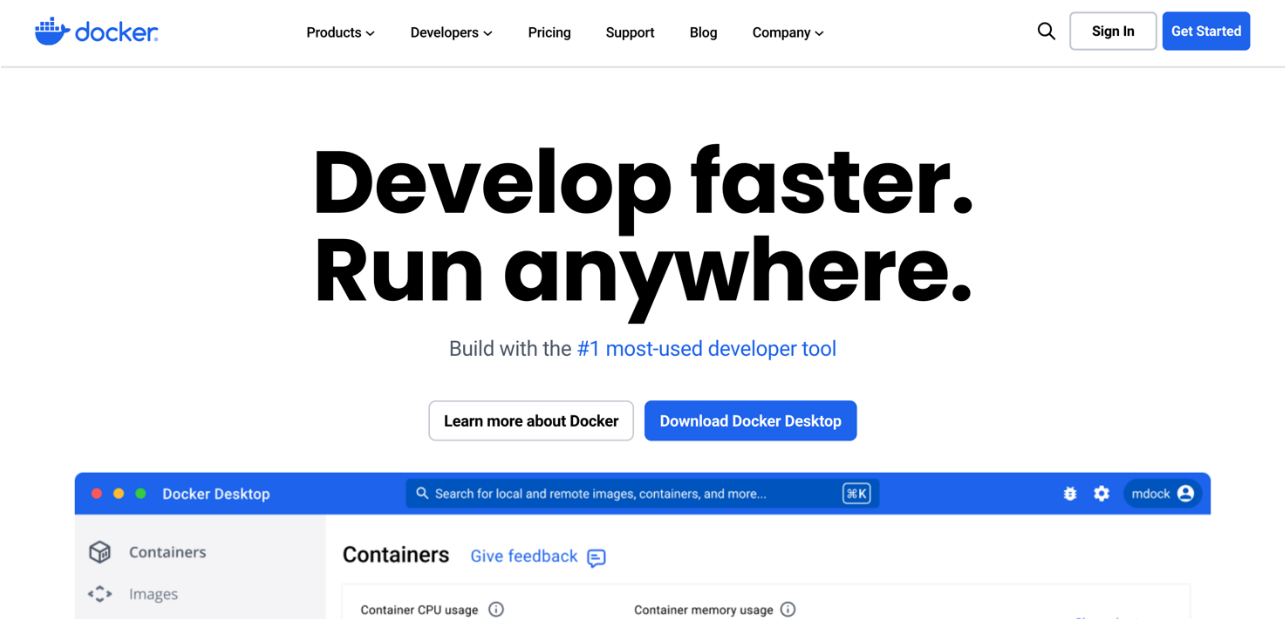
PREPARING FOR THE HANDS ON
Part 2

The Docker container

https://www.docker.com/



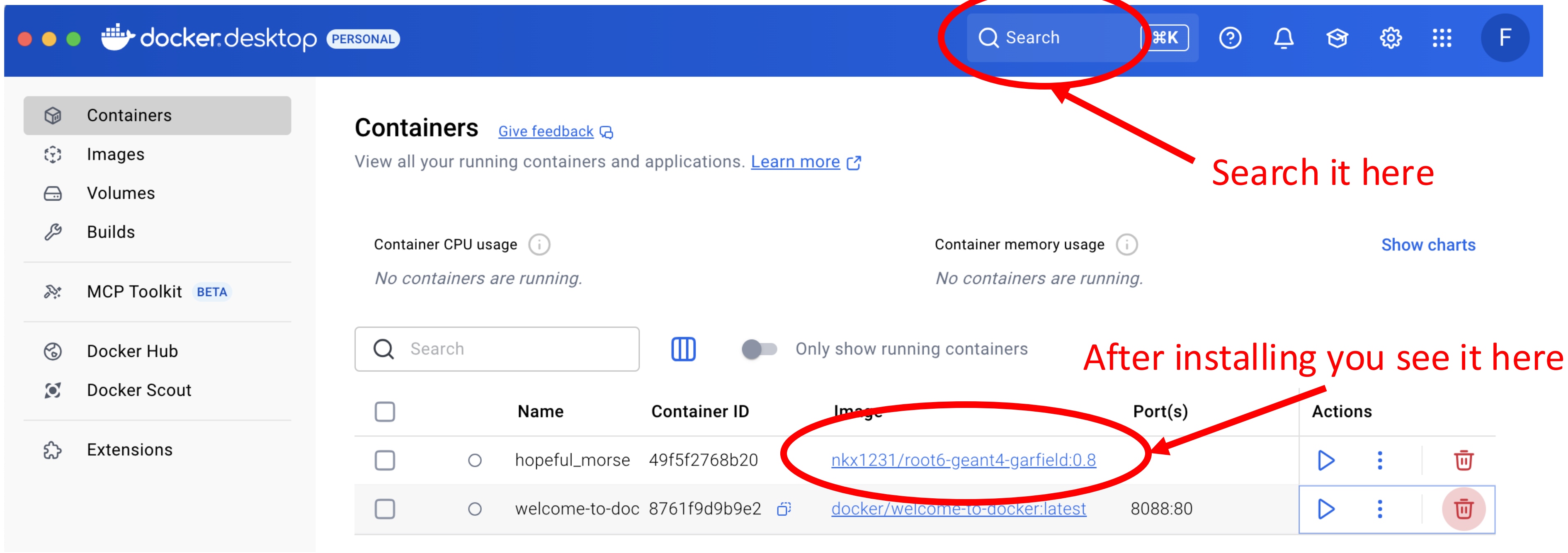
- A Docker is a container of a whole PC environment
- It allows one to run a stand-alone environment with all libraries and software preinstalled

- Download and install Docker desktop
  - Products -> Docker Desktop
- On Unix-like systems (MacOS, Linux,…) enable your **X Window System (X11)**
  - On a terminal run the following
    - `ip=$(ifconfig en0 | grep inet | awk '$1=="inet" {print $2}’)`
    - `echo $ip`
    - `xhost + $ip`
  - You should get a message like:

    $ip being added to access control list
- You should now be able to run the Docker app

Install the image

- Run your Docker app
- Search and load the Image: `nkx1231/root6-geant4-garfield:0.8`



- On unix-based systems, run your Image via prompt terminal:

› `docker run -it -v /tmp/.X11-unix:/tmp/.X11-unix -e DISPLAY=$ip:0 nkx1231/root6-geant4-garfield:0.8 /bin/bash`

Build the GEANT4 example

- You should now enter a virtual machine environment as a “root”, like:

```
[root@ccacff8eefa2:~# pwd
/root
```

- Inside `/root`, create a test directory:
  - `mkdir TestEm2_build`
  - `cd TestEm2_build`
- Build the GEANT4 directory containing the “extended” examples and chose the one named TestEm2:
  - `cmake -DGeant4_DIR=/usr/local/geant4/lib/Geant4-10.4.2 /usr/local/geant4/share/Geant4-10.4.2/examples/extended/electromagnetic/TestEm2`

Compile the GEANT4 example

- You should get a message like:

```
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /root
```

- Compile the extended example TestEm2:

  - `make -j 2 TestEm2`

Compile the GEANT4 example

- You should get a message like:

```
Scanning dependencies of target TestEm2
[  5%] Building CXX object CMakeFiles/TestEm2.dir/TestEm2.cc.o
[ 10%] Building CXX object CMakeFiles/TestEm2.dir/src/SteppingAction.cc.o
[ 15%] Building CXX object CMakeFiles/TestEm2.dir/src/StepMax.cc.o
[ 21%] Building CXX object CMakeFiles/TestEm2.dir/src/PrimaryGeneratorAction.cc.o
[ 26%] Building CXX object CMakeFiles/TestEm2.dir/src/PhysicsList.cc.o
[ 31%] Building CXX object CMakeFiles/TestEm2.dir/src/StepMaxMessenger.cc.o
[ 36%] Building CXX object CMakeFiles/TestEm2.dir/src/EventAction.cc.o
[ 42%] Building CXX object CMakeFiles/TestEm2.dir/src/SteppingVerbose.cc.o
[ 47%] Building CXX object CMakeFiles/TestEm2.dir/src/TrackingAction.cc.o
[ 52%] Building CXX object CMakeFiles/TestEm2.dir/src/PhysicsListMessenger.cc.o
[ 57%] Building CXX object CMakeFiles/TestEm2.dir/src/DetectorMessenger.cc.o
[ 63%] Building CXX object CMakeFiles/TestEm2.dir/src/EmAcceptance.cc.o
[ 68%] Building CXX object CMakeFiles/TestEm2.dir/src/ActionInitialization.cc.o
[ 73%] Building CXX object CMakeFiles/TestEm2.dir/src/DetectorConstruction.cc.o
[ 78%] Building CXX object CMakeFiles/TestEm2.dir/src/PhysListEmStandard.cc.o
[ 84%] Building CXX object CMakeFiles/TestEm2.dir/src/RunActionMessenger.cc.o
[ 89%] Building CXX object CMakeFiles/TestEm2.dir/src/Run.cc.o
[ 94%] Building CXX object CMakeFiles/TestEm2.dir/src/RunAction.cc.o
[100%] Linking CXX executable TestEm2
[100%] Built target TestEm2
```

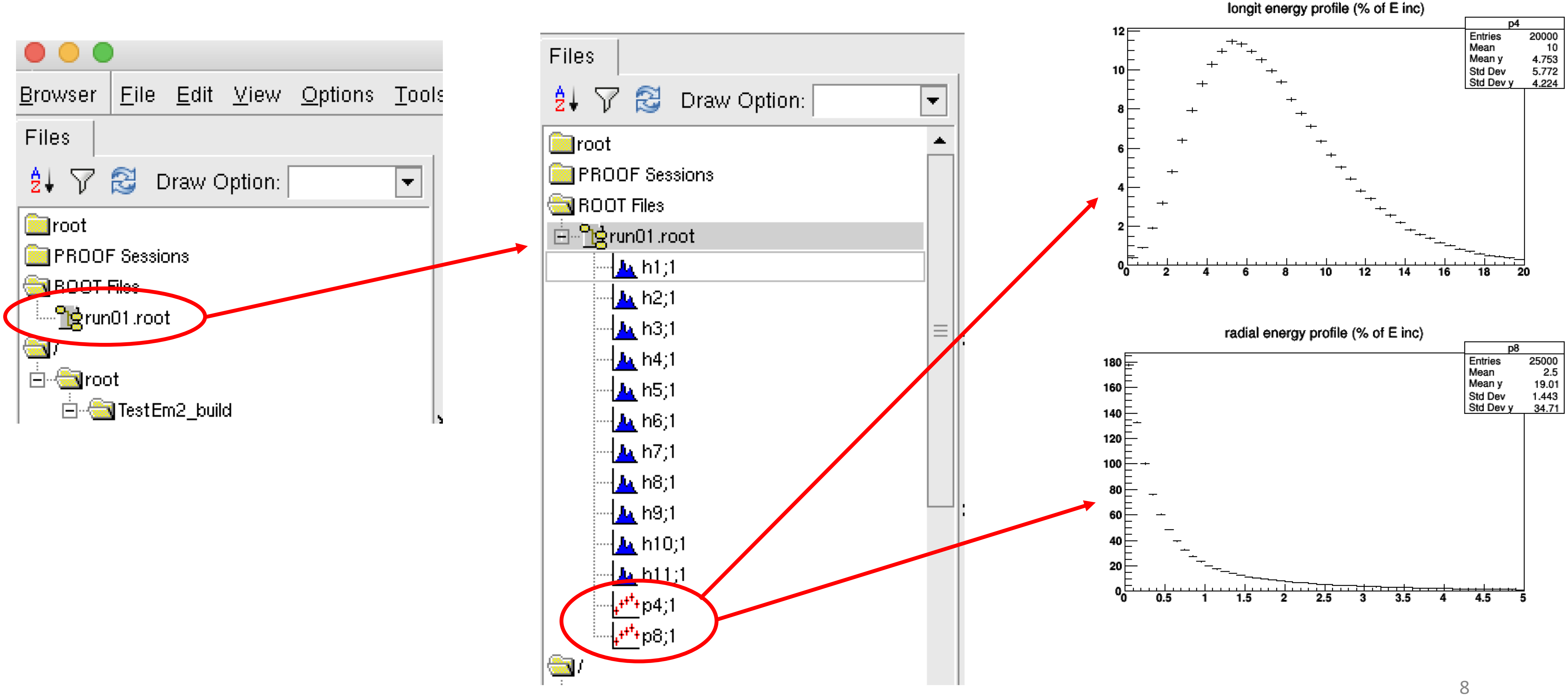
- You are now good to go!
- Remember: when you close the image all work is lost

