Table of Contents

Table of Contents

OldHLTMonitoringPage

Introduction: Tasks and Links

- **Online Monitoring**
  - Trigger Rate monitoring (Trigger Presenter)
  - Online histogram checks and comparison to references based on specific algorithms within the DataQualityMonitoringFramework DQMF to produce data quality flags
  - New DataQualityMonitoringConfigurator for DQMF/DQMD setups
  - Online histogram presenting for the shift crew based on the Online Histogram Presenter OHP
  - HLT specific online configuration files for DQMF at Point1 (for tdaq-02-00-02)
  - Description of OKS to configure the DQMF setups for Point1
  - Upgrade of DQMF and OHP for HLT online monitoring
- **Offline Monitoring**
  - standard checks during Tier0 standard reconstruction with the package TrigHLTMonitoring
  - dedicated, slice specific checks done by the experts using the CAF based on the package TrigHLTOfflineMon
  - the offline monitoring histograms from Tier0 running are available on the ATLAS DQ Monitoring page
  - HLT specific offline configuration files for offline DQMF
  - explanation how to make the han configuration files for the offline DQMF

Contact Persons and Documentation for Slices

- Coordination: **Martin zur Nedden** and **Mike Medinnis**
  - **b-Jet:** Andrea Coccaro, Fabrizio Parodi
    wiki Page for b-Jet Slice DQM
  - **B Physics:** Cristina Adorisio (online), Darren Price (offline), Julie Kirk, Pavel Reznicek
    wiki Page for BPhysics Slice DQM,
  - **Calo:** Denis Damazio, Ignacio Aracena
  - **Cosmics:** Alessandro Cerri, Maria Laura Gonzalez Silva
    wiki Page for Cosmics Slice DQM
  - **e/gamma:** Denis Damazio, Kamal Benslama, Ming Yao
    wiki Page for Egamma Slice DQM,
  - **Inner Detector:** Stefan Ask (EF), Erkcan Ozcan (L2)
    wiki Page for Tracking DQM
  - **Jet:** Ken Johns, Venkat Kaushik, Xiaowen Lei
    wiki Page for Jet Slice DQM and Jet Slice Histogram Description, References, OHP
  - **MET:** Ken Johns, Venkat Kaushik, Xiaowen Lei
    wiki Page for MET Slice DQM
  - **Minimum Bias:** Regina Kwee, Alexander Paramonov
    wiki Page for MinBias DQM
  - **Muon:** Margaritha Primavera, Akimasa Ishikawa, Yuji Yamazaki
    wiki Page for Muon Slice DQM
  - **Tau:** Sarah Demers (offline), Mansoora Shamim, Olga Igonkina, Stefania Xella
    wiki Page for Tau Slice DQM and Detailed Histogram Description for Tau

Status and Plans

DQMF online configuration

- overview of **online DQMF** histograms and checks for L2 and EF
- Full list **online DQMF** L2 and EF histograms
- Summary of **online DQMF** L2 and EF
- Table of DQMF Flag contence (with `histogram_tests` applied):

Names of Flags: *TRBCM*: beam monitor, *TRBJT*: b-jets, *TRBPH:* B-physics, *TRCAL*: calorimeter, *TRCOS*: cosmics, *TRDF*: data flow, *TRHLT*: HLT steering, *TRELE*: electrons, *TRGAM*: gammas, *TRIDT*: inner detector, *TRJET*: jets, *TRMET*: missing energy, *TRMBI*: minimum bias, *TRMUO*: muons, *TRTAU*: taus

FLAG	essential	physics	checks and algorithms
TRBCM			track and vertex info: for x/y/z-vertex `gauss_fit`; for track and vertex info `histo_not_empty`; for multiplicities `histo_mean`
TRBJT			*not yet available*
TRBPH			L2: for di-muons mass and multiplicity `histo_not_empty`; for muon eta, phi, pt, hits (barrel, endcap, RPC TGC) `histo_not_empty`
			EF: for di-muon pt, mass and mass cut `hiso_not_empty`
TRCAL	X	X	L1: for eta and phi of e/gamma, jet and tau `histo_not_empty`
			L2: for eta and phi of e/gamma, jet and tau `Bins_Diff_FromAvg`; for error histograms (eta versus conversion errors for e/gamma and jets) `Bins_GreaterThanEqual_Threshold`
			EF: for counter of cluster `histo_not_empty`; for eta/phi distributions `bins_diff_from_avarage`; for e/gamma and tau conversion errors `bin_greater_than_threshold`
TRCOS			*not yet available*
TRDF	X	X	*in preparation*
TRHLT	X	X	L2: from steering for chains, roi and active TE `histo_not_empty`; for errors `histo_effective_empy`
			EF: for process time, event size, rejection and timing `bin_filled_out_of_range`; for steering as for L2
TRELE		X	L2: for ET, phi and eta distributions `KolmogorovTest_MaxDist`; for cut counter `histogram_not_empty`
			EF: for tracks (pt, eta, phi), hits (pixel. STC, TRT mulipicities) and cluster energy `kolmogorov_MaxDist`; for cut counter `histo_not_empty`
TRGAM		X	L2: for cut counter `histo_not_empty`; for Et, eta, phi, Eratio, Rcore, dEta, dPhi and had Et `kolmogorov_MaxDist`
			EF: for isEMCluster and cut counter `hiso_not_empty`; for cluster Et, track pt, Et, eta, phi, dEta, dPhi, Eoverp, hits and outlayers (blayer, PIX, SCT and TRT) `kolmororov_MaxDist`
TRIDT	X	X	L2: for IDSCAN (pix and STC hits, number of tracks) `bins_greater_than_threshold`
			EF: for number of TRT, SCT and PIX hits `bins_greater_than_threshold`; for roi of tracks `histo_not_empty`
TRJET		X	L2: for E, Et, eta and phi `histo_not_empty` and `kolmogorov_shape_test`, additional `histo_mean` tests for eta and phi
			EF: for E, Et, eta and phi `histo_not_empty` and `kolmogorov_shape_test`; additional for eta and phi `histo_mean`
TRMET		X	L2: for Etmiss (linear scale), Summ Et and phi `histo_not_empty` and

