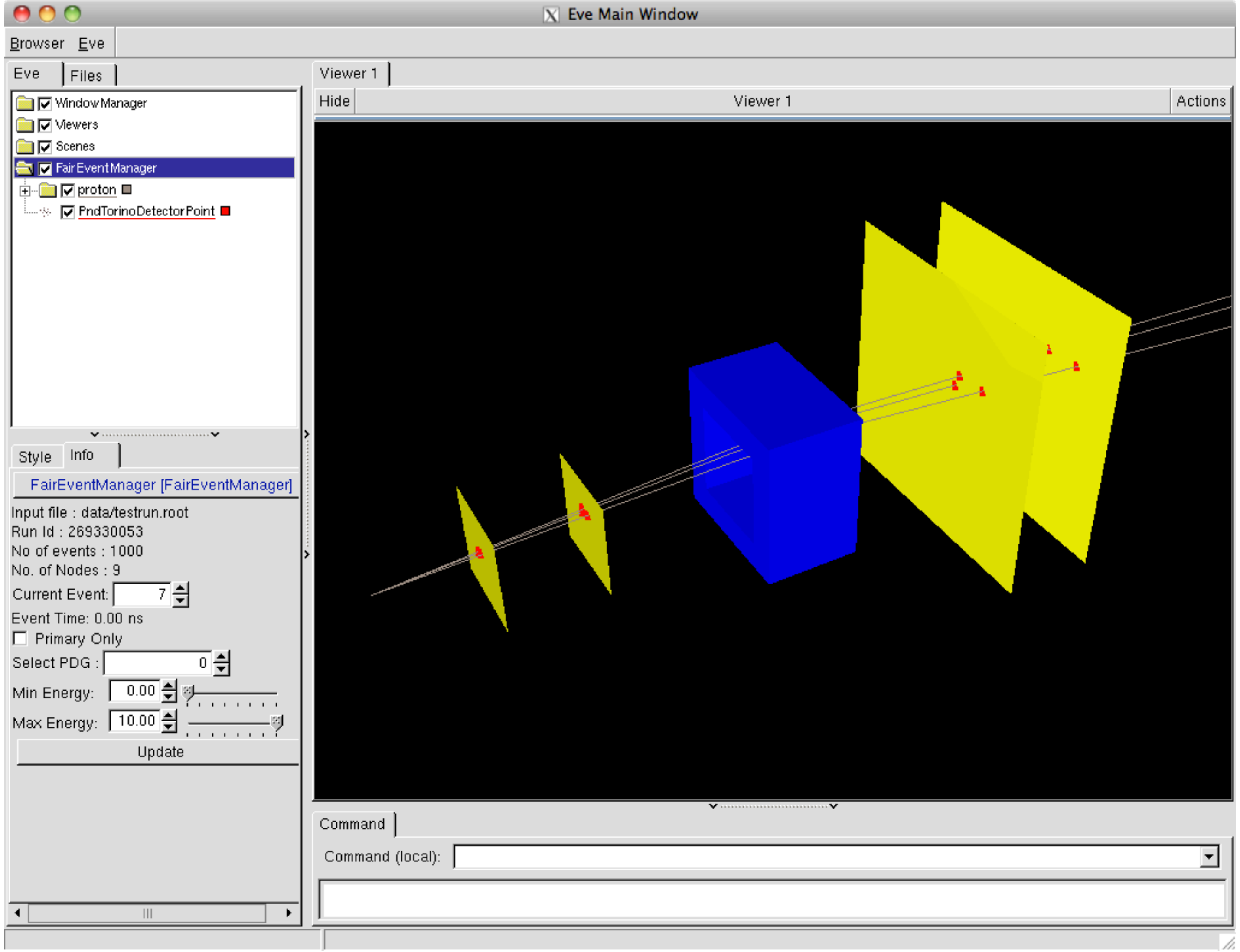
FAIRROOT SIMULATION

Panda Computing Week 2012, Torino

Example Setup (Torino Detector)



Building Blocks

- Material definitions
- Geometry
- Module/Detector class
  - Can be passive (FairModule) or active (FairDetector)
- Event generator
- Runtime Database???
- Magnetic field (optional)

Definitions

- Material
  - Represent the physical properties (mass, charge, density, radiation length, absorption length) of an element or a mixture
- Media
  - Material with tracking parameters needed for the transport (sensitivity flag, field flag, max field value ....)
  - Several media per material possible
- Volume
  - Building block of the geometry with two properties
    - Shape (Box, Tube ...)
    - Medium (needed for transport)
- Node
  - A node represent a volume positioned inside another
  - There is one top volume which has no position and contain all other nodes

