

# Table of Contents

<b>Muon Tracking Geometry</b> .....	<b>1</b>
Working with control version git (but in parallel with svn using the git svn tool).....	1
Package Analysis: Muon Tracking Geometry.....	1
Muon Tracking Geometry package validation.....	1
Tentative Strategy.....	1
Setting-up the tools.....	1
Generator/ParticleGenerator.....	2
Simulation/G4Atlas/G4AtlasApps.....	2
Simulation/ISF/ISF_Config.....	2
Simulation/ISF/ISF_Core/ISF_Services.....	5
Simulation/ISF/ISF_Geant4/ISF_Geant4Tools.....	6
Tracking/TrkDetDescr/TrkVolumes.....	6
Tracking/TrkExtrapolation/TrkExSTEP_Propagator.....	6
Tracking/TrkExtrapolation/TrkExTools.....	6
Configuration.....	9
Geantino shots.....	10

# Muon Tracking Geometry

## Working with control version git (but in parallel with svn using the *git svn* tool).

Repository from svn

Just using the assetup X.Y.Z, here the SVNROOT variable is already set. Anyway, it should be

```
SVNROOT=svn+ssh://svn.cern.ch/repos/atlasoff
```

And obtaining the code with

```
git svn clone svn+ssh://duarte@svn.cern.ch/repos/atlasoff/MuonSpectrometer/MuonDetDescr/MuonTrackingGeometry
```

Note that initially I did just: `git svn clone MuonSpectrometer/MuonDetDescr/MuonTrackingGeometry -s` assuming the previous export variable, but appears the error: *E: 'trunk' is not a complete URL and a separate URL is not specified.*

Note also that the whole process lasts some time due that git is localizing and keeping track of all branches and tags. The [svn and git](#) link explains how to work with a git repository pushing to a remote svn. The repository has been cloned in the lxplus path:

**/afs/cern.ch/user/d/duarte/work/private/MTG/CODE\_19\_2\_0/MuonTrackingGeometry.**

Apart from the *atlasoff* repository, a remote repository has been put in place in github, just to push the commits and have a double copy of the work. The forked repository can be found at [MuonTrackingGeometry github repo](#).

## Package Analysis: Muon Tracking Geometry

- Why is this package needed?
- What has this package to provide?
- Who (what classes) have to use this package?

## Muon Tracking Geometry package validation

**Problem:** Geometry defined in Geant4 presents important discrepancies with the tracking geometry (an optimized and non-detailed geometry to perform tracking).

**JIRA ticket:** <https://its.cern.ch/jira/browse/ATLASRECTS-1096>

### Tentative Strategy

- Use geantinos to map the material content as seen by the full simulation (using Geant4) and the fast simulation (using MTG package to provide the MS geometry).
  - ◆ *CAVEAT* Impossible to disentangle effects coming from bad material description in MTG and those coming from tracking extrapolation (when extrapolator does not resolve correctly the entry/exit to/from dense volume --- Shenka dixit: check this)
- Build Geant4 geometry, and MTG on top of it, checking that for each Geant4 material step there is an equivalent material defined in the tracking geometry.

### Setting-up the tools

- **Athena release:** 19.X.0 (moved from 19.2.0, missing materials in DB). Needs some fixes in the following packages:

Package	cmt version
Generators/ParticleGenerator	00-00-61
Simulation/G4Atlas/G4AtlasApps	00-07-63
Simulation/ISF/ISF_Config	00-00-54
Simulation/ISF/ISF_Core/ISF_Services	00-03-07
Simulation/ISF/ISF_Geant4/ISF_Geant4Tools	00-02-00
Tracking/TrkDetDescr/TrkVolumes	01-01-03
Tracking/TrkExtrapolation/TrkExSTEP_Propagator	01-00-29
Tracking/TrkExtrapolation/TrkExTools	03-00-34-branch

Show changes in the packages  Hide

### Generator/ParticleGenerator

```
=====
--- src/ParticleKinematics.cxx (revisión: 618359)
+++ src/ParticleKinematics.cxx (copia de trabajo)
@@ -217,7 +217,8 @@

    const HepPDT::ParticleData* particle
        = p_particleTable->particle(HepPDT::ParticleID(abs( m_pdg )));
-   m_mass = particle->mass().value();
+
+   m_mass = particle ? particle->mass().value() : 0.;

    m_time = getValue(kTime);
```

