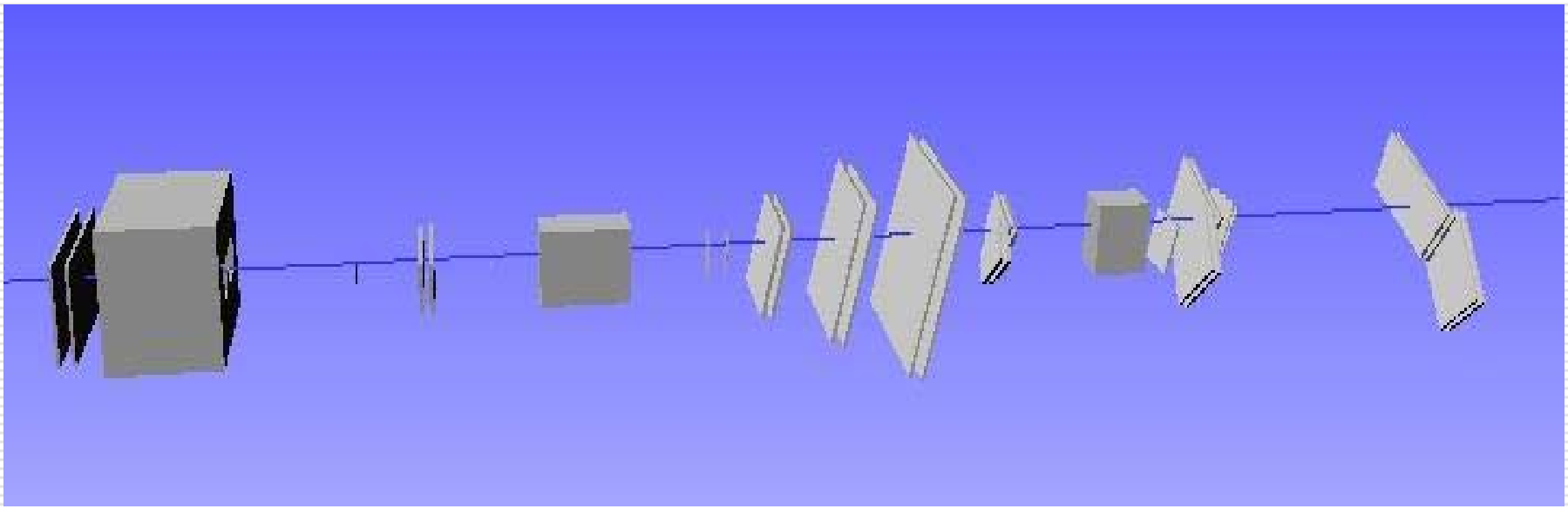
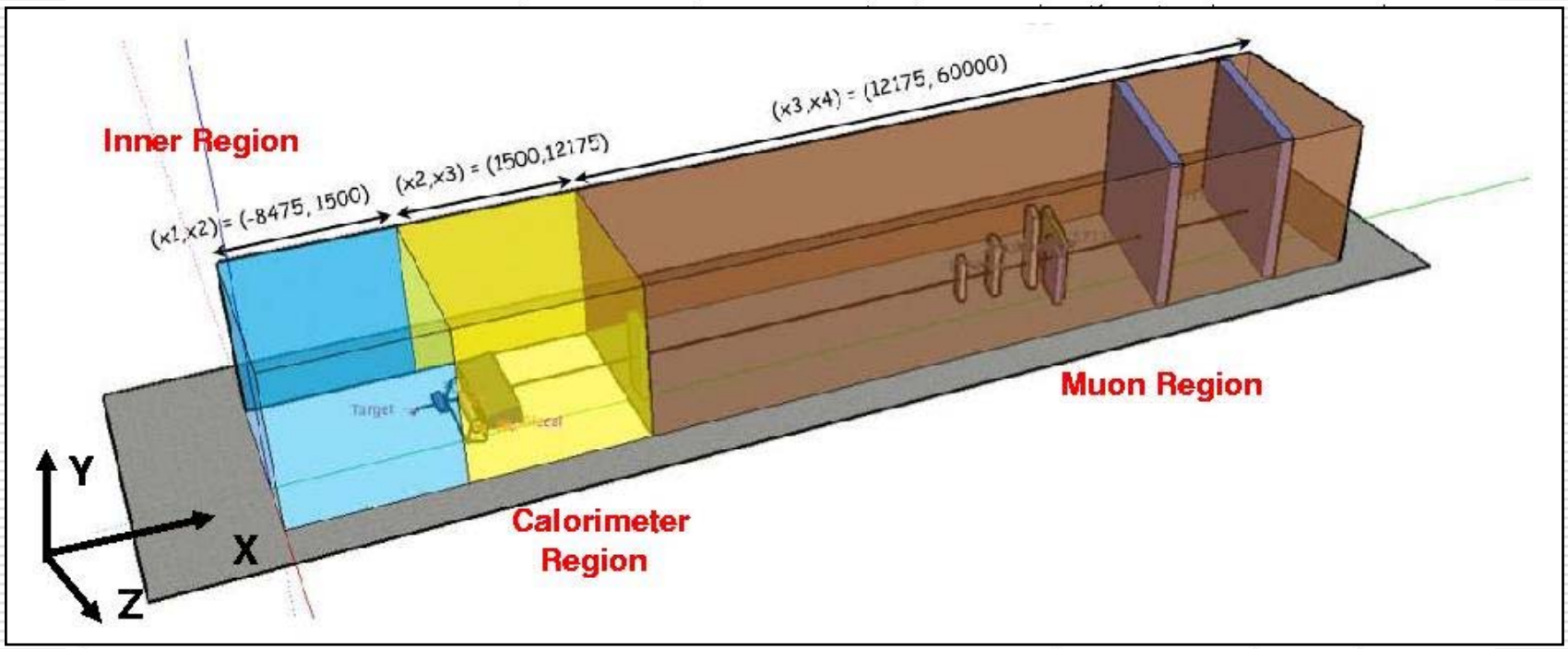
CTB_G4Sim Simulation