Material Definition

Name **Number of components** **A** **Z** **Density** **Relative weights**

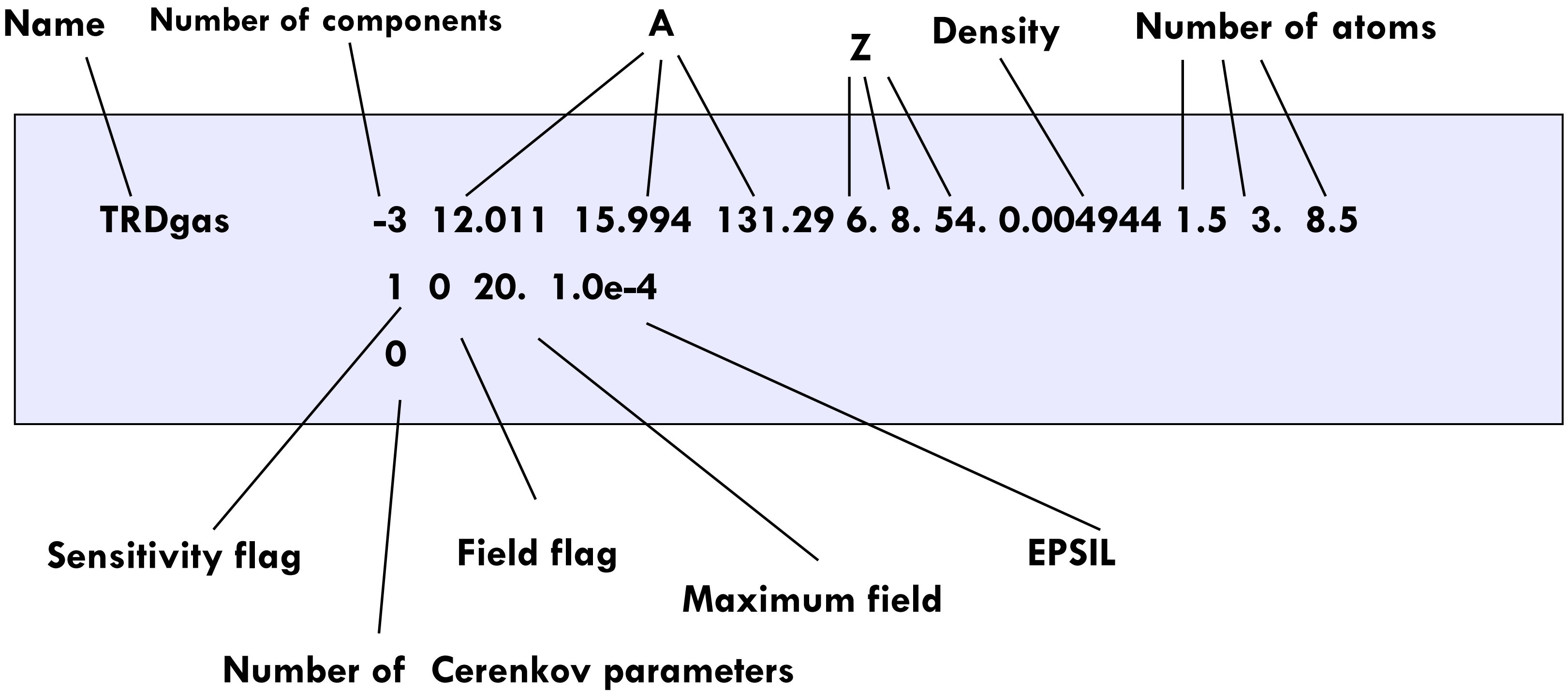
```
carbon          1  12.011  6.0  2.265
                0  1  30.  .001
                0
air             3  14.01  16.  39.95  7.  8.  18.  1.205e-3  .78 .21  .01
                0  1  20.  .001
                0
```

Sensitivity flag **Field flag** **Maximum field** **EPSIL**

Number of Cerenkov parameters

see http://fairroot.gsi.de/?q=node/34 for more details

Material Definition



see http://fairroot.gsi.de/?q=node/34 for more details

Material Definition

Name | Number of components | A | Z | Density | proportion by number of atoms

```
CsI      -2     132.9054  126.9045     55.    53.    4.53     1      1
          1   1   20.    .00001
          2
         1.77        50000.     1.0   1.0003
         10.5        50000.     1.0   1.0003
```

Number of Cerenkov parameters

photon momentum in eV

absorption length in case of dielectric and absorption probabilities in case of a metal

detection efficiency

refraction index for a dielectric, rindex[0]=0 for a metal

Material Definition

The following parameters are normally not read. The default values are -1 and the real values are automatically calculated by Geant. If you want to set these values by yourself, you must type the keyword AUTONULL in your media file. After this keyword all media must contain these additional 4 parameters at the end.

- **float madfld**
  - maximum angular deviation TMAXFD due to field
- **float maxstep**
  - maximum step permitted STEMAX
- **float maxde**
  - maximum fractional energy loss DEEMAX
- **float minstep**
  - minimum value for step STMIN

Hands On: Media Definition

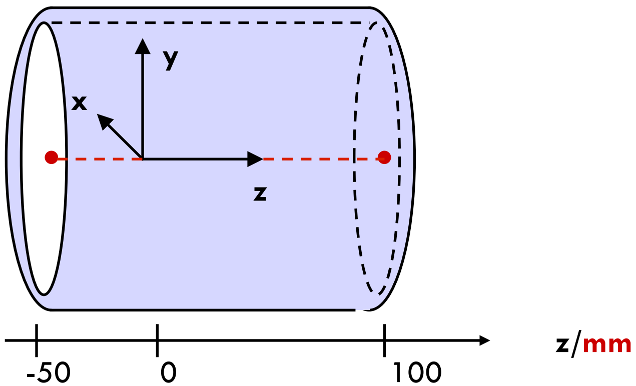
- Please create a file “media.geo” with the following materials.
  - Air (http://en.wikipedia.org/wiki/Air)
  - Silicon (http://en.wikipedia.org/wiki/Silicon)
  - Iron (http://en.wikipedia.org/wiki/Iron)

0/1 1 20. 0.001

see http://fairroot.gsi.de/?q=node/34 for more details

ASCII geometry definition (Hades)

- The position of the local coordinate system of a volume must **NOT** be the geometrical center.
- The positioning of the daughter is independent of the size and shape of the mother.
- The transformation describes the position of the local coordinate system of the daughter relative to the coordinate system of the mother.