### Simulation/G4Atlas/G4AtlasApps

Index: Simulation/G4Atlas/G4AtlasApps/python/SimAtlasKernel.py

```
=====
--- Simulation/G4Atlas/G4AtlasApps/python/SimAtlasKernel.py (revisión: 592906)
+++ Simulation/G4Atlas/G4AtlasApps/python/SimAtlasKernel.py (copia de trabajo)
@@ -266,6 +266,11 @@
    from G4AtlasApps import AtlasG4Eng,PyG4Atlas
    actions = AtlasG4Eng.G4Eng.menu_UserActions()

+
+   # PhysicsValidationUserAction
+   AtlasG4Eng.G4Eng.log.verbose('ISF_AtlasSimSkeleton::do_UserActions add ISF_G4PhysicsVa
+   ISF_G4PhysicsValidation = PyG4Atlas.UserAction('ISF_Geant4Tools','ToolSvc.ISFG4Physics
+   actions.add_UserAction(ISF_G4PhysicsValidation)
+
    # TrackProcessorUserAction
    AtlasG4Eng.G4Eng.log.verbose('ISF_AtlasSimSkeleton::do_UserActions add ISF_G4TrackPro
    ISF_G4TrackProcessor = PyG4Atlas.UserAction('ISF_Geant4Tools','ToolSvc.ISFG4TrackProce
@@ -636,7 +641,7 @@

    from G4AtlasApps.SimFlags import simFlags
    if not simFlags.ISFRun:
-       from atlas_utilities import G4SimTimer
+       from atlas_utilities import G4SimTimer
        actions.add_UserAction(G4SimTimer)

    actions.add_UserAction(G4TrackCounter)
```

### Simulation/ISF/ISF\_Config

Index: Simulation/ISF/ISF\_Config/python/ISF\_MainConfig.py

```
=====
--- Simulation/ISF/ISF_Config/python/ISF_MainConfig.py (revision 615726)
+++ Simulation/ISF/ISF_Config/python/ISF_MainConfig.py (working copy)
@@ -107,33 +107,33 @@

        'vertY: constant 0.0',
        'vertZ: constant 0.0',
        't: constant 0.0',
```

## MuonTrackingGeo < Sandbox < TWiki

```
-         'eta: flat -4.0 4.0',
-         'phi: flat  0 6.28318',
-         'pt: constant 50000']))
+         'eta: flat 0.0 1.0',
+         'phi: flat  0.3 0.5',
+         'p: constant 50000']))
    return getInput_GenericGenerator(name, **kwargs)

##### Input: pions #####
def getInput_pions(name="ISF_Input_pions", **kwargs):
    kwargs.setdefault('orders', sorted(['pdgcode: sequence 211 -211',
-         'vertX: constant 0.0',
+         'vertX: constant 1140.0',
        'vertY: constant 0.0',
        'vertZ: constant 0.0',
        't: constant 0.0',
-         'eta: flat -4.0 4.0',
-         'phi: flat  0 6.28318',
-         'pt: constant 50000']))
+         'eta: flat -0.02 0.02',
+         'phi: flat  1.55 1.57',
+         'p: constant 50000']))
    return getInput_GenericGenerator(name, **kwargs)

##### Input: photons #####
def getInput_photons(name="ISF_Input_photons", **kwargs):
    kwargs.setdefault('orders', sorted(['pdgcode: fixed 22',
-         'vertX: constant 0.0',
+         'vertX: constant 1140.0',
        'vertY: constant 0.0',
        'vertZ: constant 0.0',
        't: constant 0.0',
-         'eta: flat -4.0 4.0',
-         'phi: flat  0 6.28318',
-         'pt: constant 50000']))
+         'eta: flat  0.0 0.0',
+         'phi: flat  0 0.',
+         'p: constant 10000']))
    return getInput_GenericGenerator(name, **kwargs)

@@ -161,9 +161,9 @@
        'vertY: constant 0.0',
        'vertZ: constant 0.0',
        't: constant 0.0',
-         'eta: flat -5.0 5.0',
-         'phi: flat  0 6.28318',
-         'pt: constant 10000']))
+         'eta: flat -4.0 4.0',
+         'phi: flat  0. 6.28316',
+         'p: constant 10000']))
    return getInput_GenericGenerator(name, **kwargs)

@@ -285,9 +285,10 @@
    SimKernel.BeamPipeSimulationSelectors = [ getPublicTool('ISF_DefaultParticleKillerSelector')
    SimKernel.IDSimulationSelectors       = [ getPublicTool('ISF_DefaultFatrasSelector') ]
-   SimKernel.CaloSimulationSelectors    = [ getPublicTool('ISF_MuonFatrasSelector'),
-                                           getPublicTool('ISF_EtaGreater5ParticleKillerSimS
-                                           getPublicTool('ISF_DefaultFastCaloSimSelector')
+   SimKernel.CaloSimulationSelectors    = [ getPublicTool('ISF_DefaultFatrasSelector') ]
+   #SimKernel.CaloSimulationSelectors  = [ getPublicTool('ISF_MuonFatrasSelector'),
+   #                                     getPublicTool('ISF_EtaGreater5ParticleKillerSim
+   #                                     getPublicTool('ISF_DefaultFastCaloSimSelector')
    SimKernel.MSSimulationSelectors      = [ getPublicTool('ISF_DefaultFatrasSelector') ]
    SimKernel.CavernSimulationSelectors  = [ getPublicTool('ISF_DefaultParticleKillerSelector') ]
```

