

# Table of Contents

<b>Offline DQ Monitoring Integration with ATLAS Central System.....</b>	<b>1</b>
<b>Introduction.....</b>	<b>2</b>
<b>Manpower (people to contact).....</b>	<b>3</b>
<b>Goals.....</b>	<b>4</b>
<b>Deadlines.....</b>	<b>5</b>
<b>Tasks.....</b>	<b>6</b>
Prepare list of control distributions.....	6
1. List of parameters to be monitored.....	6
2. List of distributions to be monitored and for experts.....	6
3. List of considered but not recommended distributions.....	6
3.1. 1D distributions of occupancy per layer using hits utilised in track reconstruction.....	6
Purpose.....	6
Event selection.....	6
Description.....	6
Evaluation rules.....	6
Comments.....	6
3.2. Distribution of fraction of triggered events with no reconstructed track.....	7
Purpose.....	7
Event selection.....	7
Description.....	7
Evaluation rules.....	7
Comments.....	7
3.3. Distribution of fraction of triggered events with exactly one reconstructed track.....	7
Purpose.....	7
Event selection.....	7
Description.....	7
Evaluation rules.....	8
Comments.....	8
3.4. 1D distributions of noisy layers.....	8
Purpose.....	8
Event selection.....	8
Description.....	8
Possible problems.....	8
Evaluation rules.....	8
Comments.....	8
Prepare software environment for the new package development.....	8
1. Set up environment for monitoring package development.....	9
1.1. Testing the set-up.....	9
2. Set up and han.exe.....	10
2.1. Testing the set-up with example.....	10
2.2. Testing display of histograms with web display.....	11
2.3. SVN repository for package.....	11
Implement and test the package.....	11
1. Create directory in SVN repository.....	11
2. Create a new athena package.....	11
3. Create a new tool in the package.....	11
4. Install ALFA xAOD.....	12
5. Add call to the new class in an existing program.....	12
6. Implement distributions.....	12

# Table of Contents

## Tasks

Prepare reference distributions.....	12
Prepare defects.....	12
Prepare documentation of the package and instructions for the person doing monitoring.....	12
Prepare configuration that can be submitted to the central system.....	12
Submit the new package to the central system.....	12

## Notes.....13

Links.....	13
Presentations connected with this project.....	13
Trigger aware monitoring.....	13
Meetings.....	13
1. Diffractive meeting 15.07.2015.....	13
Properties.....	13

# Offline DQ Monitoring Integration with ATLAS Central System

# Introduction

This page is used for organisation and as a logbook for adding ALFA offline data quality monitoring to the central ATLAS offline data quality monitoring.

## **Manpower (people to contact)**

- Grzegorz Gach
- Krzysztof Janas
- Leszek Adamczyk

# Goals

1. Prepare a list of distributions to be used in the offline monitoring
2. Add relevant ALFA distributions to offline data quality web display [↗](#)
3. Add automatic distribution checks were possible
4. Define defects
5. Prepare documentation and instructions for a person doing data quality monitoring

# Deadlines

- offline monitoring shall be ready for October ALFA runs
  - ◆ development should be finished latest 15 September in order to have enough time to migrate new code to Tier0 etc.
- It is desired to finish the project as soon as possible in order to benefit from it during reprocessing of run 267358

# Tasks

## Prepare list of control distributions

### 1. List of parameters to be monitored

1. Hot fibres (covered by 2D map of hits, but better by 2D map of noise)
2. Dead fibres (covered by 2D map of hits)
3. Layer efficiencies (covered by layer occupancy)

### 2. List of distributions to be monitored and for experts

This list was moved to the official ALFA data quality site [AlfaOfflineDQHistograms](#).

### 3. List of considered but not recommended distributions

#### 3.1. 1D distributions of occupancy per layer using hits utilised in track reconstruction

##### Purpose

Identify fibres layers with hot/dead fibres. Identify potential problems with PMT settings. This distributions provide overview of the detector efficiency.

##### Event selection

All triggered events used.

##### Description

This is a set of 16 histograms (8 detectors x 2 layer types i.e. U and V). Each histogram is one dimensional and contains 10 bins, a bin per fibres layer. Histogram presents the efficiency of each layer which is calculated as a number of events with signal in a fibre used to reconstruct track in the layer divided by the total number of triggered events. Acceptable efficiency values are from 0 to 1.