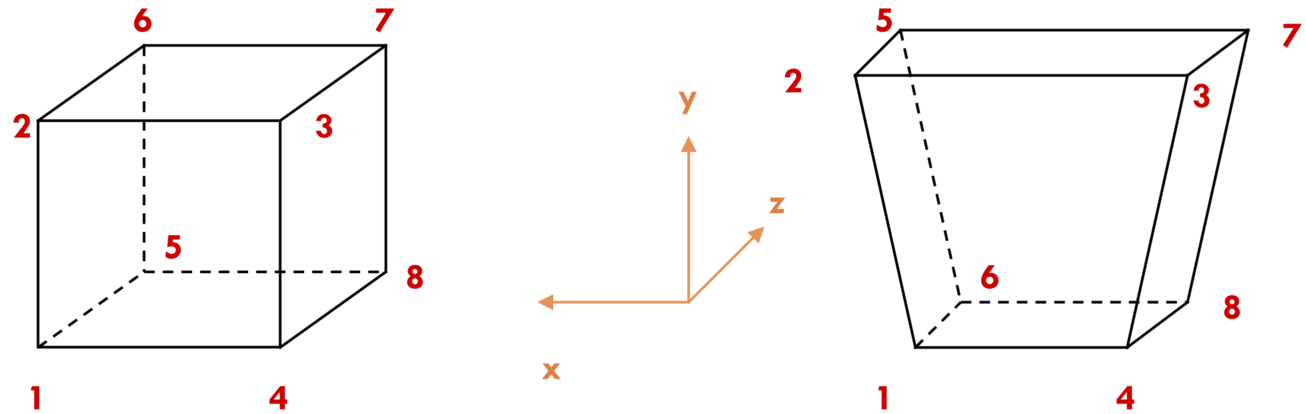



```
pipe_1
cave
TUBE
carbon
  0.  0.  -50.
  0.  5.5
  0.  0.  100.
0.  0.  0.
1.  0.  0.  0.  1.  0.  0.  0.  1.
// ***********************************************
```

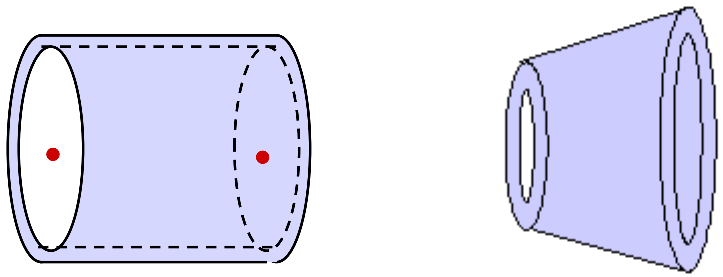
```
pipe_vac_1
pipe_1
TUBE
vacuum
  0.  0.  -50.
  0.  5.
  0.  0.  100.
0.  0.  0.
1.  0.  0.  0.  1.  0.  0.  0.  1.
// ***********************************************
target
pipe_vac_1
TUBE
gold
 0. 0. -0.25
 0. 2.5
 0. 0.  0.25
0.  0.  0.
1.  0.  0.  0.  1.  0.  0.  0.  1.
```

Geometry description (Hades)

BOX, TRAP, TRD1 described by the 8 corners



TUBE, TUBS, CONE, CONS center of front and back circle, inner and other radius



+ phi section

Advantage: more flexibility, closer to technical drawings and analysis coordinate systems

Disadvantage: more points then really needed (more typing)

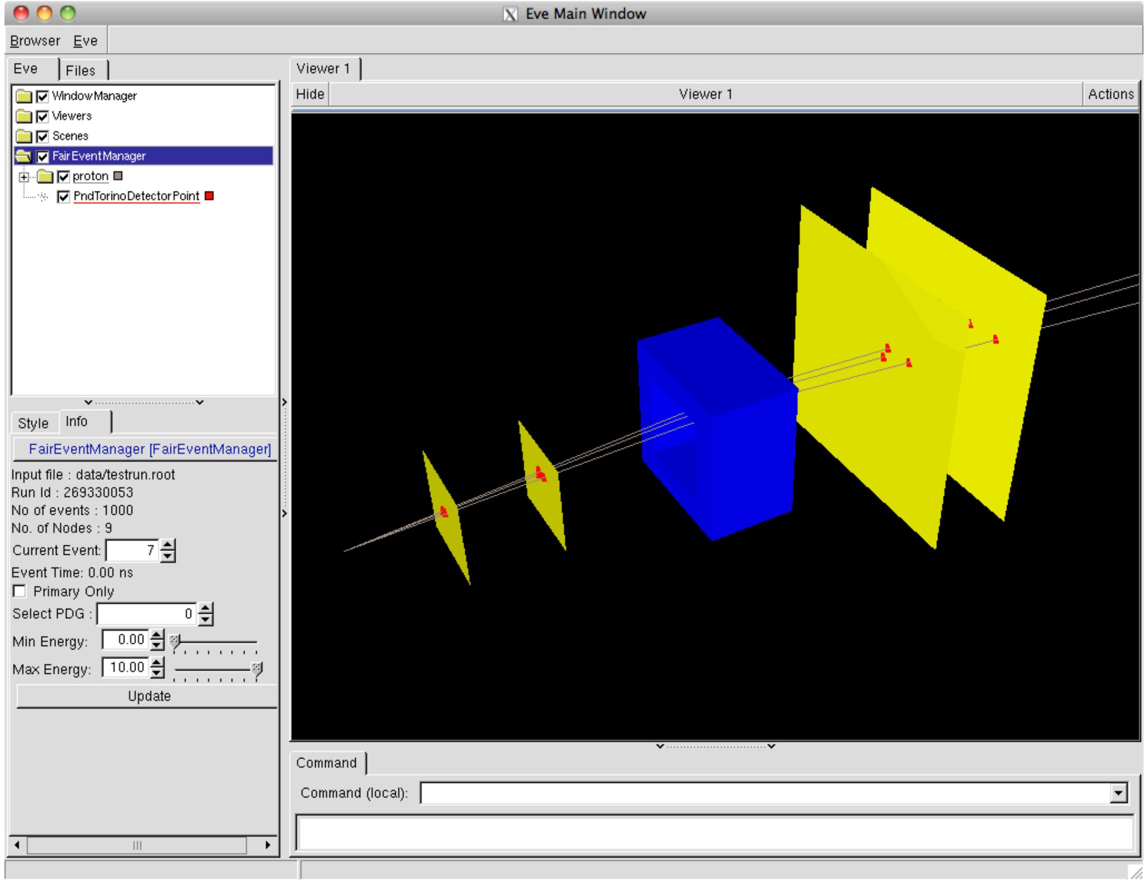
More Info: http://fairroot.gsi.de/?q=node/30