## MuonTrackingGeo < Sandbox < TWiki

```
# set the simFlags accordingly (TODO: is this even needed?)
Index: Simulation/ISF/ISF_Config/share/jobOptions_ConfGetter.py
=====
--- Simulation/ISF/ISF_Config/share/jobOptions_ConfGetter.py      (revision 615726)
+++ Simulation/ISF/ISF_Config/share/jobOptions_ConfGetter.py      (working copy)
@@ -52,8 +52,8 @@
 #ISF_Flags.OverrideInputFiles = [ '/afs/cern.ch/atlas/groups/fatras/fatras_input/singlepart_mu_p
 #ISF_Flags.OverrideInputFiles = [ '/afs/cern.ch/atlas/groups/fatras/fatras_input/singlepart_pi_p

-#from ISF_FatrasServices.FatrasTuning import FatrasTuningFlags
-#FatrasTuningFlags.MomCutOffSec = 50.
+from ISF_FatrasServices.FatrasTuning import FatrasTuningFlags
+FatrasTuningFlags.MomCutOffSec = 1000.

from G4AtlasApps.SimFlags import simFlags
simFlags.ReleaseGeoModel = False # Comment this out when not making making DCube plots
@@ -80,7 +80,38 @@

include('ISF_Config/ISF_ConfigJobInclude.py')

+from AGDD2Geo.AGDD2GeoConf import AGDD2GeoSvc
+AGDD2GeoSvc=AGDD2GeoSvc()
+AGDD2GeoSvc.PrintSections=False
+AGDD2GeoSvc.OverrideConfiguration = True
+AGDD2GeoSvc.Locked = False
+AGDD2GeoSvc.DisableSections = False
+AGDD2GeoSvc.Volumes += ["ECT_Toroids"]
+AGDD2GeoSvc.Volumes += ["BAR_Toroid"]
+AGDD2GeoSvc.Volumes += ["Feet"]
+AGDD2GeoSvc.Volumes += ["RailAssembly"]
+AGDD2GeoSvc.Volumes += ["JFSH_Shield"]
+AGDD2GeoSvc.Volumes += ["JTSH_Shield"]
+AGDD2GeoSvc.Volumes += ["JDSH_Shield"]
+AGDD2GeoSvc.Volumes += ["pp2"]
+AGDD2GeoSvc.Volumes += ["servicesAtZ0"]
+AGDD2GeoSvc.Volumes += ["MBAP_AccessPlatform"]
+AGDD2GeoSvc.Volumes += ["MBWH_BigWheels"]
+AGDD2GeoSvc.Volumes += ["SADL_CalorimeterSaddle"]
+AGDD2GeoSvc.Volumes += ["TBWH_BigWheels"]
+AGDD2GeoSvc.Volumes += ["TGC3_BigWheels"]
+AGDD2GeoSvc.Volumes += ["TGC1_BigWheels"]
+AGDD2GeoSvc.Volumes += ["MDTRail"]
+AGDD2GeoSvc.Volumes += ["HFTruckRail"]
+theApp.CreateSvc += ["AGDD2GeoSvc"]
+tsvcMgr += AGDD2GeoSvc

+from TrkExTools.AtlasExtrapolator import AtlasExtrapolator
+AtlasExtrapolator = AtlasExtrapolator()
+AtlasExtrapolator.TrackingGeometrySvc = 'AtlasTrackingGeometrySvc'
+##AtlasExtrapolator.OutputLevel = VERBOSE
+ToolSvc += AtlasExtrapolator
+
# the particle broker THist stream
if ISF_Flags.ValidationMode:
    if not hasattr(ServiceMgr, 'THistSvc'):
@@ -91,6 +122,22 @@
    ServiceMgr.THistSvc.Output += ["ISFFatras DATAFILE='ISFFatras.root' OPT='RECREATE'"]
    ServiceMgr.THistSvc.Output += ["ISFG4SimKernel DATAFILE='ISFG4SimKernel.root' OPT='RECREATE'"]

-from AthenaCommon.ConfigurationShelve import saveToAscii
-saveToAscii("config.txt")
+##from AthenaCommon.ConfigurationShelve import saveToAscii
+##saveToAscii("config.txt")

+
+##from LArTrackingGeometry.LArTrackingGeometryConf import LAr__LArVolumeBuilder
+##LArVolumeBuilder = LAr__LArVolumeBuilder()
```

## MuonTrackingGeo < Sandbox < TWiki

```
+#ToolSvc += LArVolumeBuilder
+#ToolSvc.LArVolumeBuilder.OutputLevel = VERBOSE
+
+#from TileTrackingGeometry.TileTrackingGeometryConf import Tile__TileVolumeBuilder
+#TileVolumeBuilder = Tile__TileVolumeBuilder()
+#ToolSvc += TileVolumeBuilder
+#ToolSvc.TileVolumeBuilder.OutputLevel = VERBOSE
+
+#from CaloTrackingGeometry.ConfiguredCaloTrackingGeometryBuilder import ConfiguredCaloTrackingGe
+#CaloTrackingGeometryBuilder = ConfiguredCaloTrackingGeometryBuilder(name='CaloTrackingGeometryB
+#ToolSvc += CaloTrackingGeometryBuilder
+#ToolSvc.CaloTrackingGeometryBuilder.OutputLevel = VERBOSE
```