Run the GEANT4 example

- You will find three examples of run cards:
  - `run01.mac, run02.mac, run03.mac`
- Inspect them and their contents. They simulate and E.M. Cal for different:
  - Energy on an incident electron
  - material
  - …
- Run one configuration. E.g. the first one: `./TestEm2 run01.mac`
- The output is a root ntuple: `run01.root`
- Our image contains the ROOT analysis toolkit. Open the file:
  - `root run01.root`
- On the ROOT prompt, launch a TBrowser app:
  - `new TBrowser`

Run the GEANT4 example

- From the TBrowser app you can now graphically navigate the output file



Get your hands on the code

- Familiarize with the cards
- Familiarize with the code files in:

```
/usr/local/geant4/lib/Geant4-10.4.2
/usr/local/geant4/share/Geant4-10.4.2/examples/extended/electromagnetic/TestEm2/src/
```

- Output files, e.g. root trees, can be downloaded from the docker image to your local PC if you prefer to analyze the in your local environment
- If you just work inside the docker, remember: **DO NOT** close the image, or all the work is lost
  - You can also save the image giving a new version number
- The example used here is **TestEm2** in the extended electromagnetic examples directory

gitlab.cern.ch/geant4/geant4/-/raw/master/examples/extended/electromagnetic/TestEm2/README

TestEm2 – geometry

How to do shower profiles in a homogenous medium

- GEOMETRY DEFINITION
  - The geometry consists of a cylinder of homogenous material
  - The default geometry is constructed in DetectorConstruction class, but all of the above parameters can be modified interactively via the commands defined in the DetectorMessenger class
  - Material can be chosen: Air, Water, lAr, Al, Fe, BGO, PbWO4, Pb, U
  - The cylinder is virtually sliced longitudinally (slice) and radially (ring). The size of the slices and rings are expressed in radiation length units and can be changed. E.g.:
    - /testem/det/setLbin 20 1. ---> 20 slices of 1. radl
    - /testem/det/setRbin 5 0.25 ---> 5 rings of 0.25 radl
    - /testem/det/update ---> rebuild the geometry

    (MaxBin = 500 in both directions)
  - A uniform magnetic field along the cylinder axis can be set. eg:
    - /globalField/setValue 0 0 5 tesla

TestEm2 – physics list

- PHYSICS LISTS - are based on modular design. Several modules are instantiated:
    1. Transportation
    2. EM physics
    3. Decays
    4. StepMax - for step limitation