CTB_G4Sim: Generals

- The position of the different envelopes for the sub-detectors (regions) and ancillary material is stored in the Pool file created with the simulation
  - The list of envelopes can be found in CVS (TestBeam/TBDetDescr/src/TBDetDescrManager.cxx



CTB_G4Sim: HowTo get started

Once you are sure that CMT environment is setup correctly..

- create a directory for your simulation

```
mkdir CTBSim
cd CTBSim
```

- set up the correct requirements file

```
cp ~drebuzzi/public/Tutorial/requirements .
source /afs/cern.ch/sw/contrib/CMT/v1r16p20040901/mgr/setup.sh
cmt config
source setup.sh -tag=10.0.0,opt
```

- check out the simulation package

```
cd 10.0.0
cmt co Simulation/G4Sim/CTB_G4Sim/CTB_G4Sim-*
```

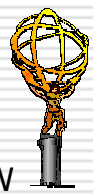
- *temporary!! To run with 10.0.0 (already fixed in the nightlies)*

```
cmt co -r G4PhysicsLists-00-00-31 Simulation/G4Utilities/G4PhysicsLists
```

HowTo run with a nightly

```
cp ~drebuzzi/public/Tutorial/requirements_nightly .
source /afs/cern.ch/sw/contrib/CMT/v1r16p20040901/mgr/setup.sh
cmt config
source setup.sh -tag=atlrel_4,opt
```

Since atlrel_3 (09-03-05) you don't need to check out G4PhysicsLists



CTB_G4Sim: HowTo get started

- compile, and build the package

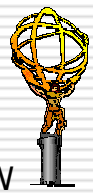
```
cd Simulation/G4Sim/CTB_G4Sim/CTB_G4Sim-*
cmt config
cmt broadcast cmt config
source setup.sh
cmt broadcast gmake
```

- run the simulation

```
cd ../run
get_files jobOptions.CTB_G4Sim.py
athena.py jobOptions.CTB_G4Sim.py
```

In ~drebuzzi/public/Tutorial the file commands_list contains all the command you should type in this tutorial, you can cut&paste from it

In the following  *= customizable, options you can select and modify*



CTB_G4Sim: Structure of the Package

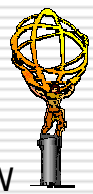
CTB_G4Sim

- **cmt**
- **data** { *parameters tables* (materials, mag field, geometry, etc.)
- **doc**
- **G4macros** { CTB_G4Sim uses in a large extend the *macro file* (*.mac) facility to set up the CTB simulation
- **run**
- **share** { jobOptions and jobOptions fragments

