Table of Contents

Environment **1**
- Code 1
- All concerned files 1
- Main changes 1
  - In ESProducer 1
  - Paths Changes 2
- Code Changes 2
- Files modified 2
  - Example of changes (on the ESHandle theCaloGeometry_ from BetaCalculatorECAL.cc/.h) : 3
  - Modification of each file 4
  - HSCParticleProducer.cc 4

Errors **6**
- Big changes 6
  - After fixing initial issues 7
- Running Step 0 and Step 1 8
  - Changes to HSCParticleProducer_cfg.py 8
  - Related informations I found online 9

Custom step 1 ..... **11**
- Analyzer/python/ParticleAnalyzerCustom_cff.py* 11
- Analyzer/test/ParticleAnalyzerCustom_cfg.py* 11
  - Color code 11
  - Sources 11

Environment

Files: AOD format are needed in order to go through steps 0 and 1 of the HSCP Analyzer :

You can find the whole process of generating your GEN-SIM for run 3, up to AOD format with a custom Hlt menu (hltMenu.py) : https://github.com/DenkMybu/DIGI2NTUPLE

This whole twiki was made for the sub-package HSCP. Changes were not made on the Analyzer/Calibration side, but the way to go will be documented.

Expect whole package to be 12_3_X ready on the 18/03/2022

Code

https://github.com/CMS-HSCP/SUSYBSMAnalysis-HSCP

All concerned files

This table shows every file that is directly impacting the migration. All these files have been modified, and further details are available for each of those in the following sections. I am focusing on the HSCParticleProducer, and all its related files.

Important files
HSCP/plugins/HSCParticleProducer.cc
HSCP/interface/BetaCalculatorMUON.h
HSCP/interface/BetaCalculatorMUON.cc
HSCP/interface/BetaCalculatorECAL.h
HSCP/interface/BetaCalculatorECAL.cc
CalibNtuplizer/plugins/calib_ntuple.cc
HSCP/plugins/HSCPDeDxInfoProducer.h
HSCP/plugins/HSCPTreeBuilder.cc
MuonTiming/BuildFile.xml
/HSCP/plugins/Skim_ReduceHcalRecHitCollectionProducer.cc

Main changes

In ESProducer

NOTE: Registration of products in ESProducers has been enforced since 11_0_0_pre12

The `setWhatProduced()` call returns an `edm::ESConsumesCollector` (the type `REC` is the Record type that the producing function given to `setWhatProduced()` takes as an argument). The collector is then used to register all the data products that may be accessed by the producing function. If the ESProducer registers multiple producing functions, the data access needs to be registered separately for each of them.

Some examples

```
class Producer: public edm::ESProducer {
  ...
  std::unique_ptr<AnyProduct> produce(SomeRecord const& iRecord);
  ...
  edm::ESGetToken<AnyProduct, SomeOrDependentRecord> token1_;
```

```
  edm::ESGetToken<SomeProduct, SomeRecord> token2_;
  edm::ESGetToken<AnotherProduct, DependentRecord> token3_;
};

...

Producer::Producer(edm::ParameterSet const& iConfig) {
  auto cc = setWhatProduced(this);

  // Register data access with type deduction (available since 11_2_0_pre6, earlier one could do
  token1_ = cc.consumes();

  // Register data access with explicit types from the same record as the produce() argument
  token2_ = cc.consumes<SomeProduct>();

  // Register data access with explicit types from a dependent record of the produce() argument
  token3_ = cc.consumesFrom<SomeProduct, DependentRecord>();
}
```

All of the functions above (`consumes()`, `consumesFrom()`) take also an `edm::ESInputTag` as an optional argument.

NOTE: the proper way to migrate from `iRecord.get(handle, label)` is `edm::ESInputTag{"", label}`.

Paths Changes

The following header files have different paths in release 12_3_X. The paths changes have been implemented in every file listed in the table above, but not every file uses evey header.

