1_0 available so far.
  - headers and documentation
  - No build necessary – unflavored, no qualifiers

Schema Evolution

- ROOT has automated schema evolution features
  - classes.h
  - classes_def.xml
- HDF5 does not store C++ class data as such – it leaves it to users to serialize and deserialize the data
- This is convenient, as our data are large, and we don't want to multiply buffer them just to convert the schema.
- But we need to store version numbers in fixed locations so we can tell unambiguously how to interpret the data.

core / artdaq-core / Data @ develop



This one doesn't have much of an i/o rule – just a switch to QuickVec at one point

The artdaq::RawFragmentHeader has evolved, however

https://cdcvs.fnal.gov/redmine/issues/23319

https://cdcvs.fnal.gov/redmine/issues/23345

Channel maps

https://github.com/DUNE-DAQ/channelmaps

has header files but no code.

Channel maps are different from the other repositories:

- Need to read in a channel map file(s)
  - channel map files need to be installed somewhere visible interactively, online and offline, and on the grid
  - Or hard-code the data in the source
- They also depend on the offline channel numbering
  - would need to include dunetpc as a dependency of the UPS product in order to guarantee reproducibility of results.
  - Offline channel numbering has already evolved for the VD Coldbox

Channel Maps

- Solution found so far: put channel maps in dunetpc/dune/VDColdbox/ChannelMaps
- dune_raw_data is now just there for archival purposes – reading in ProtoDUNE-SP-1 data
- We'll have multiple channel maps, one for each detector module and prototype
- So far, the decoder tools know which detector they're decoding, so they can call a channel map service of known name.
- But dataprep also uses the channel map for making some plots by FEMB.
- May need to make the channel map a tool – swappable at runtime.

Code Librarians

- Mike Kirby (DUNE S&C consortium co-lead) asks if code librarians have been established for the data formats repositories
- The repositories belong to the DAQ group.
- DAQ people have been responsive to bug reports.
- Offline (i.e., Tom) has been making UPS products and releases them in CVMFS and uploads tarballs to SciSoft. Not complicated, but also not automated.
- Spack is coming.