



Dcs Data Viewer



A Flexible Interface to the ATLAS DCS Data

C. Tsarouchas¹, S. Schlenker¹, G. Dimitrov¹, G. Jahn¹

¹) CERN, Geneva, Switzerland

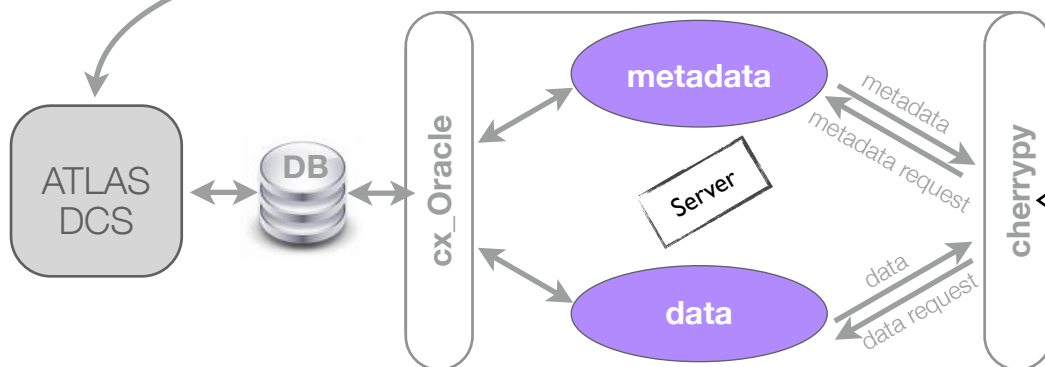
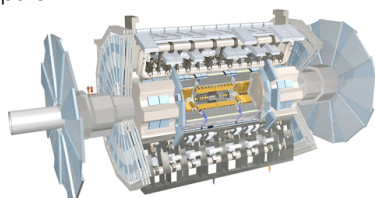
CHEP 2013
Amsterdam, Netherlands



Aim and Specifications

The DDV project aims at implementing a data viewing application for DCS data. The application targets users from the whole ATLAS collaboration allowing to access and display data from database archives. Below is the list of specifications.

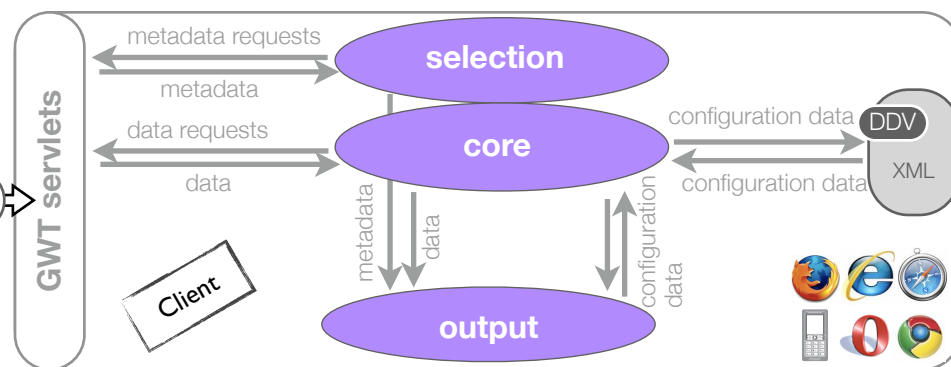
- Platform and browser independent project
- Reasonable application startup time (order of second)
- Small response time of queries (order of second)
- All possible navigation mode options (element name, alias, description)
- Well defined interfaces between server-client and client-outputs
- Request of parameters values and alarm data
- Multiple output formats (chart, tables, ascii, ROOT)
- Current configuration in XML format option
- Database protection mechanisms
- Server stand alone use



Server Client Architecture

The DDV pythonic **server** accepts requests concerning DCS information, constructs and performs database requests to the ATLAS offline DB and finally returns the results in a form of a string or of a more structured output. The server fetches both *metadata* and *data*. The DCS metadata information is copied from the database to an SQLite database cache within the DDV server, offering in this way the quickest possible response time of metadata queries and keeping the database resources as available as possible. Concerning data, a request engine with optimized SQL calls to Oracle, acquires data directly from the DB and pass it to the client through a well defined interface.

The **client** part of DDV is largely based on the Google Web Tool kit (GWT) framework for web applications where the development is done in Java while the result is a browser independent AJAX-based web application. The client interface includes the *selection* and *configuration* module where requests are constructed and outputs are configured accordingly. The final visualization of the data is done using Java Applets or JavaScript outputs which communicate with the client via a thin and well defined interface.

