# EPICS: CSS-Phoebus

**Peter Bonneau**
2023-04

I am developing an EPICS alarm system based on CS-Studio Phoebus. Phoebus will be used for new EPICS system development and will replace the existing Eclipse-based CS-Studio applications (DSG Note 2021-37).

After a system upgrade to the Linux computer used for Phoebus development (DSG Note 2023-13), the alarm annunciator failed. To isolate the fault, I checked each section of code, controls, and monitoring that contributes to the operation of the alarm annunciator.

The Phoebus alarm annunciator program is started via the alarm applications menu within the CS-Studio Phoebus alarm user interface (UI) (FIG. 1). Upon starting of the program, the UI window for the annunciator indicated that it had started, however it would not announce the new alarms that were occurring.

To troubleshoot this fault, I used the Phoebus alarm test system softIOC (DSG Note 2022-06) to generate EPICS process variables (PVs) to test the annunciator. To verify the PVs which were monitoring for alarm conditions were set correctly for the annunciator, I used the *alarm server command console* (DSG memo 2022-02) and the PV alarm configuration menu (FIG. 2) to read-back the alarm settings for the test PVs. The annunciator and the other PV alarm settings were set correctly. I further checked the log files and the *alarm_preferences.properties* file which controls the way the annunciator responds to multiple and frequent alarms. No cause for the annunciator fault was found in the any of the files.

All the Phoebus alarm system programs use (FIG. 3) Kafka Zookeeper and Kafka Server for inter-process communication. The work I've done on the programming and configuration the Kafka programs are detailed in my DSG software memos 2022-02 and 2022-03. The work I've done on the programming of the alarm server is in my DSG software memo 2022-04.

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system**
- **Developing a program to assist in debugging the alarm annunciator and the other Phoebus alarm applications**

# EPICS: CSS-Phoebus

The Phoebus alarm server monitors PVs for alarm conditions. Upon the detection of an alarm, the alarm server sends a message to the annunciator program via the unidirectional Kafka message stream called *Hall-C-NPSTalk* (FIG. 3). The alarm annunciator program translates the received Kafka message and instructs the computer to play an audible warning and writes it to the annunciator UI.

Having check all readily accessible sources for the fault, I realized I could not see the actual Kafka messages via any of the existing programs or user interfaces. The Kafka streaming messages are text-based however, Phoebus does not have a program that will directly display the text of a message. Since the annunciator messages are text, being able to read them directly could help in identifying the source of the error.

To help in the debugging of this issue, I wrote a program that could directly monitor the alarm system message streams independently of Phoebus using Kafka version 2.13-3.3.1 script commands.

When starting the Kafka message monitoring program via a Linux terminal window, the name of the message stream to be monitored is entered as a program switch. Any of the three Phoebus alarm system message streams can be monitored via the program. Since the program can read all the streams, it can be used for debugging any alarm system messaging errors.

As an option, the text being read and displayed on the Linux terminal window by the Kafka message monitoring program can also be written to a text file.

I discovered via my Kafka message monitoring program that the *Hall-C-NPSTalk* message stream was present, but no text was available to be read by the monitoring program. Upon further investigation, It was found that during the upgrade, the script that formed the message stream was incorrect and not compatible with the new Kafka version 2.13-3.3.1.

The script was corrected and the *Hall-C-NPSTalk* message stream was recreated and successfully retested via the Kafka message monitoring program. The annunciator's audible alarm and the UI monitor screen (FIG. 4) were were correctly reporting the PVs that were in alarm.

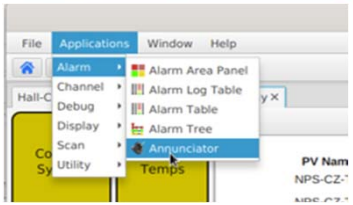


FIG.1. Phoebus Alarm Programs Menu

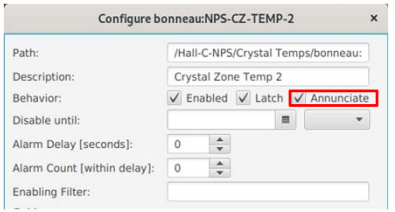


FIG.2. PV Alarm Configuration Menu

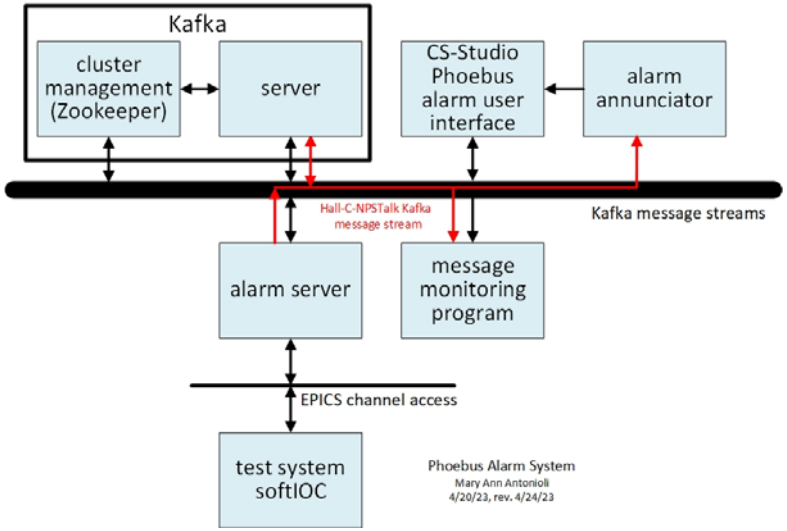


FIG.3. Phoebus Alarm Annunciator Programs

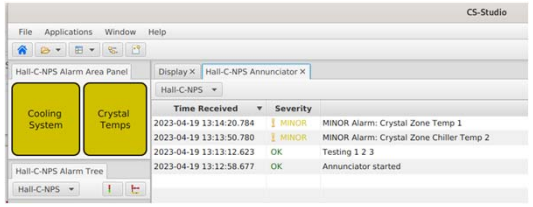


FIG.4. Phoebus Alarm Annunciator UI Monitor