



2020 LArSoft Work Plan

Last updated: December 13, 2019

Introduction

This document records the LArSoft work plan for 2020. Progress on specific items going forward are given quarterly at Steering Group meetings.

Erica and Katherine discussed priorities with each of the Offline leads in September of 2019. The experiments detailed their plans for the next year, the implied requirements for LArSoft, and how LArSoft could help, as well as what the experiments might be able to contribute to LArSoft code. Some items may be raised by LArSoft, not a particular experiment, to address general problems faced by all LArSoft users.

Based on those discussions, LArSoft proposes a plan of work for 2020 along with the relative priority of the various items. These were reviewed at the October Offline Leads meeting. A revised work plan was discussed and approved by the experiments at the December LArSoft Steering Group meeting.

The resources responsible for executing parts of the plan is listed in cases where this is known. It should be noted, however, that the LArSoft project team does not have sufficient staff at present to carry out the entire program. Some projects can and should be performed by the experiments. Allowing experiment service credit for contributions to items that benefit an experiment, but that are of a more general nature would help to fill some of this gap in effort. The more this happens, the more all experiments benefit.

Short term priorities for 2020

We define short term priorities as those items that should receive sufficient effort to be completed in CY2020 due to the urgency or timeliness of the need. The meaning of priorities listed is discussed in Appendix A.

1. Re-architecture of LArSoft code to ensure thread safety and to implement multi-threading as needed for critical production workflows. (ongoing from 2018)

Priority: high

Re-architect code in experiment production workflows to ensure thread safety, and where appropriate and cost effective, to implement multi-threading in order to control memory scaling and improve resource utilization on production platforms. Collaborate with efforts to run multi-threaded workflows on HPC resources.

<https://cdcvns.fnal.gov/redmine/issues/19288>

Resources: Saba Sehrish, Mike Wang

2. Migrate to GitHub and pull requests. (Started in June 2019)

Priority: high

Investigate moving the LArSoft central repositories from Redmine to GitHub, along with those collaboration services offered by GitHub that make the migration cost effective. There are three phases.

- Phase I: Migrate central repositories to GitHub
- Phase II: Develop necessary infrastructure to support pull requests
- Phase III: Transition the current model of merging to develop by developers and the Code Management team to a pull-request system where authorized experiment personnel handle requests after mandatory CI testing prior to merging.

This item was mostly completed in 2019. The outstanding work includes getting CI jobs to run that are triggered by GitHub; and completing the documentation needed for testing and general use.

Resources: Patrick Gartung

3. Provide a support mechanism for running LArSoft in containers

Priority: high

Develop the procedures, processes and documentation needed to allow users to develop and run LArSoft using a well-supported containerized solution. Provide on-going support for producing the required images.

Resources: SciSoft

4. **SPACK - new build system for art and LArSoft** (deferred from 2017, continued from 2018 and 2019).

Priority: high

Purpose:

- To migrate LArSoft to the Spack packaging tool, and the SpackDev development system, a standard set of packaging and build tools that have broad community support
- Via this migration, to address portability and configurability issues raised by experiments with the current build system

<https://cdcv.sfnal.gov/redmine/projects/spack-planning/issues>

Resources: Chris Green

5. **Pixel-detectors within LArSoft.** (Considered in 2018, carried forward from 2019)

Priority: high

Work with experiments to identify and implement common solutions that include LArTPCs with pixel readout. The integration should include the following:

- Adaptation or development of low-level data products
- Shared data products for 3D hits
- Geometry adaptations
- Development of pixel-based anode simulations.

Expect experiments to carry out a significant fraction of the reconstruction and simulation work.

<https://cdcv.sfnal.gov/redmine/issues/19328>

Resources: Erica Snider & ??

6. **Existing Event Display**

Priority: medium

Implement improvements to the existing LArSoft Event Display to make it faster and easier to navigate through a large number of APSs, and to identify TPCs or detector regions with data of interest. The implementation should be applicable to ProtoDUNE, DUNE FD and ICARUS.

