

Status of LArSoft beta release

L. Garren
B. Rebel
R. Snider

Fermilab

LBNE Software and Computing Meeting
Dec 20, 2013

Outline

- Status of the beta release
- Beta testing, acceptance criteria
- Off-site installs and builds
- The schedule
- LBNE code organization

(In backup: notes on how to get started)

Status of beta release

- It's out!
 - larsoft v0_02_01 -q <debug:e4 | e4:prof> (choose one)
 - Based on a snapshot of svn taken on Nov. 26
 - Documentation on the “LArSoft (beta)” site in Redmine
 - <https://cdcvs.fnal.gov/redmine/projects/larsoft>
 - Quick-start guide to using and developing under the new system
 - https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_Quick-start_guide_to_using_and_devel
 - The new development model
 - https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_The_developer_environment_
 - Explanation of the branching model and development workflow
 - Where to find which packages
 - [Overview of package organization](#)
 - [Re-factoring lists](#)

Status of beta release

- What remains to be done

- Beta testing (next topic)
- Remote installation and builds
 - On-going work testing binary installation and source build procedures
- Additional re-factoring work
 - fcl files
 - Some experiment-specific fcl files still in core LArSoft: eg, geometry.fcl
 - Better organization needed overall
 - Experiment-specific modules/services for which there is no generic analog
 - eg, SignalShapingService

None of these are critical, so are low priority prior to production transition

- Nightly build infrastructure
- Acceptance criteria / experiment sign-off

Beta testing

- Objectives
 - Technical:
 - To ensure that all programs that need to run do actually run
 - To validate the results obtained
 - To demonstrate that the software can be installed, built and run off-site
 - Usability:
 - To determine that the tools work as specified
 - To determine whether the tools provide the features needed to work effectively
 - Documentation:
 - To determine that the documentation is clear, correct, and complete
- Testing is the responsibility of LBNE
 - The LArSoft project will address problems, create new releases as needed

Beta testing

- Beta test procedure

see: https://cdcv.sfnal.gov/redmine/projects/larsoft/wiki/_Special_instructions_for_beta_testers_

- Follow the steps in the quick-start guide
- Run any job that you need to do your work
- Check the results are correct
- Report what you did and whether the test was successful
- Use the issue tracker to:
 - Report all technical problems, test failures
 - Report missing functionality
 - Suggest feature enhancements
 - Suggest documentation improvements

<https://cdcv.sfnal.gov/redmine/projects/larsoft/issues/new>

- The work for LBNE is being organized by Tom and Maxim

Acceptance criteria

- When is beta testing completed?
 - Need to agree on criteria
- Minimum criteria
 - A list of jobs that need to run
 - Define a sub-set that we will call the “test suite”
 - Will need the test suite again during the production transition
 - Verification that all run as expected
 - Validate the test suite against svn version
 - Just created an S2013.11.25 release that can be used for beta validation
 - Validation of the test suite run on an off-site binary install
 - Validation of the test suite run on an off-site source install + build
 - No “major” reported deficiencies in the tools, workflow
 - No “major” reported deficiencies in the documentation

Off-site installs and builds

- Three distribution methods

- Local binary install

- (Instructions: https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_Download_instructions_)

- Tested at SLAC (Tracy Usher)

- prodsingle_uboone.fcl worked out of the box.
 - A minor fcl change needed to standard_reco_uboone.fcl
 - Pandora was missing an xml file needed for initialization

- Nevis started testing (Bill Seligman)

- Reported an issue during installing under zsh. Work-around provided.