### Simulation/ISF/ISF\_Core/ISF\_Services

Index: Simulation/ISF/ISF\_Core/ISF\_Services/src/ParticleBrokerDynamicOnReadIn.cxx

```
=====  
--- Simulation/ISF/ISF_Core/ISF_Services/src/ParticleBrokerDynamicOnReadIn.cxx (revision 615511)  
+++ Simulation/ISF/ISF_Core/ISF_Services/src/ParticleBrokerDynamicOnReadIn.cxx (working copy)  
@@ -162,10 +162,12 @@  
    }  
  
    // setup for validation mode  
+  
    if ( m_validationOutput) {  
  
        // retrieve the histogram service  
        if ( m_thistSvc.retrieve().isSuccess() ) {  
+  
            /* this is crashing the job  
            ATH_CHECK( registerPosValTree( "push_position",  
                                         "push() particle positions",  
                                         m_t_pushPosition) );  
@@ -178,6 +180,7 @@  
            ATH_CHECK( registerPosValTree( "muonExit_pos",  
                                         "MuonExitLayer positions",  
                                         m_t_entryLayerPos[ISF::fAtlasMuonExit] ) );  
+  
            */  
        }  
  
        // error when trying to retrieve the THistSvc  
@@ -417,7 +420,7 @@  
        m_simSelectorSet.insert( m_simSelector[curGeoID].begin(), m_simSelector[curGeoID].end() );  
    }  
  
-    ATH_MSG_DEBUG("Number of unique SimulationSelctors registered: "  
+    ATH_MSG_DEBUG("Number of unique SimulationSelectors registered: "  
        << m_simSelectorSet.size() );  
    }  
  
@@ -441,6 +444,8 @@  
    ISFParticleContainer::iterator particleIterEnd = initContainer.end();  
    for ( ; particleIter != particleIterEnd; ++particleIter) {  
  
+    std::cout <<"looping over input particles:"<< std::endl;  
+  
        // identify the geoID of the particle  
        m_geoIDSvcQuick->identifyAndRegNextGeoID(**particleIter);  
        // the geoID at this point better makes sense :)  
@@ -609,6 +614,7 @@  
        ISF::EntryLayer layer = m_entryLayerToolQuick->registerParticle( *particle, layerInput);  
  
        // ---> if validation mode: fill the corresponding entry layer ROOT tree  
+ /*  
        if ( m_validationOutput) {  
            // fill the push() position TTree  
            fillPosValTree( m_t_pushPosition, *particle);
```

## MuonTrackingGeo < Sandbox < TWiki

```
@@ -616,6 +622,7 @@
    if ( validEntryLayer(layer) )
        fillPosValTree( m_t_entryLayerPos[layer], *particle);
    }
+ */
// <--- end validation output

// validation mode: check whether the particle position corresponds to the GeoID given
```

### Simulation/ISF/ISF\_Geant4/ISF\_Geant4Tools

isfgeant4tools.diff: Differences between rev. copy in table above and the local copy (Too many changes to be put explicitly.)

### Tracking/TrkDetDescr/TrkVolumes

```
Index: Tracking/TrkDetDescr/TrkVolumes/TrkVolumes/SubtractedVolumeBounds.h
=====
--- Tracking/TrkDetDescr/TrkVolumes/TrkVolumes/SubtractedVolumeBounds.h (revision 618906)
+++ Tracking/TrkDetDescr/TrkVolumes/TrkVolumes/SubtractedVolumeBounds.h (working copy)
@@ -101,7 +101,7 @@

inline bool SubtractedVolumeBounds::inside(const Amg::Vector3D &pos, double tol) const
{
- return (m_outer->inside(pos,tol) && !m_inner->inside(pos,0.) );
+ return (m_outer->inside(pos,tol) && !m_inner->inside(pos,-tol) );
}

inline Volume* SubtractedVolumeBounds::outer() const { return m_outer; }
```

### Tracking/TrkExtrapolation/TrkExSTEP\_Propagator

There is no explicit changes between local copy and rev.