Geometry description (Hades)

```
pipe1                          //Volume Name
cave                           //Mother Volume
PCON                           //Shape
carbon                         //Material
5.
0. 360.
-50.    0.  25.5
25.     0.  25.5                   //Shape parameters
200.    0.  110.5
300.    0.  15.5
1700. 0.  85.5
0.  0.  0.                         //translation relative to mother volume
1. 0. 0. 0. 1. 0. 0. 0. 1.         //rotation relative to mother volume
```

Example Setup (Torino Detector)

- Two layers of silicon detectors before a simple dipole magnet, and two after the dipole magnet



Hands On: Geometry Definition

- Cave: Box filled with Air (20m x 20m x 20m) (cave.geo)
- Detector: 4 Boxes of Silicon (detector.geo)
  - 2 before the magnet (0.5m x 0.5m x 400mu) at
    - 1m and 2m from the origin
  - 2 after the magnet (1.5m x 1.5m x 400mu) at
    - 6m and 7m from the origin
- Magnet: 4 Boxes of Iron to build a window frame magnet with the following dimensions (magnet.geo)
  - Inner size: 1m x 1m
  - Outer size: 1.6m x 1.6m
  - Length: 1m
- **More Info: http://fairroot.gsi.de/?q=node/30**

Creating Modules (passive Materials)

- The basic building block of the geometry is a Module which is responsible to construct the geometry from an input

```
ifndef PIPE_H
#define PIPE_H

#include "FairModule.h"

class CbmPipe : public FairModule {
  public:
    CbmPipe(const char * name, const char *Title="CBM Pipe");
    CbmPipe();

    virtual ~CbmPipe();
    virtual void ConstructGeometry();

  ClassDef(CbmPipe,1) //CBMPIPE

};

#endif //PIPE_H
```

Geometry construction from ASCII

```
void CbmPipe::ConstructGeometry(){
  FairGeoLoader *loader=FairGeoLoader::Instance();  // Get the geometry loader
  FairGeoInterface *GeoInterface =loader->getGeoInterface(); // Get the interface
  CbmGeoPipe *MGeo=new CbmGeoPipe();  // Create the geometry rule class
  MGeo->setGeomFile(GetGeometryFileName()); // Connect the ASCII file
  GeoInterface->addGeoModule(MGeo);  // Add the new module to the interface
  Bool_t rc = GeoInterface->readSet(MGeo);  // Read the ASCII file
  if ( rc ) MGeo->create(loader->getGeoBuilder());  // Create the geometrie
  TList* volList = MGeo->getListOfVolumes();
  ProcessNodes( volList ); // Create the geo-parameters for this detector and save it to
                           // the FairBaseParSet container, also calculate the lab
                           // transform for these nodes
}
```

Creating a detector

- The Detector class is a sub class of the module which implements extra functions which are called from the event loop of the MC to make some actions during simulation
- Creating and adding a detector is the same as creating a Module except that one has to inherit from FairDetector, and one has to implement some extra functions.

Methods Needed for a detector:


```
void Initialize()        // called during framework initialization
void ConstructGeometry() // create the geometry
Bool_t ProcessHits( CbmVolume *v=0) called for each step in detector from MC
void Register() // Registers the produced collections to the I/O Manager
TClonesArray* GetCollection(Int_t iColl) const  // Get a collections ????obsolete
void Reset() // has to be called after each event to reset the containers
```

Methods that can implemented if Needed for a detector

void ConstructOpGeometry() // To set the optical properties of media

void SetSpecialPhysicsCuts() // Can be used to set different physics cuts, switch on and off certain processes for this detector only

// Following methods will be called from the MC Application during simulation (if implimented)

void EndOfEvent()

void FinishPrimary()

void FinishRun()

void BeginPrimary()

void PostTrack()

void PreTrack()

void BeginEvent()

Minimal set of files for new detector

- For most files one can use templates
- In templates/NewDetector
- To be adapted after running rename.sh
  - CMakeLists.txt
  - NewDetectorGeo.h NewDetectorGeo.cxx
- Use as is
  - NewDetector.h NewDetector.cxx
  - NewDetectorPoint.h NewDetectorPoint.cxx
  - NewDetectorContFact.h NewDetectorContFact.cxx
  - NewDetectorGeoPar.h NewDetectorGeoPar.cxx
  - NewDetectorLinkDef.h

Hands On: Use detector template

- Add the template directory to the list of external directories and get the changes from the repository
  - svn propedit svn:externals . (In the main directory)
    - Add the following line
    - templates https://subversion.gsi.de/fairroot/fairbase/release/templates
  - svn update
- Copy the detector template
  - cp –r templates/NewDetector torinoDetector
- Rename the detector classes, variables, etc
  - cd <detector subdirectory>
  - ./rename.sh <detector name> <project name> <project prefix>
  - ./rename.sh (without parameter show help message)
  - ./rename.sh PndTorinoDetector PandaRoot Pnd
  - Change kPndTorinoDetector -> kTPC (only needed to avoid complete recompilation)

Hands on: Use detector template cont.