			kolomogorov_shape_test; additional for phi histo_mean
			EF: for Etmiss (lin and log scale), Summ Et and phi histo_not_empty and kolomogorov_shape_test; additional for phi histo_mean
TRMBI			L2: for space points (TRT, PIX and SCT), MBTS (multiplicity, time diff A-D, ocupancy and charge) histo_not_empty
			EF: for tracks z0, multipliciy and pt histo_not_empty
TRMUO	X	X	L2: for **muIso** Esum (inner/outer EC and HC) check_histo_mean; for **muComb** (IDSCAN and SI tracks) dZeta, dPhi, dEta simple_gaus_fit and deltaR check_histo_mean; for **muFast** hits (inner, middle and outer) and ot check_hitsto_mean, for residuals check_histo_res; for **muTile** eta/phi bins_less_than_threshold, phi bins_diff_fromAvg, pt bin_outofRange, nTileRDO BinsFilledOutRange and eTileROD CheckHisto_Mean
			EF: for trackbuilder and extrapolator pt, eta, phi and track chi2 checkHisto_Mean, for chi2 also chekHisto_RMS, for track combiner pt, eta, phi, z0 checkHisto_Mean; for **muGirl** segments and hits (MDT, TGC, RPC) as well for pt, cotTheta, phi, beta checkHisto_Mean
TRTAU		X	L2: for **calo** emRaius, emFrac, eta, phi, roi, isolation fraction, and strip width KolmogorovTest_Prob; for cut counter CheckHisto_Mean; for combined Et and clusters (eta/phi) KolmogorovTest_Prob
			EF: for cut counter CheckHisto_Mean; for (eta/phi), em radius, em fraction, roi of cells, isolation fraction, nr of candidates and nr of errors KolmogorovTest_Prob

Next activities

- Review of the of the online DQMF configuration in January 2010:

Person	Signatures	Comments
Antonio Sidoti	TRMUO, TRBPH, TRBSP	
Martin zur Nedden	TRHLT, TRIDT, TRMBI	
Pierre-Simon Mangeard	TRGAM, TRELE, TRCAL	
Ulla Blumenschein	TRTAU, TRJET, TRMET	

Review of Trigger Slice Monitoring November / December 2008

Reviewers:

- Coordination/general overview: **Martin zur Nedden**
  - **B Physics:** Sinead Farrington
  - **e/gamma:** Stefania Xella
    Egamma Reviewing Page
  - **tau:** Usha Mallik
  - **muon Slice and software:** David Storm
  - **Calo software:** Sefan Ask
  - **Jets:** Margherita Primavera, Jet Slice Review Page
  - **b-Jet:** Denis Damazio
  - **HLT-MET:** Andrew Hamilton, MET Slice Review Page
  - **Cosmics:** Patricia Conde
  - **Minimum Bias:** Attila Krasznahorkay
  - **ID software:** Diego Casadei (see IdMonitoringReview)

Main goals of the reviewing

- Make the **online DQMF** and **offline Tier0** checks as **coherent** as possible
  - now we have two separate world with partially different responsibilities
  - every online check should also be made offline
  - offline checks can be more sophisticated and extended, since the whole event information is available
  - take information form other slices into account: which information from other slices can be used to check a certain slice?
  - summarize all common needed checks centrally to avoid to check several times the same
- Clarify the needs for all offline checks (on Tier0) for the individual slices
  - reprocessing is running over all data: ideal place for standard checks
  - what needs to be produced for the slices on Tier0: histogram-files, nTuples, ...
  - which functionalities are missing on Tier0 for this?
  - is the CAF setup still needed? can this be implemented into Tier0? Which role should the CAF monitoring play?
- **Main Goal:** simple DQ-flags based on the DQMF checks for data analysis for each slice
  - DQMF should produce a single flag for each slice at each trigger level (ok / doubtful / bad)
  - based on this: get a single flag for L2 and one for EF
- define a **simple** but powerful **OHP setup** for each slice and for each running type
- get run type depended setups for OHP (and if possible also for DQMF). Run types are:
  - Cosmics
  - single beam
  - early beams
  - colliding beams
  - tests / ....
- get clear shift instructions for the **DQ/Trigger online shifter** each slice
  - references for all online OPH displays
  - provide *what-to-do* instructions
- get a defined task and environment for the **offline Trigger DQ shifter**
  - what kind of jobs have to be running standardly
  - which files / jobs / histograms / web-pages / outputs have to be checked by the offline shifter
  - provide instructions ad *what-to-do* instructions

