Table of Contents

Software guide to automated validation

Complete: 🟩

Introduction

Goals

The goal for automated validation is to run series of analyzers together with RelVal production. Therefore all the information produced and reconstructed is at the same time validated at the production step. Some of the developed infrastructure will be reused as part offline validation.

Offline DQM and Validation System Outline

Offline DQM core software is based on the existing software for online DQM and the same histogram registry code and output file formats are used. The system guarantees portability between online and offline, in particular online DQM modules can be run in the offline environment, with no code changes. General portability is also useful for developmentof the histogramming code which is possible in cmsRun-based interactive test cycles.

DQM histogramming code is organized in standard CMSSW modules (plain EDAnalyzers). A central (singleton) histogram registry service holds a list of all DQM histograms in the process. Histograms from different subsystems (e.g. Ecal, Hcal, ...) are distinguished by their specific root top-level folder names.

In the foreseen offline DQM workflow *one* accumulated set of histograms will be made available per run (or dataset), as soon as processing of the run has finished. No "live-monitoring" of the data is foreseen during processing.

The biggest application of the "Offline DQM" automated work flow will be prompt reconstruction at Tier-0. Other automated production areas are software and release validation, calibration jobs, MC production at Tier-X, etc.

The same webserver-based visualization GUI that is already in use for online DQM will provide access to the histogram files. A web server machine with large disk for histogram file cache is being set up and a first proof of principle test is foreseen for March.

DQM Offline Workflow

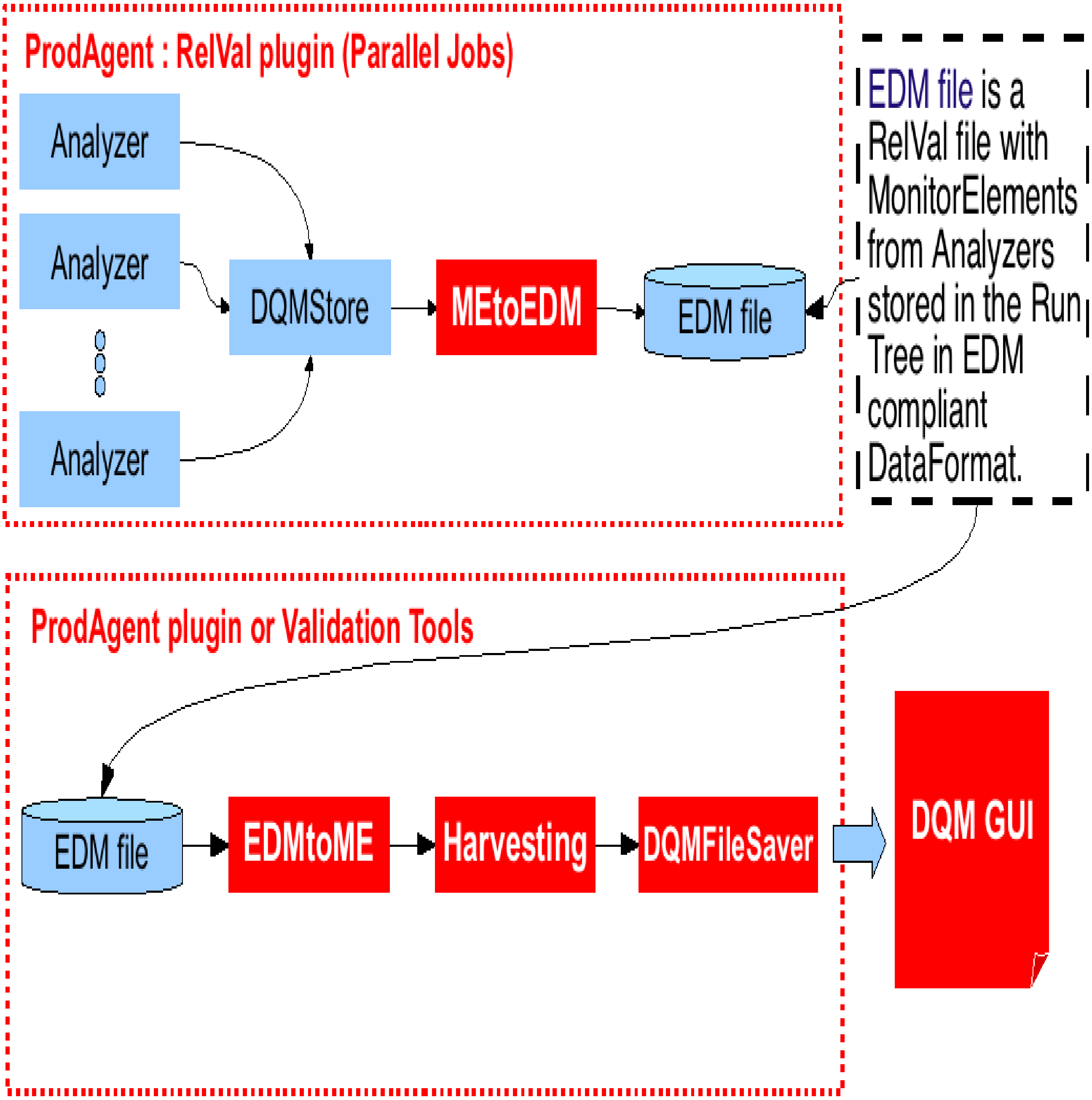
Offline event data processing is split into many single jobs, each with a relatively small number of events. At the end of each job the DQM histograms are written to the same (unmerged) EDMfile as the events and the runtree. To accumulate the full statistics, the DQM histograms then undergo three separate merge and analysis steps

- The first merge step takes place together with the event merging step (formerly EDMFastMerge). Histograms from several files are extracted from the EDMfile, summed up, and written to the merged EDM files.
- In the second merge step all merged EDM files belonging to a given run (or dataset) are input to a DQM extraction job. In this step again, the histograms are extracted from EDM, summed up and ultimately written to file. The file format of the DQM output files is plain root, like for online (TFile with internal TDirectory and TObject histogram types). These DQM output files are then stored on the webserver diskcache and their contents is visualized on request.

- Automated quality tests can be performed in a third step, using the histogram data (and reference files) that reside on the webserver disk as input. It is planned to implement the data certification algorithms in this step. The ultimate result of these data certification algorithms will be good run lists, which will be made available from DBS.

Validation Workflow

- Validation analyzers declare monitor elements/histograms via DQMStore.
- **MEtoEDMConverter** accesses the DQMStore to find what monitor elements exist.
    - At end of run, converter extracts the necessary information from the monitor element and stores it in the Run tree of the EDM file.
- **EDMtoMEConverter** accesses the information in the Run tree and uses the modified clone method of the monitor element to reconstruct the original DQM root file.
- This DQM file can then be compared against reference using standard DQM tools.
- **CMS.ProdAgent RelVal plugin** will be running analysis packages plus **MEtoEDMConverter** as part of their standard sequences.
- Another **CMS.ProdAgent plugin**, or our/custom **ValidationTools**, will be running **EDMtoMEConverter** to extract the histograms from EDM files.


ProdAgent : RelVal plugin (Parallel Jobs)
Analyzer
Analyzer
Analyzer
DQMStore
MEtoEDM
EDM file
EDM file is a RelVal file with MonitorElements from Analyzers stored in the Run Tree in EDM compliant DataFormat.
ProdAgent plugin or Validation Tools
EDM file
EDMtoME
Harvesting
DQMFileSaver
DQM GUI

Main components

- Physics and Data Quality Monitoring
- MEtoEDM and EDMtoME converters
- ValidationTools

Compliant packages

Rules

In order to be integrated into the **automated** validation process, packages need to comply with a basic set of rules.

Event by event quantities (DQM analyzer)

Most of the validation packages deal with quantities produced and analyzed event by event. What follows is the set of rules associated to this type of packages.

1. Any validation code is given as EDAnalyzer EDAnalyzer.
2. Analyzer creates validation information only as monitoring elements (ME) through DQM services.
3. Analyzer will be run as part of RelVal or standard reconstruction sequence.
4. Request to create a new RelVal dataset for a given validation needs must be negotiated with RelVal group.
5. All MEs need to be put into a directory structure with a head directory that identifies **uniquely** the source analyzer.
6. Analyzers can save MEs to DQM file only by using DQMFileSaver module.

Quantities over full statistics (Harvesting analyzer)

Some validation packages produce quantities that depend on the full statistics of a given dataset. Typical examples are packages that calculate efficiencies. Following are the rules associated to this type of packages.

1. The validation process needs to factorized in two analyzers.
2. The first analyzer produces all MEs with event by event quantities and it has to be compliant with the set of rules given previously.
3. The second analyzers calculates MEs with full statistics dependent quantities based only on the MEs produced by the analyzer.

Minimum requirement

Event by event quantities

1. A set of analyzers that use DQM infrastructure.
2. The name(s) of RelVal dataset(s) on which an analyzer will be executed.

Quantities over full statistics

1. A set of analyzer pairs that use DQM infrastructure.
2. The name(s) of RelVal dataset(s) on which the 1st analyzer will be executed.

Tutorial

- Creating a compliant DQM and harvesting analyzer

Examples

- Creating different monitoring elements .

Corresponding 'test' directory has TestValidation.cfg and EDMtoMEConverter.cfg files to run/test those MEs.

DQM Migration V3

The script will go through the base directory/any sub-directory and modify all .cc, .cpp and .h files. After this script runs in your package, the migration should be complete and ready to be committed.

```
cvs co UserCode/ksmith/Migration
cd UserCode/ksmith/Migration
```

```
./Migrate --directory=$BASE_DIRECTORY
```

Where BASE_DIRECTORY is the directory which you wish the migration script to start in.

⚠ Use this script under your on risk. We **HIGHLY** advice to make a backup copy before using this script.

Review status

Responsible: VictorBazterra

Reviewer/Editor and Date (copy from screen)	Comments
VictorBazterra - 06 Mar 2008	Page creation.
VictorBazterra - 14 Mar 2008	Adding Michael's and Andrea's information about DQM offline and validation infrstructure.
VictorBazterra - 01 Jul 2008	Updating the page with latest information about the harvesting analyzer.

This topic: CMSPublic > SWGuideValidation
Topic revision: r18 - 2008-07-01 - VictorBazterra