

Table of Contents

Perfect Geometry Analyzer & Compare DD.....	1
Goal of this page.....	1
PerfectGeometryAnalyzer.....	1
Example, IdealGeometry.....	1
Example, Ideal and Magnetic Field.....	2
TestCompareDDDumpFiles.....	2
Review Status.....	3

Perfect Geometry Analyzer & Compare DD

Complete: 

Goal of this page

Explain how to use the PerfectGeometryAnalyzer and it's companion TestCompareDDDumpFiles.

PerfectGeometryAnalyzer

The PerfectGeometryAnalyzer is designed to dump SpecPars and/or the detector geometry tree with position information.

Example, IdealGeometry

This example shows how to use it for the whole ideal geometry whether from the conditions database or from the original XML files. ([readIdealAndDump.py](#))

```
process.pAStd = cms.EDAnalyzer("PerfectGeometryAnalyzer"
                               ,dumpPosInfo = cms.untracked.bool(True)
                               ,label = cms.untracked.string("")
                               ,isMagField = cms.untracked.bool(False)
                               ,dumpSpecs = cms.untracked.bool(True)
                               ,dumpGeoHistory = cms.untracked.bool(True)
                               ,outFileName = cms.untracked.string("STD")
                               ,numNodesToDump = cms.untracked.uint32(0)
                               ,fromDB = cms.untracked.bool(False)
                               ,ddRootNodeName = cms.untracked.string("cms:OCMS")
                               )
```

The above snippet included in a config file which also includes an XMLIdealGeomtryESSource with a blank label will produce two output files: dumpSTD and dumpSpecsdumpSTD.

PerfectGeometryAnalyzer can also be used to dump the DB content since it is accessing the DDCompactView which is built either by the XML or the conditions database blob. For example the following (partial) configurations dump a simulation geometry scenario from the XML and from the DB then compare the output.

```
import FWCore.ParameterSet.Config as cms
process = cms.Process("DBGeometryTest")
process.load("FWCore.MessageLogger.MessageLogger_cfi")
process.load('Configuration/StandardSequences/GeometryDB_cff')
process.load("CondCore.DBCommon.CondDBSetup_cfi")
process.load('Configuration/StandardSequences/FrontierConditions_CMS.GlobalTag_cff')
from Configuration.PyReleaseValidation.autoCond import autoCond
process.GlobalTag.globaltag = autoCond['mc']
process.source = cms.Source("EmptySource")
process.XMLFromDBSource.label='Ideal'
process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(1)
)

process.GlobalTag.toGet = cms.VPSet(cms.PSet(record = cms.string('GeometryFileRcd'),
                                             tag = cms.string('XMLFILE_Geometry_Extended_TagXX'),
                                             connect = cms.untracked.string("sqlite_file:./myfile")
)
)
```

```

process.myprint = cms.OutputModule("AsciiOutputModule")

XMLFromDBSource = cms.ESProducer("XMLIdealGeometryESProducer",
                                rootDDName = cms.string('cms:OCMS'),
                                )

process.pDB = cms.EDAnalyzer("PerfectGeometryAnalyzer"
                             ,dumpPosInfo = cms.untracked.bool(True)
                             ,label = cms.untracked.string("Ideal")
                             ,isMagField = cms.untracked.bool(False)
                             ,dumpSpecs = cms.untracked.bool(True)
                             ,dumpGeoHistory = cms.untracked.bool(True)
                             ,outFileName = cms.untracked.string("theDBdump")
                             ,numNodesToDump = cms.untracked.uint32(0)
                             ,fromDB = cms.untracked.bool(True)
                             ,ddRootNodeName = cms.untracked.string("cms:OCMS")
                             )

```

Example, Ideal and Magnetic Field

In this example, both the IdealGeometry and the MagneticField dumped. ([readIdealAndMagDump.py](#))

```

#Produces two output files dumpGeoHistory and dumpSpecsGeoHistory
process.prod = cms.EDAnalyzer("PerfectGeometryAnalyzer"
                              ,ddRootNodeName = cms.string("cms:OCMS")
                              ,dumpPosInfo = cms.untracked.bool(True)
                              ,dumpSpecs = cms.untracked.bool(True)
                              ,dumpGeoHistory = cms.untracked.bool(True)
                              ,label = cms.untracked.string("") #actually defaults to blank and I
                              ,isMagField = cms.untracked.bool(False)
                              ,outFileName = cms.untracked.string("GeoHistory") #GeoHistory is th
                              ,numNodesToDump = cms.untracked.uint32(0) #0 means ALL, you can lim
                              )

#Produces two output files dumpMagF and dumpSpecsMagF
process.prodmag = cms.EDAnalyzer("PerfectGeometryAnalyzer"
                                 ,ddRootNodeName = cms.string("cmsMagneticField:MAGF")
                                 ,dumpPosInfo = cms.untracked.bool(True)
                                 ,dumpSpecs = cms.untracked.bool(True)
                                 ,dumpGeoHistory = cms.untracked.bool(True)
                                 ,label = cms.untracked.string("magfield") #actually defaults to bla
                                 ,isMagField = cms.untracked.bool(True)
                                 ,outFileName = cms.untracked.string("MagF") #GeoHistory is the defa
                                 ,numNodesToDump = cms.untracked.uint32(0) #0 means ALL, you can lim

```

After the above snippet is run, the following files should be in your work area: dumpSTD, dumpSpecsdumpSTD, dumpMagF, dumpSpecsdumpMagF which contain the information.

TestCompareDDDumpFiles

The idea here is to compare the output of one PerfectGeometryAnalyzer with another. The syntax is as follows with the parameters defined such that having run the above examples, you would be able to run this as well. [testCompareDumpFiles.py](#)

```

process.comparedddump = cms.EDAnalyzer("TestCompareDDDumpFiles"
                                       ,dumpFile1 = cms.string("dumpSTD")
                                       ,dumpFile2 = cms.string("dumpGeoHistory")
                                       ,tolerance = cms.untracked.double(0.0004)
                                       )

```

If there is no output, then there are no differences in the two copies. Otherwise it specifies which line(s) disagree. IF the actual in-memory hierarchy is very different, it will fail quickly and tell you to look at the dump files directly.

Review Status

Reviewer/Editor and Date (copy from screen)	Comments
MichaelCase - 27 Jan 2006	page author
JennyWilliams - 31 Jan 2007	moved text here from SWGuideDetectorDescription and edited for SWGuide

Responsible: IannaOsborne

Last reviewed by: Reviewer

This topic: CMSPublic > SWGuidePerfectGeometryAnalyzer

Topic revision: r11 - 2011-12-16 - IannaOsborne



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