Header files	New path
PixelBarrelName.h	/DataFormats/TrackerCommon/interface/
SiStripSubStructure.h	/DataFormats/TrackerCommon/interface/
PixelBarrelName.h	/DataFormats/TrackerCommon/interface/
PixelEndcapName.h	/DataFormats/TrackerCommon/interface/
PixelGeomDetUnit.h	/Geometry/CommonTopologies/interface/
CaloTopologyRecord.h	/Geometry/Records/interface/
EcalDetIdAssociator.h	TrackingTools/ TrackAssociator/ plugins/

Code Changes

The main problem in the migration between CMSSW_10_6_X to 12_3_X comes when you take a condition from the event setup

https://github.com/CMS-HSCP/SUSYBSMAnalysis-HSCP/blob/04cc3a00ff93d6780a84f61284aa0bd6902b3a58/HSC

This passes the iSetup to different functions, that must all be changed accordingly.

Files modified

File	ESHandles changed
GeomDumpForFWLite.cc	tkGeom,DtGeom,CscGeom,RpcGeom
Skim_HighPtTrackEcalDetIdProducer.cc	CaloTopology
CSCTimingExtractor_Mini.cc/.h	Propagator
DTTimingExtractor_Mini.cc/.h	Propagator,theDTGeom
MuonSegmentProducer.cc	DtGeom,CscGeom

HSCPDeDxInfoProducer.cc	tkGeom
BetaCalculatorRPC.cc/.h	RPCGeomToken
BetaCalculatorECAL.cc/.h	ecalDetIdAssociator_,bFieldToken_,theCaloGeometry_,CaloTopolog
calib_ntuple.cc	tTopoToken
HSCPTreeBuilder.cc	bFieldToken_
HSCPParticleProducer.cc	details below
HSCParticleSelector.cc	changes ?
ntuple.cc	tTopoToken_,tkGeomToken_,pixelCPEToken_
HSCPValidator.cc	rpcGeoToken_
Analyzer.cc	tTopoToken_,tkGeomToken_,pixelCPEToken_
MuonTimingProducer_Mini.cc	-
Skim_ReduceHcalRecHitCollectionProducer.cc	-
TrackProducerFromPatMuons.cc	-
Skim_UpdatedMuonInnerTrackRef.cc	-
Skim_MonoPhotonSkimmer.cc	-
Skim_HSCPFilter.cc	-
HSCPHLTFilter.cc	-

Example of changes (on the ESHandle theCaloGeometry_ from BetaCalculatorECAL.cc/.h) :

Old version

BetaCalculatorECAL.h

data members

```
edm::ESHandle<CaloGeometry> theCaloGeometry_;
```

BetaCalculatorECAL.cc

addInfoToCandidate function

```
iSetup.get<CaloGeometryRecord>().get(theCaloGeometry_);
const CaloGeometry* theGeometry = theCaloGeometry_.product();
```

New version

BetaCalculatorECAL.h

data members

```
edm::ESGetToken<CaloGeometry, CaloGeometryRecord> theCaloGeometry_;
```

BetaCalculatorECAL.cc

constructor

```
theCaloGeometry_(iC.esConsumes<CaloGeometry, CaloGeometryRecord>())
```

addInfoToCandidate function

```
auto const theGeometry = &iSetup.getData(theCaloGeometry_);
```

Modification of each file

GeomDumpForFWLite.cc

-> Changes in the constructor : added tokens and consumes before bool isInitialized

Skim_HighPtTrackEcalDetIdProducer.cc

-> Why is the private data members declared like const CaloTopology* CaloTopology; and then in the .cc aswell ?

To change all

CSCTimingExtractor_Mini.cc/.h

-> Fixed

DTTimingExtractor_Mini.cc/.h

-> To change

MuonSegmentProducer.cc

->dtGeomToken and cscGeomToken were switched to ESGetToken

HSCPDeDxInfoProducer.cc

-> Fixed

MuonTimingProducer_Mini.cc -> Line 97 : fillTiming function to investigate

BetaCalculatorRPC.cc/.h

-> The rest is functionnal under 12_3_X

Skim_ReduceHcalRecHitCollectionProducer.cc

-> trackAssociator_.associate function in line 157 to investigate : https://cmssdt.cern.ch/lxr/source/TrackingTools/TrackAssociator/src/TrackDetectorAssociator.cc?v=CMSSW_12_3_X

