# Development and Implementation of the Phoebus Alarm System Software Packages for the Laser Interlock System

Peter Bonneau, Mary Ann Antonioli, Aaron Brown, Pablo Campero, Brian Eng, George Jacobs,
Mindy Leffel, Tyler Lemon, Marc McMullen, and Amrit Yegneswaran
*Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606*
November 7, 2023

This note presents the development and implementation of the CS-Studio Phoebus alarm system software packages for the laser interlock system of the laser test lab, where the quartz bars of the Electron Ion Collider's (EIC) Detection of Internally Reflected Cerenkov Light (DIRC) detector are to be tested.

The Phoebus alarm system being developed will be tested with a real-time data source—the laser interlock system [1]. The test setup and the user interfaces to monitor and control the test have been designed [2] and developed to work with a National Instruments cRIO, an EPICS softIOC, and Phoebus alarm system software packages—core programs required for all Phoebus alarm systems, Fig. 1.

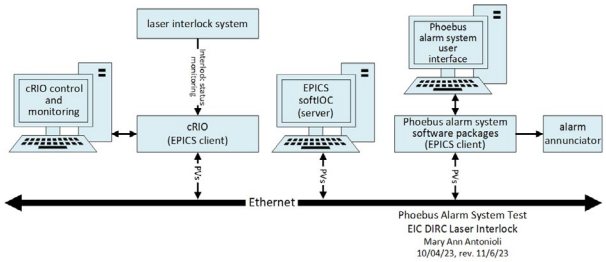


FIG. 1. Phoebus alarm system's test-setup schematic.

Code for the integrated Phoebus alarm system software packages, which work together to monitor the laser interlock status signals, consist of the Apache Kafka messaging streams infrastructure, the Phoebus alarm server, CS-Studio Phoebus alarm system user interface, and alarm annunciator, Fig. 2.

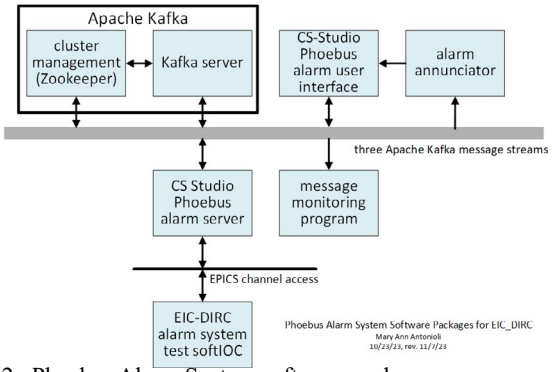


FIG. 2. Phoebus Alarm System software packages.

Phoebus alarm system applications use Apache Kafka messaging streams infrastructure for inter-process communication. Apache Kafka requires Kafka Zookeeper for cluster management and Kafka Server for hosting the alarm system messaging streams, two separate processes, running on the alarm system's host Linux computer.

For the test, a *zookeeper.properties* configuration file was written to define the Kafka Zookeeper process initialization parameters such as the client port number and the location of the transaction log file. For the initialization of the Kafka Server process, a *server.properties* configuration file was written to define the settings for the location of the message files, data buffer sizes, and communication time-out settings. The Kafka configuration files were tested by running the processes and checking the log files for errors.

A script that creates the three Kafka message streams—*state*, *command,* and *talk*—that provide communication between alarm applications was developed, Fig. 2. The script is used only once, at the creation of the alarm system to initially program the Kafka Server that defines the operational parameters of the streams, including the name of the streams, message sizes, and the location of the message log file.

To create a new alarm system, a build of Phoebus from source code was required. As a prerequisite for the build, Phoebus configuration files were written to define the alarm system's operating parameters. In the *alarm_preferences.properties* file, settings and options such as the name of the alarm server, communication port numbers, time-out limits for EPICS process variables (PVs), alarm annunciator options, and customization of the alarm user interface menus, Fig. 3, for the test were defined.


FIG. 3. Custom Phoebus Alarm System user interface.

Because Phoebus does not have an application that displays the text of Apache Kafka message streams, a script was developed to monitor the three Apache Kafka message streams, which facilitates debugging of the applications.

In conclusion, CS-Studio alarm system software packages—core programs required for all Phoebus alarm systems—for the laser interlock have been developed and implemented.

[1] T. Lemon, et al., *Design and Features of the EIC DIRC Laser Lab's Laser Interlock System*, DSG Note 2023-01, 2023.
[2] P. Bonneau, et al. *Development of the CS-Studio Phoebus Alarm System for the EIC DIRC Laser Interlocks*, DSG Note 2023-43, 2023.