- EM physics builders can be local (eg. in this example) or from G4 kernel physics_lists subdirectory
- Local physics builders: - "local" standard EM physics with current 'best' options setting. These options are explicited in PhysListEmStandard
- From `geant4/source/physics_lists/builders`:
    - "emstandard_opt0" recommended standard EM physics for LHC
    - "emstandard_opt1" best CPU performance standard physics for LHC
    - "emstandard_opt2" similar fast simulation
    - "emstandard_opt3" best standard EM options - analog to "local" above
    - "emstandard_opt4" best current advanced EM options standard + lowenergy
    - "emstandardWVI" standard EM physics and WentzelVI multiple scattering
    - "emstandardSS" standard EM physics and single scattering model
    - "emlivermore" low-energy EM physics using Livermore data
    - "empenelope" low-energy EM physics implementing Penelope models
    - "emlowenergy" low-energy EM physics implementing experimental low-energy models

TestEm2 – Primary Generator

AN EVENT: THE PRIMARY GENERATOR

- The primary kinematic consists of a single particle which hits the cylinder perpendicular to the input face
- The type of the particle and its energy are set in the PrimaryGeneratorAction class and can changed via the G4 build-in commands of G4ParticleGun class (see the macros provided with this example)
- A RUN is a set of events

VISUALIZATION

- The Visualization Manager is set in the main() (see TestEm2.cc).
- The initialisation of the drawing is done via the commands `/vis/..` in the macro `vis.mac`
- In interactive session:
  - `PreInit` or `Idle > /control/execute vis.mac`
  - The detector has a default view which is a longitudinal view of the cylinder
  - The tracks are drawn at the end of event and erased at the end of run
    - Optionally one can choose to draw all particles, only the charged one, or none. This command is defined in `EventActionMessenger` class

TestEm2 – How to run?

PHYSICS DEMO

- The particle's type and the physics processes which will be available in this example are set in `PhysicsList` class
- In addition, a build-in interactive command (`/process/inactivate procname`) allows to activate/inactivate the processes one by one
- The threshold for producing secondaries can be changed
  - E.g.: `/testem/phys/setCuts 100 microm /run/initialize`
- The shower profiles are histogramed, if histograming is activated. They can be also printed with the command: `/testem/run/verbose 1`

HOW TO START?

- Execute TestEm2 in 'batch' mode from macro files. `TestEm2 run01.mac`
- Execute TestEm2 in 'batch' mode using multi-threading. `TestEm2 run01.mac 4` (4 is # of threads)
- Execute TestEm2 in 'interactive mode' with visualization.
  - `TestEm2`
  - `....`
  - `Idle>` type your commands
  - `....`
  - `Idle> exit`

TestEm2 – macros (*cards*)

- Macros provided in this example:
  - **egs4.mac:**
    - Fe; L = 20 radl; R = 5 radl; electron 30 GeV (EGS4 simulation: Particle Data Group - Phys.Rev.D 50-3 - August94)
  - **run01.mac:** PbWO4; L = 20 radl; R = 5 radl; electron 5 GeV
  - **run02.mac:** Al; L = 13.5 radl; R = 1.35 radl; electron 1 GeV (Electron-induced cascade showers: J&H Crannel - Phys. Rev. 184-2 - August69)
  - **run03.mac:** H2O; L = 9.97 radl; R = 0.665 radl; electron 1 GeV (Electron-induced cascade showers: J&H Crannel - Phys. Rev. 184-2 - August69)
  - **test.mac:** PbWO4; L = 20 radl; R = 5 radl; electron 5 GeV
  - **stepMax.mac:** chargedgeantino in PbWO4; To illustrate step max mechanism;
  - **vis.mac:** to activate visualization

