PDV – a PVSS Data Viewer Application
D. Hoffmann, O. Pisano

To cite this version:
D. Hoffmann, O. Pisano. PDV – a PVSS Data Viewer Application. International Conference on Computing in High Energy and Nuclear Physics (CHEP 2010), Oct 2010, Taipei, Taiwan. pp.042034.
in2p3-00546068

HAL Id: in2p3-00546068
https://hal.in2p3.fr/in2p3-00546068
Submitted on 13 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Java has been made. Development is made in JDK PVSS installation at all, the choice for the machine independent, byte code generating language. In order to allow access to PVSS data to users outside of the ATLAS DCS cluster and even without DPE names are cut intelligently in order to arrange them in the tree. The DPE name display can be consulted at any time (for display and selector) from DPE names to the PVSS also contains.

**Application Features 1: Invalid data**

Invalid data tagged by PVSS through a 32 bit mask. PVSS associates validity information with each value. In case of transient errors during data acquisition, data values are still recorded, but can be flagged according to a variety of error conditions. PVSS allows to filter these data, i.e. include or exclude all or part of values with error conditions from the plots.

**Application Features 2: DPE search**

Search of DPEs with regular expression patterns on

● DPE names

● DPE comments

● DPE data and comments

Search of DPEs in the DPE Selector window

**Application Features 3: History**

User typically verify some groups of DPEs routinely or, more generally, user may want to re-make a display query that they have executed formerly. The PDV retrieves the last executed queries for each user in the user specific PVSS database (stored in the ‘data’ directory).

A scrutiny of the corresponding history menu is shown above, which also illustrates an additional feature of this history menu: All queries are stored in form of ‘history-records’ in an xml file. This file can, after acquiring some basic knowledge of the underlying XML structure, be added to history files with a standard XML editor or just any text editor, or generate a history.xml automatically.

PVSS enables the user to select any of the history records to be displayed (display record) in stamp, which allows to use PVSS for some programmed data plots semi-automatically without initial user intervention. The menu above makes use of individual identifiers (Temperatures, Dew points,...) for easy retrieval of additionally generated history records; whereas the data period is used for identification.

**Application Features 4: Export and Save Data**

The possibility of exporting displayed data after a query has been explored exhaustively and is probably the most complete feature of the PDV in its present state. Besides the standard graphical export:

- user can export the underlying data of a display as comma separated value (CSV)

Depending on level of understanding or needs, he has the choice between multiple options:

- raw data format (uncompressed from PVSS, no reconstruction)

- compressed format (compressed time wise)

- optimized compressed format where duplicate lines (cable values) are eliminated.

An interface to user is foreseen, but difficult to implement due to the requirement of machine independency, which excludes Java/Swing by principle.

**Application Features 5: Plugins**

- Accommodate subdetector/user specific requests

- Selection of DPEs by specific algorithm or interface (possibly graphical)

- DPE name/alias splitting for tree construction

- Data Export in user specific formats

- User provided code, stored in jar or class file in user directory

**Software management and frameworks**

The source code of the PDV application has been successfully managed with CVS in repository and Savannah for user feedback, bug tracking and task management. Migration from CVS to SVN has been performed by the Python script extension of it without any impact on the project.

The Eclipse IDE has not been mandatory, but turned out to be the favored tool of all developers and contributors.

Source code is open, but property of CMS/SPD2.

The initially used graphical library jChart has been replaced by JFreeChart, as we expected it to provide home support and documentation, faster reporting to bug reports and easier integration of bugs flows from our side.

**Experience and Future**

After the first year of development of the PDV application, many additional requests in addition to the basic functions have been made by the user community.

Some subdetector specific requirements have come up, which currently seems to be out of the user program's scope. They have been consistently implemented as PDV plug-ins and are ready for full integration.

The PDV is now in a maintenance phase. ATLAS has decided to develop a web based GUI for PDV requests, which uses part of the PDV code unchanged for the display of the data, which is considered a small result of open source development. The relatively low level access of the PDvss tables in Oracle is no be replaced by a common API provided by CERN-ICE. Our experience gained with the interface to the Oracle instance and the PDvss utility has left us to stick on this central development effort.

**References**

http://cern.ch/twiki/bin/view/Atlas/PDV

http://pvss.at