No ESHandle in both following BetaCalculators, everything is commented -> what is it used for ?*

Commented Files
BetaCalculatorMUON.cc/.h
BetaCalculatorTK.cc/.h

HSCParticleProducer.cc

This .cc file is calling a given function from different classes

Class
BetaCalculatorMUON
BetaCalculatorECAL

BetaCalculatorRPC
BetaCalculatorTK

The funtion addInfoToCandidate takes multiple input as following. The 4 classes are used with separate conditions, and **iSetup** is passed to all of them

```
beta_calculator_TK->addInfoToCandidate(*hscpcandidate,  iEvent,iSetup);
beta_calculator_MUON->addInfoToCandidate(*hscpcandidate,  iEvent,iSetup);
beta_calculator_RPC->addInfoToCandidate(*hscpcandidate,  iEvent,iSetup);
beta_calculator_ECAL->addInfoToCandidate(*hscpcandidate,  iEvent,iSetup);
```

HSCPTreeBuilder.cc

-> **Here we have some L1 trigger part, to save to get Eff L1 with selection**

ntuple.cc

-> changed multiple, but some handles don't use _product() (like tkGeom ?)

HSCPValidator.cc

-> No product aswell for RPC Geom ? rpcGeom used in multiple functions, redeclaration of rpcGeo using the token, or declaring it as private member ? (risk of non initialization ?)

Fix -> Redeclaration using the same cstrct token

Changes : Passing iSetup to makeSimDigiPlotsRPC, since we consume the token even pre-constructor, and he Handle is no longer in the data members (the token it though)

Analyzer.cc

-> Problems with tTopoToken, same change as before but different error

```
In file included from /grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAn
/grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAnalysis/Analyzer/plugin
  203 |   const edm::ESGetToken<TrackerTopology, TrackerTopologyRcd> tTopoToken_;
```

Fixed -> Problem occured due to the lack of esConsumes declaration prior to this (dissapeared when **ConsumesCollector.h + ic.esConsumes** were changed)

Errors

Big changes

1. GenEvent.h is missing, should be replaced

Fact : The whole hepMC was rewritten for the new release. A lot of changes are expected, see sub-section HepMC (not needed for step 0/1 I think, for now..)

https://cmssdt.cern.ch/lxr/search?%21v=CMSSW_12_3_X_2022-02-16-2300&_filestring=&_string=GenEvent.h&_ca

We see here every link between GenEvent.h and other classes > https://cmssdt.cern.ch/lxr/search?%21v=CMSSW_12_3_X_2022-02-16-2300&_filestring=&_string=GenEvent.h&_ca

Checking the HepMCPRoduct.h from different releases, noticing a few differences

10_6_20 : https://cmssdt.cern.ch/lxr/source/SimDataFormats/GeneratorProducts/interface/HepMCProduct.h?v=CMSSW_10_6_2
12_3_X : https://cmssdt.cern.ch/lxr/source/SimDataFormats/GeneratorProducts/interface/HepMCProduct.h?%21v=CMSSW_12_

Contacted Joanna Weng and Filip Moortgat.

Error code :

```
...
In file included from /cvmfs/cms.cern.ch/slc7_amd64_gcc10/cms/cmssw/CMSSW_12_3_0_pre4/src/SimData
                 from /grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAn
/cvmfs/cms.cern.ch/slc7_amd64_gcc10/cms/cmssw/CMSSW_12_3_0_pre4/src/SimDataFormats/GeneratorProdu
   11 | #include <HepMC/GenEvent.h>
...
```

2. When scram b -j8 : Warning, Invalid tool SimTrackers/Records. Please fix src/SUSYBSMAnakysis/MuonTiming/BuildFile.xml

Solution remove the following line from BuildFile.xml

```
<use   name="SimTracker/Records"/>
```

3. esConsumes() not declared ?

