THE UNIVERSITY of EDINBURGH

UK LArSoft Workshop 2026

26.10.2026

Setting up your workspace!



Lucy Kotsiopoulou



Rory at SBND-ND, Fermilab, Lucy Kotsiopoulou (2025)

Welcome to this tutorial! *What does it mean?*



- We will go through all the commands you need to run on the terminal to run… basically any software on your experiment
- On a normal day of coding for Fermilab LArTPCs (Liquid Argon Time Projection Chambers):
  - Connect to your experiment's GPVM (General Purpose Virtual Machine): shared computing nodes at Fermilab that you log into remotely to do your work
    - Rather than running heavy jobs on your own computer or having to install difficult software on it, you can work on a *virtual* machine that has everything installed for you!
  - **"Set up" the software you want to use that day - tell the computer what frameworks you will need to use**
  - Do "things" with it! E.g.:
    - Make simulation MC files (samples) of interactions in the detector
    - Analyse ready files
    - Look at raw data (for trigger studies e.g.)
    - Look at the beam simulation software
    - etc.



- **This workshop:** how to set up LArSoft (code that all LArTPCs at Fermilab are based on) to *develop* tools for looking at interactions in the detector
  - *Not* beam simulation e.g.
  - *Not* many other things that you may start out doing in your PhDs…

Contents of this presentation and how it is set up

- Commands you *should run* will be in this boxes like this, e.g.:

```
> echo DO THIS!
```

- Commands *you should not run*, but are good to know look like this:

```
> echo DON'T DO THIS!
```

- And finally, optional commands look like this:

```
> echo MAYBE DO THIS!
```

- What running the code looks like on my screen and should look like on yours:

```
[lar39@larsoft04 ~]$ echo hello
hello
[lar39@larsoft04 ~]$ echo "hello"
hello
```

Basic commands reminder!

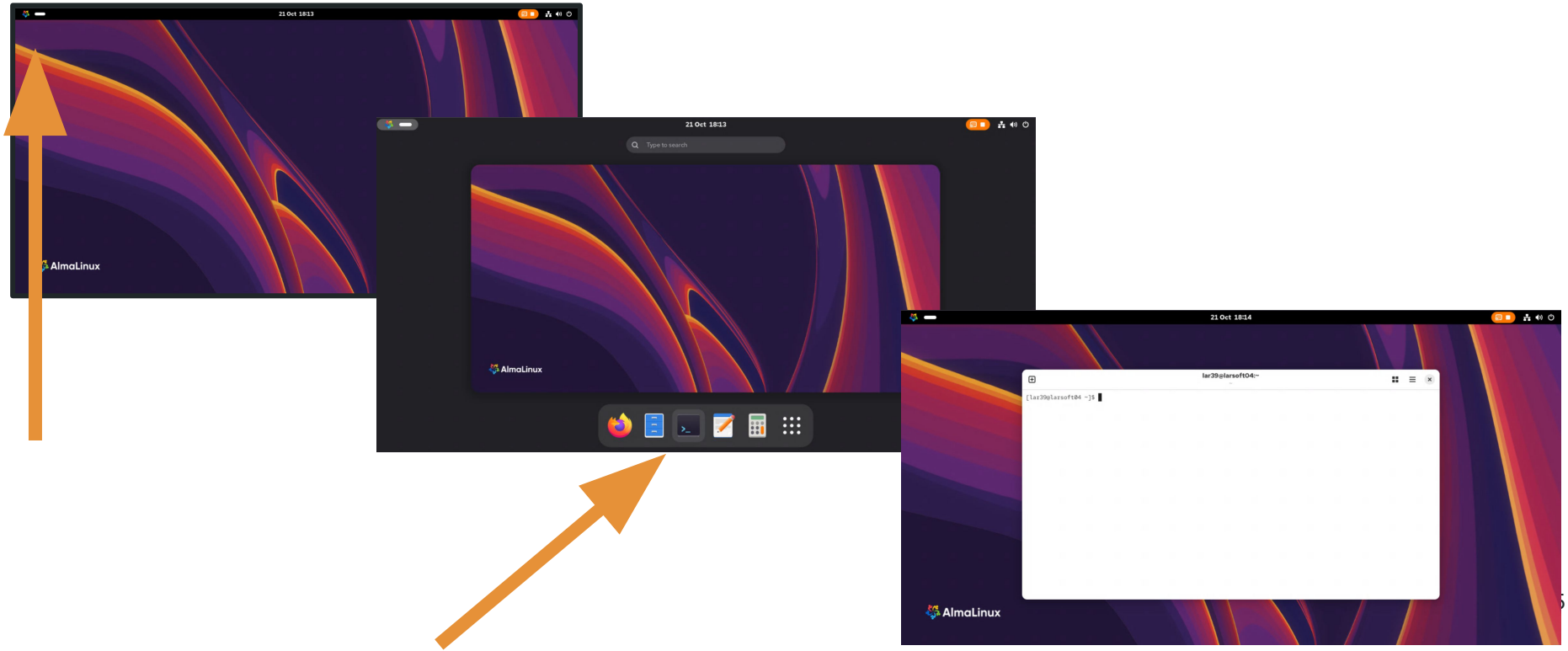
- **> mkdir my_fav_folder** : **m**a**k**e a new **dir**ectory called “my_fav_folder”
- **> cd my_fav_folder/** : **c**hange **d**irectory, go into to the folder “my_fav_folder”
  - **> cd ../** : go into the previous folder
  - **> cd ../../../../** : go into the pre-pre-pre-previous folder
- **> ls** : **l**i**s**t, shows you the contents of the folder you’re in
- **> pwd** : **p**rint **w**orking **d**irectory, tells the absolute path of where you are
- **> cp myscript.ch my_fav_folder/** : **c**o**p**y the file “myscript.sh” into my_fav_folder
- **> rm myscript.sh** : remove, delete (permanently!) the file “myscript.sh”
- **> export EASY_PATH=”/home/lar39/Videos/”** : set a path as a variable

