Table of Contents

division

This page is to collect information regarding the bandwidth division for 2015 data taking. TOC:

Data Preparation

We started from a selection of the samples prepared for the physics WGs by Conor. The min bias sample was a separate production with Conditions: Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8 (500k mag up, 500k mag down). Applying the L0 to this data goes as follows.

1. All scripts are run on slc6, after SetupProject Moore v22r0, with gaudirun,py. To recreate my environment follow the steps in /afs/cern.ch/user/e/evh/w0/forAlvaro/README.
2. The data then has to be copied from the Grid as follows:

```
lhcb-proxy-init
SetupProject LHCbDIRAC
lhcb_bkk
```

   Find the LFN of the dataset, e.g. /lhcb/MC/Dev/MCFILTER.DST/00033990/0000/00033990_00000001_1.mcfilter.dst. Copy it to some afs workspace by using Thomas' myDiracCopy.py:

```
# SetupProject LHCbDirac
# lhcb-proxy-init
from DIRAC.Core.Base import Script
Script.parseCommandLine(ignoreErrors = True )
from DIRAC.Interfaces.API.Dirac import Dirac
dirac = Dirac()
import os

if len(os.sys.argv) > 1 :
 lfn = os.sys.argv[1]
else:
# 2010 bincl MC
 lfn =  'LFN:/lhcb/MC/2010/DST/00006385/0000/00006385_00000533_1.dst'
 lfn =  'LFN:/lhcb/data/2010/DIMUON.DST/00007045/0000/00007045_00000054_1.dimuon.dst'

l_local_file = lfn.rfind('/')+1
local_file = lfn[l_local_file:]

result = dirac.getReplicas(lfn,printOutput=True)
print result
if result['OK'] :
 txt  = result['Value']['Successful']
 dict = txt[lfn.replace('LFN:','')]
 dest1 = dict.keys()[0]
 srm = dict.values()[0]
 sc = os.system('lcg-cp '+srm+' '+local_file)

 print sc
```

3. The datasets were reduced to 1k events (see /afs/cern.ch/user/e/evh/w0/bw/makesignalfiles.py):

```
#!/user/bin/env python
import GaudiPython
from LHCbConfig import *
from Gaudi.Configuration import *
from Configurables import LHCbApp
from LHCbConfig import *
from Configurables import InputCopyStream
InputCopyStream().Output = "DATAFILE='PFN:/afs/cern.ch/user/e/evh/updateL0/bw/21263002.
LHCbApp.DataType = "2012"
LHCbApp.DDDBtag = 'head-20120413'
LHCbApp.CondDBtag = 'head-20120420'
from Configurables import EventClockSvc, CondDB
EventClockSvc().EventTimeDecoder = "OdinTimeDecoder"
CondDB(IgnoreHeartBeat = True)
appConf = ApplicationMgr( OutputLevel = INFO, AppName = 'readtest',OutStream=[InputCopy
appMgr = GaudiPython.AppMgr()
sel=appMgr.evtsel()
sel.open(['root://castorlhcb.cern.ch//castor/cern.ch/user/c/chaen/bw/21263002/00033552_
appMgr.algorithm('InputCopyStream').Enable=False
evt = appMgr.evtsvc()
while nwrite<1000 :
   appMgr.run(1)
   if evt['/Event/DAQ']==None :
      print 'No Event/DAQ found'
   continue
appMgr.algorithm('InputCopyStream').execute()
```

3. Run the raw event juggler to move the raw event around for processing (see /afs/cern.ch/user/e/evh/w0/bw/runJuggler.py):

```
#
from Gaudi.Configuration import *
from Configurables import LHCbApp
LHCbApp()
from GaudiConf import IOHelper
IOHelper().inputFiles([ 'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/bwdivisio
from Configurables import GaudiSequencer,RawEventJuggler
ApplicationMgr().TopAlg+=[GaudiSequencer("Spam")]
Writer=InputCopyStream("MyStream")
IOHelper().outStream("11102003-juggled.dst", writer=Writer)
import RawEventCompat
RawEventJuggler().Input=2.0
RawEventJuggler().Output=0.0
RawEventJuggler().Sequencer=GaudiSequencer("Spam")
RawEventJuggler().WriterOptItemList=Writer
RawEventJuggler().KillExtraNodes=True
RawEventJuggler().KillExtraBanks=True
RawEventJuggler().KillExtraDirectories=True
```

4. Run the L0App to apply the L0 to the data (see /afs/cern.ch/user/e/evh/w0/bw/runL0.py):

```
#!/user/bin/env python
from Gaudi.Configuration import *
from Configurables import L0App
L0App().Simulation=True
L0App().TCK = '0x1810'
L0App().outputFile='13774002-L0.dst'
from GaudiConf import IOHelper
IOHelper().inputFiles([ 'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/bwdivisio
```