*structure of the *.mac including*

```
jobOptions.CTB_G4Sim.py
|
|__ ctb_Atlas.mac
|       |
|       |__ ctb_envelopes.mac <-- always needed
|       |
|       |__ ctb_PIXEL.mac
|       |     ctb_PIXELgeometry.mac
|       |     ctb_PIXELsensitivity.mac
|       |
|       |__ ctb_SCT.mac
|       |     ctb_SCTgeometry.mac
|       |     ctb_SCTsensitivity.mac
|       |
|       |__ ctb_TRT.mac
|       |     ctb_TRTgeometry.mac
|       |     ctb_TRTsensitivity.mac
|       |
|       |__ ctb_LArCal.mac
|       |     ctb_LiArVis.mac
|       |
|       |__ ctb_TileCal.mac
|       |     ctb_TileCalgeometry.mac
|       |     ctb_TileCalsensitivity.mac
|       |
|       |__ ctb_MuonSystem.mac
|       |     ctb_MuonSystemgeometry.mac
|       |     ctb_MuonSystemsensitivity.mac
|       |     ctb_envelopesMuonDump.mac
|       |     ctb_envelopesMuonMagnets.mac
|       |
|       |__ ctb_MagneticField.mac
|       |
|       |__ ctb_ancillary.mac
|
|__ ctb_environ.mac
|__ ctb_eventgraph.mac
|__ ctb_physicslist.mac
|       |
|       |- ctb_SCTCuts.mac
|       |- ctb_LArCalCuts.mac
|       |_ ctb_MuonSystemCuts.mac
|
|__ ctb_vis.mac
    ctb_visualization.mac
      ctb_PIXELVis.mac
      ctb_SCTVis.mac
      ctb_TRTVis.mac
      ctb_LiArVis.mac
      ctb_TileCalVis.mac
      ctb_MuonSystemVis.mac
```

***general recipe**: if you need to modify one *.mac file, copy it in the run dir before, or you should recompile the package*



CTB_G4Sim: HowTos

How to visualize the CTB Geometry

1. ***VRMLVIEW***: edit the jobOptions.CTB_G4Sim.py →
comment the line

```
G4Svc.FADSMacro =
"ctb_Atlas.mac:ctb_physicslist.mac:ctb_g4utilities-action.mac"
```

and uncomment the line

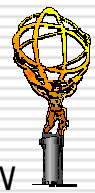
```
G4Svc.FADSMacro =
"ctb_environ.mac:ctb_Atlas.mac:ctb_physicslist.mac:ctb_vis.mac:ctb_e
ventgraph.mac"
```

and you will get a ".wrl" file → to open it, do the following

```
cp ~drebuzzi/public/Tutorial/vrmlview .
export LD_LIBRARY_PATH=/afs/cern.ch/user/d/drebuzzi/public/Tutorial:
$LD_LIBRARY_PATH
./vrmlview g4_00.wrl
```

You can customize the volumes to be visualized by changing their vis property in ctb_vis.mac or in the detector specific *_vis.mac

*But be careful! If you run an event, the *.wrl file could be ~50MB! It could take a lot of time!*



CTB_G4Sim: HowTos

How to visualize the CTB Geometry (cont'd)

2. ***HitDisplay***: edit jobOptions.CTB_G4Sim.py → add at the end the following three lines

```
theApp.Dlls+=["HitDisplay"]
HitDisplay=Algorithm("HitDisplay")
theApp.TopAlg+=["HitDisplay"]
```

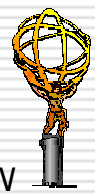
You can also visualize the hits
To save a picture type `import myFigure.jpg` (.gif, .eps, etc.) and click on the HitDisplay with the mouse left button

How to switch on/off the detectors

• *all* the detectors and ancillary materials included in the H8 experimental area are simulated *as default*, but..

*any sub-detector, ancillary detector, (muon) magnet or dump can be swithed **off** by commenting out the appropriate line in the* ctb_Atlas.mac *file*

```
# LAr calorimeters. Comment out the following line for no LAr
#/echo "CTB_G4Sim: Building the LAr calorimeter"
#/macro/execute ctb_LArCal.mac
```



CTB_G4Sim: HowTos

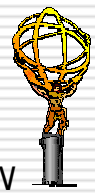
How to switch on/off the magnetic fields and the magnets

