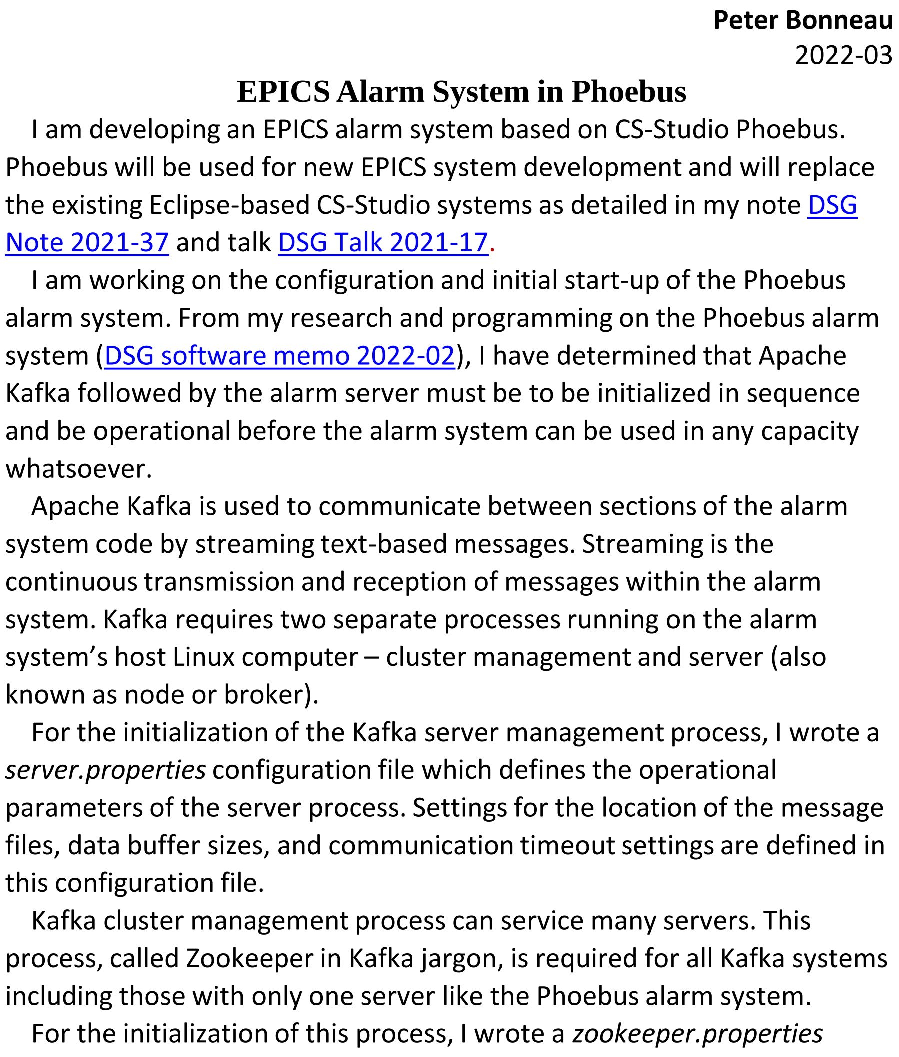
EPICS: CSS-Phoebus



Peter Bonneau
2022-03

EPICS Alarm System in Phoebus

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 systems as detailed in my note DSG Note 2021-37 and talk DSG Talk 2021-17.

I am working on the configuration and initial start-up of the Phoebus alarm system. From my research and programming on the Phoebus alarm system (DSG software memo 2022-02), I have determined that Apache Kafka followed by the alarm server must be to be initialized in sequence and be operational before the alarm system can be used in any capacity whatsoever.

Apache Kafka is used to communicate between sections of the alarm system code by streaming text-based messages. Streaming is the continuous transmission and reception of messages within the alarm system. Kafka requires two separate processes running on the alarm system's host Linux computer – cluster management and server (also known as node or broker).

For the initialization of the Kafka server management process, I wrote a *server.properties* configuration file which defines the operational parameters of the server process. Settings for the location of the message files, data buffer sizes, and communication timeout settings are defined in this configuration file.

Kafka cluster management process can service many servers. This process, called Zookeeper in Kafka jargon, is required for all Kafka systems including those with only one server like the Phoebus alarm system.