- Create the new detector PndTorinoDetector
- Change files according to the messages from the rename script
- Try to compile the new subdirectory/detector
  - Add the new subdirectory in the main CMakeLists.txt
- Create a directory in the development branch of the pandaroot repository
  - You need an SVN account

Hands On: SVN usage

- For write access one needs an SVN account
  - If you have one: Good
  - If you don't have one: Please follow the instructions at http://fairroot.gsi.de/?q=node/36 and send the file to f.uhlig@gsi.de
- Browse the repository using the web frontend at https://subversion.gsi.de/trac/fairroot/browser
- Browse the repository from the command line
  - svn ls https://subversion.gsi.de/fairroot/pandaroot
- Get help about SVN on the command line
  - svn help
  - svn help <command>

Hands On: SVN usage cont.

- Create a personal directory in the development branch of pandaroot
  - svn mkdir –parents https://subversion.gsi.de/fairroot/pandaroot/development/<your svn user name>/Torino/
- Import your new directory into the repository
  - svn import <directory> https://subversion.gsi.de/fairroot/pandaroot/development/<your svn user name>/Torino/version1 –m"Initial version."
- Browse the new directory either on the command line or with the web frontend

Running a Simulation

- A simulation session is controlled by a root macro
- After creating the shared libraries needed for simulation they have to be loaded into a root session which will steer the simulation

Simulation Macro: Define Variables


```
// Use non default gconfig and geometry directories
TString dir = getenv("VMCWORKDIR");
TString tutdir = dir + "/macro/torinoDetector";

TString tut_geomdir = tutdir + "/geometry";
gSystem->Setenv("GEOMPATH",tut_geomdir.Data());

TString tut_configdir = tutdir + "/gconfig";
gSystem->Setenv("CONFIG_DIR",tut_configdir.Data());

TString rootLogonFile = tut_configdir + "/rootlogon.C";
```

Simulation Macro – loading Libs


```
// Load basic libraries
gROOT->LoadMacro("$VMCWORKDIR/gconfig/basiclibs.C");
basiclibs(); // This will load thr ROOT and VMC libraries

// Load CbmRoot libraries
 gSystem->Load("libFairTools");
 gSystem->Load("libGeoBase");
 gSystem->Load("libParBase");
 gSystem->Load("libBase");
 gSystem->Load("libCbmBase");
 …….

 // Load basic libraries (Panda)
gROOT->LoadMacro(rootLogonFile);
rootlogon();
gSystem->Load("libPndTorinoDetector.so");
```

Simulation Macro


```
//  Create a simulation run
 FairRunSim* fRun = new FairRunSim();
// Set MC engine
 fRun->SetName("TGeant3");
// Define an output file
  fRun->SetOutputFile("data/testrun.root");
// Set media file
 fRun->SetMaterials("media.geo”)
```

Defining a magnetic field

```
// -----   Magnetic field
-------------------------------------------
  // Constant Field
  PndConstField *fMagField = new PndConstField();
  fMagField->SetField(0., 10. ,0. ); // values are in kG
  fMagField->SetFieldRegion(-50, 50,-50, 50, 350,
450);//values are in cm(xmin,xmax,ymin,ymax,zmin,zmax)
  fRun->SetField(fMagField);
  //
-------------------------------------------------------------------
```

Simulation Macro- Create Modules


```
FairModule *Cave= new PndCave("CAVE"); // create the top node
Cave->SetGeometryFileName("cave.geo"); // select  geometry file
fRun->AddModule(Cave); // add it to the Run

FairModule *Magnet= new PndMagnet("MAGNET"); // passive material only
Magnet->SetGeometryFileName("magnet.geo");
fRun->AddModule(Magnet);
```

Simulation Macro- Create Detectors


```
FairDetector *Torino= new PndTorinoDetector("TORINO", kTRUE);
Torino->SetGeometryFileName("torino.geo");
fRun->AddModule(Torino);
```

- kTRUE : Active detector
- kFALSE: The material will be taken into account during transport, but the process hit method of this detector will not be called

Simulation Macro-Event Generators


```
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
fRun->SetGenerator(primGen);

// Box Generator
FairBoxGenerator* boxGen = new FairBoxGenerator(2212, 3); // 2212 = proton; 3 = multipl.
boxGen->SetPRange(2., 2.); // GeV/c //setPRange vs setPtRange
boxGen->SetPhiRange(0, 360); // Azimuth angle range [degree]
boxGen->SetThetaRange(3, 10); // Polar angle in lab system range [degree]
boxGen->SetCosTheta(); //uniform generation on all the solid angle(default)

// boxGen->SetXYZ(0., 0.37, 0.); // origin of the vertex
primGen->AddGenerator(boxGen); // Add the particle generator to the primary generator
                               // One can add many different generators
```

Output Files

- The Simulation has by default two outputs:
  - Data File(s)
    - Tree for the events
    - Folder structure used to create the Tree
  - Parameter File
    - Field map description (optional)
    - Some geometrical stuff (optional)
    - User defined stuff
  - Geometry file (optional)

Parameter file

```
FairRuntimeDb *rtdb=fRun->GetRuntimeDb();
Bool_t kParameterMerged=kTRUE;
FairParRootFileIo* output=new FairParRootFileIo
(kParameterMerged);
output->open("data/testparams.root");
rtdb->setOutput(output);

rtdb->saveOutput();
rtdb->print();
```