- Local source install + build

- (Instructions: https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_Download_instructions_)

- Work starting at BNL (Brett Viren)

- Minor issues encountered so far

- Getting organized at U Alabama (Ion Stancu, Muhammad Elnimr)

- cvmfs

- Beta release is uploaded to cvmfs server supported by the OSG (oasis)

The plan and schedule

- Schedule summary

- Beta testing: now until Jan 20
 - Identify test suite / acceptance criteria: Dec 17 - Jan 2
 - Can we do this today, now, for the technical elements??
 - Demonstrate off-site install / off-site build: by Jan 20
 - Need this to happen as early as possible (LBNE is the driver here)
 - Sign-off by experiments: by Jan 20
 - Need this to happen **as early as possible**. The second weeks of Jan?
 - Transition dry run: TBD
 - Need to get this down to as short a time as possible.
 - Goal is two days. Not easy, so need to start practicing, preparing on Jan 2
- Develop nightly build infrastructure: Jan 1 until Jan 17
- Production transition: Jan 21 to Jan 22 (!!)
 - Production release ready by Jan 23 (Avoids collaboration meeting)

Detailed work breakdown and schedule on LArSoft Project sharepoint site

The production transition

This schedule is my guess, and is not yet approved



- LArSoft will be unavailable during this procedure
 - Freeze svn, create a release (Eric, Herb, Brian) **Late on Jan 20**
 - No more commits to svn, ever!
 - Perform svn-to-git migration **Run overnight on Jan 20?**
 - Perform re-factoring **Jan 21 pm, Jan 22 am**
 - Mostly scripted. Changes to include paths, Geometry service are not.
 - Need to identify, deal with other non-backwards compatible changes we've made
 - Build release **Jan 22 am**
 - Initial testing **Jan 22 pm**
 - Should a remote install + build + test be included here? Is not in current plan...
 - Run test suite on svn and git-based releases. **Jan 22 pm**
 - Observe identical results.
 - Release LArSoft v1_00_00: downtime over! **Jan 22 (late, so maybe effectively Jan 23 am)**

LBNE code organization

- Current structure
 - All code removed from LArSoft in a single “lbncode” repository
 - Build with mrb etc.
- The question:
 - What do we want moving into the production transition?
 - Is the current structure adequate?
 - Will LBNE have “releases”, frozen ups products, etc, like LArSoft?
 - I would recommend “yes”
 - Will analysis code go into lbncode?
 - Does everyone need a personal git repository?
- Do not need an answer today, but will soon
 - Play with the beta release, and think about it in that context.
 - Discuss after the holidays
- Stakeholders and LArSoft general meeting the 2nd week of Jan

The end

Notes on how to get started

How to get started

- An important note:
 - Changes to git repositories are allowed
 - Should limit those changes to things needed to make the release work for testing purposes. But...

