Delphes EIC Documentation

Release 0.00.01

Miguel Arratia, Stephen Sekula

Nov 24, 2021

GETTING STARTED:

CHAPTER

ONE

INSTALLATION

1.1 EJPM install

ejpm is EIC software centric package/build manager. It is the designed to be the default tool to build G4E on users machine, as it helps with:

- building dependent packages (with "right" compilation flags)
- setup environment variables to run everything
- help with what packages to install by system packet manager
- rebuild/remove/clean existing packages

  Still, ejpm is not a requirement.

First, install ejpm itself:

```
pip install --user ejpm
```

(If you have certificate problems (JLab issue), don't have pip, or have other problems, here is the detailed documentation

Install delphes and possible other dependencies packets (lhapdf, pythia8, fastjet)

```
# 1. System prerequesties (the most prerequesties are for CERN ROOT)
ejpm req centos          # get list of required OS packets. Use `ubuntu` on debian
sudo yum install ...     # install watever 'ejpm req' tells you

# 2. Where to install
ejpm --top-dir=<where-to>   # Directory where packets will be installed

# 4. Install lhapdf, pythia9 and delphes
ejpm install delphes

# 5.  Source environment
source ~/.local/share/ejpm/env.sh  # Use *.csh file for tcsh
```

You have ROOT and Geant4 and don't want EJPM to build them?(Use your installations of ROOT and Geant4)

```
# Before running 'ejpm install g4e'
ejpm set root `$ROOTSYS`     # Path to ROOT installation
ejpm set geant <path>        # Path to Geant4 installation
```

(!) If you use your version of ROOT, all packages depending on ROOT should be installed with the same C++ standard flags as root. So it it was C++11 or C++17, it should be used everywhere. To set it in ejpm`ejpm config global cxx_standard=17`

Hint (!). Run ejpm to overview all installed packets, environment and status by 'ejpm' command

Here is the sample output:

```
> ejpm

EJPM v0.01.19
top dir :
   /eic
state db :
   ~/.local/share/ejpm/db.json  (users are encouraged to inspect/edit it)
env files :
   ~/.local/share/ejpm/env.sh
   ~/.local/share/ejpm/env.csh

INSTALLED PACKETS: (*-active):
vgm:
   * /eic/vgm/vgm-v4-5 (owned)
root:
   * /eic/root/root-v6-16-00 (owned)
geant:
      /eic/geant/geant-v10.5.0 (owned)
   * /eic/geant4-10.6-betta
hepmc:
   * /eic/hepmc/hepmc-HEPMC_02_06_09 (owned)
g4e:
   * /eic/g4e/g4e-dev (owned)
```

1.2 Manual installation

Below, the environment variable ${INSTALLDIR} refers to some folder where you are putting all your EIC fast simulation code (e.g. export INSTALLDIR=${HOME}/EIC/).

1. Install PYTHIA8,
   - http://home.thep.lu.se/~torbjorn/Pythia.html,
   - Download the tarball and unpack it. ,
   - Configure it for local installation in your work area, make, install
   - If you intend to use PDFs besides the default PYTHIA PDF, you need to install LHAPDF6. The configuration instructions below assume you have done this already and want to build PYTHIA with support for LHAPDF

```
./configure --prefix=${INSTALLDIR}/EIC/ --with-lhapdf6=${INSTALLDIR}/EIC/
make -j
make install
```

   - Make sure the work area binary directory is in your PATH: `PATH=${INSTALLDIR}/EIC/bin:${PATH}`,

