

Testing of the CS-Studio Phoebus Applications and Alarm System Core Programs

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
 February 28, 2023

After the rebuild of CS-Studio Phoebus V4.6.10, tests were developed to verify the correct operation of the Phoebus applications and the alarm system core programs. The tests show that the Phoebus applications and the alarm system core programs are operating properly.

The Phoebus applications and core programs that functioned correctly in the alarm system being developed [1] with Phoebus V4.6.10, failed suddenly to connect to process variables (PVs) from the EPICS softIOC via EPICS channel access. Debugging revealed [2] that file corruption, possibly due to a failing disk drive, caused the connection failure. After rebuilding Phoebus V4.6.10, the applications and alarm core programs were tested to verify proper operation. The summary of the alarm system tests is shown in Table I.

After booting the Phoebus development computer, using the Linux *systemd* commands, the status of the alarm core programs [3]—Kafka Zookeeper, Kafka server, and the Phoebus alarm server—were checked to verify that they were sequenced correctly and running properly.

Next, the Phoebus alarm server, which uses the *procServ* interactive command wrapper to provide remote access via telnet to the alarm system command console, was tested and the program logfiles were verified to be correct for Apache Kafka Server and ZooKeeper, which generate the communication streams between the core programs.

Then, Phoebus applications and the alarm core programs were tested with the Phoebus alarm test system, Fig. 1, for which an EPICS host-based softIOC was developed [4]. The softIOC user interface, Fig. 2, was used to test the connection of the PVs generated by the softIOC to Phoebus by running the display application. The softIOC user screen displayed

the active PVs indicating that Phoebus was connecting to the softIOC.

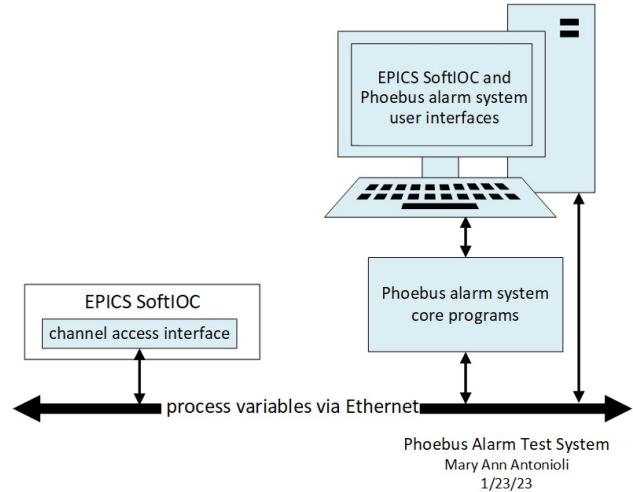


FIG. 1. Phoebus alarm test system.



FIG. 2. SoftIOC user interface for the Phoebus alarm test system.

Program name	Program function summary	Test summary
Phoebus alarm server	Monitors EPICS PVs for alarm conditions via channel access Stores alarm configuration settings for each PV	Verified monitoring of PVs from test SoftIOC Verified latching of PV value and time upon an alarm condition Verified readback values for alarm severity Verified PV alarms on HIHI, HIGH, LOW, LOLO conditions Verified alarm acknowledgment Verified the storage of PV alarm configuration settings
procServ	Provides remote access to the alarm server command console via Telnet	Verified Telnet connection to alarm server command console
Apache Kafka Zookeeper	Kafka cluster system management	Verified correct management of the Kafka server and generation of logfiles
Apache Kafka server	Hosts the alarm system message streams	Verified correct generation of the three alarm system Kafka streams
SoftIOC user interface	User alarm monitoring and system configuration	Verified correct user interface control and monitoring of softIOC and Phoebus alarm system

TABLE I. Alarm system program test summary.

Subsequently, using the EPICS softIOC user interface, alarm limits were set on the PVs generated by the EPICS soft-IOC. When an alarm condition occurred, the Phoebus alarm server correctly latched the PV value and displayed it with the timestamp of the occurrence. The alarm limits for HIHI, HIGH, LOW, and LOLO were tested for latching and alarm severity and verified to be correct, Fig. 2.

Finally, the alarm server PV alarm acknowledgement function and the alarm server configuration settings for the PVs monitored by the alarm server were verified to be working correctly.

In conclusion, after the rebuild of CS-Studio Phoebus V4.6.10, tests were developed to verify the operation of the Phoebus applications and the alarm system core programs. The tests indicate the system is running properly.

- [1] [P. Bonneau, et al., *Proposal to Implement Alarm System in Control System Studio Phoebus for the Hall C Neutral Particle Spectrometer*, DSG Note 2021-37, 2021.](#)
- [2] [P. Bonneau, et al., *Debugging CS-Studio Phoebus Version 4.6.10 Used for the Development of the Alarm System*, DSG Note 2023-03, 2023.](#)
- [3] [P. Bonneau, et al., *Automated Startup and Sequencing of the CS-Studio Phoebus Alarm System Core Programs*, DSG Note 2022-16, 2022.](#)
- [4] [P. Bonneau, et al., *Development of the EPICS Software Input/Output Controller for Testing the Phoebus Alarm System of the Hall C Neutral Particle Spectrometer*, DSG Note 2022-06, 2022.](#)