

# Table of Contents

<b>LHCb Stripping FAQ - Re-Stripping of MC in MicroDST format.....</b>	<b>1</b>
Motivation.....	1
Restrictions.....	1
How To.....	1
Did the algorithm run?.....	1
Check out Packages.....	1
Making changes to your.....	2
Making changes to the.....	2
Making Changes to CommonParticles.....	2
Automation of the above.....	3

# LHCb Stripping FAQ - Re-Stripping of MC in MicroDST format

## Motivation

Why would you want to re-strip your MicroDST?

- You want to change your stripping cuts for efficiency studies or because your stripping selection changed for data
- You want to fix a bug that can only be corrected by re-processing (like the TISTOS bug)
- You want to resample your PID and therefore need to remove all PID cuts during stripping and in the standard particles.
- some other cool idea you came up with

Although this page looks awfully long for a simple task like re-stripping, it is quite simple to accomplish. Please read carefully. Attached to this page is a working example, where you can test everything (lb-run DaVinci/vXXrXpX gaudirun.py example.py).

## Restrictions

In the case where your MC is persisted in the DST format, reconstruction information of the entire event is saved. This is not the case for MicroDST MC. It is unlikely, that re-stripping will result in the same data-sample that you would have got if you applied your stripping line during production. However, there is an algorithm (CopySignalMCParticles) in the MicroDST production that copies reconstruction information matched to signal MC-Particles. It is important to note that the candidates you can get out of MicroDST either passed any stripping line or are truth-matched (associated to the signal MC particles). This is different to DST, where you can in principal get every candidate.

Here is a short list of additional restrictions:

- All calculation requiring information of more than the candidate only, like Isolation variables in RelatedInfoTools, cannot be performed during MicroDST Re-stripping and must be left out.
- Not every TupleTool is tested in MicroDST re-stripping. If you encounter problems with one of them, contact the creator of the tool.
- (rather a warning) The presented method is not yet well validated (February 2018). It was only tested on a few Rare-Decay StrippingLines and only with DaVinci versions newer than DaVinci/v42r6p1.

## How To

### Did the algorithm run?

Firstly, one has to check whether that algorithm ran during the production (most likely yes). Therefore, look at the Option files of your production (can be found in the bookkeeping [↗](#)) and check whether the "isMC" flag is set to true in stripMicroDSTElements and in stripMicroDSTStreamConf in the SelDSTWriter algorithm. In most cases the Writer is defined in "\$APPCONFIGOPTS/DaVinci/DV-Stripping-MC-muDST.py".

### Check out Packages

**Note** that you could in principal cover all following steps within your Optionsfile (here) and would not need to checkout anything (for advanced users), but for educational purposes the manual procedure is described here.

## RestrippingMDST < LHCb < TWiki

To make the necessary changes to your stripping line (remove RelatedInfoTools and such, changing cuts), check out the right DaVinci and your desired stripping package (StrippingSelections or StrippingArchive). My example will use StrippingSelections, which should be equivalent to StrippingArchive if you checkout the right version of stripping (see release notes of your DaVinci version).

```
lb-dev DaVinci/vXXrXpX
cd ./DaVinciDev_vXXrXpX
git lb-use Stripping
git lb-checkout Stripping/vXXrXXpX Phys/StrippingSelections
```

Next up, you also need to checkout CommonParticles (CommonParticlesArchive):

```
git lb-use Phys
git lb-checkout Phys/vXXrXXpX Phys/CommonParticles
make
```

## Making changes to your

To be added/changed in your OptionsFile:

```
DaVinci().RootInTES = "/Event/AllStreams"
DaVinci().InputType = "MDST"
```

In some cases, the StrippingConf contains a "ProcStatusCheck()" as BadEventSelection. This will cause a FATAL, unless you set it's RootInTES during instantiation:

```
ProcStatusCheck(RootInTES = "/Event")
```