Second version: The set of 8 one dimensional histograms (8 detectors, layers U and V on one histogram), with 20 bins, a bin per layer.

##### Evaluation rules

The distribution of occupancy for **layers used in track reconstruction** is classified according to the following rules:

- good - efficiency of each layer is greater than 0.8
- warning - efficiency of each layer is greater than 0.7
- error - efficiency of any of the layers is smaller than 0.7

The values were selected by looking at distributions for run 267358.

##### Comments

This distribution may be biased by the track reconstruction algorithm that is why it is better to use in the data quality monitoring a distribution that is independent of the reconstruction algorithm.

### 3.2. Distribution of fraction of triggered events with no reconstructed track

#### Purpose

Monitor trigger rates (noise?) and reconstruction efficiency.

#### Event selection

Events with ALFA trigger signal.

#### Description

This is a 1D distribution with 8 bins. Each bin corresponds to one ALFA station. The histogram is filled with events with ALFA trigger signal and no reconstructed track normalised to the total number of triggered events.

It is expected that if there is a signal in ALFA trigger a particle should also be recorded by the detector thus at least one track shall be present. The fraction of events with no reconstructed track shall be small, although a significant fraction is observed in run 267358.

#### Evaluation rules

The distribution is classified according to the following rules:

- good - fraction for each station is smaller than 0.6
- warning - fraction for each station is smaller than 0.8
- error - fraction of any of the stations is greater than 0.8

The values were selected by looking at distributions for run 267358.

#### Comments

This distribution is useful only for experts, because it strongly depends on the conditions. In the set of distributions for experts there is a distribution of track multiplicity which contains more information than this one.

### 3.3. Distribution of fraction of triggered events with exactly one reconstructed track

#### Purpose

Monitor reconstruction efficiency and background.

#### Event selection

Events with ALFA trigger signal.

#### Description

This is a 1D distribution with 8 bins. Each bin corresponds to one ALFA station. The histogram is filled with events with ALFA trigger signal and exactly one reconstructed track normalised to the total number of triggered events.

Events with one track are the best events to be used in analysis, so it is desired to register a significant fraction of such events.

**Evaluation rules**

The distribution is classified according to the following rules:

- good - fraction for each station is greater than 0.4
- warning - fraction for each station is greater than 0.2
- error - fraction of any of the stations is smaller than 0.2

The values were selected by looking at distributions for run 267358.

**Comments**

This distribution is useful only for experts, because it strongly depends on the conditions. In the set of distributions for experts there is a distribution of track multiplicity which contains more information than this one.

**3.4. 1D distributions of noisy layers****Purpose**

Identify fibres layers which are noisy.

**Event selection**

All triggered events used.

**Description**

This is a set of 16 histograms (8 detectors x 2 layer types i.e. U and V). Each histogram is one dimensional and contains 10 bins, a bin per fibres layer. Histogram presents the fraction of events with more than 3 hits in a layer.

**Possible problems**

Krzysztof Janas pointed out that this distribution may be dominated by cascades and will not provide the information that is needed. This needs to be verified.	GrzegorzPawelGach	2015-07-09 - 12:57
---	-------------------	--------------------

**Evaluation rules**

The distribution is classified according to the following rules:

- good - each layer occupancy is smaller than 0.3
- warning - each layer occupancy is smaller than 0.4
- error - occupancy of at least layer is greater than 0.4

**Comments**

This distribution does not give any new information with respect to the 2D map of noisy fibres.

**Prepare software environment for the new package development**

## 1. Set up environment for monitoring package development

The monitoring package is responsible for filling the histograms. Only Athena is needed. No special other environment is needed. It is best to set up the same Athena release as is currently used at Tier0. The monitoring package is supposed to be a standard Athena package with classes deriving from `ManagedMonitorToolBase`. However it is useful to checkout the base package

```
Control/AthenaMonitoring
```

, because there is an example called

```
ManagedMonitorToolTest
```

.

