

Table of Contents

TotemDQMLite.....	1
Concept.....	1
Tools in the package.....	1
Repository and installation.....	1
Documentation.....	1
Configuration.....	1
TotemDQMLiteBatch: User's guide.....	2
TotemDQMLiteRaw: User's guide.....	2

TotemDQMLite

A package of data-quality monitoring tools, focusing on high-level reconstruction (e.g. tracks) which is performed by the Offline SW. It is complementary tool set to the Monitor.

Concept

Data processing is split into two phases:

1. *reconstruction* is performed with standard Offline SW (CMSSW), producing RECO files
2. *visualisation* is done with a separate program reading RECO files

This organisation brings several advantages

- better separation of reconstruction and visualisation code (easier to maintain)
- reconstructed data are not lost when DQM window is closed
- DQM can be used to visually inspect RECO files made for analysis

Tools in the package

- **TotemDQMLite**: monitoring tool with graphical user interface
- **TotemDQMLiteBatch**: tool to save plots as PDF files according to predefined layouts
- **TotemDQMLiteRaw**: convenience script to perform both reconstruction and visualisation with one command

Repository and installation

TotemDQMLite is a part of the Offline SW version 7.x and can be downloaded from the standard SVN repository

```
svn co svn+ssh://svn.cern.ch/repos/totem/branches/CMSSW_7_0_4/offline/cms/w/src
```

Compilation follows the standard rules of the Offline SW.

Documentation

- TotemDQMLite: User's guide
- Instructions for use in the counting room

Configuration

TotemDQMLite is configured with files with syntax very similar to the standard CMSSW python files, for example

```
import FWCore.ParameterSet.Config as cms

TotemDQMModuleDAQ = cms.PSet(
    tagRawEvent = cms.InputTag("source")
)

TotemDQMModuleRP = cms.PSet(
    tagRawEvent = cms.InputTag("source"),
    correlationPlotsFilter = cms.untracked.string("default=0,1")
)
```

)

```
#TotemDQMModuleT1 = cms.PSet()
```

If a module is commented out or not mentioned in the file, it will not be included in visualisation. At the moment, there are four modules available: names end with DAQ, RP, T1 and T2.

If no configuration file is explicitly supplied, TotemDQMLite will use the default configuration:

```
.../src/TotemDQMLite/GUI/python/TotemDQMLite_default_cfg.py
```

TotemDQMLiteBatch: User's guide

Complete help can be obtained by running `TotemDQMLiteBatch --help`. A typical use is as follows:

```
TotemDQMLiteBatch reco_file.root -l layout1.lay -l layout2.lay
```

which will process file `reco_file.root` and save plots `layout1.pdf` and `layout2.pdf` following the chosen layouts. A collection of standard layouts can be found in:

```
.../TotemDQMLite/Modules/layouts
```

TotemDQMLiteRaw: User's guide

All the options are explained by `TotemDQMLiteRaw --help`. For each of the options, one can easily set default values, see `src/TotemDQMLite/GUI/scripts/TotemDQMLiteRaw_README`. Typical usage:

```
TotemDQMLiteRaw raw_data.srs --reco-cfg RP --reco-dir /tmp
```

The script will first check the RECO-file pool (here `/tmp`) for a file corresponding to `raw_data.srs`. If not found, it will launch CMSSW with config template `RP`. These templates are stored in:

```
.../src/TotemDQMLite/GUI/scripts
```

Once the RECO file is available, it will start TotemDQMLite for visualisation.

-- JanKaspar - 2015-10-07

This topic: TOTEM > CompDQM

Topic revision: r1 - 2015-10-07 - JanKaspar



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback