

Setting up your workspace

David Marsden
Larsoft Workshop 2022

Outline

Task 1: Build sbndcode for use in further tasks

Task 2: Make a “test” build and look at more details

- Ask questions at any point
- Don't worry if you get errors
- Ask for help from me or other tutors
- **Everyone** will have built sbndcode at the end of this, we'll do it together

Before starting

Download these slides and have them open (<https://indico.ph.ed.ac.uk/event/130/>)

The \$ symbol before something indicates an environmental variable (e.g \$MRB_BUILDDIR), if things haven't been setup correctly, they might not do anything!

I put commands the you need to run in black boxes, e.g.

```
echo "hello world"
```

I also include some screenshots of terminal output, so you can check against your own.

Task 1

Step 1. Logging in

Click this link: <http://py-dom.lancs.ac.uk:8080/guacamole/#/>

Enter your username and password which you should have been provided with (shout if not!)

Open up a **terminal** (Applications → System Tools → Terminal)

Step 2. Make a work directory

```
cd $HOME  
mkdir larsoft_workdir  
cd larsoft_workdir
```

Step 3. Setup sbndcode using cvfms

```
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
```

Should look like this:

```
[dune24@py-dom larsoft_workdir]$ source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
Setting up LArSoft from "CVMFS":
- executing '/cvmfs/larsoft.opensciencegrid.org/setup_larsoft.sh'
- appending '/cvmfs/fermilab.opensciencegrid.org/products/common/db'
Setting up artdaq from "CVMFS":
- appending '/cvmfs/fermilab.opensciencegrid.org/products/artdaq'
Setting up sbn from "CVMFS":
- appending '/cvmfs/sbn.opensciencegrid.org/products/sbn'
Setting up SBND from "CVMFS":
- prepending '/cvmfs/sbnd.opensciencegrid.org/products/sbnd'
[dune24@py-dom larsoft_workdir]$ █
```

Step 4. Make a new development area with mrb

```
mrb newDev -v v09_60_00 -q prof:e20
```

Tells it the version of larsoft, in this case v09_60_00

qualifiers:

prof is to compile with optimizations and limited debug symbols

e20 codifies the compiler

Step 4. Make a new development area with mrb

```
mrb newDev -v v09_60_00 -q prof:e20
```

Should look like this:

```
[dune24@py-dom larsoft_testdir]$ mrb newDev -v v09_60_00 -q prof:e20
building development area for larsoft v09_60_00 -q prof:e20

The following configuration is defined:
  The top level directory is .
  The source code directory will be under .
  The build directory will be under .
  The local product directory will be under .

MRB_BUILDDIR is /home/dune24/larsoft_testdir/build_slf7.x86_64
MRB_SOURCE is /home/dune24/larsoft_testdir/srcs
INFO: copying /cvmfs/larsoft.opensciencegrid.org/products/larsoft/v09_60_00/releaseDB/base_dependency_database
/cvmfs/larsoft.opensciencegrid.org/products/mrb/v6_06_02/libexec/newDev.sh: line 192: unsetup: command not found

IMPORTANT: You must type
  source /home/dune24/larsoft_testdir/localProducts_larsoft_v09_60_00_prof_e20/setup
NOW and whenever you log in

[dune24@py-dom larsoft_testdir]$ █
```

Step 5. Source the local products

As it should have suggested in terminal:

```
source localProducts_larsoft_v09_60_00_prof_e20/setup
```

Should look like this:

```
[dune24@py-dom larsoft_testdir]$ source localProducts_larsoft_v09_60_00_prof_e20/setup

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v09_60_00
MRB_QUALS=prof:e20
MRB_TOP=/home/dune24/larsoft_testdir
MRB_SOURCE=/home/dune24/larsoft_testdir/srcs
MRB_BUILDDIR=/home/dune24/larsoft_testdir/build_slf7.x86_64
MRB_INSTALL=/home/dune24/larsoft_testdir/localProducts_larsoft_v09_60_00_prof_e20

PRODUCTS=/home/dune24/larsoft_testdir/localProducts_larsoft_v09_60_00_prof_e20:/cvmfs/sbnd.opensciencegrid.org/products/sbnd:/cvmfs/larsoft.opensciencegrid.org/products:/cvmfs/larsoft.opensciencegrid.org/packages:/cvmfs/fermilab.opensciencegrid.org/products/common/db:/cvmfs/fermilab.opensciencegrid.org/products/artdaq:/cvmfs/sbn.opensciencegrid.org/products/sbn
CETPKG_INSTALL=/home/dune24/larsoft_testdir/localProducts_larsoft_v09_60_00_prof_e20

[dune24@py-dom larsoft_testdir]$ █
```

Step 6. Clone sbndcode

`mrg sbndcode`

short for gitCheckout

(may need to type yes to connect)

Should take a few seconds then look like this:

```
[dune24@py-dom larsoft_testdir]$ mrg sbndcode
Cloning into 'sbndcode'...
remote: Enumerating objects: 28691, done.
remote: Counting objects: 100% (771/771), done.
remote: Compressing objects: 100% (363/363), done.
remote: Total 28691 (delta 496), reused 660 (delta 407), pack-reused 27920
Receiving objects: 100% (28691/28691), 76.05 MiB | 22.24 MiB/s, done.
Resolving deltas: 100% (19289/19289), done.
NOTICE: Adding sbndcode to CMakeLists.txt file
[dune24@py-dom larsoft_testdir]$ █
```

Step 7. Checkout the feature branch for this workshop

```
cd $MRB_SOURCE/sbndcode  
git checkout uk_larsoft_workshop_2022
```

Should look like this:

```
[dune24@py-dom sbndcode]$ git checkout uk_larsoft_workshop_2022  
branch 'uk_larsoft_workshop_2022' set up to track 'origin/uk_larsoft_workshop_2022'.  
Switched to a new branch 'uk_larsoft_workshop_2022'  
[dune24@py-dom sbndcode]$ █
```

Step 8. Set environment

`mrbsetenv` → This sets up the development environment

(check for errors)

```
[dune24@py-dom sbndcode]$ mrbsetenv
The working build directory is /home/dune24/larsoft_testdir/build_slf7.x86_64
The source code directory is /home/dune24/larsoft_testdir/srcs
----- check this block for errors -----
INFO: mrb v6 06 02 requires cetmodules >= 2.31.00 to run: attempting to configure...v3 19 02 OK
-----
To inspect build variable settings, execute /home/dune24/larsoft_testdir/build_slf7.x86_64/cetpkg_info.sh
Please use "buildtool" (or "mrb b") to configure and build MRB project "larsoft", e.g.:

  buildtool -vTl [-jN]

See "buildtool --usage" (short usage help) or "buildtool -h|--help"
(full help) for more details.

[dune24@py-dom sbndcode]$ █
```

Step 9. Build and install

```
mrbi -j4
```

↳ stands for “install”

Wait ~10 minutes and hopefully see no errors

Take a break! Or if you like, you can move on to Task 2 (but don't forget to come back for step 10.)

Step 10. Setup the local install

```
mrbslp
```

This sets up all of the products installed in the localProducts_.../ directory.

Now this work directory is setup, we can leave it ready for further tutorials!

Make a bash script (no need to install every time!)

```
cd $HOME/larsoft_workdir
```

Copy the bash script to larsoft_workdir:

```
cp srcs/sbndcode/sbndcode/Workshop/Setup/setup.sh .
```

OR you can write it yourself:

```
vim setup.sh (use emacs etc. if you prefer 😊)
```

```
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
source localProducts_larsoft_v09_60_00_prof_e20/setup
mrbsenv
mrbslp
```


Check the bash script works

Open a new terminal

```
cd $HOME/larsoft_workdir
```

```
source setup.sh
```

Hopefully see no errors

For later tutorials remember to source this script **every time** you open a new terminal!

Task 2

Make a new test directory

Open a **new** terminal

```
cd $HOME  
mkdir larsoft_testdir  
cd larsoft_testdir
```

Repeat previous steps **3 to 10**.

Make a data directory

```
cd $HOME
```

```
mkdir data
```

```
cd data
```

When working on the Fermilab gpvms, you should use:

/EXPERIMENT_NAME/**data**/ directories to store **data**

/EXPERIMENT_NAME/**app**/ directories to store **code**, these have a smaller quota but are backed up.

(There are also *scratch* directories for very large datasets, which get wiped after some time, but not going into that here.)

Create some events

```
lar -c prodsingle_sbnd.fcl -n 5 -o muons.root
```

(make sure you've done mrbslp or sourced a setup script before, or it won't run)

There will be a lot more on creating events like this in further tutorials, so don't worry about it too much here.

But briefly: -c specifies the file to run larsoft over
-n specifies the number of events
-o specifies the name of the output file

Extra stuff

Other useful commands

Look at the versions of code available

```
ups list -aK+ larsoft
```

```
ups list -aK+ sbndcode
```

Check active versions of code **currently setup** in your terminal

```
ups active
```

Better way to recompile and run

- If you're modifying code, installing and running, it can get fiddly/time consuming to do the steps to build and run
- Solution: keep two terminals open, one for building and another for running
- Solution: recompile using

```
ninja -C ${MRB_BUILDDIR} install
```

(if you're interested please ask another tutor about how ninja works 😊)

Other resources

- Previous presentations: [Iker 2020](#) , [Miquel 2022](#)
- Larsoft release documentation:
https://cdcv.sfnal.gov/redmine/projects/larsoft/wiki/LArSoft_release_list
- Try search box in [LArSoft github](#) to look for specific modules
- mrb [reference guide](#)

Good luck, you will get there!