## Making changes to the

Your stripping line must not contain RelatedInfoTools or any other algorithm that requires non-candidate information. In this example, commenting out RelatedInfoTools is sufficient.

```
self.phi2MuMuControlLine = StrippingLine(PhimumuControl_name+'Line',
                                           prescale = config['Phi2MuMuControlPrescale'],
                                           postscale = config['Postscale'],
                                           #RelatedInfoTools = config['RelatedInfoTools_Phi2MuMuC
                                           MDSTFlag = False,
                                           #RequiredRawEvents = ['Velo', 'Muon', 'Calo'],
                                           algos = [ self.selPhi2MuMuControl ]
                                           )
```

## Making Changes to CommonParticles

Additionally, changes have to be made to the CommonParticles your StrippingLine depends on. In this example we adjust StdLooseMuons, which again depend on StdAllLooseMuons.

In "Phys/CommonParticles/python/CommonParticles/StdLooseMuons.py":

```
## create the algorithm
algorithm = FilterDesktop( 'StdLooseMuons',
                           Inputs = ["Phys/StdAllLooseMuons/Particles"],
                           RootInTES = "/Event/AllStreams", #this argument has to be
                           Code = defaultCuts() )

## configure Data-On-Demand service
locations = updateDoD ( algorithm, hat = "AllStreams/Phys/" ) #the hat argument has to be

## finally: define the symbol
```

## RestrippingMDST < LHCb < TWiki

StdLooseMuons = algorithm

Proceed equivalently in StdAllLooseMuons.py:

```
## create the algorithm
algorithm = CombinedParticleMaker ( 'StdAllLooseMuons',
                                   RootInTES = "/Event/AllStreams",           #this argument h
                                   Particle = 'muon' )

# configure the track selector
selector = trackSelector ( algorithm )

# protoparticle filter:
fltr = protoFilter ( algorithm , ProtoParticleMUONFilter, 'Muon' )
fltr.Selection = [ "RequiresDet='MUON' IsMuon =True" ]

## configure Data-On-Demand service
locations = updateDoD ( algorithm, hat = "AllStreams/Phys/" )           #the hat argument has to be

## finally: define the symbol
StdAllLooseMuons = algorithm
```

## Automation of the above

**Disclaimer: The following code are quick & dirty solutions (to be improved)**

**If you know what you do**, the changes made to CommonParticles can be automated in your OptionsFile. As an example, one could use the following code snippet to set the RootInTES argument if all the CommonParticles to be adjusted are part of StandardBasic ("Phys/CommonParticles/python/CommonParticles/StandardBasic.py"):

```
import CommonParticles
algos_basic = [eval('CommonParticles.StandardBasic.{}'.format(i)) for i in dir(CommonParticles.St
for alg in algos_basic:
    alg.RootInTES = '/Event/AllStreams'
```

To adjust the DataOnDemand service (updateDoD), one could use the following snippet in the OptionsFile. Do not use this if you do not know what it does and what could possibly go wrong:

```
def automate_DoD_mdst():
    from Configurables import DataOnDemandSvc
    dod = DataOnDemandSvc()
    for key in dod.AlgMap.keys():
        if "Phys/Std" in key:
            dod.AlgMap["AllStreams/" + key] = dod.AlgMap.pop(key)

from Gaudi.Configuration import appendPostConfigAction
appendPostConfigAction(automate_DoD_mdst)
```

To automate the removal of RelatedInfoTools, one could use this snippet when creating the stream in your OptionsFile:

```
for stream in streams:
    for line in stream.lines:
        if line.name() in MyLines:
            line.RelatedInfoTools = []
            MyStream.appendLines( [ line ] )
```

For the given example, re-stripping MC MicroDST with LFPPhi2MuMuControlLine, the shown automation would cover all necessary changes without the need to check out DaVinci at all.

## RestrippingMDST < LHCb < TWiki

You are invited to improve this page over time, better automations, more convenient workarounds and so on!

-- NiklasStefanNolte - 2018-02-20

---

This topic: LHCb > RestrippingMDST

Topic revision: r7 - 2018-02-23 - NiklasStefanNolte



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors. Ideas, requests, problems regarding TWiki? Send feedback