### Tracking/TrkExtrapolation/TrkExTools

```
Index: Tracking/TrkExtrapolation/TrkExTools/src/Extrapolator.cxx
=====
--- Tracking/TrkExtrapolation/TrkExTools/src/Extrapolator.cxx (revision 617387)
+++ Tracking/TrkExtrapolation/TrkExTools/src/Extrapolator.cxx (working copy)
@@ -1153,7 +1153,9 @@
    Amg::Vector3D gp = parm.position();
    if ( vol && vol->inside(gp,m_tolerance) ) {
        staticVol = vol;
+ std::cout << vol->volumeName() <<" assigned as input static at:"<< gp.perp()<<" ,"<<gp.z()<<"
    } else {
+ if (vol) std::cout << vol->volumeName() <<"not correctly assigned:"<< gp.perp()<<" ,"<<gp.z()
    staticVol = m_navigator->trackingGeometry()->lowestStaticTrackingVolume(gp);
    const Trk::TrackingVolume* nextStatVol = 0;
    if ( m_navigator->atVolumeBoundary(currPar,staticVol,dir,nextStatVol,m_tolerance) && nextStatVol)
@@ -1525,7 +1527,9 @@
    // check missing volume boundary
    if (nextPar && !(m_currentDense->inside(nextPar->position(),m_tolerance)
        || m_navigator->atVolumeBoundary(nextPar,m_currentDense,dir,assocVol,m_tolerance)
- ATH_MSG_DEBUG( " [!] ERROR: missing volume boundary for volume"<< m_currentDense->volumeName()
+ ATH_MSG_DEBUG( " [!] ERROR: missing volume boundary for volume"<< m_currentDense->volumeName()
+ m_currentDense->inside(nextPar->position(),m_tolerance)<<" at volume boundary"
+ m_navigator->atVolumeBoundary(nextPar,m_currentDense,dir,assocVol,m_tolerance)
    if ( m_currentDense->zOverAtimesRho() != 0.) {
        ATH_MSG_DEBUG( " [!] ERROR: trying to recover: repeat the propagation step in"<< m_highVolumeName()
        m_currentDense = m_highestVolume;
Index: Tracking/TrkExtrapolation/TrkExTools/src/TimedExtrapolator.cxx
=====
--- Tracking/TrkExtrapolation/TrkExTools/src/TimedExtrapolator.cxx (revision 617388)
```

## MuonTrackingGeo < Sandbox < TWiki

```

+++ Tracking/TrkExtrapolation/TrkExTools/src/TimedExtrapolator.cxx      (working copy)
@@ -784,7 +784,10 @@
    // check missing volume boundary
    if (nextPar && !(m_currentDense->inside(nextPar->position(),m_tolerance)
        || m_navigator->atVolumeBoundary(nextPar,m_currentDense,dir,assocVol,m_toleran
-   ATH_MSG_DEBUG( "  [!] ERROR: missing volume boundary for volume"<< m_currentDense->volume
+   //ATH_MSG_DEBUG( "  [!] ERROR: missing volume boundary for volume"<< m_currentDense->volu
+   ATH_MSG_DEBUG( "  [!] ERROR: missing volume boundary for volume"<< m_currentDense->volume
+   m_currentDense->inside(nextPar->position(),m_tolerance)<<" at volume bounda
+   m_navigator->atVolumeBoundary(nextPar,m_currentDense,dir,assocVol,m_toleran
    }
    // if ( m_currentDense->zOverAtimesRho() != 0.) {
    //   ATH_MSG_DEBUG( "  [!] ERROR: trying to recover: repeat the propagation step in"<< m_h
@@ -906,15 +909,16 @@
    if ( dIter != m_denseVols.end() ) {
        currVol = (*dIter).first;
        nextVol = ( (*dIter).first->boundarySurfaces() )[index].getPtr()->attachedVolume(*nextP
-   // the boundary orientation is not reliable
-   Amg::Vector3D tp = nextPar->position()+2*m_tolerance*dir*nextPar->momentum().normalize
-   if (!nextVol || !nextVol->inside(tp,m_tolerance) ) { // search for dense volumes
+   // the boundary orientation is not reliable : step few microns in to see if good choi
+   // to be removed when boundaries smart enough ...
+   Amg::Vector3D tp = nextPar->position()+10*m_tolerance*dir*nextPar->momentum().normaliz
+   if (!nextVol || !nextVol->inside(tp,0.) ) { // search for dense volumes
        m_currentDense = m_highestVolume;
        if (m_dense && !m_denseVols.size()) m_currentDense = m_currentStatic;
        else {
            for (unsigned int i=0;i<m_denseVols.size(); i++) {
                const Trk::TrackingVolume* dVol = m_denseVols[i].first;
-   if ( dVol->inside(tp,m_tolerance) && dVol->zOverAtimesRho()!=0. ) {
+   if ( dVol->inside(tp,0.) && dVol->zOverAtimesRho()!=0. ) {
                m_currentDense = dVol;
                ATH_MSG_DEBUG( "  [+] Next dense volume found: '" << m_currentDense->volumeNam
                break;
@@ -1331,9 +1335,12 @@
        nextGeoID = Trk::GeometrySignature(Trk::Unsigned);
        return currPar;
    }
-   ATH_MSG_DEBUG( "  [+] current static volume resolved by navigator at boundary: "<< positionO
-   <<m_currentStatic->volumeName()<<"->"<< nextVol->volumeName()<<" with toleranc
-   m_currentStatic = nextVol;
+   // boundary orientation not reliable
+   if ( nextVol->inside(currPar->position()+0.01*dir*currPar->momentum().normalized(),0.) ) {
+   ATH_MSG_DEBUG( "  [+] current static volume resolved by navigator at boundary: "<< positio
+   <<m_currentStatic->volumeName()<<"->"<< nextVol->volumeName()<<" with tolera
+   m_currentStatic = nextVol;
+   }
    }
}

// current frame volume known-retrieve geoID
@@ -1592,6 +1599,8 @@
// resolve the use of dense volumes
m_dense = (m_currentStatic->geometrySignature()==Trk::MS && m_useMuonMatApprox ) || (m_current
+ // std::cout <<"current geometry signature:"<< m_currentStatic->geometrySignature() <<","<< m_
+
// reset remaining counters
m_currentDense = m_dense ? m_currentStatic : m_highestVolume;
m_navigBoundaries.clear();
@@ -1689,18 +1698,26 @@
    Amg::Vector3D nextPos = currPar->position()+dir*currPar->momentum().normalized()*m_trSurfs[
    //Amg::Vector3D halfStep = nextPos - 0.5*step*dir*currPar->momentum().normalized();

-   // check missing volume boundary
-   if ( !(m_currentDense->inside(nextPos,m_tolerance) ) ) {
-   ATH_MSG_DEBUG( "  [!] ERROR: missing volume boundary for volume"<< m_currentDense->volume
-   // new search