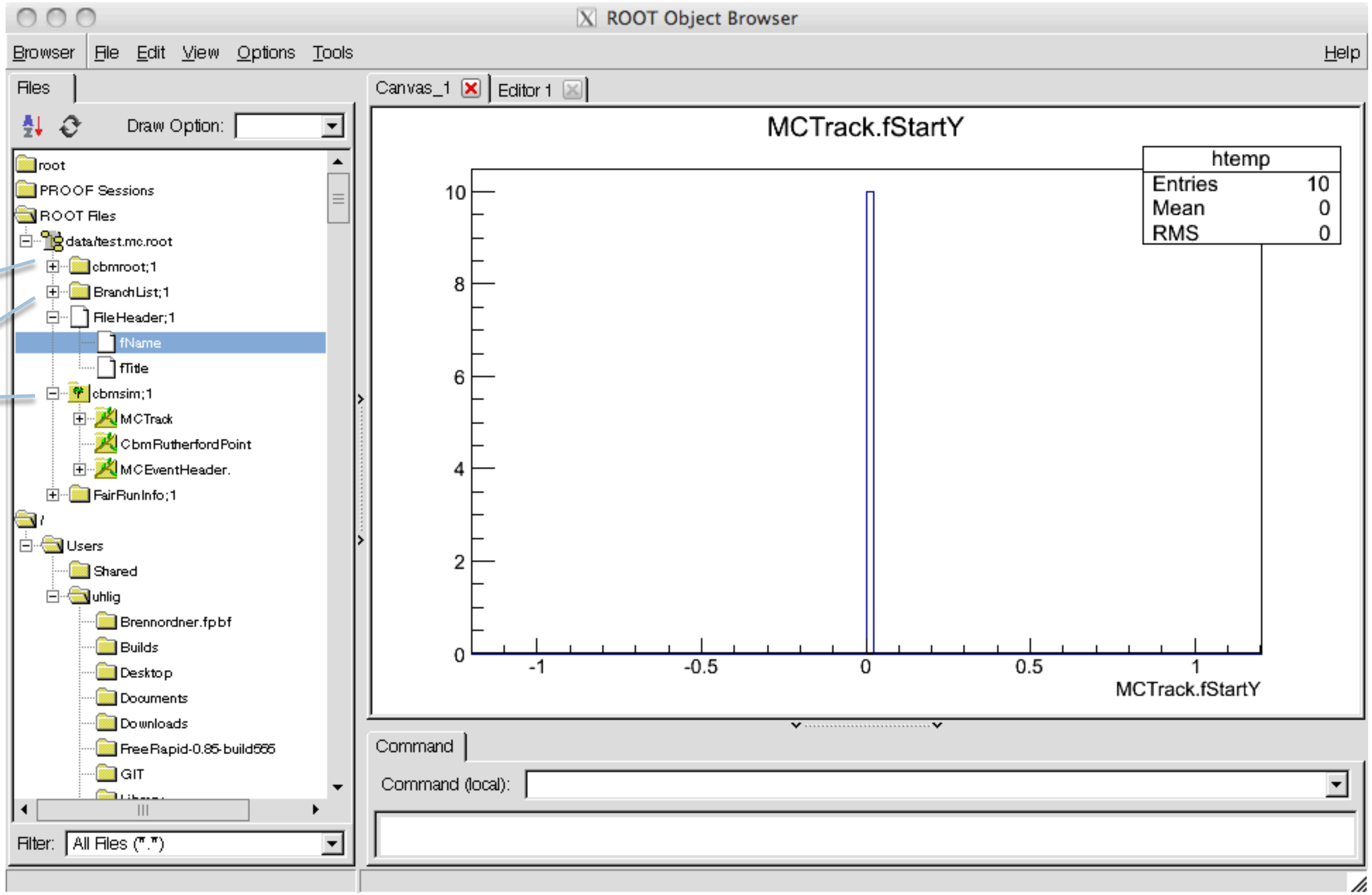
Simulation Macro- Run Simulation

- fRun->Init(); // Initialize the simulation
  - Simulation:
  - 1. Initialize the VMC (Simulation)
  - 2. Initialize Tasks (if they are used in Simulation)

