ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM011

UPGRADING AND ADAPTING TO CS-STUDIO PHOEBUS AT FACILITY FOR RARE ISOTOPE BEAMS

T. Ashwarya, J. LeTourneau, M. Ikegami, C. Morton
Facility for Rare Isotope Beams, Michigan State University, East Lansing, USA

Abstract

The Facility for Rare Isotope Beams (FRIB) has been an early adopter of the CS-Studio ecosystem for its needs for a feature-rich and user-friendly interface with EPICS and the underlying accelerator controls infrastructure. For more than a decade, FRIB has developed thousands of operator displays spanning many areas of accelerator operations and engineering using the CS-Studio display tool "BOY". CS-Studio has provided many useful control system tools like an alarm system (called "BEAST"), scan system, channels aggregator, display manager and more to support controls operations of the FRIB accelerator. In recent years, there has been a major redesign of the CS-Studio software architecture resulting in the new and upgraded CS-Studio Phoebus. Phoebus replaces the Eclipse RCP- and SWT-based Java framework with modern Java standards like JavaFX, SPI, Adapters and more. This paper details the efforts that have been made at FRIB to adapt and migrate to the upgraded CS-Studio Phoebus for FRIB's operations and engineering needs.

ALARM SYSTEM

FRIB deploys over 20 instances of the CS-Studio alarm system [1] to monitor thousands of EPICS [2] Process Variables (PVs) throughout the FRIB accelerator system [3]. Alarm systems are provided a configured list of PVs to monitor and report when the PV values are out of their normal value range or get disconnected. Operators in the FRIB control room use the alarm system to determine which PVs are not in an okay state and need attention. Some engineering groups have their own specialized alarm servers that notify them through emails and phone texts when their concerned PVs are not in the state that they should be.

The alarm system of CS-Studio changed its implementation from Apache ActiveMQ and a relational database to Apache Kafka for Phoebus [4]. At FRIB, we run a 3-node Kafka cluster to provide a fault-tolerant and load-balanced backend for the new Phoebus alarm system. Scripts provided with the Phoebus alarm server take care of correctly creating and configuring the Kafka topics for the alarm system. The alarm tree configuration, consisting of a list of PVs and their related alarm settings for legacy alarms, are automatically compatible with the new Phoebus alarm system, reducing setup efforts to a minimum.

Adopting the New Alarm System

The new Phoebus alarm system's client user interface has a very similar look and feel to the legacy alarm client. We observed a much faster performance with the importing time of the alarm tree configuration to the new Phoebus alarm server in comparison to the legacy BEAST alarm server. There are a few differences to the email/text notification behaviour of the new alarm system which were added to address the issue of duplicate alarm notifications that existed in the legacy alarm system. There are two additional alarm features available with the new Phoebus alarm system for logging the history of alarm states for all PVs and the history of alarm configuration updates. Both of these features have been determined to be effective diagnostic tools for alarm users and system maintainers.

Additional features that were requested by FRIB users to be added to the Phoebus alarm system have also been implemented. These features included a mode to disable email notifications for alarms temporarily. It's a feature used extensively by FRIB Operations to disable alarm email notifications when operators are present in the FRIB control room to live-monitor and address alarms in person. Alarm email notifications are later re-enabled so that alarms can be monitored remotely and catered to during offline operation hours. In Fig. 1, the yellow mail icon in the alarm table's toolbar is used to disable and re-enable email notifications for alarm system.

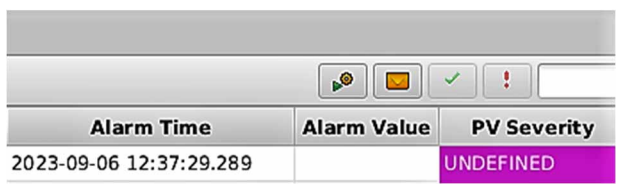

Figure 1: Mode to disable/re-enable email notifications.

The Phoebus alarm system provides an authorization mechanism to allow only selected and authorized users to be able to interact with their alarm server, change its running alarm configuration, or edit the alarm PV tree. As per FRIB's requirement for running many alarm servers within the same network, the alarm system's authorization mechanism has been extended to have authorization rules set on a per-alarm-server instance basis. This allows authorized users of an alarm server to be able to interact only with their relevant alarm server and not with alarm servers belonging to other groups or parties. This feature helps to avoid accidental updates an unauthorized user might cause to an alarm server owned and maintained by somebody else. Figure 2 shows the various alarm views supported by the Phoebus alarm system.