```



## MuonTrackingGeo < Sandbox < TWiki

```

-     m_currentDense = m_highestVolume;
-     for (unsigned int i=0;i<m_denseVols.size(); i++) {
-         const Trk::TrackingVolume* dVol = m_denseVols[i].first;
-         if ( dVol->inside(nextPos,m_tolerance)  && dVol->zOverAtimesRho()!=0. ) m_currentDense =
+ // check missing volume boundary :
+ if ( !(m_currentDense->inside(nextPos,m_tolerance)) && step>m_tolerance ) {
+     ATH_MSG_DEBUG( "  [!] WARNING: missing volume boundary for volume"<< m_currentDense->volumeName() );
+     unsigned il=0;
+     for ( ; il<int(fabs(step)/0.001); il++) {
+         if ( m_currentDense->inside(nextPos-il*0.001*dir*currPar->momentum().normalized(),0.) )
+             ATH_MSG_DEBUG("last seen inside volume before:"<< il<<" microns");
+         break;
+     }
+ }
-     if (m_dense && m_currentDense==m_highestVolume) m_currentDense = m_currentStatic;
-
-     ATH_MSG_DEBUG( "  [!] new search for dense volume : "<< m_currentDense->volumeName() );
+     if (il>10) { // reassign current dense - use 10 microns tolerance ( limited by distance )
+         // new search
+         m_currentDense = m_highestVolume;
+         for (unsigned int i=0;i<m_denseVols.size(); i++) {
+             const Trk::TrackingVolume* dVol = m_denseVols[i].first;
+             if ( dVol->inside(nextPos,m_tolerance)  && dVol->zOverAtimesRho()!=0. ) m_currentDense =
+                 dVol;
+             if (m_dense && m_currentDense==m_highestVolume) m_currentDense = m_currentStatic;
+         }
+     }
+     ATH_MSG_DEBUG( "  [!] new search for dense volume : "<< m_currentDense->volumeName() );
+ }
+ }

double tDelta = step/beta;
@@ -1726,6 +1743,9 @@

    //std::cout << "looping over intersections:"<<is<<","<< m_trSurfs[sols[is]].second<<","<<step<<endl;

+     if (fr*mDelta>0 && m_currentDense->averageZ(>0) ATH_MSG_VERBOSE( "collecting material from
+         ":thickness[X0]"<<mDelta<<" at position[R,z]"<<nextPos.perp()<<","<<nextPos.z() );
+
+     if (fr<1.) { // decay or material interaction during the step

        int process = frT < frM ? timeLim.process : m_path.process;
@@ -1772,7 +1792,7 @@
        // use global coordinates to retrieve attached volume (just for static!)
        nextVol = (m_currentStatic->boundarySurfaces())[index].getPtr()->attachedVolume(nextPar->position());
        // double check the next volume
-     if ( nextVol && !(nextVol->inside(nextPar->position()+0.01*nextPar->momentum().normalized(),0.) )
+     if ( nextVol && !(nextVol->inside(nextPar->position()+0.01*nextPar->momentum().normalized(),0.) )
        ATH_MSG_DEBUG( "  [!] WARNING: wrongly assigned static volume ?"<< m_currentStatic->volumeName() );
        nextVol = m_navigator->trackingGeometry()->lowestStaticTrackingVolume(nextPar->position());
        if (nextVol) ATH_MSG_DEBUG( "  new search yields: "<< nextVol->volumeName() );
@@ -1794,6 +1814,7 @@
        ATH_MSG_DEBUG( "  [+] Crossing position is " << nextPar->positionOutput() << " - at " << positionOutput(nextPar->position()) << "
        if (!destVol && m_currentStatic->geometrySignature()!=nextVol->geometrySignature())
            { nextGeoID=nextVol->geometrySignature(); return nextPar; }
+     m_currentStatic = nextVol;
    }
    return transportToVolumeWithPathLimit(*nextPar, timeLim, dir, particle, nextGeoID, destVol);
}
@@ -1844,23 +1865,29 @@
currVol = (*dIter).first;

    if ( m_navigator->trackingGeometry()->atVolumeBoundary(nextPos,nextPar->momentum(), currVol) )
-     if ( assocVol && assocVol->zOverAtimesRho()!= 0. ) m_currentDense = assocVol;
-     else if ( currVol->inside(nextPos+0.002*dir*nextPar->momentum().normalized()) ) m_currentDense = currVol;
+     else {
+         if ( assocVol && assocVol->zOverAtimesRho()!= 0. && assocVol->inside(nextPos+0.002*dir*nextPar->momentum().normalized()) )
+             m_currentDense = assocVol;
+         ATH_MSG_VERBOSE( " dense volume assigned by boundary association :"<< m_currentDense->volumeName() );
    }
}

