

Software Coordination Highlights

John Freeman (Fermi National Accelerator Laboratory)
Software Coordination Meeting
Jun-13-2024

Overview

- Welcome to the first ever general Software Coordination meeting!
- Basic structure:
 - We'll present some of the highlights of our activities in a relatively high-level sense (~ 1/2 hr)
 - Then we'll open the floor to feedback (~ 1/2 hr)
 - Request for a deeper dive on some of our topics
 - Ideas for how we could improve things
- Idea is to have a meeting like this about once a month

Highlights, 30,000 foot view

- Experimenting with smaller, more targeted umbrella packages than “fddaq” and “nddaq”
- Nightly integration tests for v5 line
- Switching to C++20 soon seems reasonable
- Considering dropping SL7/Centos Stream 8 support
- Doxygenated our code (Andrew will talk about this)

Other umbrella packages

- Alessandro asked me to create an “fddatautilities” package, and put it in a lightweight image
 - Directly depends only on rawdatautils and hdf5libs
 - Contained in `ghcr.io/dune-daq/nightly-release-
alma9:fddatautilities_runonly`
 - Can use this to analyze *.hdf5 files on, e.g., your laptop
- Lightweight is relative
 - 4-5 GB, rather than the 9-10 GB of the `production_v4` and `development_v5` tags of the image
- No immediate plan to make fddatautilities a “first class partner” of “fddaq” and “nddaq”

v5 Integration Tests

- Right now we only run automated nightly integration tests for the v4 line
- At this point in the evolution of the v5 line I think it's reasonable to add nightly integration tests there as well
- Eric and I did some merges ~2 weeks ago which got the listrev nightly integration test working
 - Since that time it's stopped working
 - Glass half-full: the integration test is doing its job
 - Though just *what* caused it to break it beyond the scope of this talk
- We can consult with v5 experts about what additional integration tests to add

C++20

- Has been out for a while now
- Contains potentially useful features
- Is already supported by our gcc version, 12.1.0
 - `https://gcc.gnu.org/projects/cxx-status.html`
- Making the change to the `daq_setup_environment` function in `daq-cmake` to switch from C++17 to C++20 would be trivial
- Would at least want to give our Style Guide a review if we bumped, however
 - Any C++20 specific antipatterns?
 - Any pre-C++20 constructs rendered obsolete despite still compiling?

Less is more?

- SL7 end-of-life is June 30
- CentOS Stream 8 end-of-life was (I believe) on May 31
- Currently we build the full DUNE DAQ stack every night not just for Alma9, but also for SL7
- Can we drop SL7?
 - Cuts the aggregate nightly workload in two
- Even if we keep `ghcr.io/dune-daq/sl7-slim-externals:v1.1` - based on `ghcr.io/dune-daq/sl7-spack:latest` - it still seems we can drop the unused `ghcr.io/dune-daq/c8-spack:latest`?
 - Further audit for some of our images would be useful
 - Also: the “slim” doesn’t add any useful distinguishing info

Doxygen

- Turning the floor over to Andrew...