* Work supported by the U.S. Dept. of Energy Office of Science under cooperative Agreement DE-SC0023633

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM011

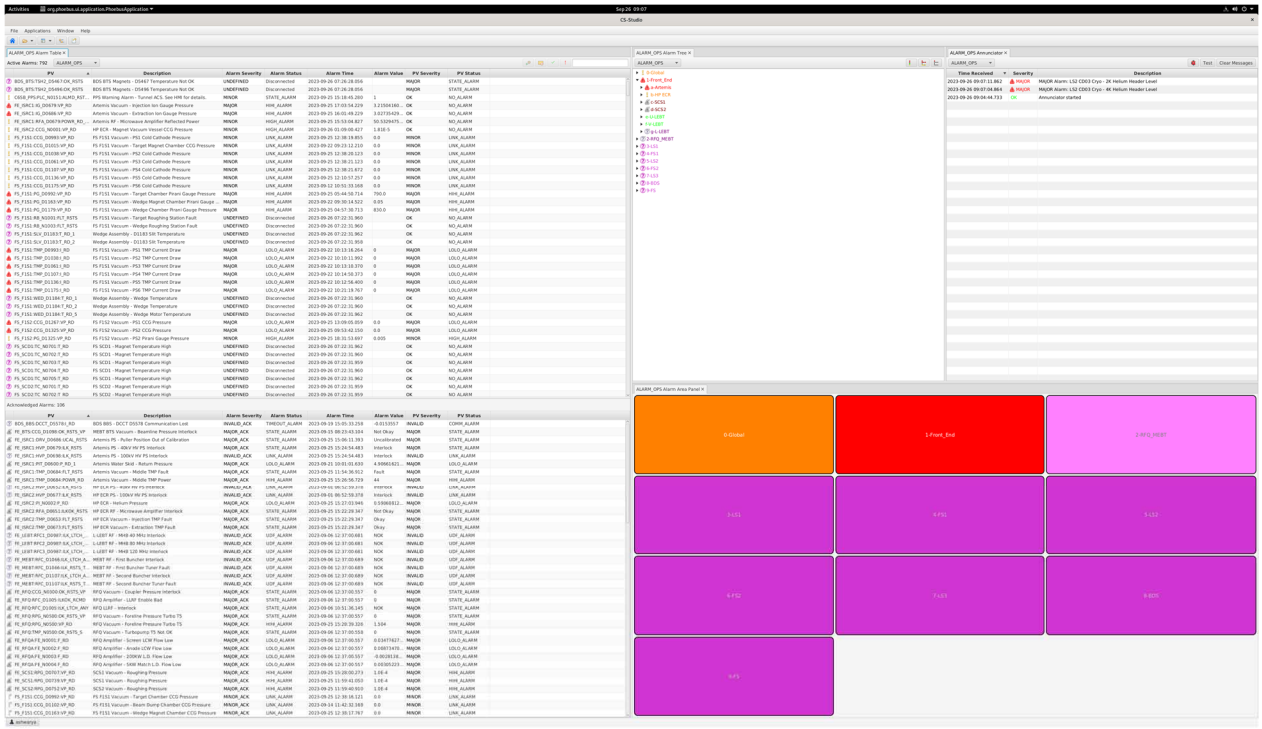

Figure 2: Phoebus showing alarm table (left), alarm tree (top center), annunciator (top right) and area panel (bottom right).

OPERATOR INTERFACE DISPLAYS

There are thousands of operator interface displays spanning across the FRIB beamline that have been developed by FRIB users over a decade with the CS-Studio BOY tool. As it is not feasible to re-develop all of these displays for Phoebus, we have utilized the auto-conversion tool provided with Phoebus to convert displays from the old ".opi" format to the new Phoebus-compatible ".bob" format. This tool has helped us convert the bulk of our displays leaving most of our effort focussed towards the testing of converted displays and addressing parts that didn't convert with the conversion tool.