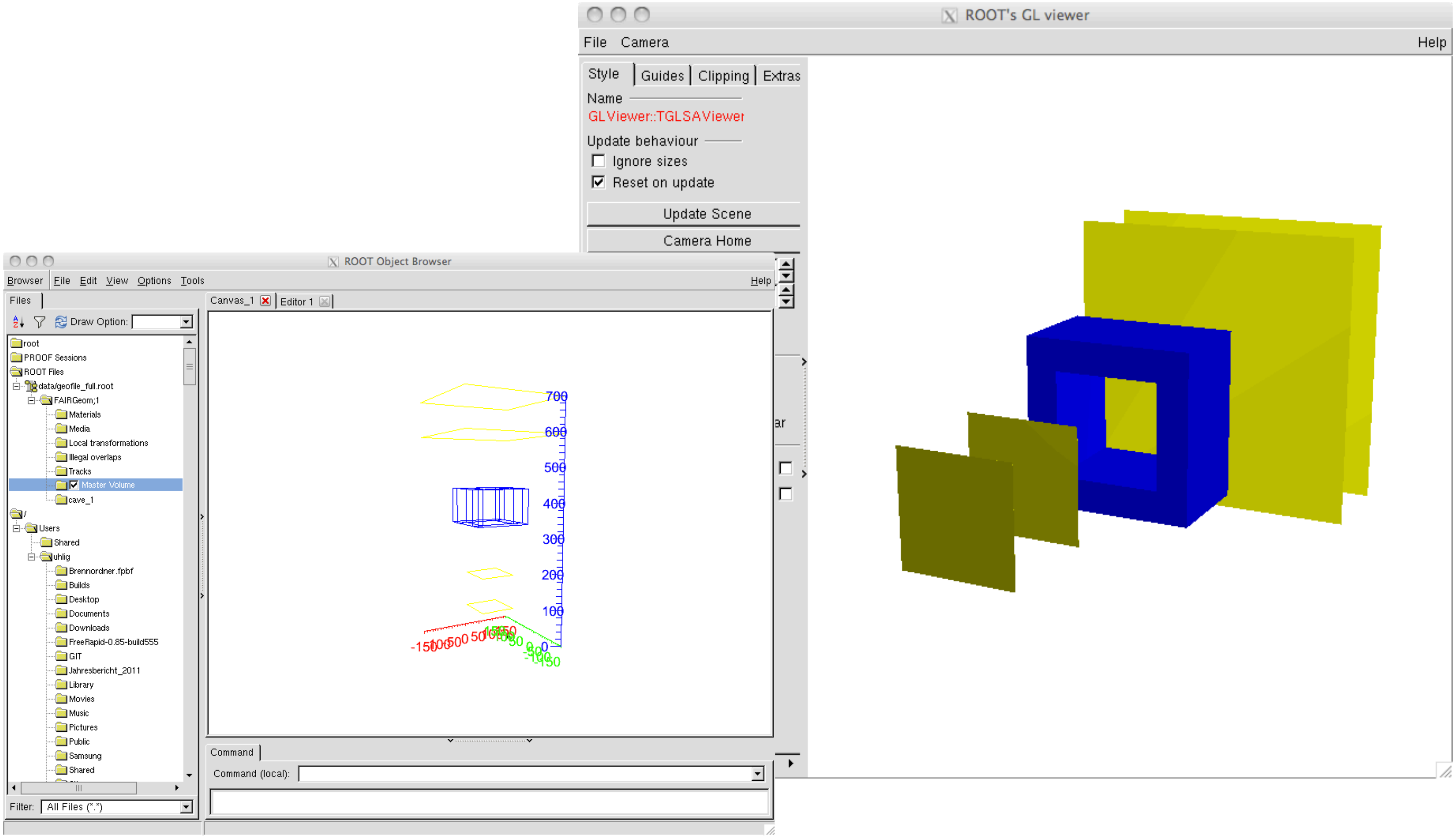
- fRun->Run(NoOfEvent); //Run the Simulation
- fRun->CreateGeometryFile(”geofile.root"); // Create a separate file with the TGeoManager