Solution : esConsumes was added in release 11_0_X, but the way it was done in the PR (https://github.com/cms-sw/cmssw/pull/36095/files) might be different. The definition must be somewhere, but didn't find it yet

FIX IN PROGRESS : **BuildFile.xml has the FWCore included**, and ConsumesCollector.h comes directly from FWCore/interface. Add the subdirectories in here once found

Need help for fixing that

```
...
/grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAnalysis/HSCP/src/BetaCa
/grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAnalysis/HSCP/src/BetaCa
   25 |   bFieldToken_(esConsumes<MagneticField, IdealMagneticFieldRecord>()),
...
```

Solution Tried many things, what works well is the following :

```
CaloTopologyToken_(iC.esConsumes<CaloTopology, CaloTopologyRecord>())
```

4. error: no matching function for call to 'TrackDetectorAssociator::getFreeTrajectoryState(const edm::EventSetup&, reco::Track&) trackAssociator_.getFreeTrajectoryState(iSetup, track)

not wrong (changed but would work)

https://cmssdt.cern.ch/lxr/source/TrackingTools/TrackAssociator/interface/TrackDetectorAssociator.h here we see the 3 ways we can call getFreeTrajectoryState, none takes iSetup as an input ?

FIX > between release 10_6_X and 12_3_X, getFreeTrajectoryState was modified and the input parameters it took before were changed. This function is called in BetaCalculatorECAL.cc for instance. See the chain call

In Skim_ProduceIsolationMap.cc (156)

Also uses the function, but magnetic field is not available there. Addition of classes/includes/tokens to make it work

Solution

Change file `HSCP/plugins/Skim_ProduceIsolationMap.cc` line 206 :

from

```
TrackDetMatchInfo info = trackAssociator_.associate(iEvent, iSetup, trackAssociator_.getFreeTraje
```

to

```
TrackDetMatchInfo info = trackAssociator_.associate(iEvent, iSetup, trackAssociator_.getFreeTraje
```

5. **SimHitShifterRun2.cc -> need to replace for run 3 ?**

6. **SiStripModuleGeometry** but not concerned for HSCParticle Producer. Might want to check this out if we want the Analyzer on 12_3_X

```
/grid_mnt/opt__sbg__cms__ui5_data1/rhaeberl/CMSSW_12_3_0_pre4/src/SUSYBSMAnalysis/Analyzer/interf
  992 |   int moduleGeometry = SSdetId.moduleGeometry();
      |                        ~~~~~~~~~~~~~~~~~~~~~~^~
      |                                              |
      |                                              SiStripModuleGeometry
```

After fixing initial issues

1. **LaunchOnCondor**

```
*** Error compiling 'src/SUSYBSMAnalysis/HSCP/python/LaunchOnCondor.py'...
  File "src/SUSYBSMAnalysis/HSCP/python/LaunchOnCondor.py", line 60
    print 'LaunchOnCondor [options]'
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print('LaunchOnCondor [options]
```

This script was made by Loic a long time ago, hide it and everything compiles fine (renamed LaunchOnCondor.pynocomp)

Running Step 0 and Step 1

Changes to HSCParticleProducer_cfg.py

1)FrontierConditions GT

```
ModuleNotFoundError: No module named 'Configuration.StandardSequences.FrontierConditions_GlobalTa

At:
  /cvmfs/cms.cern.ch/slc7_amd64_gcc10/cms/cmssw/CMSSW_12_3_0_pre4/python/FWCore/ParameterSet/Conf
  HSCParticleProducer_cfg.py(52): <module>
```

-> Fix : Find the right module :

Inspired from > https://jiafulow.github.io/blog/2017/11/08/cmssw-globaltag/

changed L52 :

```
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_condDBv2_cff')
```

to

```
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')
```

and L74-75

```
from Configuration.AlCa.GlobalTag_condDBv2 import GlobalTag
process.GlobalTag = GlobalTag(process.GlobalTag, options.GTAG, '')
```

to

```
from Configuration.AlCa.GlobalTag import GlobalTag
process.GlobalTag = GlobalTag(process.GlobalTag, '123X_mcRun3_2021_realistic_v4')
```