We have observed some significant advantages with the auto-conversion tool. First, it was able to convert old BOY widgets to new Display Builder widgets in most cases without needing any modifications from users. It reported through warnings when it came across a missing widget, missing widget property, or a missing application programming interface (API) for scripts during conversion which helped us to add the missing entity to the Phoebus Display Builder to ease the transition from BOY. The auto-conversion tool was also able to correct widget types when they were used in a wrong context in the old BOY display; for example, "Text Update" widgets without configured PVs were automatically replaced with static "Label" widgets accompanied by warnings about these replacements. The automatic patching of script imports was helpful to skip updating every widget script. Most of the manual work was restricted to fixing scripts that had a different API in Display Builder and to fixing any plot widgets that did not belong to the default converted plot widget type. The auto-conversion tool has since been expanded to address any missing mappings between a BOY widget and a new Display Builder widget that were reported by our users.

At FRIB, users developed some scripts for bulk-fixing the common issues they came across with the newly converted files. Some of the simpler fixes like updating the associated linked files within a display with newly converted ".bob" files was achieved with the help of a script. LED widgets in BOY allowed a script or rule comprising of state of multiple PVs to decide the LED status while LED widget in Display Builder restricts the LED status to depend only through the configured PV property. In order to easily transition hundreds of our LED widgets, users ran scripts to update their LED widget to utilize the PV formula to implement the logic from their old LED widget.

In the legacy CS-Studio, perspectives are used to create layouts consisting of various windows and tabs to help organize the workflow of a user. There is some user effort required in re-creating the old CS-Studio's perspectives with the new Phoebus equivalent "Layouts" to achieve the same workflow. Figures 3 and 4 show the operator displays for the FRIB LINAC overview run with Phoebus.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM011

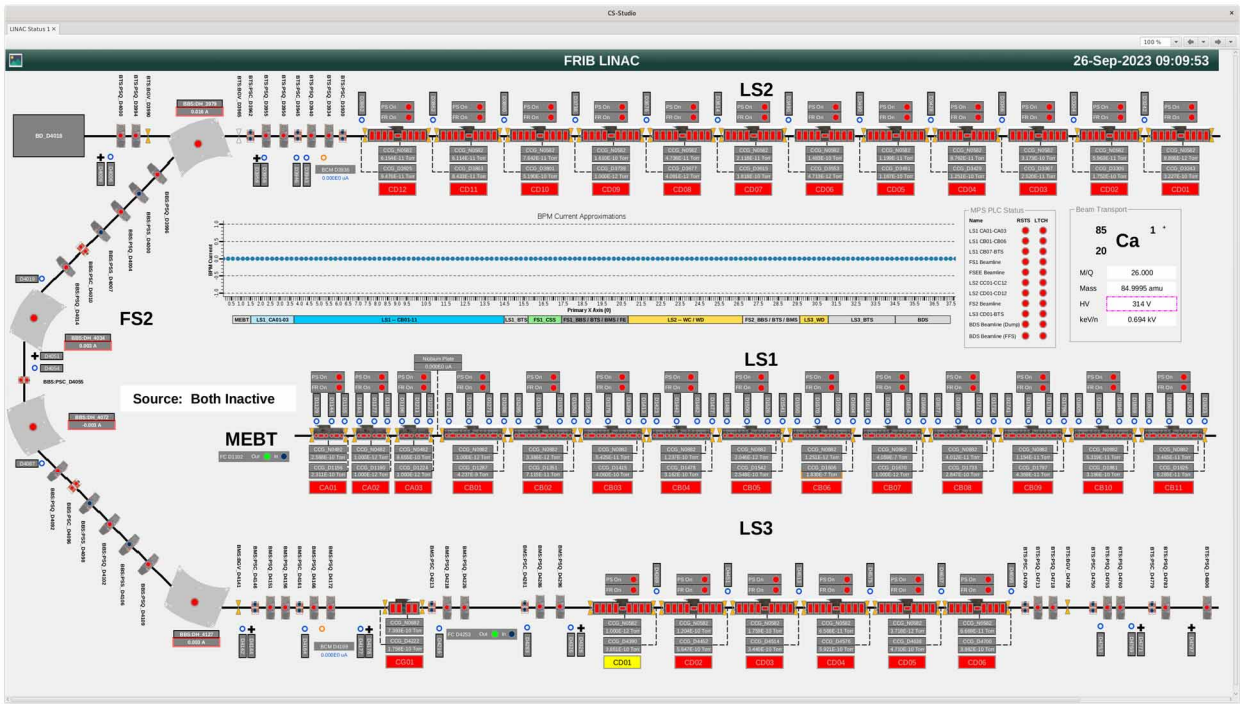

Figure 3: FRIB LINAC West in Phoebus Display Runtime.

