

EPICS ARCHIVEVIEWER

Sergei Chevtsov

SLAC, Stanford, USA

Bob Dalesio

SLAC, Stanford, USA

Kay Kasemir

ORNL, Oak Ridge, USA

Craig McChesney

LANL, Los Alamos, USA

ABSTRACT

Not long ago, members of the EPICS¹ community pointed out that existing archiving solutions lacked a user-friendly, high performance tool to browse through gigabytes of data. This paper describes EPICS ArchiveViewer, a software application that presents archived data to the user in various formats. The application is easy to install, multi-threaded, highly modular, and expandable in many ways. It is written in pure Java and is thus platform-independent.

GENERAL TERMS

Software application

ADDITIONAL KEYWORDS

EPICS, Java, data archiver, plugins

INTRODUCTION

In early 2004, ChannelArchiver² became the de facto standard solution for archiving EPICS process variables (PVs). One of the major factors was its high performance, founded in an efficient implementation of our R tree index that was presented at the EPICS meeting in Santa Fe³. Furthermore, we extended the set of features to address use cases not known to be part of other archiving solutions (either in scientific or commercial communities). Facing growing popularity of ChannelArchiver, we were determined to offer users an easy way to view and analyze the archived data. At first, we tried to extend a very popular real-time data plotter, StripTool⁴, but the prototype proved too hard to debug and maintain. We then decided to create an adequate application from scratch. After integrating an XML-RPC data server into ChannelArchiver, the server constraints on a potential client application were deemed marginal. The essential requirements were an intuitive interface, user-friendliness, and ease of installation. We chose Java for the programming language, with Swing framework providing extensible graphic widgets. As the basic plot library, JFreeChart⁵ was selected and extended. The initial look-and-feel was based on StripTool. Ultimately, we dubbed the application EPICS ArchiveViewer.

SOFTWARE ARCHITECTURE

Fig. 1 shows three major modules of the ArchiveViewer.

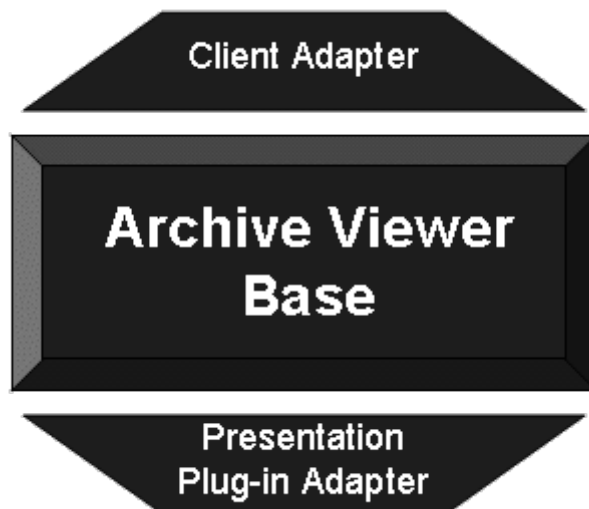


Fig. 1 Three modules of EPICS ArchiveViewer

ARCHIVEVIEWER BASE

The ArchiveViewer base is the central processing module that is responsible for interpreting user inputs to retrieve and present data in a desired way, e.g. as a chart or a spreadsheet. It incorporates a data model that is transparent from the actual data source. ArchiveViewer is thus independent of a particular server.

The user requests PV data using regular expressions and relative time ranges. Samples retrieved by a remote server are cached locally. The user then configures data to be plotted (assigning colors and/or various plot types, such as scatter or steps plots) or exported. Multiple axes are supported out-of-the-box, and there is also a mechanism to properly handle discrete data. In addition, ArchiveViewer provides a small math library for internal data analysis and manipulations (all basic arithmetical, trigonometric, and Boolean functions as well as aggregate functions are supported). While processing data, the application reports status and, in case of errors, displays comprehensible help messages. Last, but not least, each ArchiveViewer configuration can be saved to a file. We chose the XML format for its simplicity and readability.

Fig. 2 shows three major ways for the user to interact with the ArchiveViewer. The command line interface is particularly suitable for scripting. Web interface requires no installation on the user side (to a developer, ArchiveViewer provides a custom tag library for web page development). Both are limited in functionality, however. In contrast, the Swing-based XAL⁶ GUI is highly sophisticated, well documented, and contains an online help guide. It is highly customizable via preferences and features a wrapper for manipulating plots (scrolling and zooming). The XAL version is available via Java Web Start⁷, greatly facilitating installation and configuration management.