Work steps and tool

- we have a mailing list called atlas-trigger-dq-review@cernNOSPAMPLEASE.ch with archiving
- HLT DQ Review talk by Martin zur Nedden at the DQ Workshop in November 2008
- all informations will be documented here at the twiki page
- all mail exchange and documentations will be collected by Martin centrally
- all materials are collected on the Indico page of the Meeting in December 2008

Review Group Meetings (bi-weekly)

- First Meeting: 10. November 2008, 17.00, at the Phone (CERN telephone standard)
- Second Meeting: 20. November 2008, 17.30, at the Phone (CERN telephone standard)
- Third Meeting: 11. December 2008: First Reviews, 17.30 at the Phone (CERN telephone standard)
- Forth Meeting: 19. February 2009: Next Steps, Offline Monitoring, 18.30, via EVO (Phone Bridge ID 805801)

First Tests: Works for FDR2, M6/M7/M8 and 2008 Data Taking

Online Monitoring

- representatives for each slice: need >1 person, since we also need to cover DQ expert (on-call) shifts (24/7)
- **full set of DQ histograms** for each slice, which is neccessary to evaluate performance of this slice and consequences for DQ
- **detailed description of these histograms and instructions for shifters** : What should histogram look like? What do deviations from reference mean? Possible diagnosis? What to do in case it deviates?
- reference histogram for **all** histograms (not only those where reference histogram is needed for dqmf)
- **DQMF checks** for all histograms (where Histogram_Not_Empty is only used to check that histogram is not empty, not as dummy)
- **reduced small set of histograms** (1-2 !!!) per slice for by-eye-checks by shifters using **OHP**. Only possible if also decsription how histogram should look like **and** what to do if it doesnt is available in addition.

Offline Monitoring (Tier0)

In the reconstruction jobs running at Tier0 histograms for the HLT (general and slices) can be filled. The HLT is not being run at this stage, but information like HLT result or number of hits etc. as reconstructed on L2 or EF online can be extracted via StoreGate. The code is fully available for all slices.

- we have a good and extensive set of a histograms for all slices slice
- Histograms from the general Trigger Result: number of events for each L1-item, number of events for each L2/EF-chain (raw, after PT, after PS)
- **Status :** the Tier0 code (Trigger/TrigMonitoring/TrigHLTMonitoring) is running stable within the standard reconstruction

Offline Monitoring: Checks of Trigger performance and Data Quality (CAF)

Once we have taken some data constant checking of recent data will be neccessary and the online and offline histograms will not be sufficient to evaluate the quality of the data. Part of the taken data will be processed on the CAF, where the HLT can be re-runned and the full AtlasAnalysis functionalities are available

- What kind of offline checks do we need?
- What kind of trigger efficiencies should be calculated?
- What objects should be written out (histograms, branches, nTuples etc..)
- **Status :** the CAF code (Trigger/TrigMonitoring/TrigHLTOfflinMon) is available and running

M6 postmortem

Archived Histograms

Please take a look at the root files saved during M6 (e.g.look for files from the weekend - Mar 8/9) and check whether your histograms have been produced, look ok, are in the right place, etc.
Please note: There were some problems with MDA saving rootfiles during the weekend. For the following

runs the rootfiles seem to be corrupted (can not be opened in root, after revovery the Histogramming directory is empty):
43841, 43843, 43847, 43859, 43860, 43861, 43864, 43865, 43866, 43867,
43868, 43871, 43873, 43878, 43979, 44032, 44053, 44094, 44237, 44274

The histogram root-files are copied to castor
(http://pcatdwww.cern.ch/twiki/bin/view/Main/M6RunSchedule#Archived_Histograms),
some files from the weekend Mar08/09 you can also fine here/pcatr-srv1/home2/risler/m6_files/root.

Checks, Histogram analysis and open points for M6

- Unfortunately, quite some of the historgam paths given in our configuration were not given correctly and therefore the DQMF checks could not be performed. In particular in most cases the path should have contained some "CosmicAllTe" which we were not aware of beforehand. This applies e.g. for the Tau-, Jet- and Egamma-slice on L2. On the Event filter it seems that none of the histograms defined in the configuration files were produced.
- From the shift crew we got the feedback, that all histogram paths were configured incorrectly, but from comparing our configuration with the rootfiles I see, that for the MuIso histograms, everything should have been fine. not understood yet!
- Correcting for the paths for these the Tau-, Jet- and Egamma-slice I have plotted all historgams given in the dqmf configuration files available in the archived rootfiles for a few runs from the weekend Mar08/09. An example can be found here psfile with hitstograms for L2CaloJet, L2MuIso, L2Egamma and L2Tau.
- An overview of the menu (and also number of events being accepted by different chains) can be found here
- histogram review for each slice has been done
- We do need different configuration files for cosmics/techrun/data taking/...

