Setting up your workspace

Adapted form Tom Ham’s tutorial

M.Nebot, UK-Latin America School on DUNE

Goals for this session

1) Setup and compile dune software for use in later sessions. Once this has been done (successfully), don't touch it again. Make sure this has been done as you'll need it later - ask if you need help.

2) Compile another copy of dune where we can do *fun* things.

Before we start

- Things that are preceded with a $ symbol usually mean insert your own name for this e.g. your specific username or whatever you called that directory ($username=larsoft00).
- They could also refer to environmental variables which should already be set.
- The distinction will hopefully be obvious.
- For the setup I've included screenshots of what you should hopefully see so you can be sure you're on the right track.

➔ A blue box like this indicates commands you type in the terminal

- An orange box is just some extra information
- e.g. it might explain some of the commands or arguments you're using

Step 1. Login

Using a VNC:

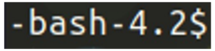
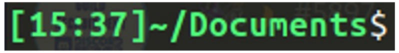
- Go to this link http://py-dom.lancs.ac.uk:8080/guacamole
- Login (with your given username *duneXY* and password)
- Open a terminal

Or ssh directly:

```
➔ ssh $username@py-dom.lancs.ac.uk
```

Optional things to make life easier

- These are based on preferences… so feel free to ignore / do something else.

- Change your terminal prompt. Go from this  `-bash-4.2$` to this  `[15:37]~/Documents$`
- Helpful so we instantly know what directory we're in and don't have to PWD all the time.
- Make a file called ".bash_profile" - note the preceding . (e.g. `vim .bash_profile` ; `s` {paste]; esc `:wq`)
- Put the following lines in your .bash_profile file:

```
export PS1="[\A]\w$ "
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\][\A]\w\[\033[00m\]$ '
```

- And then do `source ~/.bash_profile`

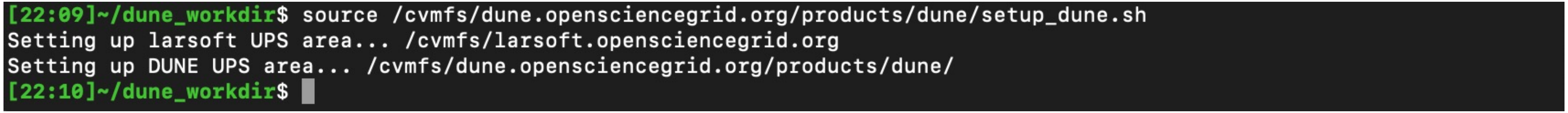
- Copy & paste instructions for the vnc: https://www.mfdemoportal.com/forum/blueshift-tips-and-tricks-staff-user/guacamole-cut-n-paste-functionality

Step 2. Make your larsoft directory and move to it

```
➔ mkdir $dune_workdir
➔ cd $dune_workdir
```