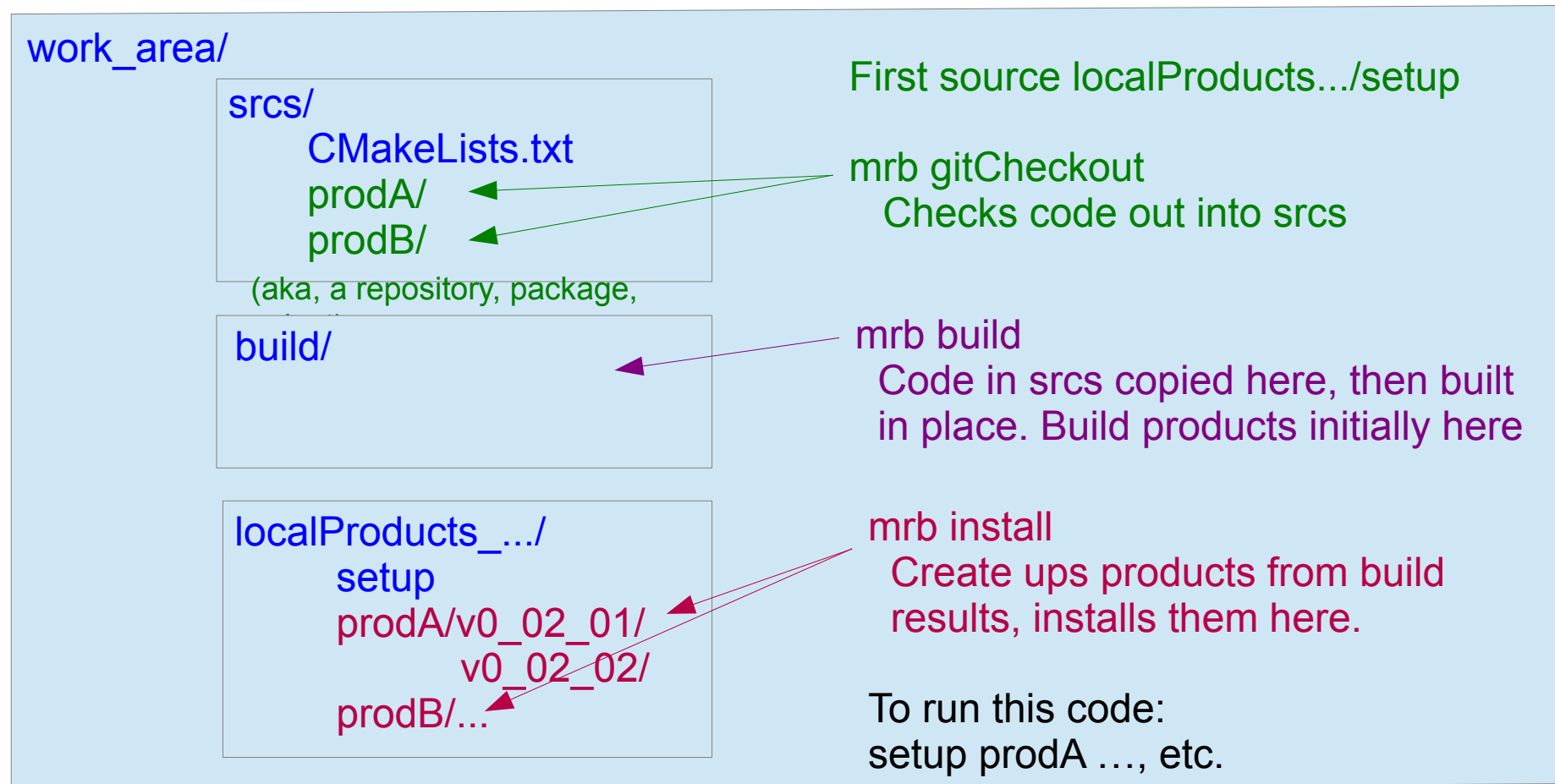
WARNING!! WARNING!! WARNING!!

- **Everything in git will be deleted** prior to the production transition
- Please **port any changes back into svn**, or **let us know about it**