It can be useful to have the code inspired by the classes based on the `IHLTMonTool`[?](#), that is why it is also useful to checkout `TrigHLTMonitoring` package. In order to get an example of the use of

```
TrigHLTMonitoring
```

a package

```
Trigger/TrigMonitoring/TrigMinBiasMonitoring
```

. It can be dangerous to develop classes based on the `IHLTMonTool`, because this is trigger class and we do not have any control other it, nor trigger should consider us when developing this class.

The above can be set using the commands

```
setupATLAS
asetup 20.1.5.11,here # or other currently used at Tier0 release
cmt co Control/AthenaMonitoring
cd Control/AthenaMonitoring/cmt/
cmt make
cd ../../../../
cmt co Trigger/TrigMonitoring/TrigHLTMonitoring
cd Trigger/TrigMonitoring/TrigHLTMonitoring/cmt/
cmt make
cd ../../../../
cmt co Trigger/TrigMonitoring/TrigMinBiasMonitoring
```

More information can be found in the page `AthenaMonFramework`. An example of the muon code can be found in the muon tutorial.

The following instructions are based on the twiki pages:

- `DQOperationalRecipes#Modify_your_configuration_locall`
- `AthenaMonFramework`

#testSetup

### 1.1. Testing the set-up

1. Create a directory for the test input data and download it

```
mkdir TestData
cd TestData
localSetupRucioClients
```

1. Set up environment for monitoring package development

```
voms-proxy-init -voms atlas
rucio download data15_13TeV:data15_13TeV.00267358.express_express.recon.AOD.r6832:data15_13
cd ..
```

## 2. Preferably create a

run

directory in

Control/AthenaMonitoring

```
mkdir Control/AthenaMonitoring/run
```

## 3. Set proper input file by setting correct path in the variable:

af.FilesInput

in the file

Control/AthenaMonitoring/share/ManagedMonitorToolTest.py

.

## 4. Set proper input file and run test algorithm in the run directory example to test if everything works

```
cd Control/AthenaMonitoring/run
athena ../share/ManagedMonitorToolTest.py
cd ../../../../
```

## 2. Set up and han.exe

It is enough to set up Athena environment to have access to

DQWebDisplay.py

and

han.exe

. It may be very useful to checkout

DataQuality/DataQualityInterfaces

, because there is an example of configuration file and data file.

```
cmt co DataQuality/DataQualityInterfaces
```

### 2.1. Testing the set-up with example

0. Enter the directory with example files

```
cd DataQuality/DataQualityInterfaces/share
```

#### 1.1. Testing the set-up

1. Generate binary configuration file based on text configuration file

```
han-config-gen.exe histogramAnalysis.config
```

2. Process the example data file using the generated configuration. This will produce result file Data\_han.root. The output file can be partially browsed with root.

```
han.exe histogramAnalysis.hcfg Data.root
```

3. Print summary of the results.

```
han-results-print.exe Data_han.root
```

Instructions based on: [HLTTrigMonHanConfigRecipesRun2](#)

## 2.2. Testing display of histograms with web display

Web display is using the same configuration file as han.exe. In order to test the display an SVN access writes are needed. Detailed instructions are available in the paragraph "How to update DQ configurations or reference histograms" in page [DQOperationalRecipes#How\\_to\\_update\\_DQ\\_configurations](#)

## 2.3. SVN repository for package

The new package shall be placed in the central SVN repository. The ALFA monitoring will be store at [atlasoff/ForwardDetectors/ALFA/ALFA\\_Monitring](#)

# Implement and test the package

## 1. Create directory in SVN repository

Directory

```
atlasoff/ForwardDetectors/ALFA/ALFA_Monitoring
```

was created with proper subdirectory structure (i.e. trunk, tags, etc). This allows checking out package using cmt command

```
cmt co ForwardDetectors/ALFA/ALFA_Monitoring
```

## 2. Create a new athena package

Description how to create a new package can be found at e.g. [SoftwareTutorialxAODAnalysisInAthena#3\\_Creating\\_an\\_analysis\\_package\\_a](#)

```
acmd.py cmt new-pkg ALFA_Monitoring  
mkdir ALFA_Monitoring/ALFA_Monitoring # dir for header files
```

## 3. Create a new tool in the package

It is possible to copy the example tool (ManagedMonitorToolTest and delete the unnecessary code (i.e. most of the code) or simply create a new class that derives from ManagedMonitorToolBase.

