A Proposed Plan for Standard Builds of LArSoft

Marc Paterno
LArSoft Coordination Meeting
10 Dec 2024

Why we need a new plan

- “In order to make it easier for non-Fermilab personnel to contribute to our shared scientific software stack, and because of the diminishing availability of personnel to maintain support of the specialized infrastructure needed, we have decided to move to using Spack as the primary tool for building and distributing the software stack.”
  - Spack is supported by a much larger community than UPS; we gain access to the work of that community. (It is one of the founding projects in the High Performance Software Foundation).
  - Many 3rd party recipes are already available.
  - It will be easier for experiments to build needed software on their own, when needed.
- We need new procedures that take advantage of the new tools are are using to build and distribute software.
- We need the new procedures to adapt to the reduced number of people available to support the procedures.

Definition of a Standard Build

- For a *package*: (a coherent body of software generally built from code in a single repository), a standard build contains code from a specific *version*, built with a specific *set of customizations and options* (a *variant*), built with a specific compiler in either debug or optimized (*profile*) mode. We will strongly limit the number of builds for any version of a package.
- For a *suite*: a standard build contains a compatible set of standard builds of the packages that belong to the suite.
- The SciSoft team will create a Spack environment for each standard build of a suite, and make the environments available through CVMFS, or through Spack *binary build caches*, or both.

Standard Builds and Releases (part 1)

1. A new LArSoft release will be created when one or more PRs are merged, or when underlying dependencies have been updated.
2. A few standard builds will be created for each release: GCC/Clang, profile/debug. We need to keep the number of compiler versions for which we made builds small.
3. A Spack environment will be created corresponding to each standard build. Experiment code developers can build against the libraries available in these environments.

Standard Builds and Releases (part 2: recommendations for experiments)

4. We recommend experiments do development builds against one of the most recent LArSoft releases.

5. We recommend experiments use some standard build as the basis for their production releases. This need not be one of the most recent.

6. Experiments can choose to create their own alternative builds of LArSoft. The SciSoft team will consult on such work on a best-effort basis.

LArSoft and Experiment Continuous Integration (CI) Needs

1. SciSoft team will use LArSoft CI to build and test the development head of LArSoft. Each PR must pass all LArSoft CI tests and other requirements as the *first step* of approval for merging.

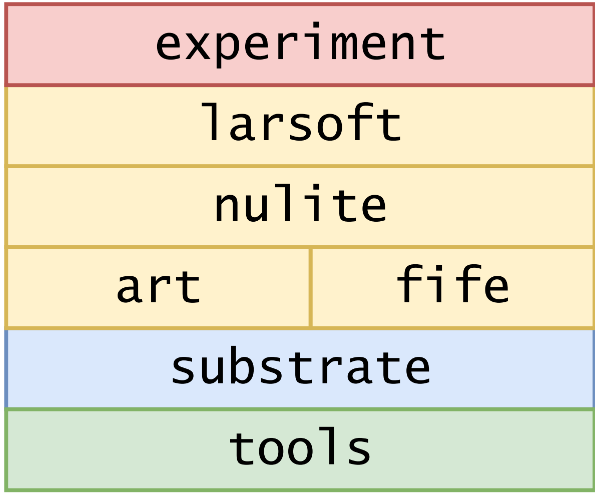
2. After LArSoft CI is passed, a PR must also pass CI for all supported experiments. It is the responsibility of the PR submitter to fix failures or to arrange to have them fixed.

3. To be included in the set of experiments Cis used to test LArSoft PRs, each experiment must keep their CI system up-to-date with the development head of LArSoft.

4. We will continue to support long-term "bug fix" branches for releases used for any experiment's production releases.

Organization of Layered Spack Environments

- Many who have used Spack have noted Spack's predilection to make new builds of previously-built packages with little provocation.
- To control this, we are planning to create *layered Spack environments*.
- We make use of a feature call `include_concrete` to help rein in the propensity of Spack to rebuild packages.
- The *substrate* level is intended to supply a large number of products, not just those required to directly built the *art* and *fife* layers. They are in this layer so that the Spack concretization picks the right build parameters to make all the packages consistent.



Final Notes

- The proposed plan (in draft version 5) is posted on the Indico page for this LCM.
- This plan has the support of the DSSL division.
- We are seeking agreement with this proposed plan from the LArSoft Collaboration experiments.