This walk-through assumes basic familiarity with the linux operating system command line, a Fermilab computing account with DUNE privileges, and familiarity with ROOT.

The goal of this walk-through is to demonstrate to a new DUNE software user the basic steps to simulating and reconstructing events, creating and analyzing events using a ROOT tree, running the event display, and checking out, compiling, and modifying the dunetpc software repository. For a much more detailed introduction, visit the DUNE-centric Guide to Using LArSoft in git/mrb world.

Confirm access

You will need a computing account with Fermilab to access the DUNE computer resources. Log into a DUNE computing node using your kerberos username and password.

```
> kinit <username>@FNAL.GOV
> ssh -Y <username>@dunegpvmXX.fnal.gov
```
(where XX is any node between 01 and 10)

If you are unable to connect, then follow the steps at Getting a Computer Account at Fermilab.

Log in

Connect to a DUNE virtual machine
```
> kinit <username>@FNAL.GOV
> ssh -Y <username>@dunegpvmXX.fnal.gov
```
(where XX is any node between 01 and 10)

Change directory to your user workspace
```
> cd /dune/app/users/<username>
```
(If this directory does not exist, then create it.)
```
> mkdir /dune/app/users/<username>
```

Create a new directory to work in during this tutorial.
```
> mkdir tutorial-160916
> cd tutorial-160916
```

Set up the DUNE software environment

Now we're ready to set up the various DUNE environment variables (e.g. paths) so that we can use the DUNE implementation of LArSoft.
```
> source /grid/fermiapp/products/dune/setup_dune.sh
```

For this first example, we will just use an existing ("frozen") release of the dunetpc repository. This contains all of the DUNE-specific code within LArSoft; we don't have to check out and compile any code. We will have access to the DUNE-specific software and the parts of the LArSoft framework that it relies upon.

Optional: If you would like to see all of the versions of dunetpc that are available, type

```
> ups list -aK+ dunetpc
```

We will use a frozen release that we know will work for the scope of this tutorial. Let's use the optimized version ("prof") instead of the debug version ("debug") since we're not yet developing and debugging code.

```
> setup dunetpc v06_05_00 -q e10:prof
```

Now all of the larsoft bits that we need are set up and available. We can run larsoft with a FHiCL configuration by typing: `lar -c my_fcl_file.fcl`

Running an example FHiCL file

[To-Do: A good example FHiCL file.]

Stepping through the simulation and reconstruction chain

Once you're environment is set up, you can simulate DUNE events through the following stages:
– Event Generation
– Event Simulation (Geant4)
– Detector Simulation
– Reconstruction
– Analysis Output (AnaTree)

The MC challenge FHiCL files are mostly up-to-date and likely will work "out of the box" for you. These can be found in the dunetpc repository in `/fcl/dunefd/`. We are currently using the central version of this repository, so our environment variable `$DUNETPC_DIR` will point us in the right direction.

```
> ls $DUNETPC_DIR/source/fcl/dunefd/
detsim  g4  gen  mergeana  reco
```

Simulate some single-electron events

The MC challenge FHiCL files are in our path, so we can just specify them outright at the command line. To generate five single-muon events between

100 MeV and 5 GeV sprinkled throughout the DUNE Single-Phase 10kt 1x2x6 geometry (a small section of the far detector TPC volume), do:

```
> lar -c prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6.fcl -n 5
```

This will create the file
`prod_eminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen.root`
which is the ART file with the single-particle generator output. Generally, a histogram file for validation purposes is also created. Here, `single_hist_dune.root` was made but is empty in this example.

Now we can step through the rest of the simulation and reconstruction chain. The parameter
`-c <filename.fcl>` specifies the configuration (fcl) file and `-s <filename.root>`
specifies the source file. Each should be self-explanatory.

```
> lar -c standard_g4_dune10kt_1x2x6.fcl -s prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen.root

> lar -c standard_detsim_dune10kt_1x2x6.fcl -s prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen_g4.root

> lar -c standard_reco_dune10kt_1x2x6.fcl -s prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen_g4_detsim.root

> lar -c standard_ana_dune10kt_1x2x6_hist.fcl -s prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen_g4_detsim_reco.root
```

The final step outputs the file `nue.root` which contains a ROOT tree for basic analysis. (It's a preliminary format for the electron neutrino appearance analysis, hence the name.)

Let's make some histograms!!! What should we draw? Let's do it via the ROOT command line.

```
> root -l nue.root
root [1] TBrowser b
```
(Launch a TBrowser to view the tree's contents.)

```
root [2] nueana->Draw(“hit_summedADC:hit_shwkey”,”hit_shwkey>=0 && event==1”,”COLZ”)
```

Run an event display

We can run the DUNE event display on the reconstruction output. It will display the three wire-plane views for one TPC drift volume at a time.

```
> lar -c evd_dune.fcl -s prod_muminus_0.1-5.0GeV_isotropic_dune10kt_1x2x6_gen_g4_detsim_reco.root
```

To view the reconstructed hits in the TPC, select the “Reconstructed” radio button at the bottom of the event display. In this example, event #4 shows up clearly passing through TPC #4. You can navigate with the “Next ----->” button at the top and the TPC # [   ] box on the left panel.

Check out and build dunetpc

Let's start fresh. Log out and log back in to a dunegpvmXX computer. Then set up the DUNE software environment as before.

```
> source /grid/fermiapp/products/dune/setup_dune.sh
> cd /dune/app/users/<username>/tutorial-160916
```

Now let's create a new directory and set it up for software development.

```
> mrb newDev -f -v v06_05_00 -q e10:debug
```

Then set up the environment variables so that they will point to our local installation.

```
> source localProducts_larsoft_v06_05_00_e10_debug/setup
```

Next, let's check out a copy of the dunetpc repository and compile it.

```
> cd srcs
> mrb g -t v06_05_00 dunetpc
> cd ../build_slf6.x86_64/
> mrbsetenv
> mrb i -j4
```

Note: If you log into dunebuild01.fnal.gov you can use `-j16` for much faster building.
But only build your software there! No running jobs!

```
> cd ../
> mrbslp
```

After you've built the software once, you only need to set up the environment next time you log in:

```
> source /grid/fermiapp/products/dune/setup_dune.sh
> cd tutorial-160916
> source localProducts_larsoft_v06_05_00_e10_debug/setup
> mrbslp
```

Now we're using our own local copy of everything in the dunetpc repository and we can try out changes to the code.