

git/mrb cutover plans

Eric, Yale, 16-Jan-2014

Schedule

- Rick's team would like sign-off in next 2 days, so as to go live with production version of new git/mrb-based LArSoft.
- What is planned is svn repository will be frozen to svn ci's. It is to be svn2git'd on **Jan 21**. The new git repository will be opened for business by **Jan 24**, it is hoped.

At that point, absent disaster, we will be in git/mrb world only!

slightly more detail

- Rather than getting your Makefiles all handed to you as in SRT, you get CMakeList.txt files on which cmake is run to produce the Makefiles. You should not have to tinker with these, as you did not either in SRT.
- Everything is ups-ified. Meaning, libraries live in precisely versioned areas, and they must be set up. Even those you make yourself in your own area. We used to have one foot in this world and the other in a world of a hodgepodge of directories.

Work till now

- Herb, Wes, I have been the guinea pigs exercising the new environment
- bugs reported, most fixed
- Validation needed
 - I've made a start on that, but only a start
- Documentation at quick-start link is good.
Under construction: https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_Quick-start_guide_to_using_and_developing_LArSoft_code_

email exchange

- There are a few serious concerns we have, even if we are sanguine about the results of the validation and feel we can grant approval for cutover.

(1) We need to know that there's a nightly build ready to pick up on that next day; we can not have all ~20 active uB Tools people updating and building the whole code base in their own directory as a matter of routine.

The plan has always called for a nightly build to be available. We have some work to do on this yet, so I will not promise at the moment that a nightly build will be available from day one. Although I think it is important to have a nightly build, I am basically willing to push ahead with the transition provided that we believe we can deliver a nightly build shortly after the transition. I would guess that in the mean time, most people can get by without updating and building everything on a daily basis, though I could be convinced that this is wrong.

(2) The build and run environments are not seamless; we can get by for now, but would like some assurance that consideration is going toward making that smoother. Many times this past week I found myself re-building whole large packages because of some small change I made somewhere.

Suffice it to say, we (the project) are committed to making the user experience as painless as possible,...

(3) The explicit way in which we are to work on branches has not been demonstrated. I'm sure that will work, but we need explicit documentation. We once said we could all get by with O(8) git commands at our disposal. It might be nice to lay those out.

Adding information on the branching model is already on my list of documentation changes to make. I'll try to get that out for review soon.

(4) We need to know how to build uboonecode in our area in the prescribed manner

I think this is also already on my list of places where documentation needs to be improved.

15-Jan LArSoft mtg

- LArSoft principals have thrown the resources at solving all the previous complaints before next Thursday.

Recourse

- Do we want to bail out on a cutover in January?
- We would need to get SCD to not freeze svn.
- We will have to commit to helping w yet another svn2git later.
- It is not what SCD wants.
- The longer we put this off, the closer we get to data. I'm not entirely convinced it's a good idea.

Validation

Excuses

- We are trying to validate S2013.11.25 with ART1.06 in SRT vs a git snapshot on that date, which was then updated to work with ART 1.08.10.
- many mrb issues, bug fixing cycles, which we think are largely wrapped up
- The modern era fcl files didn't all work back then, so we've had to slowly whittle them back and rediscover what ran:
 - opticalDigitization didn't work, e.g.
 - trackkalsps didn't produce Assns, so calo didn't work with trackkalsps, e.g.
- **GENIE job dies with a runtime error**, as of last night
- (Wes and) I have not forced random seeds ==, and run the SRT jobs.

What *does* work now

- I can produce single particles and recon them with a reasonable suite of modules

Our uB-specific stuff is in `uboonecode`!

- I pulled AnalysisTree out of unbooneoffline, made it build, and git pushed it to

