Table of Contents

SUSYDPDMaker

Introduction

The SUSYDPDMaker package is a utility created for the SUSY WG users to make primary DPDs and, eventually, also to create secondary and tertiary DPDs. It takes ESDs, AODs and also SUSYDPDs as input and creates slimmed/thinned/skimmed DPDs as output in the form of pool.root files. The AOD->DPD job (as well as D1PD->D2PD->D3PD) is configured by python scripts. Slimming/thinning/skimming of containers/events are driven by the same SUSYDPD_jobOption.py file but are all independent from each other and will create different Streams and separate outputs. PyParticleTools are used to create the primary and secondary DPDs. The slimming algorithm that are used to create the primary DPDs are those defined in the DPDUtils package.

The current version of SUSYDPDMaker is 00-00-02. This includes all the slimming/thinning/skimming scripts/examples defined below.

Any comment/suggestions can be posted to the SUSY WG hypernews or directly to Fabrizio Salvatore.

Installing SUSYDPDMaker at CERN

In the following you can find the instructions on how to install SUSYDPDMaker in a 13.0.40 release at CERN.

- First, prepare your account to run the package; in the following, the package is installed in a directory called "test" that is defined in the "testarea" in your $HOME direcotry:

```
> mkdir $HOME/testarea
> mkdir $HOME/testarea/test
> mkdir $HOME/testarea/test/cmthome
> cd $HOME/testarea/test/cmthome
> source /afs/cern.ch/sw/contrib/CMT/v1r20p20070720/mgr/setup.sh
```

- Make the requirements file in cmthome, then run config and setup.sh:

```
#-------------------------------------------------------
set   CMTSITE         CERN
set   SITEROOT        /afs/cern.ch
macro ATLAS_DIST_AREA ${SITEROOT}/atlas/software/dist
macro ATLAS_TEST_AREA ${HOME}/testarea/test
apply_tag setup
apply_tag simpleTest
use AtlasLogin AtlasLogin-* $(ATLAS_DIST_AREA)
#-------------------------------------------------------

> cmt config
> source setup.sh -tag=13.0.40,32
> cd ..
> mkdir $HOME/testarea/test/13.0.40
> cd $HOME/testarea/test/13.0.40
```

- Then, get (and compile) the required packages:

```
> cmt co -r DPDUtils-00-00-09 PhysicsAnalysis/DPDUtils
> cd PhysicsAnalysis/DPDUtils/cmt
> source setup.sh
> make
> cd -

> cmt co -r PyParticleTools-00-00-26 PhysicsAnalysis/PyAnalysis/PyParticleTools
> cd PhysicsAnalysis/PyAnalysis/PyParticleTools/cmt
> source setup.sh
> make
> cd -

> cmt co -r McParticleTools-00-10-04 PhysicsAnalysis/TruthParticleID/McParticleTools
> cd PhysicsAnalysis/TruthParticleID/McParticleTools/cmt
> source setup.sh
> make
> cd -

> cmt co -r SUSYDPDMaker-00-00-03 PhysicsAnalysis/SUSYPhys/SUSYDPDMaker
> cd PhysicsAnalysis/SUSYPhys/SUSYDPDMaker/cmt
> source setup.sh
> make
> cd ..
```

- Now create a 'run' directory that you will use to run your AOD-> DPD jobs:

```
> mkdir run
> cd run
```

At this point, the installation is finished and you can configure your job to make the DPDs.

Running SUSYDPDMaker at CERN

The main script to run the AOD->DPD job is share/SUSYDPDMaker.py.

share/SUSYDPD_PoolOutput.py is the script that contains the list of containers that are selected for the primary (and also secondary/tertiary) DPDs. A list of all the available AOD/ESD containers can be found on the Atlas.AODClassSummary page.

The jobOptions file share/SUSYDPD_jobOptions.py is used to customize the job and is sourced by the SUSYDPDMaker.py script.

To create or modify pool.root DPDs, examples of slimming, thinning and skimming are provided in SUSYDPDMaker/share. For primary DPD production, only the slimming algorithms will be applied. The thinning and skimming algorithms that are described below are only examples of what could be done in order to produce secondary and tertiary DPDs and are provided to help users to write their own thinning/skimming algorithms.

Slimming

By slimming the containers, object that are not of interest for the analysis are removed in order to reduce the overall size of the container itself (e.g. error matrix for tracks with Pt < 5 GeV). In SUSYDPDMaker the jet and track slimming scripts defined in DPDUtils/share and the truth slimming script defined in SUSYDPDMaker/share/TruthSlimming.py are used. Example of slimming can be seen in share/SUSYDPD_PoolOutput.py. If one wants to add and then execute more slimming scripts in the job definition, the slimming scripts have to be defined in SUSYDPDMaker/share/SUSYDPD_Slimming.py.

The environment variable:

```
ApplySlimming = True
```

is defined in SUSYDPD_jobOptions.py in order to run the jet and track slimming in the AOD->DPD job. If also

```
WriteTruthInfo = True
```

the truth slimming is applied as well to obtain the primary DPD. The "slimmed" containers are then added to a

```
GenericSUSYStream.ItemList[]
```

in SUSYDPD_PoolOutput.py . The slimming algorithms have to be registered to an output stream in SUSYDPD_PoolOutput.py like this:

```
GenericSUSYStream.RequireAlgs=["slimJets", "slimTacks"]
```

Thinning

Thinning consist of writing out only a fraction of a container. The "thinned" containers are added to a

```
ThinnedSUSYStream.ItemList[]
```

stream in SUSYDPD_PoolOutput.py (as for the "slmming" above).