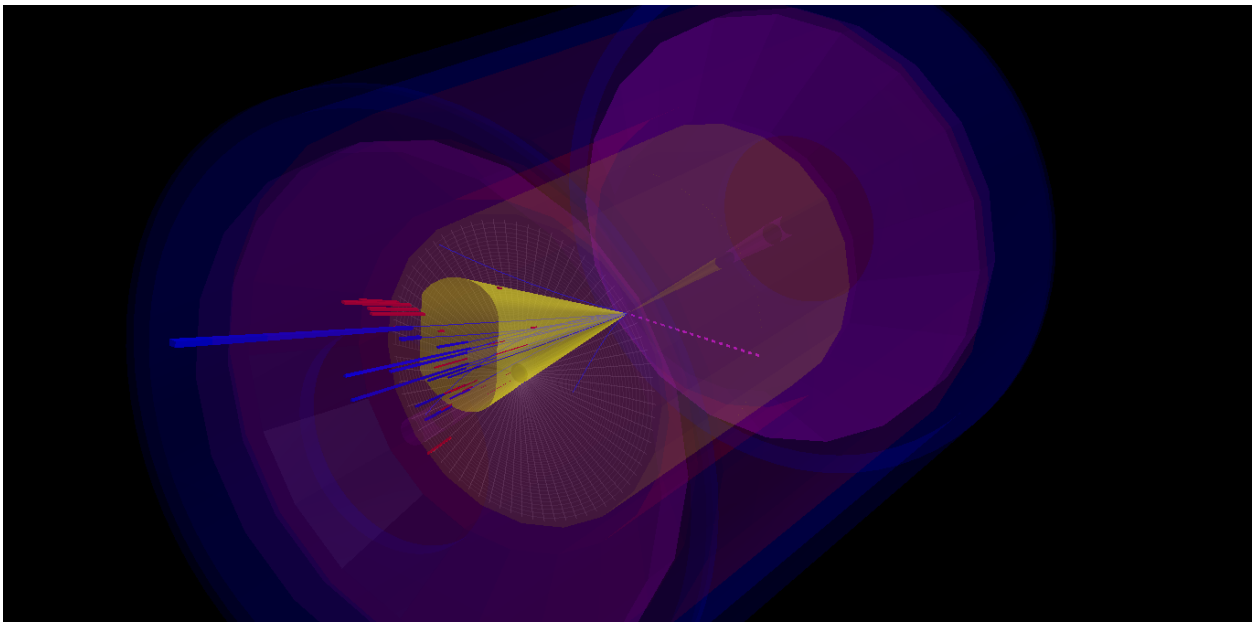
2. Install Delphes,
    - https://github.com/delphes/delphes,
    - Clone the project and make sure you are on the master branch,
    - Make sure ROOT is available in your path, e.g. `lsetup \"root 6.18.04-x86_64-centos7-gcc8-opt\"`,
    - Compile with PYTHIA8: `HAS_PYTHIA8=true PYTHIA8=${INSTALLDIR}/EIC ./configure --prefix=${INSTALLDIR}/EIC/`,
    - Build, install

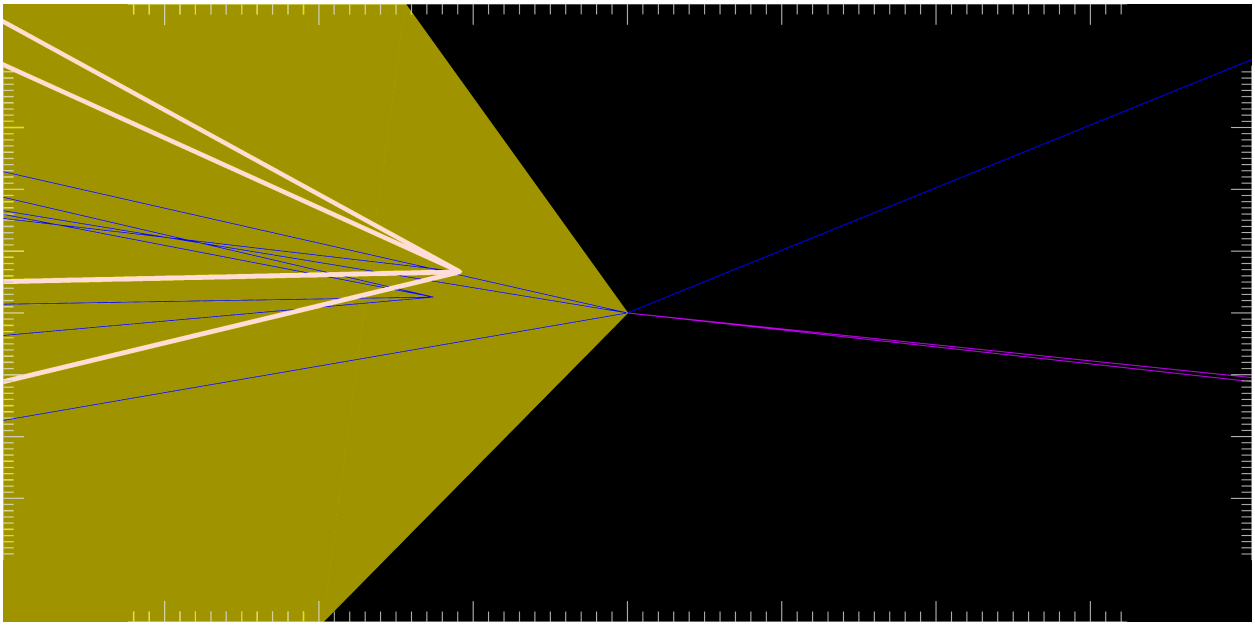
```
make -j
make install
```

3. Get the Delphes/EIC code for simulation and analysis of a detector baseline/configuration.,
    - https://github.com/eic/delphes_EIC,
    - Clone the repository locally,
    - Follow the instructions to run the example and generate a ROOT file.