Step 3. Setup sbndocde

- ➔ `source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh`

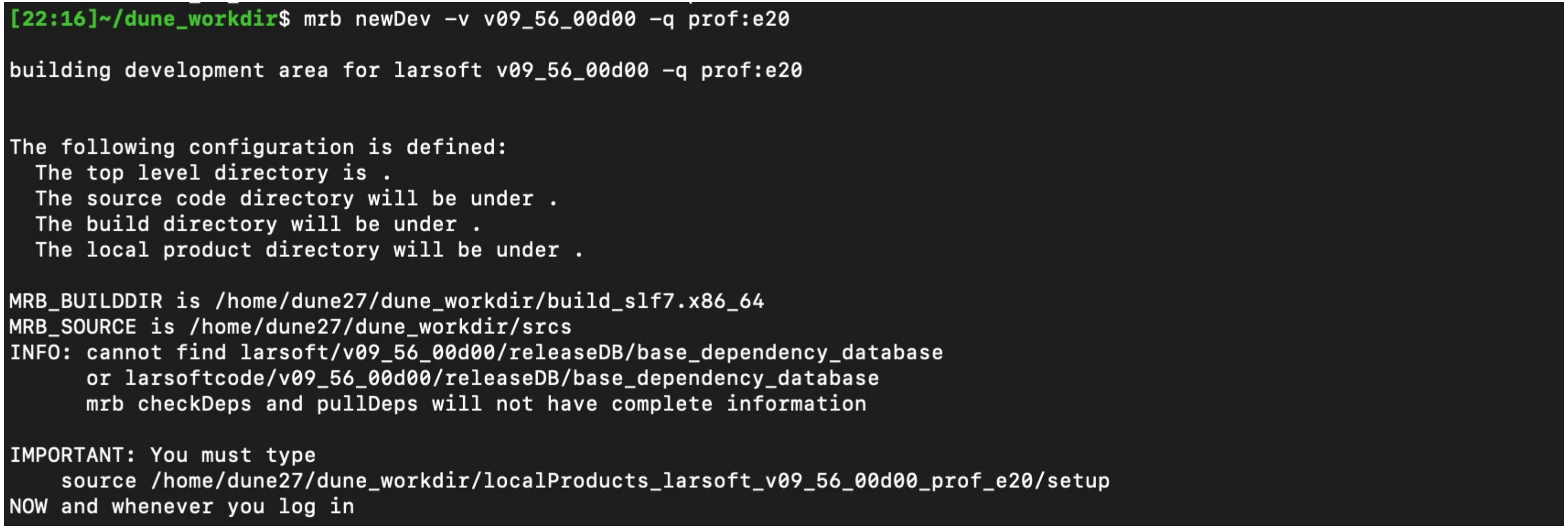


```
[22:09]~/dune_workdir$ source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
Setting up larsoft UPS area... /cvmfs/larsoft.opensciencegrid.org
Setting up DUNE UPS area... /cvmfs/dune.opensciencegrid.org/products/dune/
[22:10]~/dune_workdir$
```

Step 4. Make a new development area

```
➔ mrb newDev -v v09_56_00d00 -q prof:e20
```

- -q; is for qualifiers
- prof; compile with optimisation and limited debugging.
- Could use ‘debug’ instead of ‘prof’ for full debugging.
- e20; Sets the compiler



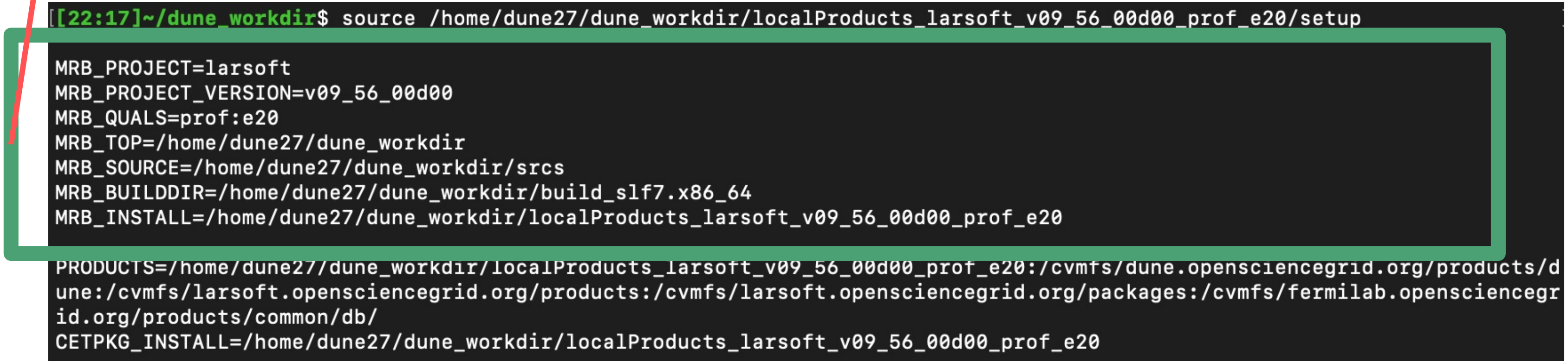
```
[22:16]~/dune_workdir$ mrb newDev -v v09_56_00d00 -q prof:e20

building development area for larsoft v09_56_00d00 -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/dune27/dune_workdir/build_slf7.x86_64
MRB_SOURCE is /home/dune27/dune_workdir/srcs
INFO: cannot find larsoft/v09_56_00d00/releaseDB/base_dependency_database
      or larsoftcode/v09_56_00d00/releaseDB/base_dependency_database
      mrb checkDeps and pullDeps will not have complete information

IMPORTANT: You must type
    source /home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20/setup
NOW and whenever you log in
```

Step 5. Source local products

- Do as you’re told and source the local products.
- We don’t need the whole path since we’re already in the dune_workdir directory.

```
➔ source localProducts_larsoft_v09_56_00d00_prof_e20/setup
```

- These set some variables so if you type e.g. “echo $MRB_PROJECT” it will print “larsoft”.
- Can be useful to help navigate many layers of directories quickly.