In this version of SUSYDPDMaker there are 2 different ways to perform the thinning of containers.

Thinning using the ThinningSvcWrapper in DPDUtils

To use this method, set the **UseAtlas.ThinningSvc** variable to **True** SUSYDPD_jobOptions.py. There is one example of thinning algorithm in SUSYDPDMaker/share/EleThinAlgo.py that can be used as a template. This algorithm will thin the ElectronAODContainer by selecting only those candidates with:

```
pt > 5 GeV
author = 1
```

The thinning algorithms have to be declared in share/SUSYDPD_Atlas.ThinningSvc.py and then registered to an output stream in SUSYDPD_PoolOutput.py, like this:

```
ThinnedSUSYStream.AcceptAlgs=["EleThinAlg"]
```

The thinned container in the output DPD will have the same name of the original container (e.g. ElectronAODContainer in the example provided). Only the following containers can be thinned using this service:

```
INavigable4MomentumCollection
ElectronContainer
Analysis::MuonContainer
ParticleJetContainer
Rec::TrackParticleContainer
CaloClusterContainer
```

Thinning using the src/DPDThinner.cxx method (deep copy)

To use this method, set the **UseAtlas.ThinningSvc** variable to **False** SUSYDPD_jobOptions.py.

An example of thinning can be found in SUSYDPDMaker/src/DPDThinner.cxx, which is used to only select candidates in the Electron, Photon, Muon Tau and Jet containers that satisfy a certain requirement (e.g. Pt > 5 GeV). The thinning algorithms can be controlled via the jobOptions file SUSYDPDMaker/share/SUSYDPD_Thinning.py.

The resulting output containers are a 'deep-copy' of the original containers and the name of the thinned containers will have to be different to the name of the original ones (e.g. in case of thinning of the ElectronAODContainer, the thinned container in the DPD will have to be called for example NewElectronAODContainer)

Skimming

Skimming is a method to filter out the events to be written in the output DPD. Some skimming algorithms are currently available in SUSYDPDMaker to skim events using specific cuts on Electron, Photon, Muon, MissingET, Tau, Jet and/or Track containers. Examples can be found in:

```
share/EleFilterAlg.py
share/GammaFilterAlg.py
share/MuFilterAlg.py
share/MissEtFilterAlg.py
share/TauFilterAlg.py
share/JetFilterAlg.py
share/TrackFilterAlg.py
```

For the skimming algorithms to be effective, they need to be registered to an output stream in SUSYDPD_PoolOutput.py, like this:

```
SkimmedSUSYStream.AcceptAlgs=["EleFilterAlg, MuFilterAlg, TauFilterAlg"]
```

if they are to be applied in OR, or like this:

```
SkimmedSUSYStream.RequireAlgs=["EleFilterAlg, MuFilterAlg, TauFilterAlg"]
```

if they are to be applied in AND. Filter algorithms have to be declared in SUSYDPDMaker/share/SUSYDPD_Skimming.py in order to be applied in the poolOutput script. The "skimmed" containers are then added to a

```
SkimmedSUSYStream.ItemList[]
```

in SUSYDPD_PoolOutput.py (same as for the "slimming" and "thinning" above).

Running SUSYDPDMaker interactively for test purposes

For making some interactive AOD->DPD tests an input AOD and output DPD file can be defined in this SUSYDPD_jobOptions.py file. Once the job is set up, go in in the 'run' directory and do:

```
> athena SUSYDPDMaker/SUSYDPDMaker.py
```

This is useful to make an interactive test that the installation and set up works correctly. To run primary DPD production the Grid tools will be used (see below for an example).

Making primary DPDs using PandaTools

An easy way to produce the DPDs and make them available to everyone via dq2 is to run SUSYDPDMaker with "pathena". In order to be able to do it, PandaTools have to be installed in your "testarea" following the instructions below.

- Go in the test area:

```
cd $HOME/testarea/test/13.0.40
```

- Install the Atlas.Panda tools:

```
> cmt co -r PandaTools-00-00-08 PhysicsAnalysis/DistributedAnalysis/PandaTools
> cd PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt
> source setup.sh
> make
> cd -
```

- Go in your SUSYDPDMaker/cmt directory

```
> cd PhysicsAnalysis/SUSYPhys/SUSYDPDMaker/cmt
```

- Modify the "requirements" by un-commenting the following line:

```
#use PandaTools                  PandaTools-00-00-08      PhysicsAnalysis/DistributedAnalysis
```

- Run config, setup.sh and then make:

```
> cmt config
> source setup.sh
> make
```

- Go in your 'run' directory and run "pathena":

```
> cd ../run
> pathena SUSYDPDMaker/SUSDPDMaker.py --inDS <my_input_dataset> --outDS <my_output_dataset>
```

where

```
 <my_input_dataset>
```

is the input AOD dataset to be processed and

```
 <my_output_dataset>
```

is the corresponding output dataset. The name of the output dataset has to be of the type:

```
user.MyGridLoginName.datasetname
```

Once the job has finished, the output dataset will be available on dq2. Please see the workbook instructions on DQ2 to get info about how to retrieve a dataset using the dq2 commands.

More information on Atlas.Panda can be found here.

Major updates:
-- FabrizioSalvatore - 17 Mar 2008

%RESPONSIBLE% FabrizioSalvatore
%REVIEW% **Never reviewed**

This topic: AtlasArchive > SUSYDPDMaker
Topic revision: r4 - 2008-06-24 - StephenHaywood