- **Up arrow key** : see previous command
- **ctr+r** : search command history, keep pressing to keep going up
- **Tab :** autocomplete a command, e.g. **cd my_**



How to get started!

- Click on the link! [Guacamole server for working tutorials during workshop]
- Type in your username and password to enter the computer
- Open the terminal!



DO NOT log out!

- NEVER log out of the computer
- You can:
  - Close the terminal
  - Close the internet browser tab or window
  - Turn of your computer
- You CANNOT:
  - Log off
  - :)



Let's begin! Enter a container

You should be in your home directory automatically when you open a terminal. Like this:

```
> pwd
```



The Linux operating system of this computer is AlmaLinux - but our code doesn't work here! So we need to enter a "box" with a different operating system, SL7 - A CONTAINER:

```
> source /Workshop/setting_up/setup_container.sh
```

```
[lar39@larsoft04 ~]$ source /Workshop/setting_up/setup_container.sh
(CONTAINER) [larsoft04@~]$
```

* This command only works for *this workshop*; see the backup slides for how to do it in the real world.

Make your working folder

Make the directory you want to work with for the rest of the week!

```
> mkdir larsoft_workshop_stuff
```

You can now see the folder you made was created:

```
> ls
```

```
(CONTAINER) [larsoft04@~]$ mkdir larsoft_workshop_stuff
(CONTAINER) [larsoft04@~]$ ls
Desktop    Music     Templates              setting_up       test2
Documents  Pictures  Videos                 setting_up_test  test_container_github
Downloads  Public    larsoft_workshop_stuff test1            workshop_2026
```

Now go into to this directory:

```
> cd larsoft_workshop_stuff/
```

```
> ls
```

Load the SBND software environment

Source (run) this setup script to tell your computer what LArSoft is:

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

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ 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'
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

- This command loads all the SBND software environment and tells your computer *what LArSoft is*, where to find its commands and points to the right tools and libraries.
- CVMFS (CernVM FileSystem): a network file system CERN runs that distributes software to physics labs worldwide, so you don't have to install it locally
- Also tells your computer what ROOT and python is e.g.