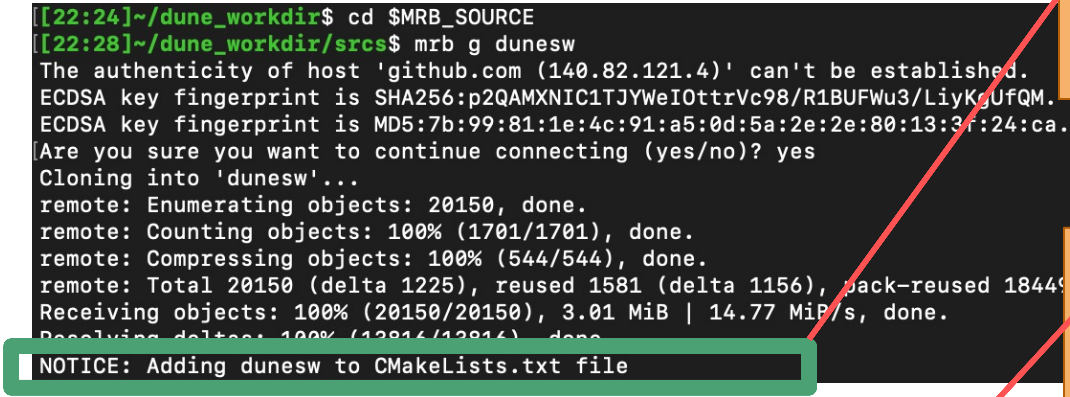


```
[[22:17]~/dune_workdir$ source /home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20/setup

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v09_56_00d00
MRB_QUALS=prof:e20
MRB_TOP=/home/dune27/dune_workdir
MRB_SOURCE=/home/dune27/dune_workdir/srcs
MRB_BUILDDIR=/home/dune27/dune_workdir/build_slf7.x86_64
MRB_INSTALL=/home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20

PRODUCTS=/home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20:/cvmfs/dune.opensciencegrid.org/products/d
une:/cvmfs/larsoft.opensciencegrid.org/products:/cvmfs/larsoft.opensciencegrid.org/packages:/cvmfs/fermilab.opensciencegr
id.org/products/common/db/
CETPKG_INSTALL=/home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20
```

Step 6. Clone dunesw (+others) from github

```
➔ cd $MRB_SOURCE
➔ mrb g dunesw
➔ cd dunesw
```

- May get prompted with some *key fingerprints* Just type “yes” and continue



```
[[22:24]~/dune_workdir$ cd $MRB_SOURCE
[[22:28]~/dune_workdir/srcs$ mrb g dunesw
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
ECDSA key fingerprint is MD5:7b:99:81:1e:4c:91:a5:0d:5a:2e:2e:80:13:3f:24:ca.
[Are you sure you want to continue connecting (yes/no)? yes
Cloning into 'dunesw'...
remote: Enumerating objects: 20150, done.
remote: Counting objects: 100% (1701/1701), done.
remote: Compressing objects: 100% (544/544), done.
remote: Total 20150 (delta 1225), reused 1581 (delta 1156), pack-reused 1844
Receiving objects: 100% (20150/20150), 3.01 MiB | 14.77 MiB/s, done.
Resolving deltas: 100% (13816/13816), done.
NOTICE: Adding dunesw to CMakeLists.txt file
```

- You can clone the repository *normally* with “git clone …”, but the CMakeLists.txt file would not be updated automatically.
- “mrb uc” will update the CMakeLists.txt file if you do things this way.

```
➔ git checkout origin/lalarworkshop2022
  -b lalarworkshop2022
```

- This is the name of the remote dunesw branch we want.
- This will be the name of the branch on our machine (could call it anything, but to avoid confusion let’s give it the same name)