- there are three magnets called ***MBPSID***, ***MBPL*** and ***MBPS2*** along the beam direction
  1. ***MBPSID***, the first in the line, hosts the *Pixel* and *SCT* detectors
  2. ***MBPL*** and ***MBPS2*** magnets are included in the Muon region (to have them, the MuonSystem must be active!)
- the configuration of the magnetic fields is done in the ctb_MagneticField.mac macro file and uses the data stored in the ctb_magfield.xml file
  - the field is set in the corresponding magnetic-gap region (**MBPSID**, **MBPL** and **MBPS2**) to the three magnet volumes (**MAGBOXMBPSID**, **MAGBOXMBPSL**, **MAGBOXMBPS2**)

- to swicth ***off*** the magnetic fields (the default is ***on***), edit ctb_Atlas.mac and comment the following lines:

```
/echo "CTB_G4Sim: Setting up the magnetic field"
/macro/execute ctb_MagneticField.mac
```

If you want to ***remove*** one magnet, edit ctb_envelopesMuonMagnet.mac and comment out the corresponding lines



CTB_G4Sim: HowTos

How to use the ParticleGenerator

CTB_G4Sim is using the ParticleGenerator package, which generates single particles and puts them into the HepMC transient data store

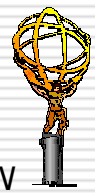
Note: the time is fixed together with the start X position of the beam. This will provide t=0 at X=0 (the Inner Detector hits must have positive time)

```
#--- ParticleGenerator -----------------------------------
# PDGCodes
# e--11, e+=-11, mu--13, mu+=-13, pi+=211, pi-=-211, pi0=111
# gamma=22, geantino=999
# Energy and momentum now in MeV!!!!
theApp.Dlls +=["ParticleGenerator"]
# If you want to change the random number seed for your
#ParticleGenerator, uncomment the three following lines
#and replace second number in the AtRndmGenSvc.Seeds
#command with the one you want.
theApp.ExtSvc += [ "AtRndmGenSvc" ]
AtRndmGenSvc= Service("AtRndmGenSvc")
AtRndmGenSvc.Seeds = [ "SINGLE 2040160768 443921183" ];
# If you want to read seeds from a file uncoment the next line
#AtRndmGenSvc.ReadFromFile = TRUE;
ParticleGenerator = Algorithm( "ParticleGenerator" )
ParticleGenerator.orders = [
 "pdgcode: constant 11",
 "energy: constant 54300",
 "vertX: constant -27500.0",
 "vertY: flat   -15.0 15.0",
 "vertZ: flat   -15.0 15.0",
 "t: constant -27500.0",
 "momX: fixed 1",
 "momY: fixed 0",
 "momZ: fixed 0"
]
```

default parameter choice

parameter choice optimized for the 250 GeV momentum beam, only the muon system in the setup

```
ParticleGenerator.orders = [
 "pdgcode: constant 13",
 "energy: normal 166800 8830",
 "vertY: normal   -20.0 35.0",
 "vertZ: constant   0.0 ",
 "t: constant -27500.0",
 "phi: constant 0.0",
 "theta: normal
1.57651232679489661923132169163980.001048"]
```



CTB_G4Sim: HowTos

How to write hits in POOL

- edit the jobOptions.CTB_G4Sim.py and uncomment the line:

```
include( "CTB_G4Sim/HitAthenaPoolWriteOptions.py" )
```

→ at the end of the athena job, the root file ctb_MyOutputFile.root together with PoolFileCatalog.xml are produced

You can change the name of the root output file inside the HitAthenaPoolWriteOptions.py file → you should copy it in your run dir and modify the line above in `include( "HitAthenaPoolWriteOptions.py" )` in the jobOptions.CTB_G4Sim.py (or recompile the package)

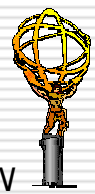
How to change the database for the muons

- edit the jobOptions.CTB_G4Sim.py and add

```
NovaCnvSvc.Version = 8
```

after the line `include ("MuonGeoModel/GeoModelInit.py")`, for instance, to use the amdb b.01 "as-built" database

But be careful! You should use the same database also in the digitization and reconstruction (unless explicitly decided)!