Output Files: Root Browser

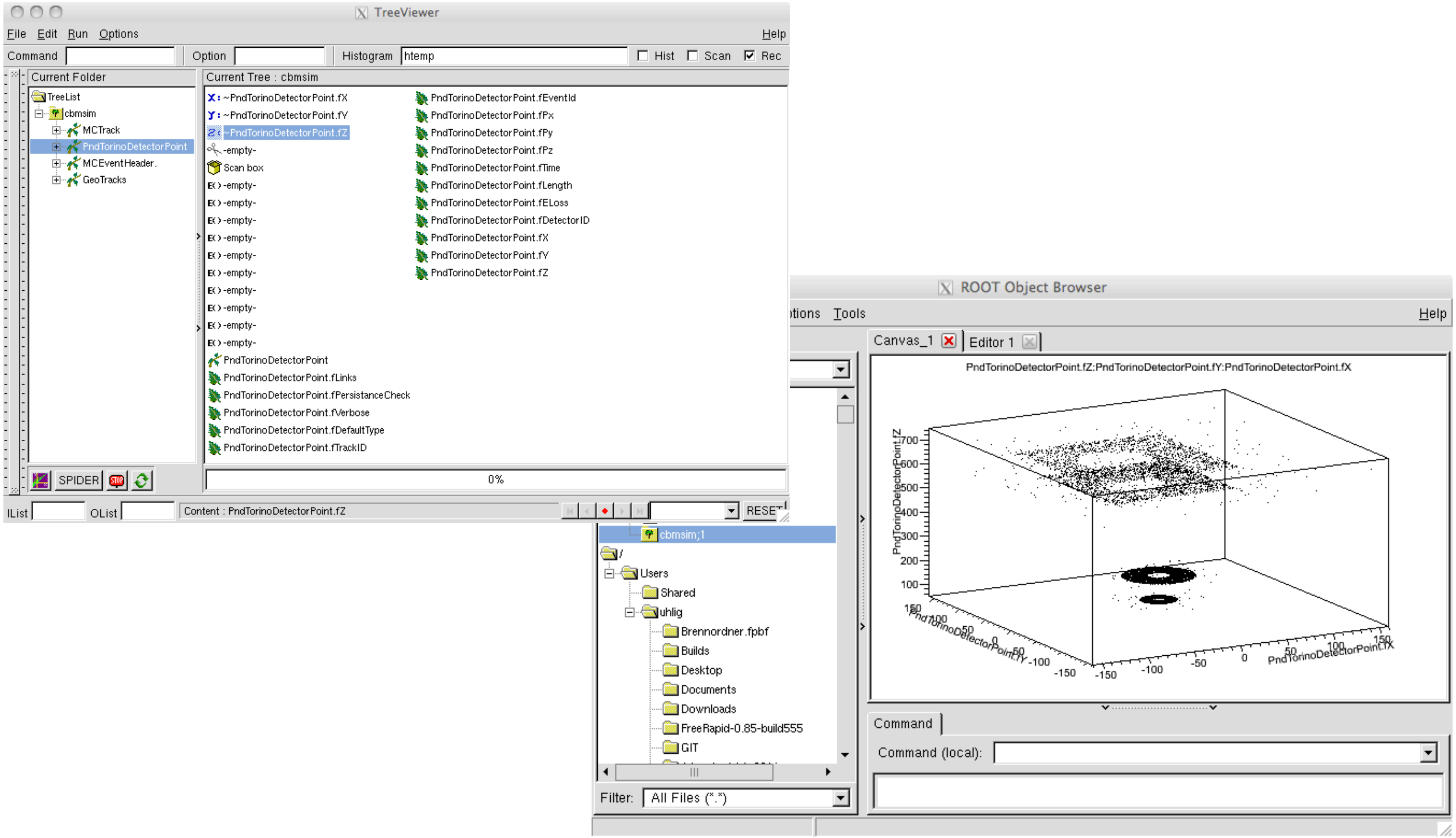
Folder
Branchlist
Data



Geometry in Root Browser



Tree Viewer


TreeViewer
File Edit Run Options Help
Command Option Histogram htemp Hist Scan Rec
Current Folder
TreeList
cbmsim
MCTrack
PndTorinoDetectorPoint
MCEventHeader.
GeoTracks
Current Tree : cbmsim
X: ~PndTorinoDetectorPoint.fX
Y: ~PndTorinoDetectorPoint.fY
Z: ~PndTorinoDetectorPoint.fZ
-empty-
Scan box
PndTorinoDetectorPoint
PndTorinoDetectorPoint.fLinks
PndTorinoDetectorPoint.fPersistanceCheck
PndTorinoDetectorPoint.fVerbose
PndTorinoDetectorPoint.fDefaultType
PndTorinoDetectorPoint.fTrackID
PndTorinoDetectorPoint.fEventId
PndTorinoDetectorPoint.fPx
PndTorinoDetectorPoint.fPy
PndTorinoDetectorPoint.fPz
PndTorinoDetectorPoint.fTime
PndTorinoDetectorPoint.fLength
PndTorinoDetectorPoint.fELoss
PndTorinoDetectorPoint.fDetectorID
PndTorinoDetectorPoint.fX
PndTorinoDetectorPoint.fY
PndTorinoDetectorPoint.fZ
SPIDER
0%
IList OList Content : PndTorinoDetectorPoint.fZ RESET
ROOT Object Browser
Tools Help
Canvas_1 Editor 1
PndTorinoDetectorPoint.fZ:PndTorinoDetectorPoint.fY:PndTorinoDetectorPoint.fX
PndTorinoDetectorPoint.fZ
PndTorinoDetectorPoint.fY
PndTorinoDetectorPoint.fX
cbmsim;1
Users
Shared
uhlig
Brennordner.fpbf
Builds
Desktop
Documents
Downloads
FreeRapid-0.85-build555
GIT
Filter: All Files (*.*)
Command
Command (local):

Hands on: Run a simulation

- Create a macro which transports some events through the PndTorinoDetector experiment
- Check if the geometry is what you expect
- Check the distribution of the stored MC points

Magnetic Field

- The Framework deliver a base class which can handle the magnetic field in the simulation or analysis (reconstruction)
- The User has to derive from this class and implement
  - Init() :
    - Intialization. E.g. read in the field map. If needed, to be implemented in the concrete class.
  - GetFieldValue(const Double_t point[3], Double_t* bField);
    - Get magnetic field value at a certain point.
    - point Coordinates [cm]
    - bField (return) Field components [kG]
    -

Cbm field implementations

- Constant field
  - PndFieldConst
- Fieldmaps
  - PndMultiField
  - ????

Physics settings

- Default settings for all media can be found in the subdirectory gconfig
- Normally it is not necessary to change these settings
- Geant3
  - g3Config.C and SetCuts.C
- Geant4
  - g4Config.C and SetCuts.C
  - One has do define physics lists. There are no default settings which cover 90% of all cases

Special physics settings

- If you need special settings for a material dedicated to your detector, don't change the default settings
- Each Detector class can implement a function SetSpecialCuts

```
void CbmTrd::SetSpecialPhysicsCuts(){
   FairRun* fRun = FairRun::Instance();
   if (strcmp(fRun->GetName(),"TGeant3") == 0) {
     // Get Material Id and some material properties from the geomanager
     Int_t mat = gGeoManager->GetMaterialIndex("TRDgas"); // search by name
     TGeoMaterial *trdgas =  gGeoManager->GetMaterial(mat);
     // Get the material properties for material with id+1
     // (of-by-one problem) from the Virtual Monte Carlo
     Int_t matIdVMC = mat+1;
     // Set new properties, physics cuts etc. for the TRDgas
     gMC->Gstpar(matIdVMC,"STRA",1.0);
  }
}
```

HowTo use example implementation

- Full running implementation of the example detector
  - https://subversion.gsi.de/fairroot/pandaroot/development/uhlig/Torino/version1
  - The torinoDetector directory contain the source code
  - The macro directory contain the macros, geometries, etc.
- Use my version of the torinoDetector source code
  - svn switch https://subversion.gsi.de/fairroot/pandaroot/development/uhlig/Torino/version1/torinoDetector torinoDetector
- Use my version of the macros
  - cd macro
  - svn propedit svn:externals .
  - Add there
    - torinoDetector https://subversion.gsi.de/fairroot/pandaroot/development/uhlig/Torino/version1/macro/torinoDetector

Useful SVN command

- Get help
  - svn help
- Get detailed help about a specific command
  - svn help <command>
- Get info about the status of your local working copy (WC). Which files have been changed, deleted, …
  - Differences between the state of your WC and the state of the last svn revision you checked out
    - svn stat
  - Differences between the state of your WC and the state of the head revision on the central server
    - svn stat –u

Useful SVN commands (cont.)

- See differences between different versions of the code
  - svn diff
- Create a directory
  - svn mkdir
- Move/Rename a file or directory
  - svn mv
- Put files and directories under version control. They will be added in next commit.
  - svn add

Useful SVN command (cont.)

- Send changes from your working copy to the repository
  - svn commit -m”Useful comment”
  - svn commit <file> -m”Useful comment”
  - svn commit <directory> -m”Useful comment”
- Bring changes from the repository into the working copy
  - svn update