Datasets (prefix /castor/cern.ch/user removed)

- BWDivisionDatasetsForDec14LHCbWeek

Event type	Channel	Raw from Grid	1k events
11102003	B02Kpi	/c/chaen/bw/11102003/00033993_00000002_1.mcfilter.dst	/e/evh/bwdivision/11102003.c
11114001	B2KstMuMu	/c/chaen/bw/11114001/00033856_00000005_1.mcfilter.dst	/e/evh/bwdivision/11114001.c
11124001	Bd2Kstaree	/c/chaen/bw/11124001/00033106_00000007_1.mcfilter.dst	/e/evh/bwdivision/11124001.c
13774002	Bs2Dsmuantinu	/c/chaen/bw/13774002/00033582_00000006_1.mcfilter.dst	/e/evh/bwdivision/13774002.c
12103035	Bplus2KKPi	/c/chaen/bw/12103035/00033974_00000008_1.mcfilter.dst	/e/evh/bwdivision/12103035.c
12165106	Bplus2DK	/c/chaen/bw/12165106/00032641_00000006_1.mcfilter.dst	/e/evh/bwdivision/12165106.c
11874004	B2Dmuantinu	/c/chaen/bw/11874004/00033666_00000005_1.mcfilter.dst	/e/evh/bwdivision/11874004.c
21263002	D2KKpi	/c/chaen/bw/21263002/00033552_00000004_1.mcfilter.dst	/e/evh/bwdivision/21263002.c
30000000	min bias	/c/chaen/bw/xdigi/00034698_00000001_1.xdigi	N.A. file contains 5015 evts
30000000	min bias	/c/chaen/bw/xdigi/00034698_00000002_1.xdigi	N.A. file contains 11397 evts

Larger samples:

Event type	Channel	Nb of events	Path L0 applied
11102003	B02Kpi	28169	/e/evh/bwdivision/11102003/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
11114001	B2KstMuMu	24149	/e/evh/bwdivision/11114001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
11124001	Bd2Kstaree	25245	/e/evh/bwdivision/11124001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
13774002	Bs2Dsmuantinu	24723	/e/evh/bwdivision/13774002/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
12103035	Bplus2KKPi	24273	/e/evh/bwdivision/12103035/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
12165106	Bplus2DK	25101	/e/evh/bwdivision/12165106/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
13102201	Bsphigamma	24768	/e/evh/bwdivision/13102201/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
13112001	Bs2MuMu	31239	/e/evh/bwdivision/13112001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
13144001	Bs2JpsPhi	21207	/e/evh/bwdivision/13144001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
13264021	Bs2Dspi	21899	/e/evh/bwdivision/13264021/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
11874004	B2Dmuantinu	22471	/e/evh/bwdivision/11874004/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-1
21263002	D2KKpi	23568	/e/evh/bwdivision/21263002/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-2
30000000	min bias	11397	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	11497	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	12432	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	1247	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	17806	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	18822	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	19050	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	2019	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	20696	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	21257	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	22121	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	22308	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	23829	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	26197	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	26959	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27043	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27177	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27181	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27208	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27279	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27319	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27343	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27360	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3

30000000	min bias	27372	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27406	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27496	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27667	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27721	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27738	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	27868	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	28074	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	28324	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	28720	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	3924	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	5115	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	6735	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	8183	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	8577	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-3
30000000	min bias	17734	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27047	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27533	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27638	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27729	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27932	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	7174	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	11497	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	1247	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	17806	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	20696	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	27043	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	6735	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia
30000000	min bias	8577	/e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagDown-Nu1.5-25ns-Pythia

L0

```
* Minbias and rate:
  o In MC we only write xings with at least one pp-int to tape (checked with plotting Gen/Coll
     In real data NoBias events, we write also the empty xings.
  o MC: to convert number of events to rate:
     - (1.-exp(-nu))*nr-bunches * 11.245 kHz.
     - 25 ns: nr-bunches is 2330 (Massi thinks ~2400), hence nu=1.5 gives 20.355 MHz
     - 50 ns: nr-bunches is 1296, hence with nu=2.7 gives 13.594 MHz
  o In real data NoBias events: rate=nr-bunches*11.245 kHz
* Scripts
   1. The steering program.
   2. The Minuit FCN.
* Plots
 o for the channels in the above table, i.e. minbias nu=1.5,
    (Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8)
    made some plots as a function of the cut applied.
* Maximum Efficiencies obtained for the signal channels for L0 only:
```