Muon Databases

Athena version	Amdb version	Nova Version
8.8.0	a.04	NovaCnvSvc.Version= 6
9.0.0	a.04	NovaCnvSvc.Version= 4
9.3.0	a.05	NovaCnvSvc.Version= 7
9.3.0	b.01	NovaCnvSvc.Version= 8
9.3.0	b.02	NovaCnvSvc.Version= 9

} *"as built" databases with the X-tomo information*

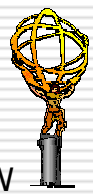
- ***b.01 vs a.05:*** *MDT inner structure*

a.05

```
W MDT 1 .090 4 30.0350 14.6000 123.0682 27.0000
               30.0175 56.0286 82.0396 108.0507
               30.0350 15.0175 30.0350 15.0175 0.4000
W MDT 2 .090 4 30.0350 14.6000 123.0682 27.0000
               15.0175 41.0286 67.0396 93.0507
               15.0175 30.0350 15.0175 30.0350 0.4000
```

b.01

```
*BIL 2 Jura-side (RM1003-BIL6A01)
W MDT 9 .090 4 30.0350 14.6000 123.0852 27.0000
               30.0173 56.0332 82.0493 108.0625
               30.0350 15.0175 30.0350 15.0175 0.4000
W MDT 10.090 4 30.0334 14.6000 123.0799 27.0000
               15.0167 41.0322 67.0477 93.0632
               15.0167 30.0334 15.0167 30.0334 0.4000
```



CTB_G4Sim: HowTos

How to run the digitization and write digits in POOL

The jobOptions.CTB_G4Dig.py in the share directory *reads the POOL file with hits and produces another POOL file with digits*

- to run the digitization you should type from your run dir

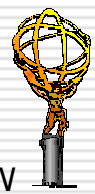
```
athena.py CTB_G4Sim/jobOptions.CTB_G4Dig.py
```

or (if you have plans to modify it, for instance)

```
get_files jobOptions.CTB_G4Dig.py
athena.py jobOptions.CTB_G4Dig.py
```

the default jobOptions creates the digit POOL file ctb_MyOutputFileDig.root reading from the POOL hits file ctb_MyOutputFile.root

You can modify these names in the jobOptions.CTB_G4Dig.py



CTB_G4Sim: HowTos

How to digitize an already generated hit file on castor

• if you want to read an already generated hit file on castor → edit the jobOptions.CTB_G4Dig.py and modify the InputCollection name

```
EventSelector.InputCollection =
"rfio:/castor/cern.ch/atlas/ctb/test/monte_carlo/simulation/muon-
/preprod_g4sim.CTB_G4Sim_mu-_180_GeV_eta_0.2_Mag_0.v1.900.00001.root"
```

and copy the corresponding PoolFileCatalog in your run dir

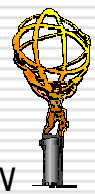
```
rfcp /castor/cern.ch/atlas/ctb/test/monte_carlo/simulation/muon-/
PoolFileCatalog.preprod_g4sim.CTB_G4Sim_mu_180_GeV_eta_0.2_Mag_0
.v1.900.00001.xml .
```

the list of the available hit files already generated can be gotten by typing

```
rfdir /castor/cern.ch/atlas/ctb/test/monte_carlo/simulation
```

or at the following web page (not yet updated)

http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/testbeam/simulationCTB/productionMC.html



CTB_G4Sim: Digitization Options

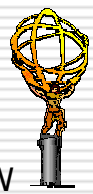
The digitization can be customized by changing the DetFlags in jobOptions.CTB_G4Dig.py

```
# - Select detectors
DetFlags.ID_setOff()
DetFlags.Calo_setOff()
DetFlags.em_setOff()
DetFlags.Tile_setOff()
DetFlags.Muon_setOn()
DetFlags.Truth_setOff()
#DetFlags.LVL1_setOn()

# - Switch off tasks
DetFlags.pileup.all_setOff()
DetFlags.simulate.all_setOff()
DetFlags.makeRIO.all_setOff()
DetFlags.writeBS.all_setOff()
DetFlags.readRDOBS.all_setOff()
DetFlags.readRIOBS.all_setOff()
DetFlags.readRIOPool.all_setOff()
DetFlags.writeRIOPool.all_setOff()
#DetFlags.writeRDOPool.all_setOff()
```

Properties or the Muon Digitization, *to be optimized for the testebeam setup*

```
RPC_Digitizer.WindowLowerOffset = -1000
RPC_Digitizer.WindowUpperOffset = 1000
RPC_Digitizer.CTB2004 = TRUE;
MDT_Digitizer.OffsetTDC = 0.
MDT_Digitizer.BunchCountOffset = -200
MDT_Digitizer.UseTof = FALSE
MDT_Digitizer.UseProp = FALSE
```



CTB_G4Sim: Muon Digitization

Two digitization tools available:

1. ***MDT_Response_DigiTool*** (default): detailed simulation of MDT response including cluster size fluctuations, diffusion and the ADC response (slewing correction)

