

Goal: provide a central web service and graphical interface to CMS conditions databases

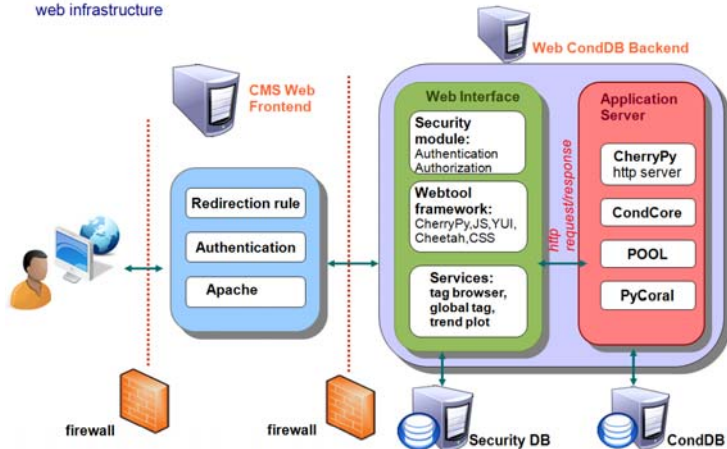
Introduction: The web application service as part of the conditions database system serves applications and users outside the event-processing. The application server is built upon conditions python API in the CMS offline software framework. It responds to http requests on various conditions database instances. The main client of the application server is the conditions database web GUI which currently exposes three main services.

The tag browser allows user to see the availability of the conditions data in terms of their version (tag) and the interval of validity (IOV).

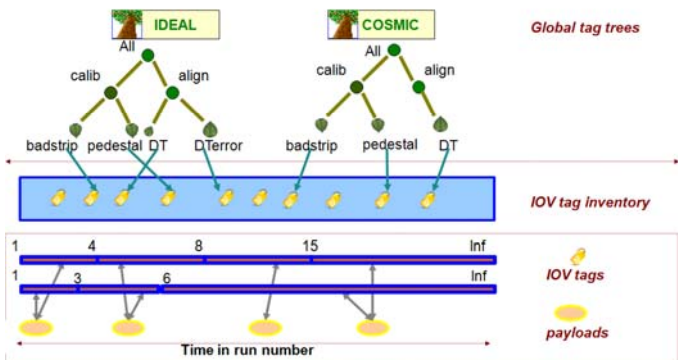
The global tag component is used by physicists to inspect the organization of the tags in a given data taking or data production while production managers use the web service to produce such tag hierarchy.

History chart plotting service creates dynamic summary and distribution charts of the payload data in the database. Fast graphical overview of different information greatly helps physicists in monitoring and validating the calibration data stored in the condition database.

1. Architecture: component-based multi-tier service over HTTP that resides behind common CMS web infrastructure



2. Conditions DB content: time-variant non-event data (payload) are stored together with their different versions of Interval Of Validity (IOV). IOV versions are identified by tags (IOV tag) which are organized as trees (tag tree). A tag tree can be transversed from any node (global tag).



3. IOV Tag Browsing Service

The screenshot shows the web interface for IOV tag management. Key features include:

- CondDB: IOV tag management:**
 - Buttons: "dynamically select data locations", "read data directly from ORACLE database or via Frontier-cache".
 - Form: Task (STRIP), Service (cms_offprod), Schema (CMS_COND_21X_STRIP), Via cache (checkbox).
 - Buttons: "List all tags", "Browse conditions data in terms of Interval Of Validity (IOV) and the version (tag)", "export the selected data from production databases to private sqlite files".
- Available tags:** A table listing tags with columns for Name, Since, and Till.
- Export selected data with optional parameters:** Form with fields for "as tag:", "first since:", and "last till:".

4. Global Tag Management Service

The screenshot shows the Global Tag Management Service interface. Key features include:

- select database account:** A dropdown menu to choose the database account.
- Global tag trees in the selected account:** A tree view showing the hierarchy of tags.
- Select a tag tree and browse:** A button to select a specific tag tree for browsing.
- IOV tags in the selected account:** A list of IOV tags for the selected account.
- Production managers can edit the tag tree and its node:** A note indicating that production managers have edit access.

5. History Data Quality Monitoring: A Python-based application for historical database inspection gives the user a quick overview of:

1. The detector conditions and performance over time
2. Source of information: Data Quality Monitoring (DQM) histograms
3. Flow chart of the procedure: short description of the procedure

Use cases: The aim is to assess the detector performance over time but also to give prompt feedback about the stability of the data taking.

Architecture and design of the historic DQM tool:

The historic DQM consists of three steps:

1. Extraction of the relevant information from the DQM monitoring elements (MEs)
2. Storage of the relevant information in the historic database
3. Creation and visualisation of the trend charts



GUI: For each database account and tag the following information can be queried: the full collection of IOVs, the values stored according to the different quantities and sub-detector regions. It is possible to apply preselection cuts on the stored quantities in the database and to black list given IOVs in order to chose the IOVs entering the trend chart.

The screenshot shows the GUI configuration for a trend chart:

- Tag:** historicFromT0_V9_ReReco
- Sub-detector regions:** Tracker
- What:** DistanceOfClosestApproach_RSTk, NumberOfRecHitsPerTrack_CKFTk, NumberOfRecHitsPerTrack_CosmicTk, NumberOfRecHitsPerTrack_RSTk, NumberOfTracks_CKFTk
- y-axis:** min: 0, max: 0
- Select Runs:** Begin: 6500, Last: 70500
- Apply preselection cuts:** Y_values >= 1, Backlisting: (empty)

Trend plots: This chart shows an example of trend chart created during the Cosmic Data taking of CMS in Autumn of 2008: the mean number of events vs time.

The mouse location helps the user to interact directly with the chart, displaying the exact run number and corresponding value.