Event type	Channel	Nb of events	Max Eff (L0 only)	Throttle
11102003	B02Kpi	28169	0.38	0.097
11114001	B2KstMuMu	24149	0.86	0.036
11124001	Bd2Kstaree	25245	0.56	0.092
13774002	Bs2Dsmuantinu	24723	0.62	0.072

12103035	Bplus2KKPi	24273	0.34	0.11
12165106	Bplus2DK	25101	0.29	0.11
13112001	Bs2MuMu	31239	0.92	0.03
13144001	Bs2JpsPhi	21207	0.87	0.035
13264021	Bs2Dspi	21899	0.32	0.11
11874004	B2Dmuantinu	22471	0.66	0.056
21263002	D2KKpi	23568	0.18	0.11

- see attached slides for the optimised efficiencies and rates

Efficiencies' table for various scenarios.

Event type	Channel	max eff	eff/(max eff)	eff/(max eff)	eff/(max eff)	eff/(max eff)	eff/(max eff)
			no CPU limit	CPU limit	4X w(Kstee)	4X w(charm)	adding phigamma
13112001	Bs2MuMu	0.92	0.86	0.87	0.79	0.80	0.90
13144001	Bs2JpsPhi	0.87	0.79	0.80	0.67	0.64	0.83
11114001	B2KstMuMu	0.86	0.78	0.79	0.66	0.64	0.83
11874004	B2Dmuantinu	0.66	0.66	0.67	0.44	0.42	0.67
13774002	Bs2Dsmuantinu	0.62	0.61	0.61	0.37	0.35	0.61
11124001	Bd2Kstaree	0.56	0.34	0.43	0.80	0.25	0.50
11102003	B02Kpi	0.38	0.76	0.71	0.53	0.84	0.61
12103035	Bplus2KKPi	0.34	0.73	0.71	0.50	0.82	0.59
13264021	Bs2Dspi	0.32	0.75	0.72	0.50	0.84	0.59
12165106	Bplus2DK	0.29	0.72	0.72	0.48	0.83	0.57
21263002	D2KKpi	0.18	0.67	0.61	0.33	0.72	0.44
13102201	Bsphigamma	0.60	-	-	-	-	0.48

- Thresholds for the above options:

	eff/(max eff)	eff/(max eff)	eff/(max eff)	eff/(max eff)	eff/(max eff)
L0Channel	no CPU limit	CPU limit	4X w(Kstee)	4X w(charm)	adding phigamma
L0Electron	200	170	117	240	170
L0Photon	200	170	117	240	190
L0Hadron	210	210	250	208	238
L0Muon	60	60	112	116	67
L0DiMuon	950	940	1440	1970	665
SPD mu,had,ele		480	530	570	540
SPD dimu (fixed)		900	900	900	900

http://dijkstra.web.cern.ch/dijkstra/BW-15/compare.ps

Hlt1

- Modifying the thresholds of Hlt1 lines at runtime

Vanya has modified the HltUnit code to allow for changable parameters (in the svn head of Phys/LoKiTrigger) via a dictionary Params. For instance introduce a parameter for PtMin in the Hlt1TrackAllL0 line as follows:

```
    def hlt1TrackNonMuon_Streamer( self, name, props ) :
        from Hlt1Lines.Hlt1GECs import Hlt1GECUnit
        from Configurables import LoKi__HltUnit as HltUnit
        props['name'] = name
        props['forward'] = 'LooseForward' if name.find('Photon') > -1 else 'TightForward'
        if props['ValidateTT'] :
            props['forward'] = "ValidateWithTT >>" + props['forward']
        Params=  {'PtMin': props.get('PT',0) }
        lineCode = """
        VeloCandidates
        >>  ( ( TrIDC('isVelo') > %(Velo_NHits)s ) &
        ( TrNVELOMISS < %(Velo_Qcut)s ) &
        ( Tr_HLTMIP ( 'PV3D' ) > %(IP)s * mm) )
        >>  tee  ( monitor( TC_SIZE > 0, '# pass VeloQ/IP', LoKi.Monitoring.ContextSvc ) )
        >>  tee  ( monitor( TC_SIZE    , 'nVeloIP' , LoKi.Monitoring.ContextSvc ) )
        >>  %(forward)s
        >>  tee  ( monitor( TC_SIZE > 0, '# pass Forward', LoKi.Monitoring.ContextSvc ) )
        >>  tee  ( monitor( TC_SIZE    , 'nForward' , LoKi.Monitoring.ContextSvc ) )
    >> ( (TrTNORMIDC > %(TrNTHits)s ) &
        ( TrPT > PARAM('Params[PtMin]')) &
        ( TrP  > %(P)s  * MeV ) )
        >>  tee  ( monitor( TC_SIZE > 0, '# pass P/PT', LoKi.Monitoring.ContextSvc ) )
        >>  tee  ( monitor( TC_SIZE    , 'nP' , LoKi.Monitoring.ContextSvc ) )
        >>  FitTrack
        >>  tee  ( monitor( TC_SIZE > 0, '# pass TrackFit', LoKi.Monitoring.ContextSvc ) )
        >>  tee  ( monitor( TC_SIZE    , 'nFit' , LoKi.Monitoring.ContextSvc ) )
        >>  ( ( TrCHI2PDOF < %(TrChi2)s ) & \
        ( Tr_HLTMIPCHI2 ( 'PV3D' ) > %(IPChi2)s ) )
        >>  tee  ( monitor( TC_SIZE > 0, '# pass TrackChi2/IPChi2', LoKi.Monitoring.ContextSvc )
        >>  tee  ( monitor( TC_SIZE    , 'nChi2' , LoKi.Monitoring.ContextSvc ) )
        >> SINK( 'Hlt1%(name)sDecision' )
        >> ~TC_EMPTY
        """ % props
        hlt1TrackNonMuon_Unit = HltUnit(
            'Hlt1'+name+'Unit',
            Preambulo = self.hlt1Track_Preambulo( name ),
            Params= Params,
            Code = lineCode
            )
        from HltTracking.HltPVs import PV3D
        return [ Hlt1GECUnit( 'Loose' ), PV3D(), hlt1TrackNonMuon_Unit ]
```