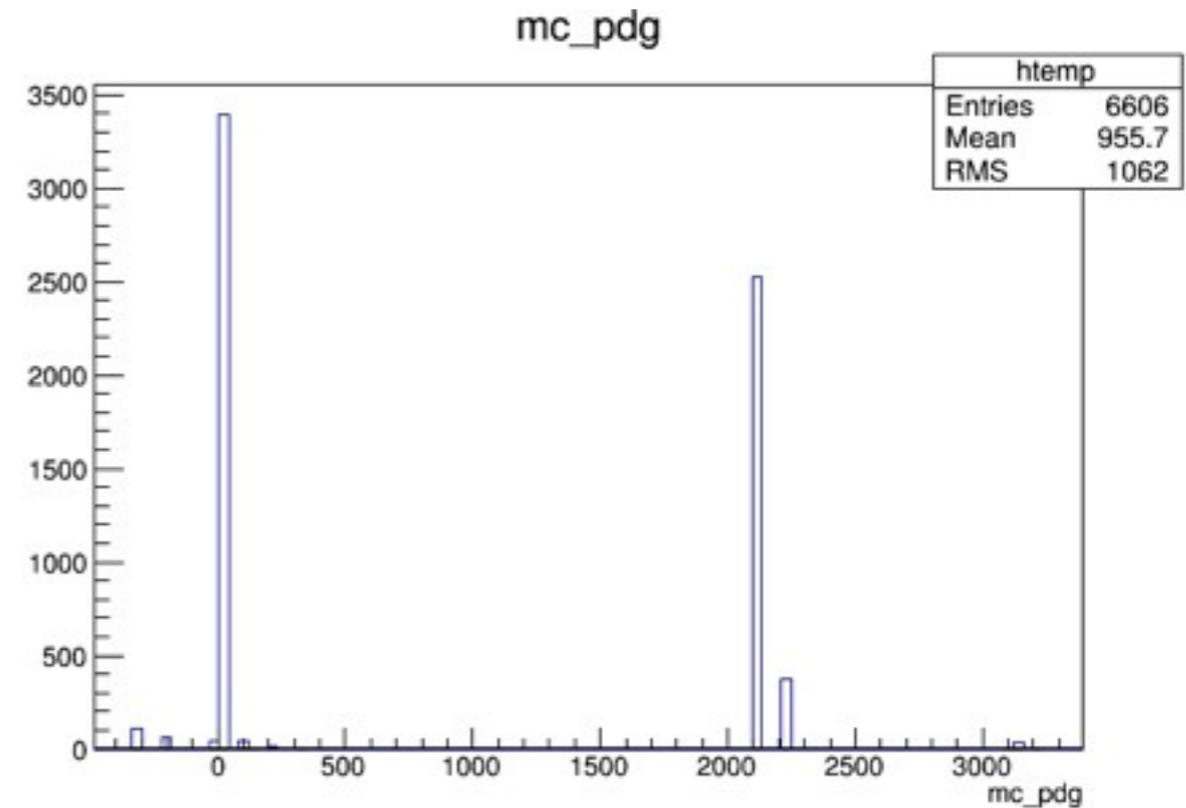
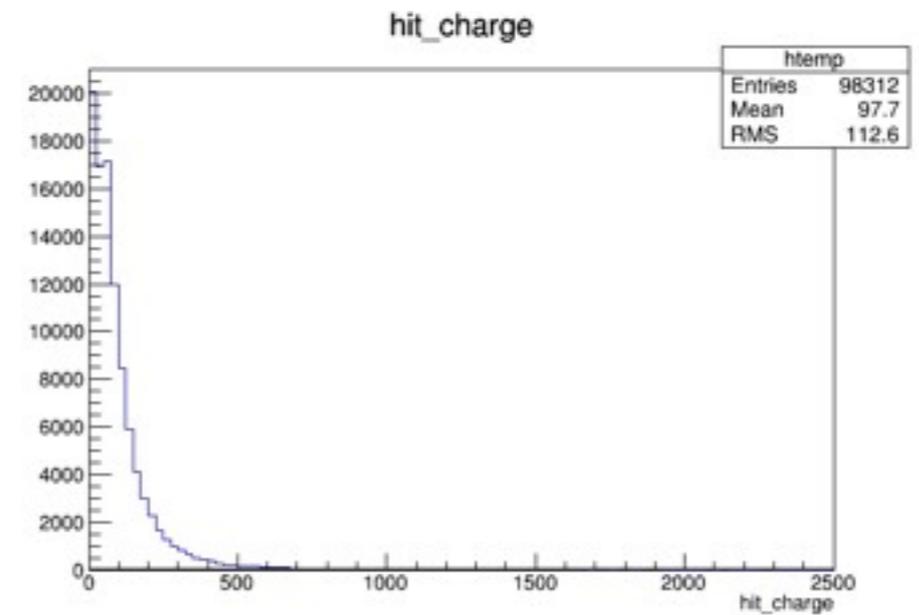
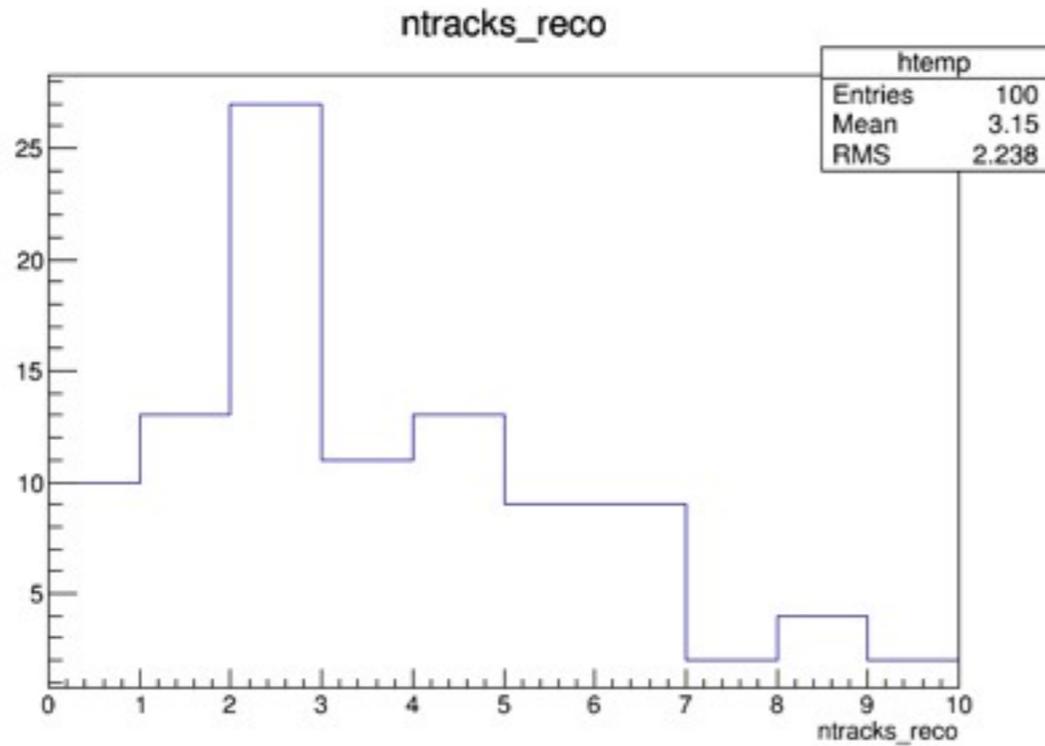
`uboonecode`. I used it in `/uboone/app/users/`