How to get started

- Structure of the working area

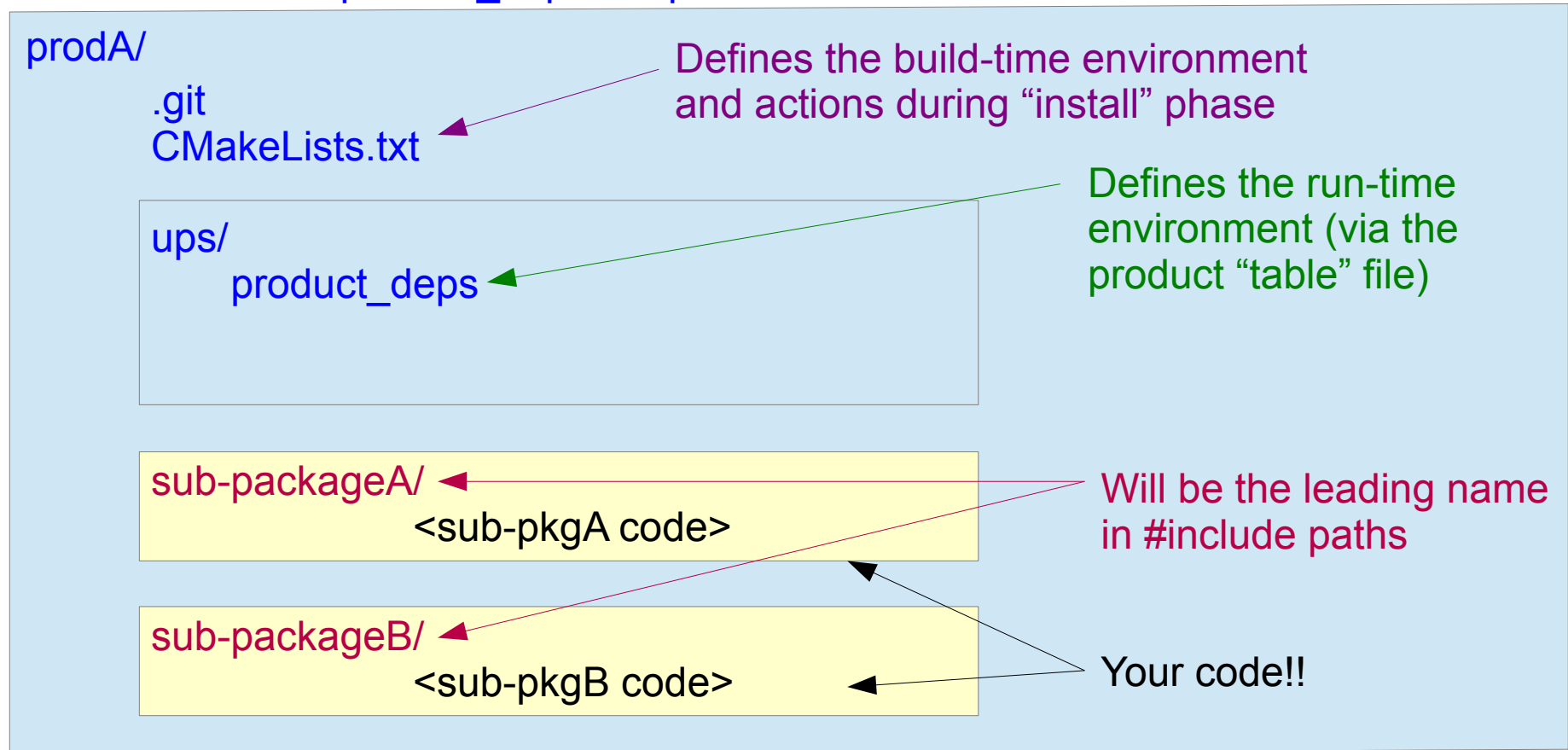
“mrB newDev” creates empty structure + localProducts.../setup + CMakeLists.txt



How to get started

- Structure of a repository

“mrb newProduct” creates skeleton of product/repository area + top-level CMakeLists.txt + product_deps templates



How to get started

- The quick-start summary

(see https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/_Quick-start_guide_to_using_and_developing_LArSoft_code_)

```
. /grid/fermiapp/uboone/software/setup_uboone.sh
mkdir <work_dir>
cd <work_dir>
mrb newDev -v v0_02_01 -q debug:e4      (or e4:prof for optimized)
. localProducts...xxx/setup
cd srcs
mrb g <repository-name>
cd <repository name>
<work.....>
cd ../../build
source mrb setEnv
mrb install
cd ..
setup <repository-name> vx_yy_zz -q debug:e4  (or whatever)
```