## 4. Install ALFA xAOD

ALFA xAOD are not available in the production release, yet, so they have to be installed separately. They can be checked out and compiled with the following commands

```
cmt co Event/xAOD/xAODForward/
cd Event/xAOD/xAODForward/cmt
cmt make
cd ../../../../
```

## 5. Add call to the new class in an existing program

In order to test the code it has to be executed. It can be done by adding the call to the new algorithm to the example provided in

```
Control/AthenaMonitoring/run
```

(see Testing the set-up). In order to do it the file

```
Control/AthenaMonitoring/share/ManagedMonitorToolTest.py
```

should be modified by adding the following lines at the end of the file:

```
from ALFA_Monitoring.ALFA_MonitoringConf import ALFAMonTool
ALFATest = ALFAMonTool (
name          = "ALFATool",
PreScale      = 0,
EnableLumi    = False,
ProcessNEvents = 40
## The next line is for cases when the AlgTool cannot be attached directly to AthenaMonManager
# ManagerName = "AthMonTestMan"
)
ALFATest.OutputLevel = INFO

ToolSvc += ALFATest
monMan.AthenaMonTools += [ ALFATest ]
```

## 6. Implement distributions

### Prepare reference distributions

### Prepare defects

### Prepare documentation of the package and instructions for the person doing monitoring

### Prepare configuration that can be submitted to the central system

### Submit the new package to the central system

# Notes

## Links

- Instructions at data quality page
- Documentation of Athena monitoring framework
- DQHistogramAnalyzer
- Muon tutorial and instructions
- HLT tutorial - Muon one is probably better
- Tier0 Update

## Presentations connected with this project

- Analysis of run 267358 [↗](#)
- Analysis of run 267358 (next presentation) [↗](#)

## Trigger aware monitoring

Every monitoring tool is prepared to select events based on trigger chain. The same histograms can be filled with different trigger chains, this requires C++ tools to have a possibility to save results to customized output directory in order not to overwrite the histograms. Presentation about this feature of the monitoring tool was given in data quality meeting [↗](https://indico.cern.ch/event/397895/contribution/4/material/slides/0.pdf): <https://indico.cern.ch/event/397895/contribution/4/material/slides/0.pdf> [↗](#)

## Meetings

### 1. Diffractive meeting 15.07.2015 [↗](#)

- (Wolfgang) Look on the 2D maps only for the very first luminosity blocks.
- (Wolfgang) Make a distribution of total number of noisy fibres for all stations in one distribution and do the checks on it.
- Keep the 2D maps only for experts and do not make any checks on it.
- Use the same procedure as for noise fibres for efficiency plots.
- (Wolfgang) It could be useful to remove bin with 0 hits in a plot in distributions of hits multiplicity and present it in separate distribution.
- (Wolfgang) The distribution of layer occupancy is with at least one hit is surely not biased by the track reconstruction, so it is a little bit safer to use it.
- (Krzysztof) Replace layer occupancy plots with 20 bins plots containing both U and V layers.
- (Leszek) Make distribution of number of tracks only for experts.
- 2D plots of track multiplicity as a function of number of hits will make things less clear and it was studied before that 3 and 5 are the best values.
- (Leszek) Keep the track length distributions for the experts and do not do any automatic checks on it.
- (Krzysztof) Make automatic comparison between U and V planes using the track length distribution.

WebNotify

## Properties

Show details  Hide details

- Set TODOMARK = (**T**ODO)
- Set INPROGRESSMARK = (**I**N **P**ROGRESS)

- Set DONEMARK = **(DONE)**
- Set QUESTIONMARK = **(question to the expert)**
- Set DISCUSSION = **(OPEN DISCUSSION)**
- Set ALLOWTOPICCHANGE = GrzegorzPawelGach LeszekAdamczyk1 KrzysztofWojciechJanas

-- GrzegorzPawelGach - 2015-06-18

---

This topic: Main > AlfaOfflineDQMonitoringIntegrationToDo

Topic revision: r33 - 2015-07-30 - GrzegorzPawelGach



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