```

## MuonTrackingGeo < Sandbox < TWiki

```
+         } else if ( currVol->inside(nextPos+0.002*dir*nextPar->momentum().normalized(),0.) ) {
+         m_currentDense = currVol;
+         ATH_MSG_VERBOSE( " dense volume assigned by continuation (current) :"<< m_currentDense->volumeName() );
+     } else {
+         // new search
+         m_currentDense = m_highestVolume;
-         if ( m_useMuonMatApprox && !m_denseVols.size() ) m_currentDense = m_currentStatic;
+         if ( m_currentStatic->geometrySignature()==Trk::MS && m_useMuonMatApprox && !m_denseVols.size() )
+         else {
+             for (unsigned int i=0;i<m_denseVols.size(); i++) {
+                 const Trk::TrackingVolume* dVol = m_denseVols[i].first;
-                 if ( dVol->inside(nextPos+0.002*dir*nextPar->momentum().normalized(),m_tolerance) )
+                 if ( dVol->inside(nextPos+0.002*dir*nextPar->momentum().normalized(),0.) && dVol->volumeName() != m_currentDense->volumeName() )
+                     m_currentDense = dVol;
+             }
+         }
+         ATH_MSG_VERBOSE( " dense volume assigned by new search:"<< m_currentDense->volumeName() );
+     }
+ }
+ }
+ } else { // detached volume bounds - not relevant ?
-
+     ATH_MSG_VERBOSE( " detached volume bouns: not relevant ? " );
+ }
+
+ throwIntoGarbageBin(nextPar);
```

### Configuration

- Configuration file: jobOptions\_ConfGetter.py

The

**Sarka e-mail (2-10-14)** the default builds just endcap toroid for dead material and BME stations for active volumes, you can easily change the configuration ). The same script allows you to change the output level for different tools ( for debugging ) and to switch on the VP1 to visualize both geometries ( GeoModel under 'Geo', TrackingGeometry under 'TrkGeo', after loading plugin "All studies", when asked for the name of TrackingGeometrySvc and TrackingGeometryName, fill in : ISF\_FatrasTrackingGeometrySvc and ISF\_FatrasTrackingGeometry ).

The input tracks are shooted in ranges defined in  
mtg/Simulation/ISF/ISF\_Config/python/ISF\_MainConfig.py.

The ntuples contain tree "particles" which gives you, among others, the amount of the material traversed by the track (X0) in a given subdetector (geoID=1 (ID), 3(Calo), 4(MS) ). This is the info used to plot the integrated material maps ( as function of eta and phi ).

Please try to produce these ntuples and make some rough comparisons ( for some of discrepancies you will see I have a fix already, some are not yet understood, we can discussed this next week over skype ).

Let me know if something does not work , or if I forgot to explain, ok ?

**Sarka email (3-10-14)** Hi Jordi,

very well. I have added 2 more packages which contains some fixes ( Tracking/TrkExtrapolation/TrkExSTEP\_Propagator and Tracking/TrkExtrapolation/TrkExTools ). With those, I see a rather good agreement in the MS barrel, there are still differences in endcaps.

The problems with geantino scans is that they do not distinguish between bugs in the tracking geometry and bugs in the extrapolation ( when extrapolator does not resolve correctly the entry/exit to/from dense volume). To have a better handle on the tracking geometry buildup, I have started to code additional information in

[/afs/cern.ch/work/t/todorova/public/mtg\\_detailed](https://afs.cern.ch/work/t/todorova/public/mtg_detailed)

the idea is to run Geant4 only, but to load muon tracking geometry on top of it, and for each G4 material step, verify that there is equivalent material defined in the tracking geometry. The code doing this is in Simulation/ISF/ISF\_Geant4/ISF\_Geant4Tools/PhysicsValidationUserAction , for the moment there is just a simple printout with material found by Geant4 and by the tracking geometry ( line 498 and beyond ) . The script is in the subdirectory run/jo\_g4\_geant. The output ( for 10 geantinos processed ) in in log\_g4\_geant, immediately one can see there are problems with volumes named JDSF\_\* and some parts of endcap toroid / ECT\_\* ).

This is the most direct comparison we can do, so I think we should turn this into permanent validation option (after debugging). The material should be the same in GeoModel and in the tracking geometry , except for known approximations ( conical shieldings are approximated by cylinders at the moment ). It would be very useful if you can play a bit with this code and map the discrepancies ( save them in a separate ntuple tree ). Once we know where the discrepancies come from , we can start to fix them one-by-one in MuonInertTrackingGeometryBuilder.