TestEm2 – Histograms

HISTOGRAMS

- TestEm2 produces several histograms. Content of these histo:
    1. energy deposit per event
    2. charged track length per event
    3. neutral track length per event
    4. longitudinal energy profile
    5. rms of longitudinal energy profile
    6. cumulated longitudinal energy profile
    7. rms of cumulated longitudinal energy profile
    8. radial energy profile
    9. rms of radial energy profile
    10. cumulated radial energy profile
    11. rms of cumulated radial energy profile
- Define the output file name with histograms via UI command : "`/analysis/setFileName name`”
- The histogram file format can be: **root (default),** xml, csv, by selecting `g4nnn.hh` in `RunAction.hh`

Example cards: run01.mac

```
# $Id: run01.mac 83431 2014-08-21 15:49:56Z gcosmo $
#
# Macro file for "TestEm2.cc"
# (can be run in batch, without graphic)
#
# PbWO4 L = 20 radl  R = 5 radl; electron 5 GeV
#
/control/verbose 2
/run/verbose 2
#
/testem/det/setMat G4_PbWO4
/testem/det/setLbin 40 0.5
/testem/det/setRbin 50 0.1
#
/testem/phys/addPhysics  emstandard_opt0
#
/run/setCut 1 mm
#
/run/initialize
#
/gun/particle e-
/gun/energy 5 GeV
#
# histogram file
/testem/histo/setFileName run01
#
/run/printProgress 50
/testem/run/verbose 1
#
/run/beamOn 500
```

The example describes an e.m. calorimeter with the following setup:

- A homogeneous medium of cylindrical shape
  - length has been fixed to 20 radL
  - radius value corresponds to 5 radL
- Material: PbWO4
- The cylinder has been divided into 100 slices longitudinally (0.2 radL each) and into 50 rings radially (0.1 radL each)
- 500 electrons with beam energy fixed at 5 GeV, entering the detector along its main axis
- G4 Physics List: emstandard_opt0

Example of a ROOT macro

```
{
    gStyle->SetOptTitle(0);
    gStyle->SetOptStat(0000);

    TFile* f1 = new TFile("run01-1gev.root");
    TFile* f2 = new TFile("run01-10gev.root");
    TFile* f3 = new TFile("run01-100gev.root");

    f1->cd();
    TH1D* h_0 = p4->ProjectionX();
    h_0->GetYaxis()->SetTitle("Energy");
    h_0->GetXaxis()->SetTitle("L (X0)");
    h_0->SetMarkerSize(0.8);
    h_0->SetMarkerStyle(20);
    h_0->SetMarkerColor(2);
    h_0->Draw();

    f2->cd();
    TH1D* h_1 = p4->ProjectionX();
    h_1->SetMarkerSize(0.8);
    h_1->SetMarkerStyle(20);
    h_1->SetMarkerColor(3);
    h_1->Draw("same");

    f3->cd();
    TH1D* h_2 = p4->ProjectionX();
    h_2->SetMarkerSize(0.8);
    h_2->SetMarkerStyle(20);
    h_2->SetMarkerColor(4);
    h_2->Draw("same");

    TLegend *legend = new TLegend(0.5616046,0.7052632,0.8925501,0.8905263,NULL,"brNDC");
    legend->AddEntry(h_0,"Ee = 1 GeV","p");
    legend->AddEntry(h_1,"Ee = 10 GeV","p");
    legend->AddEntry(h_2,"Ee = 100 GeV","p");
    legend->Draw();
}
```

Possible studies

- longitudinal energy profile – at different particle energy, measure:
  - Energy fraction (%) vs depth in radiation length (X_0)
- radial energy profile – for different materials, measure:
  - energy fraction (%) vs depth radiation length (X_0)
- Vary the physics model
- Example of possible particles: electron, photon
- Example of possible energies: 0.1 MeV, 1 MeV, 1 GeV, 10 GeV, 100 GeV, 1 TeV
- Example of possible materials:
  - lead tungstate PbWO4, a scintillating crystal with X0 = 0.89 cm RM = 2.2 cm, characterised by a quick response
  - bismuth germanate Bi4Ge3O12
  - liquid Argon
  - liquid Xenon
  - liquid Kripton