Meetings and Minutes

- Phone Meeting 12. 04. 2007, Minutes2007April12
- Phone Meeting 17. 09. 2007, Transparencies from Christiane
- Phone Meeting 22. 10. 2007, Minutes2007October22
- Phone Meeting 12. 11. 2007, Minutes2007November12
    - Experiences form M5, **Preparation Technical Run:**
    - release ~ 13.0.3.30 + patches
    - Lets put all histos produced in slices at least in a OHP configuration (dedicated to HLT) and as much as possible into DQMF !
    - Please provide DQMF/OHP config files (or a list of histograms and publish path).
- Phone Meeting: 10. 12. 2007
- Phone Meeting: 20. 2. 2008
    - Anoncement2008February20
    - agenda for phone meeting,
    - Eduardos Slides
- Report in the Trigger Core Software Meeting, CERN 30. 6. 2008 by Martin zur Nedden
- Report in the Trigger Core Software Meeting, CERN 1. 9. 2008 by Martin zur Nedden
- Report in the Trigger Operations Meeting, CERN, 4. 9. 2008 by Martin zur Nedden
- Report in the Trigger Core Software Meeting, CERN, 20. 10. 2008, by Martin zur Nedden
- Report in the Trigger Trigger Operations Meeting, CERN, 29. 10. 2008, by Martin zur Nedden
- Report in the general Data Quality Workshop, CERN, 12. 11. 2008, by Martin zur Nedden
- Report in the ATLAS TDAQ-Week, CERN, 20. 11. 2008, by Martin zur Nedden

Workshops, Tutorials and Presentations

- DQMF tutorial
- TDAQ Data Quality Workshop at DESY-Zeuthen, 28. 2. - 1. 3. 2007
- DQ Data Quality Workshop at CERN, 15. - 17. 10. 2007
- Trigger Data Quality Assurance Workshop at CERN, 6. 5. 2008
- Trigge Efficiency Ad hoc Workshop at CERN, 1. 7. 2008

Developing Code for HLT Monitoring at Tier0

How to write code for Tier0 monitoring

- replace **`SliceName`** by your slice, i.e. **Muon**, **Tau**, **MET** etc.
- Code resides in the Offline SVN Repository at **`Trigger/TrigMonitoring/TrigSliceNameMonitoring`**
- Usual CMT package:
  - ♦ Header (.h) files in **`/SliceName`**
  - ♦ Implementation (.cxx) files in **`/src`**
- tool name **`HLTSliceNameMonTool`**
- for the standard running, the histograms should be the same as for the online monitoring.
- register your histogram path in `ManagedMonitorToolBase` in `HLTSliceNameMonTool::book()`

```
addMonGroup( new MonGroup(this,"HLT/SliceNameMon",shift,run) );
```

- register the histogram itself at the same place

```
addHistogram( new TH1F("Histo_Name", "Title", nBin,BinMin,BinMax) );
```

- in `HLTSliceNameMonTool::fill()` extract the StoreGate key (example for muon slice)

```
const DataHandle<MuonFeature> muonFeature, muonFeaturesEnd;
StatusCode sc_muFast=m_storeGate->retrieve(muonFeature, muonFeaturesEnd);
```

  loop over the objects and fill the histograms

```
hist("Histo_Name") ->Fill(var);
```

- update the **`cmt/requirements`** file in case of any dependencies

- Tool in initialized using a python snippet **`TrigSliceNameMonitoringConfig.py`** in **`/python`**, e.g. **`TrigBphysMonitoringConfig.py`**

```
from AthenaCommon.AppMgr import ToolSvc
def TrigBphysMonitoringTool():
        from TrigBphysMonitoring.TrigBphysMonitoringConf import HLTBphysMonTool
        HLTBphysMon = HLTBphysMonTool(name               = 'HLTBphysMon',
                                      histoPathBase      = "/Trigger/HLT")
        from AthenaCommon.AppMgr import ToolSvc
        ToolSvc += HLTBphysMon;
        list = [ "HLTBphysMonTool/HLTBphysMon" ];
        return list
```

- Tools are managed by a central Tool at **`Trigger/TrigMonitoring/TrigHLTMonitoring`**:
  - ♦ in **`/python/HLTMonFlags.py`** flags are defined to switch the Tools on and off
  - ♦ The flags are set in **`/share/HLTMonitoring_topOptions.py`**:
    - ◊ to run a tool during RAW->ESD step switch if off (**`HLTMonFlags.doSliceName = False`**) under **`DQMonFlags.monManEnvironment == 'tier0ESD`**
    - ◊ to run a tool during ESD->AOD step switch if off (**`HLTMonFlags.doSliceName = False`**) under **`DQMonFlags.monManEnvironment == 'tier0Raw`**
  - ♦ **`addMonTools.py`** includes the **`*Config.py`** to add the tools to the algorithm

How to write code for offline extended checks

- use the package **`Trigger/TrigMonitoring/TrigHLTOfflineMon`**, where all tools are defined centrally