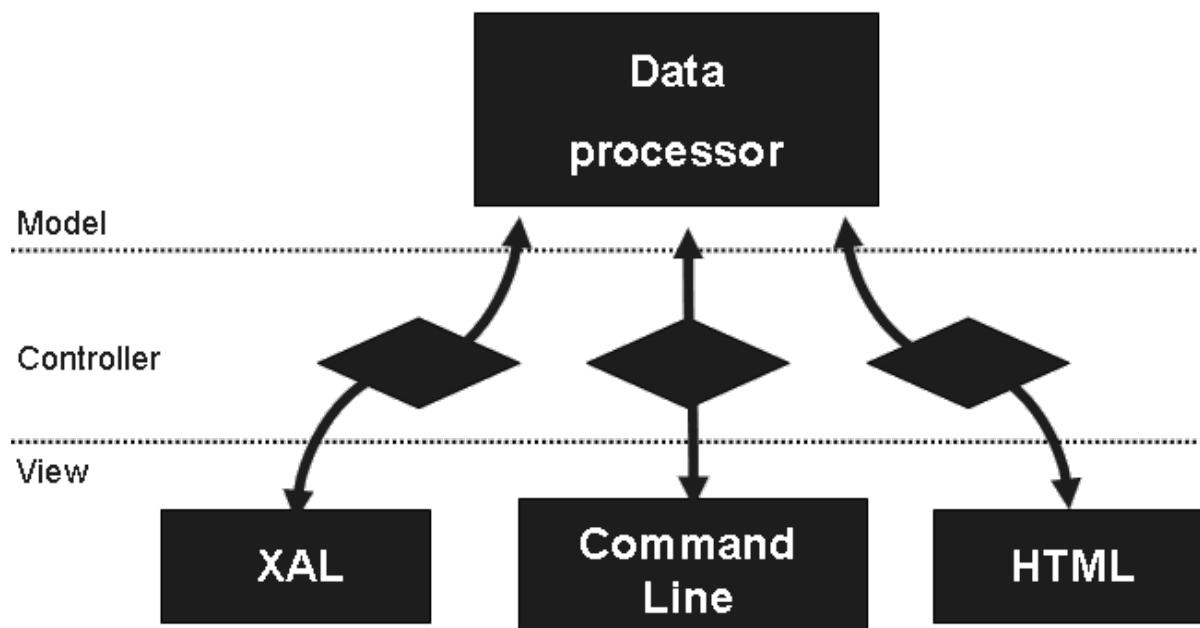


Fig. 2 ArchiveViewer's Model- View- Controller design

CLIENT PLUGINS

At the beginning, we thought of ArchiveViewer as the cutting-edge client for ChannelArchiver only. When ArchiveViewer evolved beyond initial scope, the functionality was transformed into a client plugin.

The ChannelArchiver client plugin provides full support for ChannelArchiver's XML-RPC interface. It has become the example of high-performance applications in the Java/RPC community. We conducted the following measurements: In a LAN environment, data from four PVs with 1000 values each can be retrieved and plotted in just under four seconds. ChannelArchiver's binning algorithm and the R tree index ensure that these numbers are independent of requested time ranges.

Since ChannelArchiver organizes PV data into directories, ArchiveViewer supports PV directories per default, too. In the particular case of the ChannelArchiver however, there are some limitations concerning data interpolation and alignment, otherwise fully supported by the ArchiveViewer base. For example, when using ChannelArchiver data in formula calculations, the user must ensure the PVs are archived to a common directory.

Recently, it came to our attention that other client plugins have been developed in accelerator labs all over the world (at DESY, JLab, and SLAC).

PRESENTATION PLUGINS

There are basically two ways how archived data can be presented through ArchiveViewer- values can either be plotted, or exported as text.

Export Plugins

Export of data in CSV ("comma-separated values") format is fully supported. Data in this format can be read by virtual all spreadsheet programs, such as MS Excel.

After converting binary data into string values, ArchiveViewer can write the text directly to a file, or display it on screen first, leaving the user the option to dump it to a file later. The CSV export plugin supports only a single time axis, but can handle multiple PVs of different types at the same time.

Plot Plugins

Plot plugins are the major ingredient in success of the ArchiveViewer, being in some ways the flagship that drives the entire application.

The fundamental time plot plugin renders PV values against multiple time and range axes. Range axes can be configured to display linear or logarithmic scale. If no meaningful numeric values can be assigned to a data sample (e.g. if the status information indicates invalidity), it is not discarded, but plotted as a clickable artifact underneath the main plot.

The resulting plot image can be resized, undocked to a separate window, saved to a file, and/or printed.

A waveform plot plugin can be used to plot data from waveform PVs. A data sample from a waveform PV contains an array of values that is stored and processed as one unit. The waveform plot plugin acts like a virtual oscilloscope, allowing the user to play recordings of archived waveform data with various speeds. Multiple time and range axes are supported.

As of now, the final and most important plot plugin is the correlator. It aligns non-waveform PV samples in time. After the user assigns one PV to the domain axis, remaining PVs are plotted in correlation to it. All basic functionality of other plot plugins (such as zooming etc.) is preserved.

VERSION 2.0

EPICS ArchiveViewer is currently released in version 1.2. In the immediate future, the plan is to fully integrate ArchiveViewer into Eclipse⁸ framework as part of the EPICS Office project. Additionally, we have following new features in view: runtime deployment of clients (currently, clients are deployed at compile time); an initial set-up guide; an extension of the math library; and support for German language.

REFERENCES

¹ EPICS main web site, <http://www.aps.anl.gov/epics/>

² ChannelArchiver main web site, <http://ics-web1.sns.ornl.gov/~kasemir/archiver/>

³ R tree index presentation, EPICS meeting, Santa Fe, 2004,
<http://lansce.lanl.gov/EPICS/presentations/Sergei%20Chevstovrt5E824.ppt>

⁴ StripTool main web site, <http://www.aps.anl.gov/epics/extensions/StripTool/index.php>

⁵ JFreeChart main web site, <http://www.jfree.org/jfreechart/index.php>

⁶ XAL main web site, <http://www.sns.gov/APGroup/appProg/xal/xal.htm>

⁷ EPICS ArchiveViewer, Java Web Start descriptor
http://ics-web1.sns.ornl.gov/archive/viewer/web_start/archiveviewer.jnlp

⁸ Eclipse main web site, <http://www.eclipse.org/>