and then change it at runtime as follows:

```
#!/user/bin/env python
from Gaudi.Configuration import *
from Configurables import Moore
from LHCbKernel.Configuration import *
from Gaudi.Configuration import *
from Configurables import EventClockSvc, CondDB
EventClockSvc().EventTimeDecoder = "OdinTimeDecoder"
CondDB(IgnoreHeartBeat = True)

Moore().UseTCK = True
Moore().InitialTCK = '0x007b0044'
Moore().CheckOdin = False
Moore().EvtMax = 10
Moore().DDDBtag = "dddb-20120831"
Moore().CondDBtag = "cond-20120831"
Moore().Simulation = False
Moore().DataType = '2012'
Moore().Split = "Hlt1"
Moore().WriterRequires = ["Hlt1"]
Moore().inputFiles = [
'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/117770/117770_0x0044_NB_L0Phys_00.raw']
print Moore()
```

```
from Configurables import EventSelector
EventSelector().PrintFreq =1
import GaudiPython
gaudi = GaudiPython.AppMgr(outputlevel = 3)
gaudi.initialize()

def setupEvent(gaudi,Loop) :
  from GaudiKernel.SystemOfUnits import MeV
  alg_list = gaudi.algorithms()
  for a in alg_list:
    if a == "Hlt1TrackAllL0Unit" :
      line = gaudi.algorithm("Hlt1TrackAllL0Unit")
      props=line.properties()
      for p in props:
    value=props[p].value()
    print "Property  Name/Value:   '%s'/%s " % ( p, value )
      line.OutputLevel=2
      if Loop == 1 :
         line.Params= {'PtMin': 5000.0 * MeV}
    props=line.properties()
    for p in props:
      value=props[p].value()
      print "After changing. Property  Name/Value:   '%s'/%s " % ( p, value )
      if Loop == 2 :
         line.Params= {'PtMin':500.0 * MeV}
    props=line.properties()
    for p in props:
      value=props[p].value()
      print "After changing. Property  Name/Value:   '%s'/%s " % ( p, value )
      line.OutputLevel=3

import gaudigadgets
nEvents=10
Loop=1
while (Loop<3):
   if Loop==2: gaudigadgets.panorewind()
   setupEvent(gaudi,Loop)
   for i in range(0,nEvents):
     gaudi.run(1)
     print "event number ",i," Loop ", Loop
   Loop=Loop+1
```

These scripts plus instructions on how to get the right environment can be found in /afs/cern.ch/user/e/evh/w0/forAlvaro. #TMVA

TMVA

- Setting a discrete range of variables for the TMVA Genetic algorithm is done as follows:

```
std::vector<TMVA::Interval*> parameterRanges;
parameterRanges.push_back(new TMVA::Interval(.5,1.,6)); //means: Interval(.5,1.,6)= .5, .6., .
```

-- EricvanHerwijnen - 31 Mar 2014

This topic: LHCb > BWDivision
Topic revision: r28 - 2015-08-25 - ConorFitzpatrick