Quick starting point for your work on ePIC development! Below is a quick guide, but you can find more information on landing page

Getting the environment

First you need to get `eic-shell` to set up your working environment:

```language-sh
curl --location https://get.epic-eic.org | bash
```

In order to get a specific version you need to use eg:

```language-sh
curl --location https://get.epic-eic.org | bash -s -- -c jug_xl -v 25.10.1
main-stable
```

If you have access to `CVMFS` (typically on SDCC at BNL) you can list the available container images:

```
/cvmfs/singularity.opensciencegrid.org/eicweb/jug_xl
```

The above options will get you the ePIC container image, with all the software you need to start working. No need to install anything extra!

More info can be found on ePIC tutorial page and eic-shell itself.

Now you just need to run:

```
./eic-shell
```

And set up the basic environment variables:

```language-sh
source /opt/detector/setup.sh
```

The above line sets the default variables for the main branch of epic repository. This means the default version of ePIC libraries and geometry. If you need a specific version then you need to check out a specific branch of the repository and modify it locally.

Compiling epic repository

```language-sh
git clone https://github.com/eic/epic
cd epic
cmake -B build -S . -DCMAKE_INSTALL_PREFIX=install
cmake --build build -j8 -- install
```

```
cd ../
source epic/install/setup.sh
```

You can make changes, recompile and run `setup.sh`. Now the simulations will include the changes you made.

Other links

- New tutorials (in progress)
- Tutorials collection
- Old tutorial

Running simulations with dd4hep

The dd4hep toolkit is a modular package with tool for detector description and simulations used in HEP. With dd4hep you can start detector development.

The `eic-shell` uses `npsim`, which is a clone of original `ddsim`. To check available options type `npsim --help`. To run particle gun, use the example below:

```
npsim --compactFile ${DETECTOR_PATH}/${DETECTOR_CONFIG}.xml --numberOfEvents
${N_EVENTS} --random.seed ${SEED} --enableGun \
--gun.particle proton --gun.thetaMin 170*degree --gun.thetaMax 180*degree --
gun.distribution uniform \
--gun.energy ${ene}*GeV --outputFile output_E${ene}GeV.edm4hep.root
```

The `${DETECTOR_PATH}` should be set by the `setup.sh` script, but you can and just it yourself. The `${DETECTOR_CONFIG}` sets the compact geometry file to use for the simulation and has to be set by hand. You can view the `epic/compact` or `epic/install/share/epic/` for the available detector subsystem configurations. It's an easy way of selecting through multiple setups. Alternatively you can just point to the compact file directly rather than use the environment variables.

More info in running simulations tutorial

Analyzing simulation output

Now you can use ROOT to read the output edm4hep data files (edm4hep GitHub). A good starting point is the example in repository ePICSimDataAnalysisTest

Just get it and run the macro:

```
root -l -b -q readTreeSim.C+ | tee run.log                    language-sh
```

More information can be found in the analyzing simulation output tutorial.

Running reconstruction with eicrecon

To run reconstruction, just call `eicrecon`, set input `$DDSIM_FILE`, set output `${EICRECON_FILE}` and number of events.

```sh
eicrecon $DDSIM_FILE -Ppodio:output_file=${EICRECON_FILE} -jana:nevents=${NUMBER_OF_EVENTS}
```

This will produce all available output collections, however you can specify the ones you want by adding:

```sh
-Ppodio:output_include_collections="MCParticles,HcalEndcapNRawHits,HcalEndcapNRecHits"
```

To list plugins:

```sh
eicrecon --list-default-plugins
eicrecon --list-available-plugins
```

Changes to eicrecon

Get the repository and follow the instructions here.

Analyzing reconstructed data

The analysis of reconstructed data can be done with similar macro as for the simulated data, however the format is different. It uses edm4eic (GitHub repository).

Submitting jobs on SDCC

The BNL SDCC cluster uses condor batch system. Example job description and execution scripts are located here

To run do following steps.

Prepare directories

```sh
./mkdirCondor.sh
```

Submit simulation jobs

```sh
condor_submit submitSim.job | tee sim.log
```

Check the status:

```sh
condor_q
```

Submit reconstruction jobs (remember to adjust input!)

```sh
condor_submit submitReco.job | tee reco.log
```

Selecting a specific container image version

Replace the line in condor job description file:

```sh
+SingularityImage="/cvmfs/singularity.opensciencegrid.org/eicweb/jug_xl:nightly"
```

with the following to select version eg. `23.11-stable`:

```sh
+SingularityImage="/cvmfs/singularity.opensciencegrid.org/eicweb/jug_xl:23.11-stable"
```

To list available image versions:

```sh
ls /cvmfs/singularity.opensciencegrid.org/eicweb/
```

More info

https://htcondor.readthedocs.io/en/lts/admin-manual/singularity-support.html