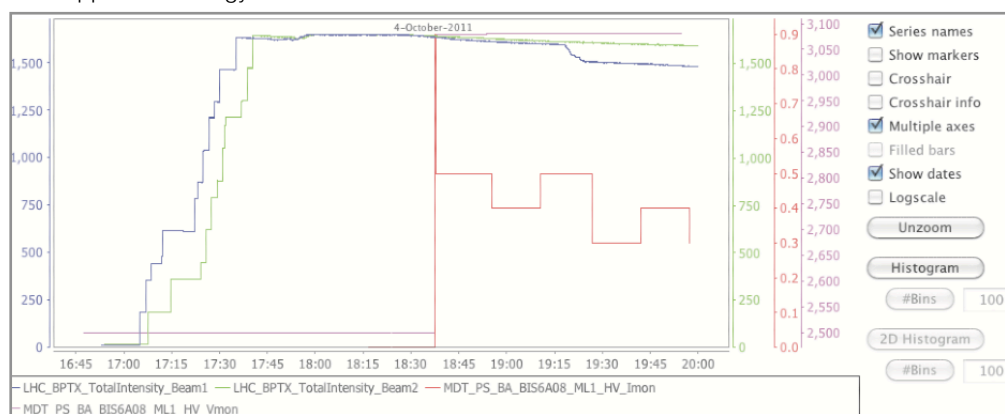


Web User Interface

The main browser interface of DDV. Below is shown the user friendly date/time selection options and the column-tree metadata navigation mode. Protection against web security attacks is foreseen and authentication constrains have been taken into account. Moreover, each user has its own dedicated repository space where request configuration files can be saved and called in a future time.

Outputs

• **Chart (Applet):** The default output for graphical representation of data. It is implemented using Java Applet technology and based on the JfreeChart libraries.



• **Chart (Java Script):** A light, highly universal graphical output implemented using JavaScript which can be visualized in environments that do not support Java Applets (e.g. smart phones).

• **Values Table (Java Script):** Table output displaying the information of metadata, date, time and recorded value. Supports filtering, sorting options.

• **ROOT:** Output in ROOT format to cover the needs of elaborate data manipulations.

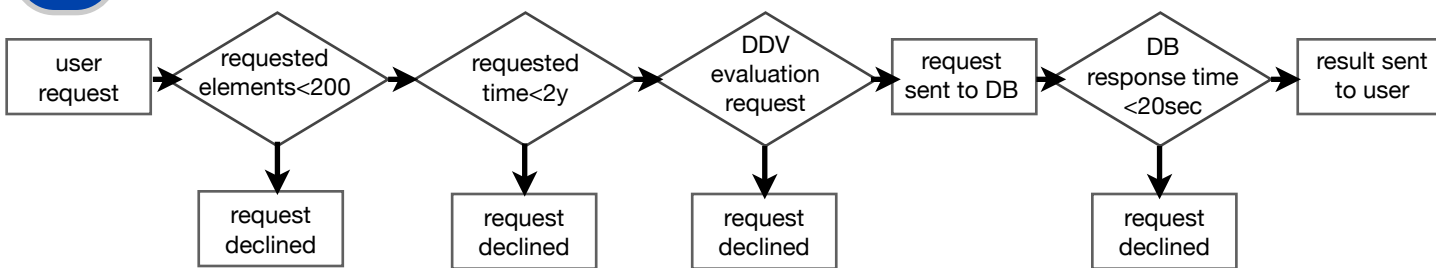
• **ASCII:** Simple plain text format.

• **Alarms Table:** Table output displaying alarms information, severity, date, time, alarm text/class, corresponding value and acknowledgement state. Supports filtering, sorting options.

| Abbr | Dir | Item | Alarm Text | Value | Class | Ack | Timestamp |
|------|------|--|------------|--------|---------------|-----|-------------------------|
| E | came | CIC RackControl DAQ Y0402D2 OutletWaterTemperature | HIGH_HIGH | 25.031 | fwErrorNack | | 16:01:24.279 2013-06-28 |
| W | went | CIC RackControl DAQ Y0404D2 OutletWaterTemperature | HIGH | 21.999 | fwWarningNack | | 11:05:22.513 2013-07-01 |
| W | went | CIC RackControl DAQ Y0504D2 OutletWaterTemperature | HIGH | 21.999 | fwWarningNack | | 11:06:23.223 2013-07-01 |
| W | went | CIC RackControl DAQ Y1004D2 OutletWaterTemperature | HIGH | 21.876 | fwWarningNack | | 11:10:27.110 2013-07-01 |
| E | went | CIC RackControl DAQ Y1006D2 OutletWaterTemperature | HIGH_HIGH | 24.889 | fwErrorNack | | 11:17:27.148 2013-07-01 |
| W | went | CIC RackControl DAQ Y0804D2 OutletWaterTemperature | HIGH | 21.999 | fwWarningNack | | 11:39:30.659 2013-07-01 |
| E | went | CIC RackControl DAQ Y0402D2 OutletWaterTemperature | HIGH_HIGH | 24.889 | fwErrorNack | | 11:41:28.066 2013-07-01 |

Features

- **Search Engine:** The big number of archived parameters (10^6) can create difficulties to non-expert users. For such cases DDV provides a *search engine* for DCS metadata.
- **Relational Queries:** Covers requests for subsets of data (e.g. case of a spike) where an accepted range of values is specified.
- **Configuration in XML.** The various configurations of the DB request and the output selections can be stored in an XML file
- **External Requests.** DDV can be called from another application by pointing at the same time a pre-saved configuration file.
- **Data Base Protection Mechanism.** DDV validates each request with a minimum-response-time cost and if all quality check propagate it to the DB (see details in the decision diagram below).



Usage