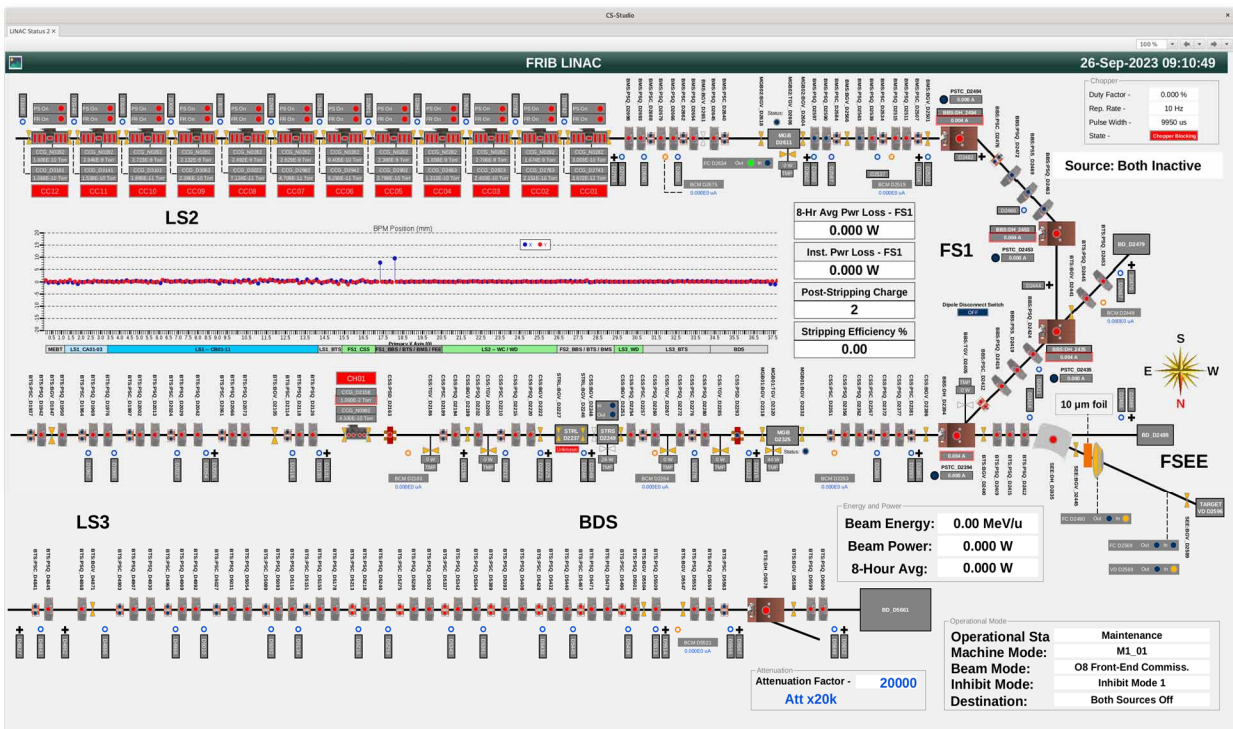

Figure 4: FRIB LINAC East in Phoebus Display Runtime.

SAVE-AND-RESTORE

Phoebus Save-and-Restore is a tool to take snapshots of a pre-configured list of PVs at a specific time and write values of a snapshot back to PVs at a later time. The old Save-and-Restore tool has Git Repository or a relational database engine for its backend where it stores the beamline savesets and their snapshots. The backend for Phoebus Save-and-Restore has been designed with Elastic Search for the storage for beamline savesets and snapshots. The transition to the Phoebus Save-and-Restore required a method to migrate savesets and snapshots from Git to the new Elastic Search backend. At FRIB, we used the Git migration tool provided with the Phoebus Save-and-Restore

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM011

service to migrate hundreds of beamline savesets and snapshots to Phoebus. Figure 5 shows the migrated git repository to a new directory in the Phoebus Save-and-Restore tree.