Step 6. Clone dunesw (+others) from github

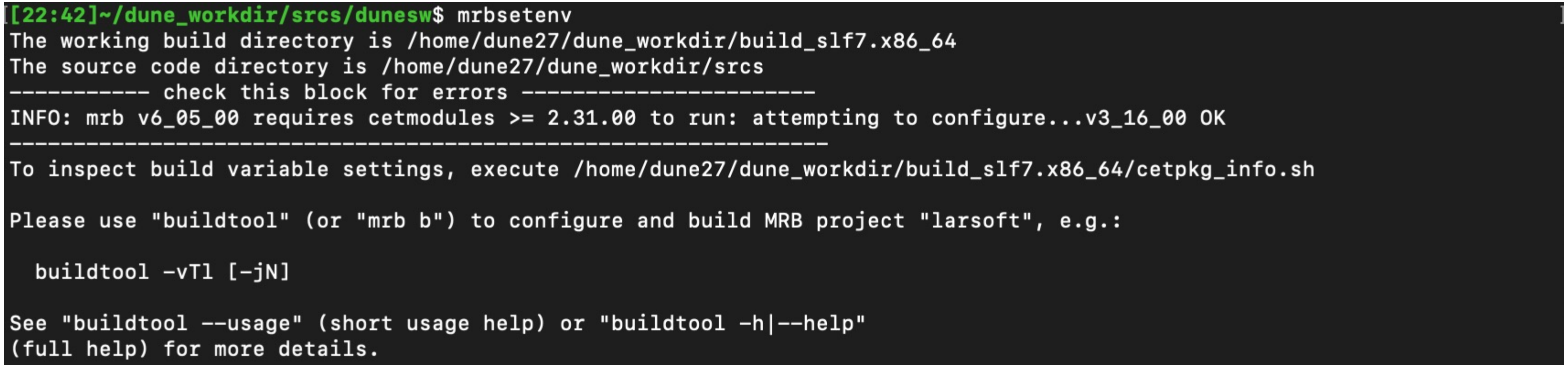
```
➔  cd $MRB_SOURCE
➔  mrb g dunereco
➔  cd dunereco
```

```
➔  git checkout origin/lalarworkshop2022
   -b lalarworkshop2022
```

- And now duneana

Step 7. Setup environment

```
➔ mrbsetenv
```



```
[22:42]~/dune_workdir/srcs/dunesw$ mrbsetenv
The working build directory is /home/dune27/dune_workdir/build_slf7.x86_64
The source code directory is /home/dune27/dune_workdir/srcs
----------- check this block for errors -----------------------
INFO: mrb v6_05_00 requires cetmodules >= 2.31.00 to run: attempting to configure...v3_16_00 OK
----------------------------------------------------------------
To inspect build variable settings, execute /home/dune27/dune_workdir/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.
```

Step 8. Build and Install

```
➔ cd $MRB_BUILDDIR
➔ mrb install  -j3
```

- -j sets the number of threads to use i.e. speeds up the build process.
- The “lscpu” command will give info about how many cores etc you have available.
- Don’t use more than 3 in this case, otherwise we might kill the system with everyone running this at the same time.

- Don’t wait for this to finish, move to the next slide.

Step 8.5. While we wait..

- While we wait for that to finish, let’s start compiling another copy of dunesw (+) where we can play around a little and not worry if we break something.

- We aren’t quite done with the first copy of dunesw(+) however, so don’t forget to come back (step 9) to finish it off.

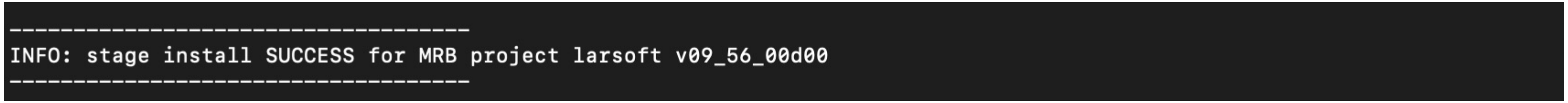
- Open a new terminal (don’t close the current one since it’s still running).

- Starting from step 1, repeat everything again (in step 2, you will need to make a “dunesw_workdir” with a different name.)

