

# THE DESIGN OF A PYTHON-BASED LIVE & ARCHIVED DATA VIEWER

Y. Yazar\*, T. Summers, J. Bellister, Z. Domke, F. Osman  
SLAC National Accelerator Laboratory, Menlo Park, USA

## Abstract

A new open-source PyQT-based archive viewer application is under development at SLAC National Accelerator Laboratory (see Fig. 1). The viewer's main purpose is to visualize both live values and historical Process Variable (PV) data retrieved from the EPICS Archive Appliances. It is designed as both a stand-alone application and to be easily launched from widgets on PyDM operator interfaces. In addition to providing standard configurability for things like traces, formulas, style and data exporting, it provides post-processing capabilities for filtering and curve fitting. The development teaming is current working on a release which will support standard enumerated and analog data types as well as waveforms. Extension of this support to EPICS7 normative data types such as NTable and NTNDArray is to follow.

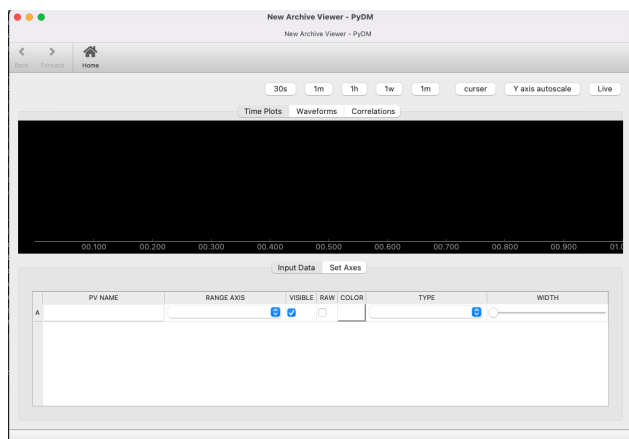


Figure 1: New Archive Viewer Under Development.

## INTRODUCTION

SLAC has a need to replace its current Java-based archive-viewer. This need was brought about through a combination of a need for new functionality to view archived and live data along with a degradation of the current archive viewer over the last decade or so.

### Problems

One of the main impetus to replace the current java-based archive viewer (see Fig. 2), which was developed around 2005 at SNS, is that it is no longer maintained at SLAC, this has lead to a loss in functionality. In addition to the degradation, there is a controls department desire to coalesce around a smaller set of technologies, in this case part of its ongoing effort to move many of the labs older Java-based

\* email: yazar@slac.stanford.edu

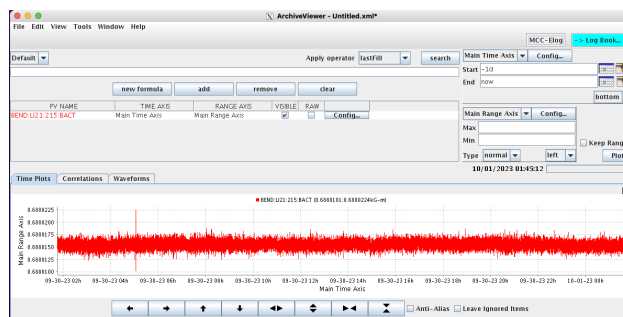


Figure 2: Current Java-Based Archive Viewer.

application to python-based ones. In addition to these two main points, there are a slew of additional challenges facing the lab with its current archive viewer:

- New functionality needed of the archive viewer going forward
- A slew of applications have been developed/used at SLAC to access archived data through the Archiver Appliance API, and although the objective is not to replace them all, there is a desire to include enough functionality to reduce the total number of applications being maintained by the lab
- The current viewer is less than intuitive with new users needing ample instructions (see Fig. 3 for some examples of this)
- The code is not well documented, making modification to the existing code challenging

These collective points are behind the decision to build a new archive viewer.

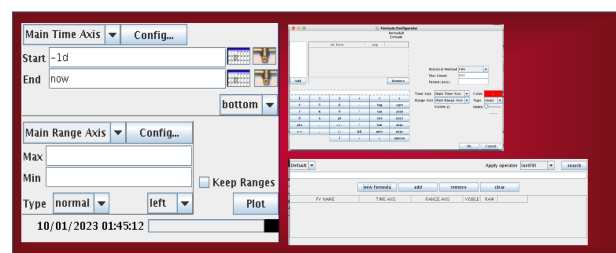


Figure 3: Current Viewers less than intuitive design.