Create a fresh *development area* to work in

Create the area:

```
> mrb newDev -v v10_14_02_04 -q prof:e26
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ mrb newDev -v v10_14_02_04 -q prof:e26

building development area for larsoft v10_14_02_04 -q prof:e26


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/lar39/larsoft_workshop_stuff/build_slf7.x86_64
MRB_SOURCE is /home/lar39/larsoft_workshop_stuff/srcs
INFO: copying /cvmfs/larsoft.opensciencegrid.org/products/larsoft/v10_14_02_04/releaseDB/base_dependency_database
/cvmfs/larsoft.opensciencegrid.org/products/mrb/v6_09_13/libexec/newDev.sh: line 192: unsetup: command not found

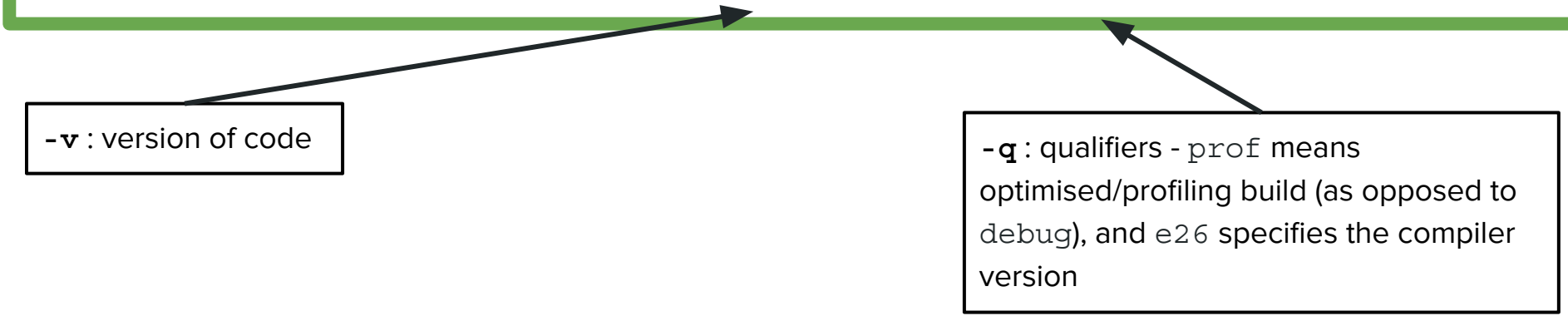
IMPORTANT: You must type
    source /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/setup
NOW and whenever you log in

(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

Create a fresh *development area* to work in

Create the area:

```
> mrb newDev -v v10_14_02_04 -q prof:e26
```



- This is basically creating a folder structure where you can use and modify LArSoft code without touching the central installed version
- **mrb**: **M**ulti-**R**epository **B**uild — a tool Fermilab made specifically for managing and building LArSoft code
- LArSoft is split across many separate repositories (=big folder with code, and its history in it) and building them all together in the right order with the right dependencies (=code that your code relies on to work) is complicated. `mrb` handles all of that for you.

Create a fresh *development area* to work in

Create the area:

```
> mrb newDev -v v10_14_02_04 -q prof:e26
```

So now, your folder has things in it! You can look at them:

```
> ls
```

```
> ls srcs/
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls
build_slf7.x86_64  localProducts_larsoft_v10_14_02_04_prof_e26  srcs
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls srcs/
CMakeLists.txt
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

Setup you local development area

Source the dev area's setup now: (it told you to do this in the terminal anyway…)

```
> source localProducts_larsoft_v10_14_02_04_prof_e26/setup
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ source localProducts_larsoft_v10_14_02_04_prof_e26/setup

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v10_14_02_04
MRB_QUALS=prof:e26
MRB_TOP=/home/lar39/larsoft_workshop_stuff
MRB_SOURCE=/home/lar39/larsoft_workshop_stuff/srcs
MRB_BUILDDIR=/home/lar39/larsoft_workshop_stuff/build_slf7.x86_64
MRB_INSTALL=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26

PRODUCTS=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26:/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.o
rg/products/sbn
CETPKG_INSTALL=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26

(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls
```

- This line tells your terminal "use *my* local versions of things, not just the central ones from cvmfs." So if you build and install your own modified code, your terminal will find your version first.

A note on environment variables

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ source localProducts_larsoft_v10_14_02_04_prof_e26/setup

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v10_14_02_04
MRB_QUALS=prof:e26
MRB_TOP=/home/lar39/larsoft_workshop_stuff
MRB_SOURCE=/home/lar39/larsoft_workshop_stuff/srcs
MRB_BUILDDIR=/home/lar39/larsoft_workshop_stuff/build_slf7.x86_64
MRB_INSTALL=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26

PRODUCTS=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26:/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.o
rg/products/sbn
CETPKG_INSTALL=/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26

(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls
```

a.k.a. What do these mean???

You can set paths that are useful to you as variables, so as to not have to type out the path every time, e.g.:

```
> export EASY_PATH="/home/lar39/Videos/"
```

```
> echo $MRB_TOP
```

Copy the repositories from Github to your dev area

Clone (copy) the sbndcode repository (big folder) into your dev area's `srcs` folder:

```
> mrb g sbndcode
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ mrb g sbndcode
Cloning into 'sbndcode'...
remote: Enumerating objects: 74798, done.
remote: Counting objects: 100% (832/832), done.
remote: Compressing objects: 100% (321/321), done.
remote: Total 74798 (delta 640), reused 556 (delta 510), pack-reused 73966 (from 3)
Receiving objects: 100% (74798/74798), 510.29 MiB | 35.44 MiB/s, done.
Resolving deltas: 100% (53321/53321), done.
NOTICE: Adding sbndcode to CMakeLists.txt file
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

```
> mrb g larpandora
```

```
> mrb g larpandoracontent
```

- `g` stands for git - it copies the folder from Github
- Github is a website for storing code so that everyone can collaborate on it together. See this amazing Git for Dummies video

Copy the repositories from Github to your dev area

Now we have things in our `srcs` folder!

```
> ls srcs/
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls srcs/
CMakeLists.txt  larpandora  larpandoracontent  sbndcode
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

This is the SBNSoftware website where ALL the repositories are stored, like:

- sbndcode
- larpandora
- larpandoracontent
- and MANY others (~95 repos...)

So we've just "downloaded" these relevant repositories for our tutorials onto our computers, into the dev area and told mrb to look for them.

Checkout the workshop git branches

Go into one of our git cloned (copied/downloaded) repositories:

```
> cd $MRB_SOURCE/sbndcode
```

Git has branches: version of code that have diverged from the “main” branch without affecting it.

You can see what branch we are on now:

```
> git branch
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/sbndcode]$ git branch
* develop
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/sbndcode]$
```

You can see all the branches that have made in this repository like this:

```
> git branch -a
```

Checkout the workshop git branches

Checkout (=start working on this branch) the larsoft workshop branch we made:

```
> git checkout uk_larsoft_workshop_2026
```

And now you can see the branch we are on has changed!

```
> git branch
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/sbndcode]$ git checkout uk_larsoft_workshop_2026
branch 'uk_larsoft_workshop_2026' set up to track 'origin/uk_larsoft_workshop_2026'.
Switched to a new branch 'uk_larsoft_workshop_2026'
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/sbndcode]$ git branch
  develop
* uk_larsoft_workshop_2026
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/sbndcode]$
```

Checkout the workshop git branches

Do the same again for larpandora:

```
> cd $MRB_SOURCE/larpandora
```

```
> git checkout LARSOFT_SUITE_v10_14_02_02
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandora]$ git checkout LARSOFT_SUITE_v10_14_02_02
Note: switching to 'LARSOFT_SUITE_v10_14_02_02'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a93b252 larpandora v10_01_09_02 for larsoft v10_14_02_02
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandora]$
```

Checkout the workshop git branches

...and *kinda* the same for larpandoracontent:

```
> cd $MRB_SOURCE/larpandoracontent
```

```
> git remote add pandora https://github.com/PandoraPFAOrg/LArContent.git
```

```
> git fetch pandora
```

```
> git checkout pandora/uk_larsoft_workshop_2026
```

```
> git switch -c uk_larsoft_workshop_2026
```

- The previous two repositories (sbndcode and larpandora) were in the same organisation, on the SBNSoftware GitHUb
- This *particular* branch of larpandoracontent that we want is in a *completely different* organisation: PandoraPFAOrg

Set up your environment for building!

Basically mrb checking your dev area and preparing everything so that it is ready for installation

```
> mrbsetenv
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandoracontent]$ mrbsetenv
The working build directory is /home/lar39/larsoft_workshop_stuff/build_slf7.x86_64
The source code directory is /home/lar39/larsoft_workshop_stuff/srcs
----------- check this block for errors ----------------------
------------------------------------------------------------
To inspect build variable settings, execute /home/lar39/larsoft_workshop_stuff/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.

(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandoracontent]$
```

Build!

Build and install all the code in your dev area:

```
> mrb install --generator ninja -j 3
```

```
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/f
/mergesimsources_sbnd.fcl
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/f
/simdrift_sbnd.fcl
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_tools_TrackHitEfficiencyAnalysis_tool.so
-- Set runtime path of "/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_1
02_04/slf7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_tools_TrackHitEfficiencyAnalysis_tool.so" to ""
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/i
lude/sbndcode/TPC1DSignalProcessing/tools/IHitEfficiencyHistogramTool.h
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
rce/sbndcode/TPC1DSignalProcessing/tools/IHitEfficiencyHistogramTool.h
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
rce/sbndcode/TPC1DSignalProcessing/tools/TrackHitEfficiencyAnalysis_tool.cc
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/f
/analysis_tools_sbnd.fcl
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_CalWireSBND_module.so
-- Set runtime path of "/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_1
02_04/slf7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_CalWireSBND_module.so" to ""
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_HitEfficiencyAna_module.so
-- Set runtime path of "/home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_1
02_04/slf7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_HitEfficiencyAna_module.so" to ""
-- Installing: /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26/sbndcode/v10_14_02_04/s
7.x86_64.e26.prof/lib/libsbndcode_TPC1DSignalProcessing_ROIFinderStandardSBND_tool.so
```

Build!

Build and install all the code in your dev area:

```
> mrb install --generator ninja -j 3
```

`-generator ninja` : use Ninja as the build tool instead of the default Make. Ninja is faster

`-j 3` : uses 3 CPU cores at once to speed things up - you can tune this to higher but **NOT HERE at the workshop**

- **Takes about 10 minutes :)** - LArSoft is a big codebase!
- Subsequent builds are faster because it only recompiles what changed.

Building “just with mrb” is also widely used for building your code (at least in SBND!), so just as an FYI, this is it:

```
> mrb i -j3
```

Build!

The final step! With this we check that everything is set up correctly:

```
> mrbslp
```

```
------------------------------------
INFO: stage install SUCCESS for MRB project larsoft v10_14_02_04
------------------------------------

(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandoracontent]$ mrbslp
local product directory is /home/lar39/larsoft_workshop_stuff/localProducts_larsoft_v10_14_02_04_prof_e26
----------- this block should be empty -----------------
------------------------------------------------------
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff/srcs/larpandoracontent]$
```

- **`mrbslsp`** : mrb show local products

All 21 commands we ran!

```
 1.  source /Workshop/setting_up/setup_container.sh
 2.  mkdir larsoft_workshop_stuff
 3.  cd larsoft_workshop_stuff
 4.  source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
 5.  mrb newDev -v v10_14_02_04 -q prof:e26
 6.  source localProducts_larsoft_v10_14_02_04_prof_e26/setup
 7.  mrb g sbndcode
 8.  mrb g larpandora
 9.  mrb g larpandoracontent
10.  cd $MRB_SOURCE/sbndcode
11.  git checkout uk_larsoft_workshop_2026
12.  cd $MRB_SOURCE/larpandora
13.  git checkout LARSOFT_SUITE_v10_14_02_02
14.  cd $MRB_SOURCE/larpandoracontent
15.  git remote add pandora https://github.com/PandoraPFAOrg/LArContent.git
16.  git fetch pandora
17.  git checkout pandora/uk_larsoft_workshop_2026
18.  git switch -c uk_larsoft_workshop_2026
19.  mrbsetenv
20.  mrb install --generator ninja -j 3
21.  mrbslp
```

Only four are necessary to “resume” your work!

```
source /Workshop/setting_up/setup_container.sh
    mkdir larsoft_workshop_stuff
    cd larsoft_workshop_stuff
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
    mrb newDev -v v10_14_02_04 -q prof:e26
source localProducts_larsoft_v10_14_02_04_prof_e26/setup
    mrb g sbndcode
    mrb g larpandora
    mrb g larpandoracontent
    cd $MRB_SOURCE/sbndcode
    git checkout uk_larsoft_workshop_2026
    cd $MRB_SOURCE/larpandora
    git checkout LARSOFT_SUITE_v10_14_02_02
    cd $MRB_SOURCE/larpandoracontent
    git remote add pandora https://github.com/PandoraPFAOrg/LArContent.git
    git fetch pandora
    git checkout pandora/uk_larsoft_workshop_2026
    git switch -c uk_larsoft_workshop_2026
mrbsetenv
    mrb install --generator ninja -j 3
mrbslp
```

If you close the terminal, everything we did to set up the *environment* is forgotten. So we need to run these again:

Only four are necessary to “resume” your work!

If you close the terminal, everything we did to set up the *environment* is forgotten. So we need to run these again:

```
source /Workshop/setting_up/setup_container.sh
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
source localProducts_larsoft_v10_14_02_04_prof_e26/setup
mrbsetenv
mrbslp
```

Instead of having to remember them or have these slides open… you can make a bash script that will run them for you:

```
> cd $MRB_TOP
```

```
> emacs setup.sh &
```

& keeps terminal active

Working with EMACS for now (see backup for key controls): Use the GUI for what you can (paste, save, close)

Only four are necessary to “resume” your work!

So paste the following lines into the script:

```
source /Workshop/setting_up/setup_container.sh
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
source localProducts_larsoft_v10_14_02_04_prof_e26/setup
mrbsetenv
mrbslp
```

Try this now!

Except for the container script source! If you do, once it enters the container, i will just forget about the rest of the commands… I’ve tried :)

Paste the lines.

Save the script.

CLOSE THE TERMINAL!

Try this again now!

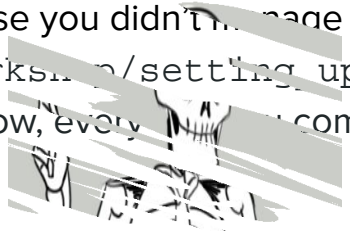
So EVERY time you come back to the terminal now, do:

```
> cd larsoft_workshop_stuff/
```

```
> source /Workshop/setting_up/setup_container.sh
```

```
> source setup.sh
```

- Wherever you source the container from, you will continue to be in that folder afterwards; so it doesn't matter where you are when you source it :)
- In case you didn't manage to make this setup script, you can find a ready one I made here: `/Workshop/setting_up/backup_devArea_setup.sh`
- So now, every time you come back to the workshop, or after a break if you have closed your terminal, do ⬆





CONGRATULATIONS! That's it!

Other useful commands!

This will show you all the all the UPS (Unix Product Support) packages that are currently set up in your environment - basically everything that's been activated via setup commands!

```
> ups active
```

- **> screen -S <screen_name> :** opens a screen session on your terminal; you can close it and things will keep running there
  - **> screen -r <screen_name> :** resume screen session
  - **ctrl+a+d :** leave the screen session (things keep running)
  - **> screen -XS <screen_name> quit :** kill the session, when not in it

Text editors you can use!

Thank you Holly Parkinson!

Good things to know: text editors

vim

- `vim <filename>` to open file (creates file if it doesn't exist)
- `i` to edit (insert)
- `esc` to stop editing
- `:wq` to save and quit, or `:q` to quit without saving

emacs

- `emacs –nw <filename>` to open file (creates file if it doesn't exist; nw is 'no window')
- move your cursor to where you want and start typing
- `Ctrl-X Ctrl-C` to quit: when asked to save, type `y` for yes or `n` for no
- `Ctrl-X Ctrl-S` to save without closing

or, if you have another text editor that you like, use that!

Let’s simulate some things to test it!

- Close the terminal
- Open the terminal again
- Make a new folder for this new test to not contaminate the previous one for the workshop

```
> mkdir test_sim
```

- Set up a new dev area from scratch by following slides 7-24 . Yes... again :)
- Now let’s use it!

Make some neutrino events!

- ***Generate* (i.e. simulate Monte Carlo) events with a *generator fhicl*:**

```
> lar -c prodgenie_corsika_proton_rockbox0p1_sbnd.fcl -n 10 -o output.root
```

`lar`: LArSoft's command for "do" something

`-c <fcl>` :which configuration file to use - tells it which fhicl! Watch that there's not path to this fhicl, it just *knows* how to find it!

`-o <file_name.root>` : output, what to name (and where to save) your output file

`-n <number>` : number, how many (neutrino) events to generate

Make some neutrino events!

- ***Generate* (i.e. simulate Monte Carlo) events with a *generator fhicl*:**

```
> lar -c prodgenie_corsika_proton_rockbox0p1_sbnd.fcl -n 10 -o output.root
```

- What does this *actually simulate*?
  - Neutrinos from our neutrino beam using the GENIE neutrino generator
  - Cosmic muons using the CORSIKA generator (our main background!)
    - Rockbox is a wider volume than the active volume (AV), so that we simulate neutrinos that not only interacted in the detector but around the *dirt* surrounding SBND. External neutrinos need to deposit a certain amount of energy (0.1Mev -> 0p1) for this - there have been many studies to define this number!
- This very fhicl is what is centrally used in SBND for *Production*!

Make some neutrino events!

- ***Generate*** **(i.e. simulate Monte Carlo) events with a** ***generator fhicl*****:**

```
> lar -c prodgenie_corsika_proton_rockbox0p1_sbnd.fcl -n 10 -o output.root
```

```
TimeReport ---------- Time summary [sec] -------
TimeReport CPU = 216.923104 Real = 255.763587

MemReport  ---------- Memory summary [base-10 MB] ------
MemReport  VmPeak = 7701.82 VmHWM = 2142.38

%MSG-w MCTruthEventAction:  PostEndJob 24-Oct-2026 01:03:00 BST ModuleEndJob
The following unknown PDG codes were present in the simb::MCTruth input.
They were not processed by Geant4.
   Unknown PDG code = 2000000101, seen 2 times.  (GENIE specific)
%MSG
Art has completed and will exit with status 0.
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

- “Exit with status 0” means it worked! Any other number is bad :(

Make some neutrino events!

- **See what it made:**

```
> ls
```

```
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$ ls
build_slf7.x86_64  hists_prodgenie_cosmic_rockbox_sbnd_GenieGen-20261024T000300.root  messages.log  srcs
cputime.db         localProducts_larsoft_v10_14_02_04_prof_e26                        output.root
errors.log         memory.db                                                          setup.sh
(CONTAINER) [larsoft04@~/larsoft_workshop_stuff]$
```

- It made:
  `output.root`
  `hists_prodgenie_cosmic_rockbox_sbnd_GenieGen-20261024T000300.root`
  `errors.log`
  `cputime.db`
  `memory.db`
  `messages.log`
  The ROOT files are the interesting ones! You can open one like this: (more in the next tutorials...)

```
> root
> root [0] TBrowser b
```

Three ways to set up your workspace

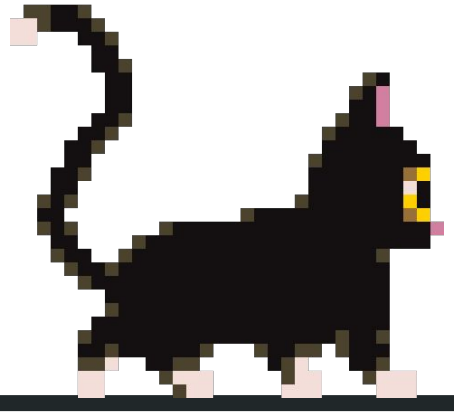
So there are three cases for setting up your workspace that you will need:

1. **Starting completely from scratch; you want to copy ready versions of code from elsewhere to your area, with the end goal of modifying it slightly your way:**
   Use all 21 commands for this! Not needed again for this workshop.
2. **Returning to an already set up area:**
   Use all 4 commands we put in the setup script before. You will need this every time you close the terminal during the workshop!
3. **Registering "modified code" within your devArea: you changed some of the code that you originally set up before! Now you need to tell mrb to see and integrate that change into its system. After making the change, you will need to rebuild:**
   `mrbsetenv`
   `ninja -C $MRB_BUILDDIR sbndcode/install -j 3` OR `mrb i -j3`
   `mrbslp`
   They will tell you how to do this in the next tutorials though!
4. **EXTRA: just run the simulation! If you're not planning on *modifying* the existing code at all, you don't need to do any of this. e.g.:**

```
source /Workshop/setting_up/setup_container.sh
source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
setup sbndcode v10_14_02_04 -q e26:prof
lar -c prodgenie_corsika_proton_rockbox0p1_sbnd.fcl -n 10 -o output.root
```

How to follow these commands for your experiment: SBND

```
> sh /exp/$(id -ng)/data/users/vito/podman/start_SL7dev_jsl.sh
> source /cvmfs/sbnd.opensciencegrid.org/products/sbnd/setup_sbnd.sh
> mrb newDev -v v10_14_02_04 -q prof:e26
> source localProducts_larsoft_v10_14_02_04_prof_e26/setup
> mrb g sbndcode
> cd $MRB_SOURCE/sbndcode
> git checkout <maybe_your_own_branch>
> mrbsetenv
> mrb i -j4
> mrbslp
```



How to follow these commands for your experiment: uBooNE

I owe you this!

