

# CONTROL SYSTEM STUDIO (CSS) DATA BROWSER\*

K.U. Kasemir, ORNL, Oak Ridge, TN 37831, U.S.A.

## Abstract

The “Data Browser” is an interactive strip-charting and plotting program for both live and archived control system data. Implemented within the Eclipse-based Control System Studio (CSS) environment, it interfaces nicely with other CSS tools. Users can seamlessly access samples from various data sources. We describe the current features and discuss the benefits as well as difficulties that result from CSS/Eclipse.

GUIs based on Java sidestep portability issues. A framework like Eclipse [1] promises consistency as well as integrated online help, preferences and data exchange between applications that are developed by various implementers at different sites.

CSS is a collaboration between accelerator sites to develop new control-system (CS) tools. It extends Eclipse with CS specific support [2].

## ECLIPSE-BASED CONTROL SYSTEM APPLICATIONS, CSS

Traditional control system applications were based on Unix/X11, which makes them inherently hard to port to office PCs. Even graphical user interface (GUI) applications for the same toolkit like the Experimental Physics and Industrial Control System (EPICS) often lack consistent look-and-feel and only offer limited interoperability.

## DATA BROWSER

Given a Process Variable (PV) name, the Data Browser will display the PV’s data over time (Fig. 1). PV names can be entered directly, obtained by an archive search, or transferred from other CSS applications via drag & drop or context menus.

In addition to the plot there are panels for configuration, archive search, data export, waveform inspection, and more.

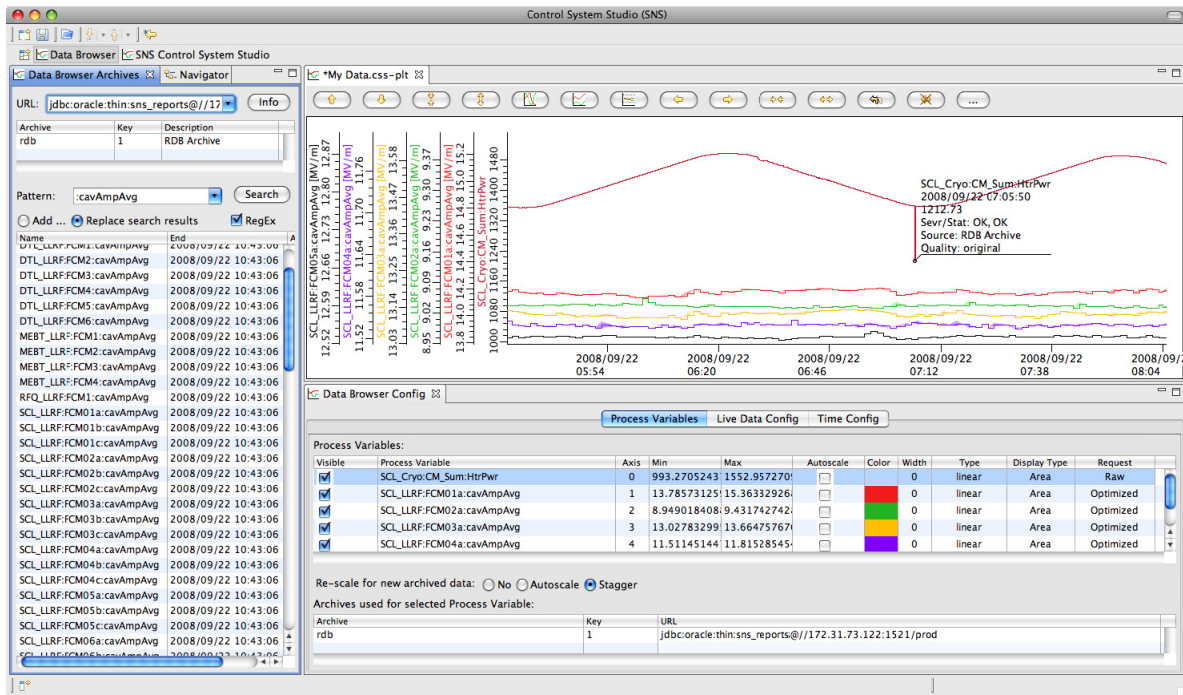


Figure 1: CSS Data Browser Plot with Configuration and Archive Search View.

## USABILITY

The Data Browser offers features similar to Strip Tool

\* SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.

[3] and the Archive Viewer [4], but with emphasis on better usability.

## Interactive

The Data Browser is highly interactive. A toolbar above the plot allows basic zoom and pan operations. Per

default they affect the whole plot, but after selecting a “Y” axis by clicking on it, operations will only target the respective axis. Users can also “rubberband” with the mouse to zoom in. There is multi-level Undo for zoom & pan operations.

The configuration panel (lower right of Fig. 1) can be freely positioned. When multiple plots are created, they share the configuration panel; it will always reflect the settings of the currently selected plot.

Changes to axis limits etc. in the configuration panel typically have an immediate effect - no need to click “Apply” or to otherwise manually trigger an update of the plot.

Whenever the addition of a PV or the selection of a new time range requires the retrieval of archived data, this occurs in background threads. The plot stays responsive, received data is displayed as it arrives.

When using drag & drop to place a new PV on the center of the plot window, it will show on a newly created axis. When dropping it onto an existing axis, it will instead be added to that axis. In any case, the axis assignments of all PVs can be adjusted in the configuration panel.

### Seamless Data Access

While running, the Data Browser periodically collects “live” samples from the control system at a configurable rate. A ring buffer of adjustable size is used to prevent memory exhaustion. Each PV can have additional sources. Initially, all PVs use configurable default archive data sources. Users can remove data sources from the configuration panel, or add new ones from the archive search panel via drag & drop.

In the plot data from all sources will appear as one trace even though samples from different time ranges might have different sources. The use of multiple data sources is in fact so transparent that users are often unaware of it until they add “Markers”, which are created by clicking on a sample in the plot, to indicate the source (Fig. 2).

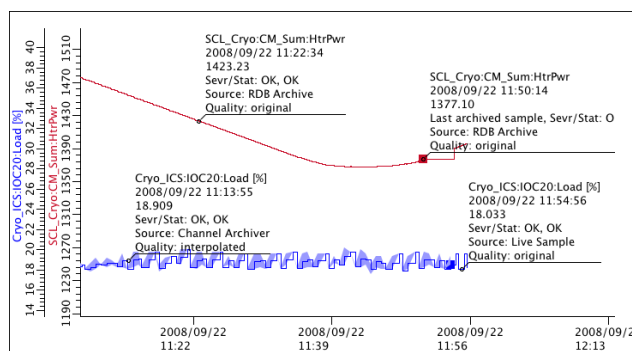


Figure 2: Markers showing values from different data sources

### Formulas

Formulas can be added to a plot. Their inputs are the other PVs or formulas of the plot. Basic computations as well as most single-input routines from java.lang.Math

are supported. Formulas are evaluated across data source boundaries, be it live data from one PV or archived data from another.

### Support for Data Reduction

When there are many archived samples for a given time range, the archive data provider will typically reduce the amount of data by sending min/max/average values. When zooming in far enough, the data provider will supply the original samples. The blue trace in Fig. 2 shows one way of displaying the optimized data: a light blue area indicates the min/max outline, while the darker trace represents the average values. “Markers” can show the data quality: Original or interpolated.

## INTEGRATION

The key benefit of using a framework like Eclipse and collaborating on CSS lies in the resulting integration of the tools.

### Data Exchange

Since the Data Browser deals with PVs, it integrates with all other PV-aware CSS tools: When right-clicking on a PV in the Data Browser, a context menu allows users to send this PV to any of the other PV-aware tools, and vice versa.

In addition, Data Browser samples can be sent to the CSS Post Analyzer for basic curve fitting and correlation, or exported to files in spreadsheet-type format.

Snapshots of a Data Browser plot can be saved as images (PNG format). Images with added comments can also directly be entered into an electronic logbook.

### Preferences, Online Help

The Data Browser has various settings that are adjustable in a preference GUI. It offers many pages of online help. Maybe even more important is that these GUIs also include the preferences and online help of the underlying data interfaces, as well as those of other CSS tools with which the Data Browser can interact.

## DATA INTERFACES

To allow for different control systems, all access to data uses interface layers, and implementations are provided in separate plug-ins. Applications like the Data Browser automatically detect the installed plug-ins on startup without need to recompile any code.

In principle, Eclipse users could even disable plug-ins and load new ones at runtime to change from one control system protocol to another without restart of the application. The current applications, however, do not support this because we see no need for it at this time.

When for instance the Logbook implementation is missing, the related features are obviously not available, but the application will otherwise continue to function.

### *Live Samples*

Access to live samples uses a “PV” interface layer. So far implemented for EPICS Channel Access, it supports both the pure Java CAJ library and the JNI binding to the EPICS base binaries [5]. Installation of the EPICS plug-in automatically adds preference pages and online help for EPICS network client configuration to CSS.

### *Archived Data*

Archived data is accessed via an API with implementing plug-ins available for the XML-RPC based Channel-Archiver data server, the DESY AAPI archive server, the DESY archive record, the DESY Data Logger, and the new RDB-based Channel Archiver (Oracle, MySQL). Live and archived data samples use the same Value interfaces for numeric, enumerated and string values, scalars and arrays.

### *Logbook*

Support for electronic logbooks has been implemented for the SNS “ELog”.

## ISSUES

The usability, interoperability, and portability of CSS tools is great for the end user, but it requires significantly more development time than one-of, standalone, non-portable tools.

A related issue is packaging and deployment. The Data Browser itself is not site specific, but to be most useful, it should be offered with other CSS tools and site-specific preferences and extensions like Logbook access and database lookups. In the past, accelerator sites have used for example TCL/TK or Python to glue various tools together. CSS is a better glue, but still requires somebody at each site to perform the gluing. End-user deployment is currently via download from a web page; Java web start is unfortunately not supported.

## REFERENCES

- [1] “Eclipse an open development platform”, <http://www.eclipse.org>.
- [2] Jan Hatje et al, “Control System Studio (CSS)”, ICALEPCS2007, Knoxville, 2007, MOBFB03.
- [3] StripTool documentation and download, <http://www.aps.anl.gov/epics/extensions/StripTool/index.php>
- [4] Java ArchiveViewer for Channel Archiver, <http://ics-web.sns.ornl.gov/archive/viewer> (no longer maintained)
- [5] Cosylab Channel Access for Java, <http://caj.cosylab.com>