For the initialization of this process, I wrote a *zookeeper.properties*

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system - to be implemented on Hall C detectors**

- **Debugging startup of Kafka message streaming system**

- **Plan to implement alarm system message streaming interface between the Phoebus user interface and the alarm server**

EPICS: CSS-Phoebus

configuration file which defines the process operational parameters such as the location of the log file.

When starting the server using my *server.properties* file, an error occurred stating: *Session 0x0 for server localhost/:2181, Attempting reconnect (org.apache.zookeeper.ClientCnxn)*. I was expecting an error because I know from my research that both Kafka processes must be running.

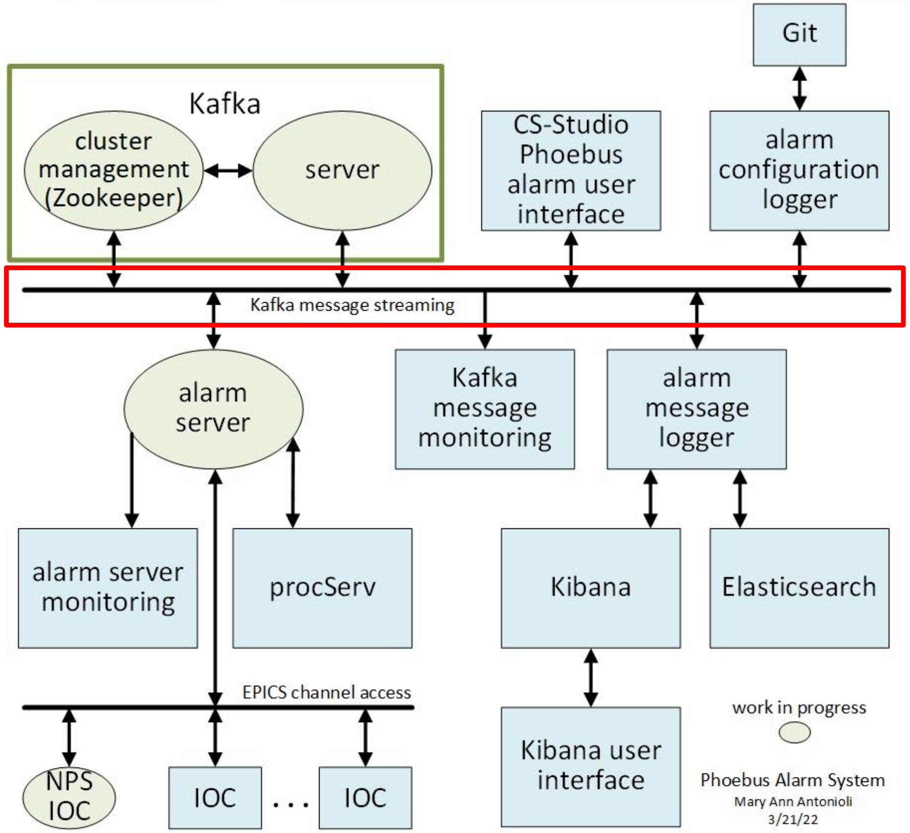
While the server was still running, I started the Zookeeper process using my *Zookeeper.properties* file. The process reported ~ 100 lines of startup information with no errors listed. Following the startup of Zookeeper, the server process listed ~ 1200 new lines of information, warnings, and errors before crashing. The last error stated: *Error while creating /brokers/ids/0, node already exists and owner does not match current session.*

I then restarted the Zookeeper process followed by the restart of the server process. I studied the 680 lines of startup information from these two processes and found no errors. The Phoebus alarm server process was started following Kafka processes.

The startup sequence requirement of first starting Zookeeper process followed by the server process was not clearly mentioned anywhere. The Kafka and alarm server initialization sequence (Zookeeper, server, and alarm server) was successful.

I've determined that the automatic startup and the sequencing of the Kafka followed by the alarm server process will be required on the Linux computer that hosts the Phoebus alarm system. To accomplish this task, I am writing scripts that will automatically execute upon booting of the host computer.

I plan to implement a message streaming interface between the Phoebus user interface and the alarm server as the next step in the development of the alarm system.



Phoebus alarm system programs