How to follow these commands for your experiment: DUNE

I owe you this!

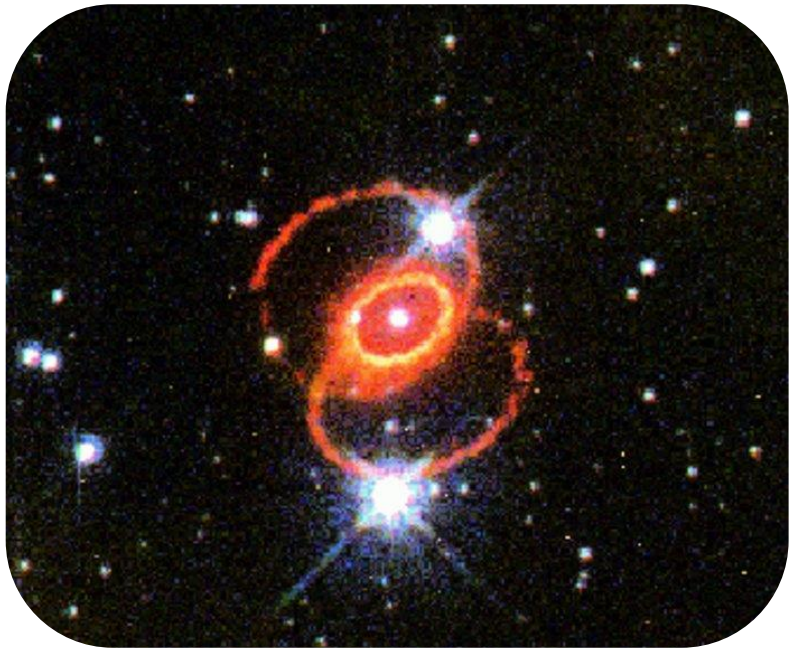
A note on *The Grid* and *Production*

- **The Grid:**
  - In the beginning, you will probably start by simulating small samples (-n 100 or -n 10000) for the analysis you're working on
  - But soon you will need many more statistics! The GPVMs will crash and be very slow if you simulate too many events...
  - So you can send large jobs to a centralised system of many CPUs working in parallel where your simulation will run faster and more efficiently: The Grid!
  - Makes a difference of days (gpvms) to hours (grid)!
  - Not covered in this workshop, but there are some instructions here: Using project.py for grid jobs
    - Or ask any of us!
    - Or ask in the SBN Slack in `#nostupidquestions`
- **Production**
  - MC samples made centrally by your collaboration that everyone can use!
  - Instead of everyone making their own samples... (it won't be universal, and there will be mistakes) - there is a Production Team that makes HUGE statistics samples that you can analyse and finally use in your analysis
  - This happens a few times a year to keep up with code versions (Gen1 and Gen2 so far in SBND!)
  - This Team also processes real data from the detector!

A little bit about me!

- **I love HALLOWE'EN**
- **On SBND and DarkSide-20k (dark matter experiment in Italy)**
- **General theme of my PhD is supernova neutrinos!**
- **On SBND:**
  - **Working on a cross section measurement of low energy (<100MeV) ν_e from muons decaying-at-rest (μDAR)**
    - Beam (BNB) simulation using G4BNB
    - Added in new trigger configuration for μDARs
    - Cut selection analysis
  - Calibration database manager
  - Slow Controls (DCS - Detector Control System) expert - one of many!
- **You will end up doing many things too!**





A photo of Supernova 1987A's remnants as seen in 1995 by the Hubble space telescope, **GSFC/NOAO/NASA/ESA/STIS Instrument Definition Team** (1995)

Thank you!



Please ask questions! Especially “stupid” ones...

I am around on Slack for more conversations :)

Backup slides