- The branching model

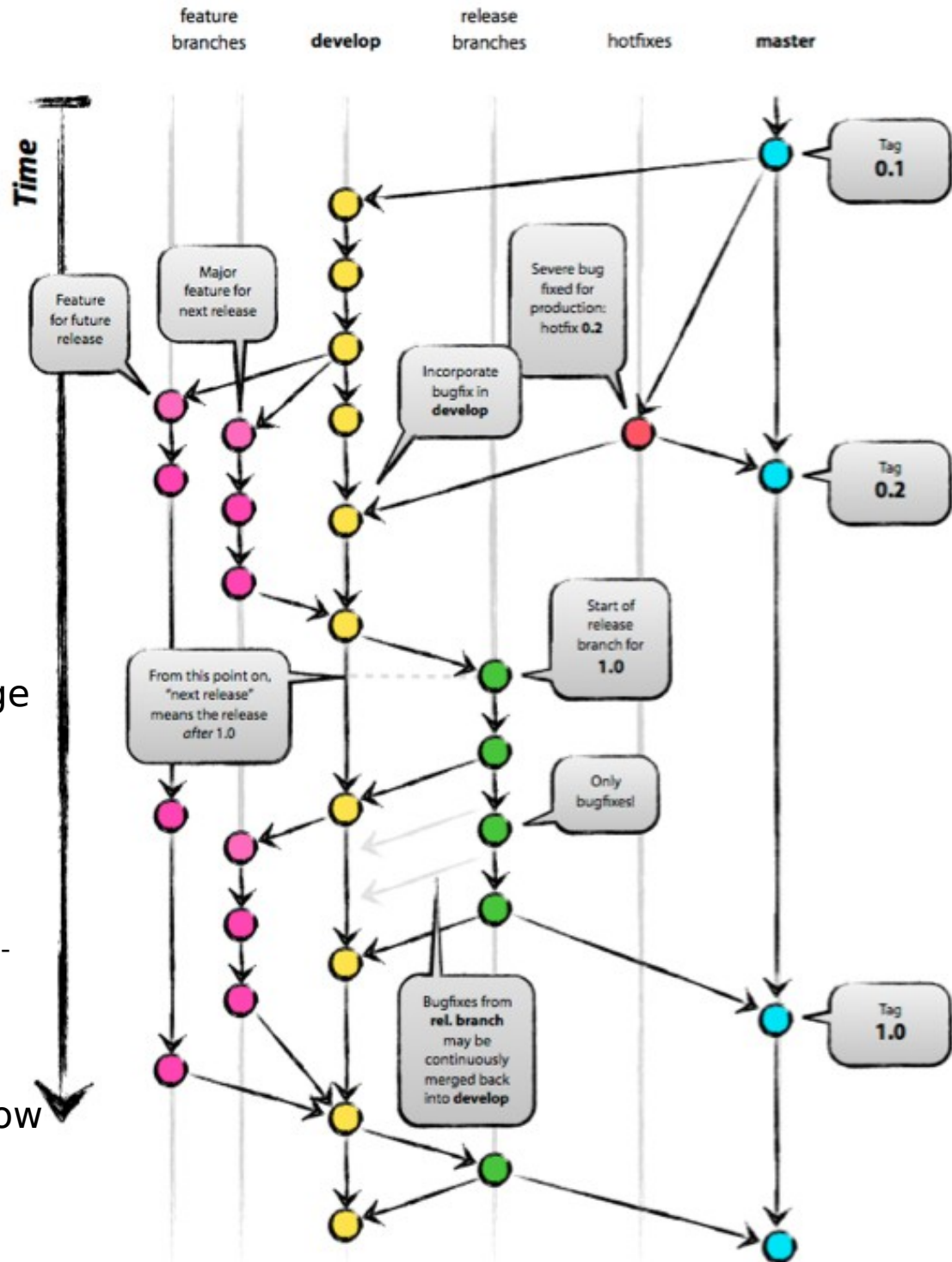
(see <http://nvie.com/posts/a-successful-git-branching-model/>)

- Vertical lines are branches
- Circles represent commits = a state of the repository
- Arrows are actions that change the state: branching, committing, merging

- Git flow

(See https://cdcvs.fnal.gov/redmine/projects/cet-is-public/wiki/Git_flow_quick_start)

- A tool to assist with working within this branching / workflow model



- Using git flow

git flow feature start <name>
<work...>

To share this feature:

git flow feature publish <name>
<more work>

To get work from others

git pull origin feature/<name>

To finish up:

git commit ...
git flow feature finish
<merges everything into develop>
git push origin develop