2)itervalues (no idea why it is here)

```
----- Begin Fatal Exception 18-Feb-2022 16:21:54 CET-----------------------
An exception of category 'ConfigFileReadError' occurred while
   [0] Processing the python configuration file named HSCParticleProducer_cfg.py
Exception Message:
 unknown python problem occurred.
AttributeError: 'FixedKeysDict' object has no attribute 'itervalues'

At:
  HSCParticleProducer_cfg.py(187): <module>
```

Fix :

-> Changes in the config file (HSCParticleProducer_cfg.py)

dict.itervalues() was removed from Python3, use instead dict.values()

#187 and #189

```
for mod in process.producers_().values():
```

The migration is complete, and the HSCParticleProducer compiles

3)Lack of TriggerResults::HLT leads to nonsense results

When running the code, new error to fix

```
%MSG-e HLTHighLevel:  HLTHighLevel:HSCPTrigger 21-Feb-2022 15:57:17 CET  Run: 1 Event: 500
TriggerResults product TriggerResults::HLT not found - returning result=false!
```

Wrong Fix (but still important) : -> Added the collections inside HLT2AOD.py

```
# Input source
process.source = cms.Source("PoolSource",
    fileNames = cms.untracked.vstring('file:mitL1/hltOutput_withL1_DIGIrun3.root'),
    inputCommands = cms.untracked.vstring(
        'keep *'
    ),
    secondaryFileNames = cms.untracked.vstring()
)
```

The inputCommands (line 4) was not there initially, and that was causing the issue

Right fix

From HLT to HLTX

```
process.HSCPTrigger.TriggerResultsTag = cms.InputTag( "TriggerResults", "", "HLTX" )
```

4)MustUseEsGetToken


```
Begin processing the 1st record. Run 1, Event 1, LumiSection 1 on stream 0 at 22-Feb-2022 11:24:1
----- Begin Fatal Exception 22-Feb-2022 11:24:22 CET-----------------------
An exception of category 'MustUseESGetToken' occurred while
   [0] Processing  Event run: 1 lumi: 1 event: 1 stream: 0
   [1] Running path 'HSCPTuplePath'
   [2] Calling method for module MuonSegmentProducer/'MuonSegmentProducer'
Exception Message:
Called EventSetupRecord::get without using a ESGetToken.
 While requesting data type:DTGeometry label:''
See https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideHowToGetDataFromES
for instructions how to migrate the calling code
----- End Fatal Exception -------------------------------------------------
Another exception was caught while trying to clean up files after the primary fatal exception.
```

Fix : -> Changed ESHandle dtGeeom and cscGeom

The whole step 0 was migrated towards CMSW_12_3_X, compiles and results are coherent with expectencies

Related informations I found online

https://github.com/cms-sw/cmssw/issues/31061 (Migrate modules for the use of esConsumes)

https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideFrameWork (general SW framework guide, very dense)

Custom step 1

Once step 0 done, one needs to do the reconstruction on the HSCP candidates, to then access the values of L1 seeds efficiencies aswell as HLT paths, with the given selection (see table 1 below)

Either : Make new EDAnalyzer from scratch, or run step 1 ?

Analyzer/python/ParticleAnalyzerCustom_cff.py*

Change 1 > L.22 put HLTX to be consistent with the prior changes

```
,triggerResults = cms.InputTag("TriggerResults","","HLTX")
```

Change 2 > L.33-36

```
,TypeMode         = cms.untracked.uint32(0) # 0:Tk only, 1:Tk+Muon, 2:Tk+TOF, 3:TOF onlypwd, 4:Q>1
    ,SampleType       = cms.untracked.uint32(0) # 0:Data, 1:Background, 2:Signal, 3:Signal Systema
    ,SampleName       = cms.untracked.string("BaseName")
    ,Period           = cms.untracked.string("2016")
```

Analyzer/test/ParticleAnalyzerCustom_cfg.py*

Color code

= problem is fixed and I am confident in the fix

= problem is fixed but the solution needs review

= problem has no solution yet

= warning but no errors

Sources

Big thanks to Tamas Almos Vami for answering a lot of my questions and helping out on this

-- Main.RaphaelJulienHaberle - 2022-02-13

This topic: Main > HSCPMigrationTo123X
Topic revision: r17 - 2022-02-23 - RaphaelJulienHaberle