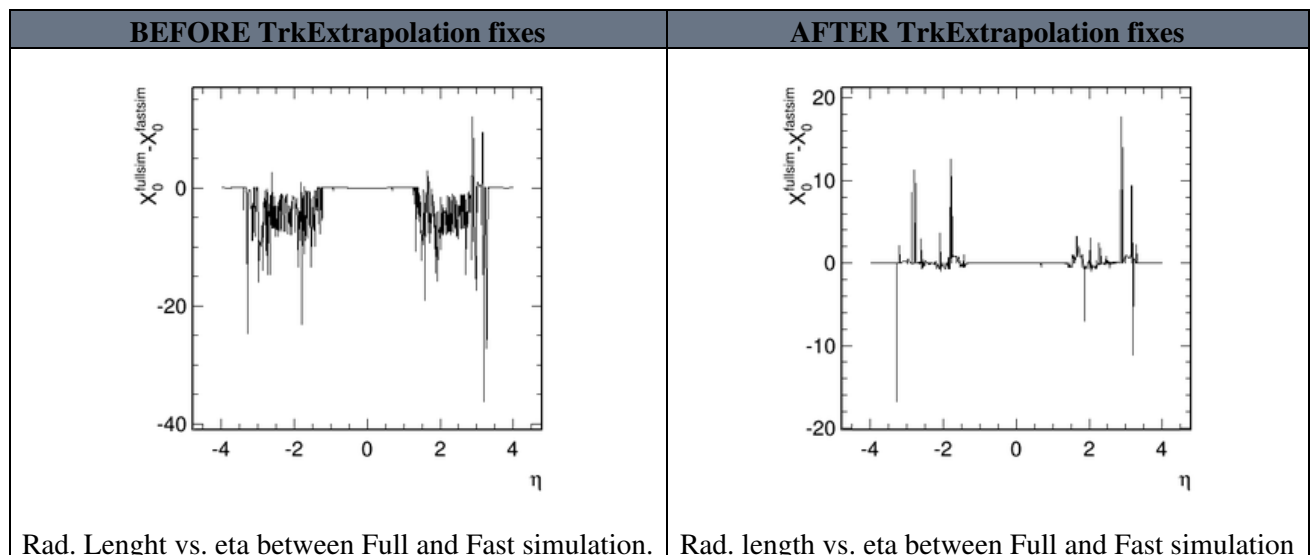
( Sorry the code is a bit chaotic, it is my private version which also does -optionally- detailed validation of Calo volumes and of the energy loss - these I switched off by default )

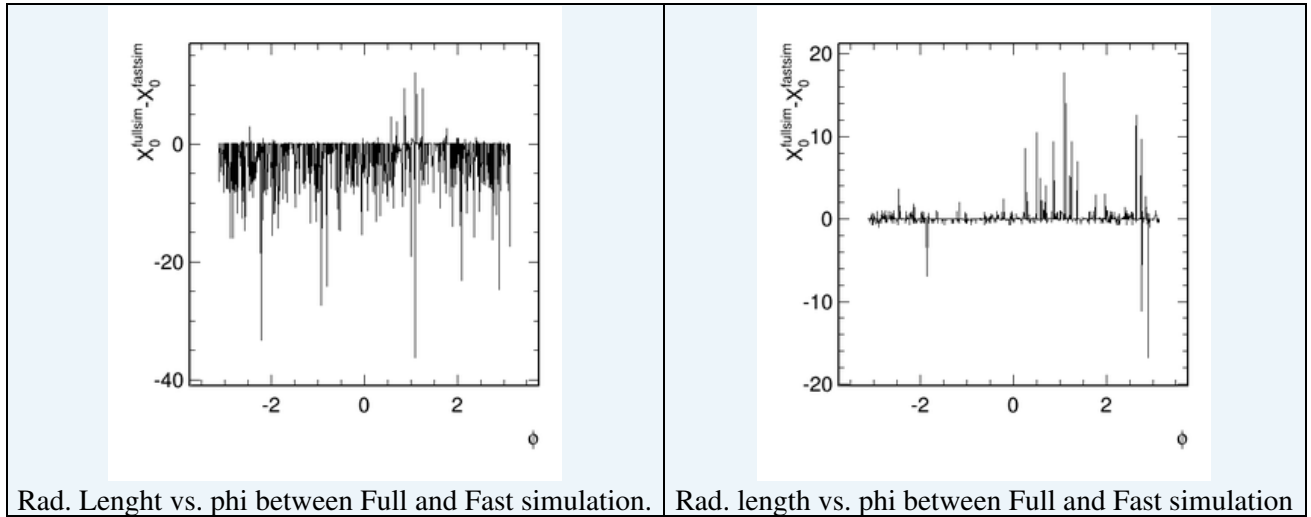
As Jochen explained, we'll need to move to nightlies to be able to work with the latest layout, I'll update the setup early next week if necessary.

cheers,

Sharka =====

**Geantino shots**





Rad. Length vs. phi between Full and Fast simulation.

Rad. length vs. phi between Full and Fast simulation

-- JordiDuarte - 02 Oct 2014

This topic: [Sandbox > MuonTrackingGeo](#)

Topic revision: r8 - 2014-10-20 - JordiDuarte



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)