- to run in CAF
- toolname **`HLTSliceNameOfflineTool`**
- Here, all more sophisticated checks and analysis as trigger efficiencies, trigger simulations and rerunning the trigger should be made.
- Check whether the corresponding HLTOfflineMonFlags exists in **`/python`**
- add your tool initialization in **`/share/addMonTools.py`**
- Tools can be switched on/off in **`/share/HLTOfflineMon_topOptions.py`**

Testing Code for Offline Monitoring at Tier0

Current Release:

- Currently testing should be done using release 15.6.X

Workspace setup:

- Setup your environment using (replace X in **`rel_X`** with 0-6 for the nightlies Sunday through Saturday)

```
source /afs/cern.ch/sw/contrib/CMT/v1r20p20090520/mgr/setup.sh
cmt config
source setup.sh -tag=15.6.X,rel_X,AtlasOffline
```

Tier0 packages

- replace `USERNAME` with your username to save the ESD and AOD files in yout tmp directory

```
Reco_trf.py inputBSFile=/afs/cern.ch/user/g/gencomm/w0/RTT_INPUT_DATA/CosmicATN/daq.ATLAS
```

- For debugging you can switch tracing and VERBOSE output on using **`--athenaopts='-s -lVERBOSE'`**
- If you wish to test against another release try one of the older test jobs below

Extended Packages (CAF)

- replace `USERNAME` with your username

```
export STAGE_SVCCLASS=atlcal
Reco_trf.py inputBSFile=/castor/cern.ch/grid/atlas/DAQ/2009/00142402/express_express/data
```

Workspace setups for older releases

Setup for release 14.1

* code repository for the whole package * the final tag for **14.1** is **`TirgHLTMonitoring-00-01-09`** * make your requirements file (example at **`~nedden/public/FDR2/cmthome`**) * set up your enviroment for release 14.1.0.14 with:

```
source /afs/cern.ch/sw/contrib/CMT/v1r20p20080222/mgr/setup.sh
cmt config
source setup.sh -tag=setup,AtlasPoint1,14.1.0.14,releases,32,opt
```

* make your working directory in the testarea, go there and check out the code

```
mkdir testarea/AtlasPoint1-14.1.0.14
cd testarea/AtlasPoint1-14.1.0.14
cmt co  -r TrigHLTMonitoring-00-01-09 Trigger/TrigMonitoring/TrigHLTMonitoring
```

* for release 14.1 a special tag is needed for TrigCaloEvent (used by Missing ET slice):

```
cmt co -r TrigCaloEvent-00-02-12-01 Trigger/TrigEvent/TrigCaloEvent
```

* compile the code

```
cd Trigger/TrigMonitoring/TrigHLTMonitoring/cmt
cmt config
source setup.sh
cmt bro gmake
```

* **General tools**: all common tool needed for the coding are contained in **`IHLTMonTool.cxx`** from which all tools inherits. Take an existing tool (for example from the muon slice) as an example.

Setup for release 14.2

* code repository for the whole package * set up your enviroment for release 14.2.X.Y (for nightly release X=0,1,..,6) with:

```
source /afs/cern.ch/sw/contrib/CMT/v1r20p20080222/mgr/setup.sh
cmt config
source setup.sh -tag=setup,14.2.2X.Y-VAL,AtlasTier0,rel_X,32
source $AtlasArea/AtlasTier0RunTime/cmt/setup.sh
```

* The X.Y release is **in the test phase using nightlies**, change accordingly for other nightlies (**`rel_X`**). * make your working directory in the testarea, go there and check out the code

```
mkdir testarea/AtlasTier0-rel_X
cd testarea/AtlasTier0-rel_X
cmt co  Trigger/TrigMonitoring/TrigHLTMonitoring
```

* recent tag is **TrigHLTMonitoring-00-02-06** * compile the code

```
cd Trigger/TrigMonitoring/TrigHLTMonitoring/cmt
cmt config
source setup.sh
cmt bro gmake
```

* **General tools**: all common tool needed for the coding are contained in **`IHLTMonTool.cxx`** from which all tools inherits. Take an existing tool (for example from the muon slice) as an example.

Test Jobs for Tier0 Tools against older releases

Test job over FDR2 data for rel 14.1

* An example job is given at **`~nedden/public/FDR2/run`**. Go to the `run` directory and modify the shell scripts for your environment. Make also the directory **`/tmp/$USER/run`** * To get the data files do

```
data2tmp.sh
```

* The python script to configure the monitoring job is in **`~nedden/public/FDR2/run/DataQualityTools/`** * To let the job run do (change the script beforehand accordingly)

```
run-hlt-mon.sh
```

This will produce for you a root file with the name **monitoring.root** containing the HLT monitoring histograms.

