

# EPICS: CSS-Phoebus

Peter Bonneau

2023-02

An EPICS alarm system based on CS-Studio Phoebus is under development. Phoebus will be used for new EPICS system development and will replace the existing Eclipse-based CS-Studio systems as detailed in note [DSG Note 2021-37](#) and talk [DSG Talk 2021-17](#).

While working on the development of the alarm system with Phoebus V4.6.10, the applications that had previously functioned correctly failed to connect to process variables (PVs) from a softIOc via EPICS channel access. The debugging revealed (software memo [2023-01](#)) that file corruption, possibly due to a failing disk drive was the cause of the EPICS channel access connection failure.

After the successful rebuild of Phoebus V4.6.10, the applications and alarm core programs were tested to verify proper operation. First tested was the automatic start and sequencing of alarm core programs ([DSG Note 2022-16](#)) upon Linux system boot.

After booting of the Phoebus development computer was completed, the status of the alarm core programs was tested using Linux [systemd](#) commands. The core programs Kafka Zookeeper, Kafka Server, and the Phoebus alarm server were all correctly sequenced and were running. Figure 1 shows the program status for the Phoebus alarm server.

Phoebus applications and the functionality of the alarm core programs was tested next with the Phoebus alarm test system (figure 2). An EPICS host-based softIOc was developed for the test system ([DSG Note 2022-06](#)). The softIOc user interface (figure 3) was used to test the connection of the PV's generated by the softIOc to Phoebus by running the display application. The softIOc user screen successfully displayed the active PVs verifying that Phoebus was connecting to the softIOc.

The functionality of the alarm system was also verified with the Phoebus alarm test system. Using the softIOc user interface, alarm limits were set

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system**
- **Testing Phoebus applications after system rebuild**

# EPICS: CSS-Phoebus

on the PVs generated by the softIOc. 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.

The alarm server PV alarm acknowledgement function was tested and verified to be correct. The alarm server configuration settings for the PV's monitored by the alarm server was also verified to be working correctly. The summary of the alarm system tests is shown in Table 1.

Program Name	Program Function Summary	Test Summary
Phoebus alarm server	<ul style="list-style-type: none"> <li>Monitors EPICS process variables (PVs) for alarm conditions via channel access</li> <li>Stores alarm configuration settings for each PV</li> </ul>	<ul style="list-style-type: none"> <li>Verified the monitoring of PVs from test SoftIOc</li> <li>Verified latching of PV value and time upon an alarm condition</li> <li>Verified read-back values for alarm severity</li> <li>Verified PV alarms on HIHI, HIGH, LOW, LOLO conditions</li> <li>Verified alarm acknowledgment</li> <li>Verified the storage of PV alarm configuration settings</li> </ul>
proServ	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
CS-Studio Phoebus alarm test system user interface	User alarm monitoring and system configuration	Verified correct UI control and monitoring of IOc and Phoebus alarm system

Table 1. Alarm System Program Test Summary

```
[bonneau@fedora Desktop]$ sudo systemctl status alarm_server.service
[sudo] password for bonneau:
● alarm_server.service - Phoebus Alarm Server
   Loaded: loaded (/etc/systemd/system/alarm_server.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2023-02-11 14:50:03 EST; 21min ago
   Main PID: 1220 (procServ)
     Tasks: 68 (limit: 37873)
    Memory: 256.7M
       CPU: 9.583s
   CGroup: /system.slice/alarm_server.service
           └─ 1220 /usr/bin/procServ --foreground --noautorestart --name alarm-server --chdir /home/bonneau/Downloads/phoebus-master
           └─ 1885 /bin/sh ./alarm-server.sh -config Hall-C-NPS
           └─ 1887 java -jar ./target/service-alarm-server-4.6.10-SNAPSHOT.jar -config Hall-C-NPS
           └─ 2395 /usr/lib/jvm/java-11-openjdk-11.0.17.0.8-2.fc35.x86_64/bin/java -classpath ./target/service-alarm-server-4.6.10-S
Feb 11 14:50:03 fedora systemd[1]: Started Phoebus Alarm Server.
```

FIG.1. Phoebus Alarm Server Program Status

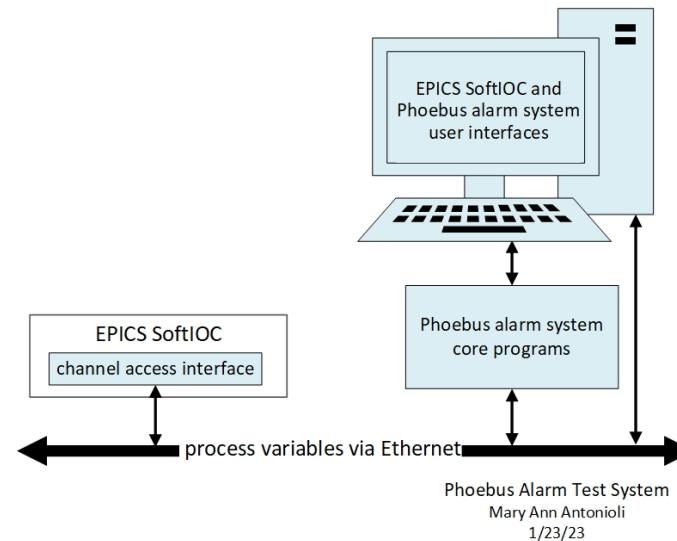


FIG.2. Phoebus Alarm Test System

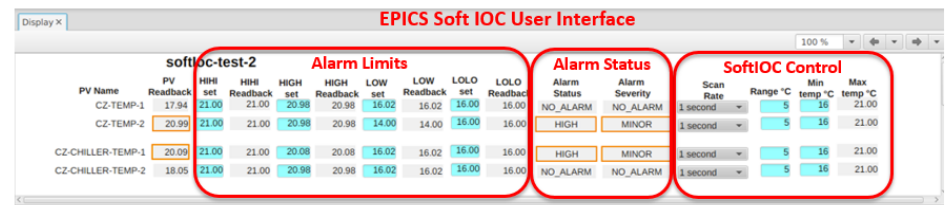


FIG.3. SoftIOc User Interface for the Phoebus Alarm Test System