- While this 2nd copy of dunesw is compiling let’s finish the first one (hopefully it’s built by now)

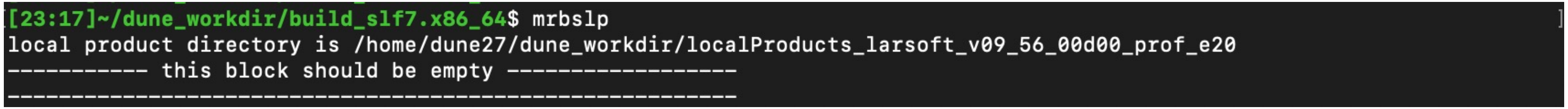
Step 9. Setup installation

- Hopefully it will have installed successfully and you’ll see the following message;



```
------------------------------------
INFO: stage install SUCCESS for MRB project larsoft v09_56_00d00
------------------------------------
```

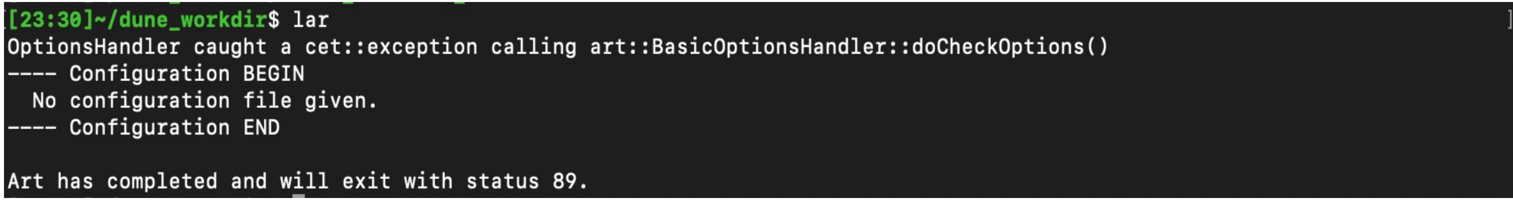
```
➔ mrbslp
```



```
[23:17]~/dune_workdir/build_slf7.x86_64$ mrbslp
local product directory is /home/dune27/dune_workdir/localProducts_larsoft_v09_56_00d00_prof_e20
----------- this block should be empty ------------------
---------------------------------------------------------
```

- Double check things are working (check we have the “lar” command).

```
➔ cd ..
➔ lar
```



```
[23:30]~/dune_workdir$ lar
OptionsHandler caught a cet::exception calling art::BasicOptionsHandler::doCheckOptions()
---- Configuration BEGIN
  No configuration file given.
---- Configuration END

Art has completed and will exit with status 89.
```

Next steps

- Once your first copy of dunesw(+) has been properly compiled (steps 1-9), don’t touch it again for now. You’ll need a working copy for the upcoming sessions.

- We’ll cover a bit more explanation and setup and then look at generating a few events etc. using your second copy of dunesw. Go at your own pace, it doesn’t matter if you don’t get everything done.

Setup now we have compiled code

- We don’t need to compile each time we login.

- Close your terminal and start a fresh one.

- Navigate back to your 2nd working directory. Then, all you need to do is the following.

```
➔ source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
➔ source localProducts_larsoft_v09_56_00d00_prof_e20/setup
➔ mrbsetenv
➔ mrbslp
```

- Even better is if you can make these commands into a shell script.
- Then all you have to do is move to your working directory and source the setup script.

- “mrbslp” - Setup the products installed in the localProducts_xxx directory.
- “mrbsetenv” - Setup the development environment.

Generate some events

```
➔ lar -c prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6.fcl -n 3 -o muons_gen.root
```

- -c specifies the file we want to run
- -n specifies the number of events (in this case 3)
- -o sets the name of the output file.

It doesn’t matter in which directory you generate your events, but it’s probably advisable to generate them outside the srcs/ directory i.e. generate them in your larsoft_workdir2 or if you want to be organised make another directory to hold all your events.

- If you’re a root veteran you may be keen to open muons_gen.root in a TBrowser and have a look around.
- This of course is fine, but these types of files aren’t really for “human consumption”, so don’t expect to see anything too meaningful in there.