You can tune many parameters, e.g. the electonic threshold

```
ToolSvc = Service( "ToolSvc" )
ToolSvc.MDT_Response_DigiTool.Threshold = 20
```

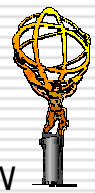
2. ***RT_Relation_DigiTool***: transforms r→t + smearing in time using external rt relations, from Garfield, from data

If you want to select this → add the following line and uncommented it

```
# Uncomment the following line if you want to use MDT digitization
# with external RT relation (default is Nikhef algorithm digi.)
#MDT_Digitizer.DigitizationTool =   "RT_Relation_DigiTool"
```

the default rt relation which is taken is the ArCO2.rt file in MuonSpectrometer/MuonDigitization/MDT_Digitization/share

Be careful because the t→r conversion in the reconstruction must be selected consequently!



CTB_G4Sim: Muon Digitization

How to set a customized rt relation

If you want to change the rt relation, using your customized one

1. check that you have both the rt and the tr file, in the standard format "à la Calib"
2. select the RT_Relation_DigiTool in the jobOptions.CTB_G4Sim.py, as explained in slide 14
3. check out the MDT_Digitization

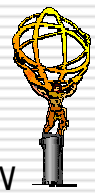
```
cmt co MuonSpectrometer/MuonDigitization/MDT_Digitization
```

4. in the share directory, replace the file ArCO2.rt with your rt file (r, t, resolution on r), for example

```
cp ~drebuzzi/public/Tutorial/ArCO2_ludo.rt MuonSpectrometer/
MuonDigitization/MDT_Digitization/MDT_Digitization-*/share
```

5. compile the package
6. be sure to pass to the reconstruction the same tr relation→ for example

```
MuonTBCalibrationSvc.RT_InputFiles = [ "/afs/cern.ch/user/d/
drebuzzi/public/Tutorial/ArCO2_ludo.tr" ]
```



CTB_G4Sim: Exercise 1

create a sample of 100 muon events @ 250GeV, commenting out all the detectors and ancillaries but the muons

visualize the setup to be sure that the chambers are hit

re-read the hits using the jobOptions ReadMuonSimHitOptions.py
→ from the run dir

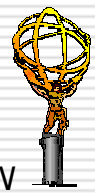
```
cp ~drebuzzi/public/Tutorial/ReadMuonSimHitsOptions.py .
athena.py ReadMuonSimHitsOptions.py
```

run the digitization on the generated hit file, using the default digitization tool

re-read the digits with the jobOptions ReadMuonDigitOptions.py

```
cp ~drebuzzi/public/Tutorial/ReadMuonDigitsOptions.py .
athena.py ReadMuonDigitsOptions.py
```

re-read digits from RecExTB



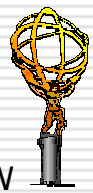
CTB_G4Sim: Exercise 2

run the digitization and the reconstruction on the fly on the hit file simul.10.0.0.muon_only.b.01.root, selecting the a.05 database in the jobO

```
cp ~drebuzzi/public/Tutorial/simul.10.0.0.muon_only.b.01.root .
cp ~drebuzzi/public/Tutorial/PoolFileCatalog.xml .
cp ~drebuzzi/public/Tutorial/jobOptions.CTB_G4Dig+Rec.py .
```

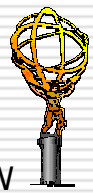
analyze the reconstruction root file (ntuple.root) and check the sagitta and the residuals for the barrel chambers

compare the results with the ones obtained on the 250GeV data sample analyzed before



Useful Links and Documentation

- ***HowTo for the CTB (2004) simulation***
  http://mgallas.home.cern.ch/mgallas/ctb_atlas/ctb_howto.html
- ***CTB_G4Sim progress by tag***
  http://mgallas.home.cern.ch/mgallas/ctb_atlas/ctb_progress.html
- ***Status of Simulation production***
  http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/testbeam/simulationCTB/productionMC.html
- ***Main CTB web page***
  http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/testbeam/testbeam.html
- ***The 2004 Atlas Muon Test Beam in H8***
  http://atlas/tb/muon.web.cern.ch/atlas-tb-muon
- ***"ATLAS Barrel Combined Run in 2004 Test Beam Setup and its evolution"***, B.Di Girolamo, M.Gallas and T.Koffas, EDMS Note: ATC-TT-IN-0001



Acknowledgements

Many thanks! to

Ketevi – Giuseppe – Manuel - Daniele