Welcome to the **Delphes EIC** project. This code aims to deliver a fast-simulation model of baseline or specific detectors for studies to support the Electron-Ion Collider (EIC) project. This code has been used in a few publications and papers and parts of it are available for citation through Zenodo:

- Charm jets as a probe for strangeness at the future Electron-Ion Collider. Miguel Arratia (UC Riverside amd Jefferson Lab), Yulia Furletova (Jefferson Lab), T. J. Hobbs (Southern Methodist U. and Jefferson Lab), Frederick Olness (Southern Methodist U.), Stephen J. Sekula (Southern Methodist U.). https://arxiv.org/abs/2006.12520 [hep-ph]
- A Delphes card for the EIC yellow-report detector. Miguel Arratia (UC Riverside amd Jefferson Lab) and Stephen J. Sekula (Southern Methodist U.). https://arxiv.org/abs/2103.06886 [physics.ins-det]. DOI: https://doi.org/10.5281/zenodo.4592887.
- Science Requirements and Detector Concepts for the Electron-Ion Collider: EIC Yellow Report.R. Abdul Khalek, A. Accardi, J. Adam, D. Adamiak, W. Akers et al. e-Print: https://arxiv.org/abs/2103.05419 [physics.ins-det]





CHAPTER

TWO

WHAT'S NEW?

- EIC PID code has been used to create IdentificationMaps for the mRICH, barrelDIRC, and dualRICH. No more external code needed!

CHAPTER

THREE

INSTALLATION INSTRUCTIONS

1. OPTIONAL (but recommended): Install LHAPDF
2. Install PYTHIA8.3 (if you installed LHAPDF, build with LHAPDF support)
3. Install Delphes3 following: https://github.com/delphes/delphes

CHAPTER

FOUR

EIC YELLOW-REPORT DETECTOR MODELS

The current model we recommend is `delphes_card_allsilicon_3T.tcl`. Is is based on detector studies from the EIC Yellow Report (https://arxiv.org/abs/2103.05419). This model incorporates all-silicon tracking, as well as an EMCAL and HCAL. PID system responses are provided by efficiency maps based on EIC PID code (https://gitlab.com/preghenella/pid/).

- Magnetic field: 1.5 T or 3.0 T
- Solenoid length: 2.0 m
- Tracker radius: 80 cm
- An HCAL and and EMCAL
- Particle ID systems: based on the mRICH, barrel DIRC, and dual RICH concepts articulated by the EIC community; implemented using IdentificationMaps

We currently simulate DIS using Pythia8 within Delphes. The command file (ending in *.cmnd*) shown here and available in the project is suitable for DIS at EIC.

CHAPTER

FIVE

USING THE PROJECT

5.1 Generating Events

We recommend PYTHIA 8.3 for event generation, and the latest version of Delphes builds and links against 8.3 without incident, allowing for seamless generation and then detector response modeling all in one step using the `DelphesPythia8` binary:

```
DelphesPythia8 delphes_card_allsilicon_3T.tcl pythia8cards/CC_DIS.cmnd out.root
```

For examples of other detector configurations, analysis code, etc. see the main Delphes project site.

5.2 Visualizing Events

Run the Delphes visualization script on your ROOT file, using the input Delphes card (TCL) file to help it structure the detector in the display:

```
root -l examples/EventDisplay.C'("delphes_card_allsilicon_3T.tcl","out.root")'
```

The two examples shown here are for neutral-current and charged-current event for beam energies of 10 GeV electron on 100 GeV proton (63 GeV center-of-mass energy).

5.3 EIC Collider Variations

Beam energy recommended benchmarking points are (the order is hadron on lepton):

- 275 on 18 GeV
- 100 on 10 GeV
- 100 on 5 GeV
- 41 on 5 GeV

5.4 The “SimpleAnalysis” framework

See the dedicated SimpleAnalysis Documentation.