Test Job over real data (Cosmics) for rel 14.2

* An example job is given at **~nedden/public/HLTMon/run_14.2_cosmics**. You do not have to change anything * copy the directory to you srcatch0 directory at lxplus * To let the job run do

```
run_job
```

This will produce for you many files file, but also a root file with the name **HIST.root** containing the HLT monitoring histograms. * An alternative way how to run the job over Cosmics 08 data is descibed on the Trigger at Tier0

page

Test Job over FDR data for rel 14.2

* An example job is given at **~nedden/public/HLTMon/run_14.2_fdr**. Go to the `run_14.2_fdr` directory and modify the shell scripts for your environment. Make also the directory **/tmp/$USER/run** * To get the data files do

```
data2tmp.sh_14.2.20
```

* The python scripts to configure the monitoring job is in **~nedden/public/HLTMon/run_14.2.20_aod/DataQualityTools/** * To let the job run do (change the script beforhand acordingly)

```
run-hlt-mon_14.2.20.sh
```

This will produce for you a root file with the name **Monitor.root** containing the HLT monitoring histograms.

Test Job for 14.4.0.1

* This follow **this information**: RecoRealData, Using_the_transform * use the tag TrigHLTMonitoring-00-02-06 * source setup.sh -tag=14.4.0.1,AtlasTier0,32 * source $AtlasArea/AtlasTier0RunTime/cmt/setup.sh * check out and compile `cmt co -r RecExCommission-00-03-82 Reconstruction/RecExample/RecExCommission` *

```
CmdToPickledDic.py Reco_trf.py inputBSFile=/castor/cern.ch/grid/atlas/DAQ/2008/87863/physics_BPTX
```

* To change the configuration (input data file, number of events, etc), you have to do it within the CmdToPickledDic.py command above. You can also drop the AOD and DPD parts. * `Reco_trf.py --argdict=input.pickle 1>&2 | tee Log.txt`

Test Job for 14.5

* This follow **this information**: RecoRealData, Using_the_transform * check out the tag TrigHLTMonitoring-00-02-10-01 (own branche): **cmt co -r TrigHLTMonitoring-00-02-10-01 Trigger/TrigMonitoring/TrigHLTMonitoring** to get the head version of this brance * source setup.sh -tag=14.X.0-VAL,rel_5,32 * source $AtlasArea/AtlasOfflineRunTime/cmt/setup.sh *

```
Reco_trf.py inputBSFile=/castor/cern.ch/grid/atlas/DAQ/2008/90275/physics_IDCosmic/daq.ATLAS.0090
```

Test Job for Cosmics with 15.0

* check out the HEAD version of TrigHLTMonitoring **>cmt co Trigger/TrigMonitoring/TrigHLTMonitoring** (or a tag larger than **-00-03-00**) * **>source setup.sh -tag=15.X.0-VAL,rel_1** (take the most recent nightly release) * **>source $AtlasArea/AtlasOfflineRunTime/cmt/setup.sh** *

```
 Reco_trf.py inputBSFile=/afs/cern.ch/user/g/gencomm/w0/RTT_INPUT_DATA/CosmicATN/daq.ATLAS.009190
```

Test Job for Cosmics with 15.2

* check out the tag 00-03-19 of TrigHLTMonitoring **>cmt co Trigger/TrigMonitoring/TrigHLTMonitoring** * **>source setup.sh -tag=15.2.0,AtlasOffline,32 * ==>source $AtlasArea/AtlasOfflineRunTime/cmt/setup.sh** *

```
 Reco_trf.py inputBSFile=/afs/cern.ch/user/g/gencomm/w0/RTT_INPUT_DATA/CosmicATN/daq.ATLAS.009190
```

Test Job for Cosmics with 15.4

* check out the tag 00-03-30 of TrigHLTMonitoring (or the HEAD version) **>cmt co Trigger/TrigMonitoring/TrigHLTMonitoring** * **>source setup.sh -tag=15.X.0-VAL,rel_x (where x=0,1,..,6 stands for the most recent nightly * compile ... * ==>source $AtlasArea/AtlasOfflineRunTime/cmt/setup.sh** *

```
 Reco_trf.py inputBSFile=/afs/cern.ch/user/g/gencomm/w0/RTT_INPUT_DATA/CosmicATN/daq.ATLAS.009190
2>&1 | tee Monitoring.log
```

* eventually you have also to check out: **>cmt co -r InDetRecExample-01-17-59 InnerDetector/InDetExample/InDetRecExample**

Testing online DQMF setup

The procedure to test the DQMF online on the preseires is described here. For info on the preseries refer here (the updated part).

How to log in

- ssh -Y preseriesgw
- enter a name of a preserie machine at prompt (e.g. pc-preseries-xpu-001)

Instruction for first time preseries users

Since you have on the preserie a separate home than the one at point1 you may want to require to have also your point1 home on the preserie. To do that you need to do (only once) on one point1 machine (NOT the preserie):

$> sudo -u atdadmin /daq_area/tools/sync/remote_sync.sh -x -t p1_home the point1 home will appear on /atlas-home/<0.OR.1>//P1_home

YourPoint1 will be resynced every ~30 min to the one in the preserie.