`echurch/lgm/build/reco_1w_FULL.fcl`.

- reco: [caldata, gaushit, fuzzycluster, cccluster, trackkalmanhit, spacepointfinder, trackkalsps, beziertracker, calo]

100 isotropic 1 GeV/c

K-'s



no cluster info yet in
AnalysisTree, I think.

Herb's lessons

- photon look-up lib, pandora fcl files not picked up => fixed
- ubtools (project.py, condor_lar.sh) now fully mrb-compliant!
- knows if you're in SRT or mrb world by xml config file

Wes's lessons

- like me, mrb behaviour
 - Particularly, incremental builds work less reliably than full simultaneous builds.
- Come to workshop tomorrow am

environment(s)

- We recommend a **build** setup and separately a **run** setup

below presumes you've already made your work area

- `alias ulgmb='source /grid/fermiapp/uboone/software/setup_uboone.sh; cd /uboone/app/users/echurch/lgm/build; setup larsoft v0_02_01 -qe4:prof; source /uboone/app/users/echurch/lgm/localProducts_larsoft_v0_02_01_e4_prof/setup; source mrb setEnv;'`
- `alias ulgmr='source /grid/fermiapp/uboone/software/setup_uboone.sh; cd /uboone/app/users/echurch/lgm/build; source /uboone/app/users/echurch/lgm/localProducts_larsoft_v0_02_01_e4_prof/setup; source mrb slp;'`