Resources: not yet identified

7. **New Event Display development plan.**

Priority: medium

Phase 1: Create a plan for developing a new LArSoft event display in light of the results of the 2019 ED technology review (see <https://cdcv.sfnal.gov/redmine/issues/19038>), and the [ED requirements document](#) with identified resources to execute the plan. The new ED may leverage existing code and effort from outside institutions.

Phase 2: Design and implement an event display framework following the requirements and investigation of tools described above, consulting with the experiments at regular intervals to ensure that evolving needs are met.

Resources:

8. NuTools product re-factoring

Priority: medium

Re-factor NuTools and the relevant LArSoft packages so as to separate the explicit dependence of LArSoft versions from explicit versions of Genie. The re-factoring should introduce a new product that interfaces LArSoft to Genie via common, stable data products in NuTools, and sources of geometry information in LArSoft. Upon completion, this framework will be used to integrate the NuWro and other generators into LArSoft. At lower priority, the same technique should be used to decouple LArSoft versions from explicit versions of Geant4.

Resources: Robert Hatcher

Long term and continuing priorities for 2020

We define long term priorities as those that we do not anticipate completing within CY2020 due to the nature of the work involved. Unless explicitly indicated, a project defined as a long-term priority should not be construed as implying it is “low priority”. A “continuing priority” is a set of tasks that by construction represent a continual, recurring stream of work that may or may not span more than a calendar year.

1. Places where 2020 support from LArSoft team has been requested

Priority: high as needs arise. Low otherwise

- a. Support for migrating to the refactored LArG4
 - b. Wire-cell integration
 - c. Geometry (e.g., cases where ICARUS breaks implicit assumptions: non-vertical collection wires, existence of horizontal wires, multiple channels per geometric wire)
 - d. Extend the NuWro direct integration model to include the GiBUU event generator framework and the NEUT event generator.
 - e. Investigate need for magnetic field map within LAr volume. Design and develop the interface for such a service. -- Make this a named project. Assign architect, etc.
 - f. Coordinate between GArSoft and LArSoft
 - i. Ensure sharing of data products and geometry (and with other detectors)
 - ii. Ensure sharing of any services, should that be needed (eg, for magnetic field)
 - iii. Possible introduction of non-uniform magnetic field to LAr volume
 - g. SBND data reduction strategies
 - h. Participate in the discussion of how the DUNE data streams (SN + possibly others/all) will be handled within *art* / LArSoft. May involve moving away from root-based I/O.
2. Support for running select LArSoft workflows on HPC resources.
 - a. SBND has time allocations on Theta at Argonne, and are looking to run LArSoft there
 - b. LArIAT has requested that they be able to run LArG4 on GPUs.
 3. Architecture work to support common framework for data preparation (??)
 4. Architecture work to provide full integration of DL networks
 5. **Ongoing work on bug fixes, emergency feature requests, building software, helping users.**

Priority: high, or as indicated by the request

Resources: SciSoft team, primarily Erica Snider, Saba Sehrish

6. **Architecture-dependent libraries (2018).**

Priority: low

Follow-on work to SIMD vectorization or other identified use case. Support is needed to allow coexisting builds that include / exclude, support for different computing backends under a given OS flavor.

- a. An immediate goal will be to allow the standard setup procedure to support a generic library as well as one built with a single SIMD extension set
- b. More generally, there are a number of areas that might benefit from this feature, such as allowing use of GPU backends when available for external libraries, or selecting the optimal SIMD instruction set for a given computing node. The types of backend support required needs to be studied, and appropriate tools adapted or developed to allow the required setup procedures.

Resources: ??

7. **Other topics under consideration to work on include the list of accepted, but not assigned, redmine issues.** These can be found under [accepted redmine issues](#).

Priority: low or as indicated in the respective ticket

Appendix A: Definition of priorities

The priorities listed above have the following meaning. The impacts noted are those negotiated with the experiments.

- High: the most urgent projects that have high impact to the community, and that should be fully staffed at all times until completion
- Medium: a project that has a significant impact to the community, and that should be staffed sufficiently to ensure completion within the calendar year
- Low: a project that does not have immediate impact to the community, could be rolled into the next planning period without imposing undue burdens, and should be staffed after ensuring all other obligations are met.