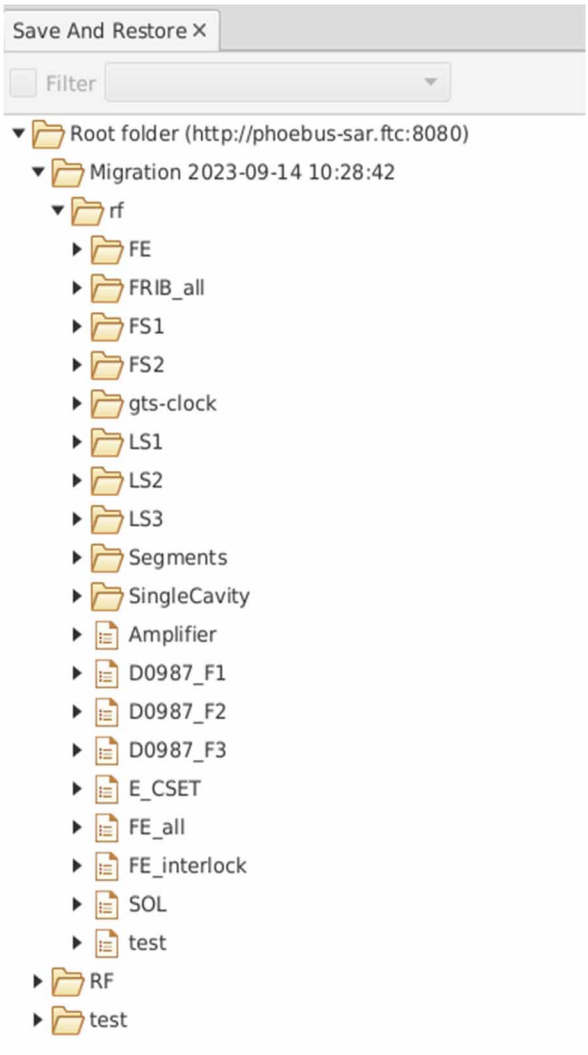

Figure 5: Migrated Save-and-Restore.

OTHER TOOLS

Phoebus provides many other controls system tools that share a similar user interface and functionality as the old CS-Studio but have been improved for user experience and system maintainability. For instance, the Probe tool has been extended to display PV values in formats other than the default and a new GUI version of Probe called "Probe Display" is available with Phoebus. A new tool "3D Viewer" has been added to allow users to configure 3 dimensional structures using spheres, cylinders and boxes which can be rotated, zoomed and moved when rendered on screen.

SUMMARY

FRIB is in the process of transitioning to the new and upgraded CS-Studio Phoebus. We are utilizing a combination of the Phoebus auto-conversion tool, user scripts and manual testing to migrate our large number of displays to Phoebus Display Builder. We have deployed multiple instances of the Phoebus alarm server across the FRIB beamline that has been robustly providing alarm monitoring to the FRIB Operations and various engineering groups. In the coming months, we plan to transition all of our displays for all FRIB beamlines to Phoebus and to decommission the old CS-Studio and its services entirely.

ACKNOWLEDGEMENT

We thank the CS-Studio collaboration for their contributions towards the development, maintenance, and education of CS-Studio Phoebus. We also thank the FRIB operators and engineers who have contributed towards the conversion efforts, provided feedback, and led the acceptance of the new CS-Studio.

REFERENCES

[1] M. R. Clausen, C. H. Gerke, M. Moeller, H. R. Rickens, and J. Hatje, "Control System Studio (CSS)", in *Proc. ICALEPCS'07*, Oak Ridge, TN, USA, Oct. 2007, paper MOPB03, pp. 37-39.

[2] EPICS Base releases, https://epics-controls.org/resources-and-support/base/epics-7/

[3] K.-U. Kasemir, "CS-Studio Alarm System Based on Kafka", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1511. doi:10.18429/JACoW-ICALEPCS2019-WESH2001

[4] K.-U. Kasemir and M. L. Grodowitz, "CS-Studio Display Builder", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 1978-1981. doi:10.18429/JACoW-ICALEPCS2017-THSH303