How to run a test partition

1. log on preserie machine
2. source the offline **`$> source /sw/atlas/cmtsite/setup.sh -tag=AtlasP1HLT,15.5.6.1,opt,32,setup`**
3. source the tdaq: **`$> source /sw/tdaq/setup/setup_tdaq-02-00-03.sh`**
4. check that the $TDAQ_DB_PATH variable is /preseries/oks/tdaq-02-00-03:/atlas/oks/tdaq-02-00-03 by **`$> echo $TDAQ_DB_PATH`**
5. you will find on the preseries a directory with the correct files: /atlas-home/0/sidoti/hlt_dqm
6. set the python path in order to use the pythons scripts you have in the directory **`$> export PYTHONPATH=/det/tdaq/hlt/pm:/atlas-home/0/sidoti/hlt_dqm:$PYTHONPATH`**
7. the following two points might be skipped if you use an already generated partition file: e.g. dqmhlt_test.data.xml
8. re-generate your data file (need to erase TDAQ produced data): **`$> rosconf-from-data.py --py --ignore '^0x007[35bc]' | egrep -v 0x00760001\|0x00770001 > robhit.py`**
9. generate your xml partition file with: **`$> hltpm_part_l2ef.py -F l2efopt.py -Z useCoralProxy -Z addDQM`**. Pay attention in "l2efopt.py" that the release 15.5.6.1 is used (release used here must be

the same as the one you setup at step 2.).

1. Run your partition with **`$> setup_daq -p dqmhlt_test -d dqmhlt_test.data.xml`**

Known problems and things to know

- to copy from outside to preseries do: scp my_file.txt @preseriesgw:
- OKS works differently wrt point1: * TDAQ_DB_REPOSITORY variable must be empty. * Configuration files are taken from /preseries/oks/tdaq-02-00-0x (if they don't exixt they are looked for in /atlas/oks/tdaq-02-00-0x) * Files have to be manually copied from /atlas/oks/tdaq-02-00-0x to /preseries/oks/tdaq-02-00-0x (no oks-checkout) * You can also use OKS files stored in your working directory (to check)
- DQM_HLT segment takes ages to start

- To add your own libs and bins add the following line to the l2efopt.py file option['repository-root'] = '/atlas-home/0/...'
- Note that in the l2efopt.py you might have some offline versions "carved" in stone (in l2efopt.py) that might be incompatible with your setup. This is fixed now but better to remind for the next future

Implementation of Online DQM Histograms into HLT-Code (for DQMF)

- Gudielines from Trigger Validation Please follow this Guidelines, this are the main reference for using trigger generic monitoring tools for FEX and HYPO algorithms.
- ***Code example:*** The code was done for the implementation of the monitoring of the Muon-Slice based on rel. 13.0.X of *TrigMooreFEX* algorithm iside Trigger/TriggerAlgorithms/TrigMoore/
  - **Implementation for the header (.h) file:** If the variable to me monitored is already member of hte algorithm class, you do not have to change the header file any more, other wihise define the variable. As example, the implementation in *Trigger/TriggerAlgorithms/TrigMoore/TrigMoore/MooHLTAlgo.h* is shown:

```
// e. g. std::vector<float> tgc_phi_res;
float pt_moore;
float phi_moore;
float eta_moore;
```

  - **Implementation for the source code (.cxx) file:** In the constructor or in the initialization the variables to be monitored have to be declared in the source code file. The following example is taken from *Trigger/TriggerAlgorithms/TrigMoore/src/MooHLTAlgo.cxx* :

```
declareProperty("histoPathBase",m_histo_path_base="/EXPERT/");
declareMonitoredStdContainer("tgc_phi_res", tgc_phi_res);
declareMonitoredVariable("pt_moore", pt_moore);
declareMonitoredVariable("phi_moore", phi_moore);
declareMonitoredVariable("eta_moore", eta_moore);
```

    For safty reasons, you should set all variables to *unphysical* values at the beginning of the execution to avoid remainig values from previous calls. This is the only part to be done, the rest is done automatically. Example:

```
tgc_phi_res.clear();
```

    Afterwards, just do your routine calculations as before. Example:

```
pt_moore = fabs(1./perigee.inverse_pt())/1000.;
phi_moore = perigee.phi();
eta_moore = -log(tan(atan(1./fabs(perigee.cot_theta()))/2.));
if (perigee.cot_theta()<0.) eta_moore = -eta_moore;
```

  - **Implementation for configurarion python file:** The followig example is from *Trigger/TrigAlgorithms/TrigMoore/python/TrigMooreConfig.py* :

```
 from TrigMoore.TrigMooreMonitoring import *
 e.g. class TrigMooreConfig_MS (MooHLTAlgo):
 __slots__ = []
 ........
self.histoPathBase = ""
validation = TrigMooreValidationMonitoring()
online = TrigMooreOnlineMonitoring()
self.AthenaMonTools = [ validation, online ]
```

  - **Implementation for phython (.py) file** Finally, in the python file for the job, you can define histogram bins, limits etc., enable or disable histograms at run time. Again, the example is taken from *Trigger/TrigAlgorithms/TrigMoore/python/TrigMooreMonitoring.py* :

```
from TrigMonitorBase.TrigGenericMonitoringToolConfig
import defineHistogram, TrigGenericMonitoringToolConfig

class TrigMooreValidationMonitoring(TrigGenericMonitoringToolConfig):
def __init__ (self, name="TrigMooreValidationMonitoring"):
super(TrigMooreValidationMonitoring, self).__init__(name)
self.defineTarget("Validation")

self.Histograms += [ defineHistogram('tgc_phi_res', type='TH1F',
title="Hit phi residual TGC; Moore",
xbins=100, xmin=-5., xmax=5.) ]
self.Histograms += [ defineHistogram('pt_moore', type='TH1F',
title="Muon pt; Moore",
xbins=150, xmin=0., xmax=150.) ]
self.Histograms += [ defineHistogram('phi_moore', type='TH1F',
title="Muon phi; Moore",
xbins=100, xmin=-5., xmax=5.) ]
self.Histograms += [ defineHistogram('eta_moore', type='TH1F',
title="Muon eta; Moore",
xbins=100, xmin=-5.5, xmax=5.5) ]

class TrigMooreOnlineMonitoring(TrigGenericMonitoringToolConfig):
def __init__ (self, name="TrigMooreOnlineMonitoring"):
super(TrigMooreOnlineMonitoring, self).__init__(name)
self.defineTarget("Online")

self.Histograms += [ defineHistogram('tgc_phi_res', type='TH1F',
title="Hit phi residual TGC; Moore",
xbins=100, xmin=-5., xmax=5.) ]
self.Histograms += [ defineHistogram('pt_moore', type='TH1F',
title="Muon pt; Moore",
xbins=150, xmin=0., xmax=150.) ]
self.Histograms += [ defineHistogram('phi_moore', type='TH1F',
title="Muon phi; Moore",
xbins=100, xmin=-5., xmax=5.) ]
self.Histograms += [ defineHistogram('eta_moore', type='TH1F',
title="Muon eta; Moore",
xbins=100, xmin=-5.5, xmax=5.5) ]
```

DQMF: Instructions and Helps

DQMF twiki pages

DataQualityMonitoringFramework Twiki
DataQualityWorkbenchTutorial

DQMF offline Configuration

- This section describes how to setup and test the offline DQMF. For beginners, the tutorial OfflineDQMFTutorial is a good starting point. The example given below was used to test the HLTjet slice configuration. Other trigger signature groups could

use this as a template to test their configuration.

- Setup
- Packages required
- Preparing a configuration file
- Adding checks, algorithms and defining output
- Run/Testing configuration
- Output

DQMF Configuration files examples

The HLT DQ DQMF configuration files at point 1 are in /db/tdaq-01-08-03/daq/segments/DQM/ . Small example files can be found at /afs/cern.ch/user/r/risler/public/DQMF_db_examples.

```
 <include>
 <file path="dqm_config/schema/DQM.schema.xml"/>
 <file path="dqm_config/data/DQM_algorithms.data.xml"/>
 </include>

 <obj class="DQParameter" id="name_of_DQParameter">
<attr name="InputDataSource" type="string">"Histogramming-EBEF-Segment-iss.EF-EBEF-Segment-
<attr name="Weight" type="float">1.0</attr>
<attr name="Action" type="string">""</attr>
<rel name="Algorithm">"DQAlgorithm" "Histogram_Not_Empty"</rel>
<rel name="AlgorithmParameters" num="0"></rel>
<rel name="References" num="0"></rel>
<rel name="GreenThresholds" num="0"></rel>
<rel name="RedThresholds" num="0"></rel>
</obj>
<obj class="DQParameter" id="another_DQParameter">
<attr name="InputDataSource" type="string">"Histogramming-EBEF-Segment-iss.EF-EBEF-Segment-
<attr name="Weight" type="float">1.0</attr>
<attr name="Action" type="string">""</attr>
<rel name="Algorithm">"DQAlgorithm" "Histogram_Not_Empty"</rel>
<rel name="AlgorithmParameters" num="0"></rel>
<rel name="References" num="0"></rel>
<rel name="GreenThresholds" num="0"></rel>
 <rel name="RedThresholds" num="0"></rel>
</obj>

<obj class="DQRegion" id="name_of_your_DQRegion">
<attr name="InputDataSource" type="string">""</attr>
<attr name="Weight" type="float">1.0</attr>
<attr name="Action" type="string">""</attr>
<rel name="Algorithm">"DQAlgorithm" "SimpleSummary"</rel>
```

```
<rel name="AlgorithmParameters" num="0"></rel>
 <rel name="References" num="0"></rel>
<rel name="GreenThresholds" num="0"></rel>
<rel name="RedThresholds" num="0"></rel>
<rel name="DQRegions" num="0"></rel>
<rel name="DQParameters" num="2">
 "DQParameter"  "name_of_DQParameter"
 "DQParameter"  "another_DQParameter"
 </rel>
</obj>
```

Example OHP configuration files

OHP configuration examples can be found in /afs/cern.ch/user/r/risler/public/OHP_conf_example.

Major updates:
-- MartinZurNedden - 26 Sep 2007

RuthHerrberg
Never reviewed

This topic: Sandbox > OldHLTMonitoringPage
Topic revision: r1 - 2010-04-09 - RuthHerrberg