## REQUIREMENTS

The requirements for the new archive viewer can be succinctly summarized as follows:

- Replicate the functionality of the current Archive Viewer
- Provide enhanced post-processing capabilities
- Include EPICS 7 data type integration

- replace a good portion of both live data and archived data GUI displays

These requirements are targeted to be developed over two stages of development. The first stage is to provide a full feature replacement of the current Java Archive Viewer. The second stage would be to introduce new post-processing capabilities and EPICS 7 data type integration.

## SOLUTIONS

After a period of requirements gathering, and discussions with some of the current archive viewer users, a set of approaches and solutions were formulated to build the new archive viewer.

The new Archive Viewer is being developed in PyDM, a display manager tool-set developed, and currently being used to replace EDM screens, at SLAC. PyDM is a PyQt based tool-set. The reason for this choice was to select a set of technologies which the controls department had ample developer talent to better maintain and expand the new Archive Viewer overtime. In addition to having the new archive viewer to be feature complete with the current viewer there is a need for new functionality, this includes:

- The new viewer will also replace both live data and archived data GUI displays, by combining both in one application
- It will expand on the post processing capabilities (the full list is still being worked out)
- Provide the ability to display EPICS 7 data types (this is also currently being worked out)

Another main focus for the new archive viewer is to build an application that is both more user and developer friendly (see Fig. 4). For example the intent is for the new design to be intuitive, which was a major complaint of the old Viewer. Some of the solutions that will help address this are:

- Removing plot button and plotting automatically when a PV is entered into the PV table
- Removing multiple time axes
- Building out a flexible and extensive formula editor
- The new archive viewer will be archiver agnostic, we are aiming to provide ample documentation for users to add support for their own archiver

There are also steps being taken to make the new viewer more friendly to developers, with the intent being that maintaining the project will be more sustainable and also easier for developers to add new features. Some solutions being pursued to address these two objectives:

- Selecting python, which has a large developer base.
- Create ample documentation within the code base
- Hosting a public repository for the open-source code and being active on pull requests and issues.

## IMPLEMENTATION

Following the gathering of requirements, a initial mock-up was created to have a blueprint to follow for the new archive viewer. The project is actively working towards building a first working version following this mock-up, adjusting the

design as necessary. Phase one is to tackle a replication of the current archive viewer, while checking in on key users to ensure design decisions follow the requirements and stated desire to have an intuitive design. This has entailed integrating in a PV Search functionality. Working on expanding the functionality of the PyDMArchiveViewer widget, which supports live + archived data, to also handle an only archived data mode. Other steps have been to set up GitHub Actions to:

- Run unit tests
- Establish a site for documentation (<https://slaclab.github.io/Archive-Viewer>)
- Enforce code formatting standards for all pull requests going forward

The project is moving forward now with connecting the different key parts (retrieving and displaying data, taking user input, formula editor, etc.) to test a rough working version, where it will then be tested with users to garner more feedback on the design, before repeating until all requirements are satisfied.

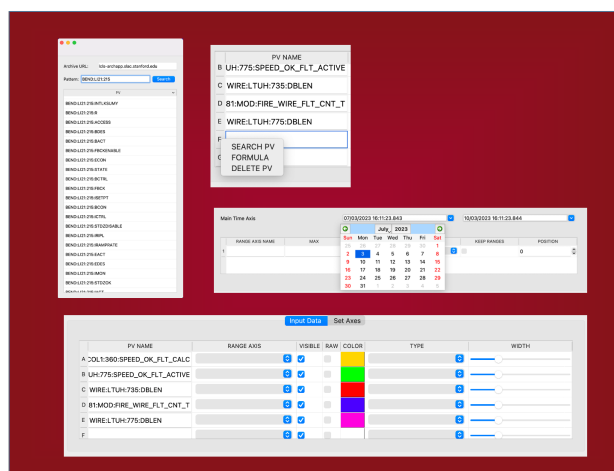


Figure 4: Elements of The New Archive Viewer.

## FINAL NOTES

The Python-based Archive Viewer is actively in development, targeting a mid-December 2023 release for the initial version, the full replacement of the current archive viewer functionality without the enhanced post processing or the EPICS 7 support. Anyone interested in keeping up with the progress can watch the GitHub repository (<https://github.com/slaclab/Archive-Viewer>) Feel free to reach out to me through my email if you have any questions or would like an update on the project. This is an open-source project, so if you are interested in contributing to the project, feel free to make issues on the GitHub repository or create pull requests. I will try my best to reply to all PRs and issue within a week.

## ACKNOWLEDGMENTS

The authors would like to thank all those who provided feedback for the gathering of requirements along with feed-

back and thoughts on the mock-up design of the new archive viewer.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI