

EPICS: CSS-Phoebus

Peter Bonneau
2022-06

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](#).

The Phoebus alarm system are client applications that monitor predetermined process variables (PVs) from Input / Output Controllers (IOCs) for alarm conditions. To test the recently integrated software components detailed in my DSG software memos [2022-01](#), [2022-02](#), [2022-03](#), and [2022-04](#), an IOC is required. The alarm software components to be tested include the Apache Kafka infrastructure and the Phoebus alarm server (FIG. 2).

To start the development of an EPICS IOC, the EPICS base must be built on the alarm system computer. The EPICS base is the set of core software, i.e. the components of EPICS without which EPICS will not function. I downloaded the zipped tar source file distribution for EPICS base version 3.14 which is currently in use in the Jlab Physics Division.

According to the EPICS documentation, the dependencies required to build the EPICS 3.14 base include *Perl* version 5.8.1 or later, *GNU-make* version 3.81 or later, and *GNU-readline*.

I verified and installed when needed the required the EPICS build dependencies. The Linux operating system on the alarm system development computer already had the latest versions of *Perl* and *GNU-make* installed and these could be found through the default *PATH* environmental variable in Linux. However, *GNU-readline* was not installed.

I downloaded, installed, and tested the *GNU-readline* library. This library is used by the EPICS IOC shell to provide command line editing and command line history recall.

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system - to be implemented on Hall C detectors**
- **Building and debugging EPICS base**
- **Plan to develop an IOC to test the Apache Kafka infrastructure and the alarm server (block diagram on slide #2)**

EPICS: CSS-Phoebus

Also required for the build is the configuration of the Linux environment variables that are used by the EPICS base. I determined the EPICS `EPICS_HOST_ARCH` variable by running the `EpicsHostArch.pl` Perl script. I also determined the values for the EPICS Channel Access (CA) environmental variables and entered them into the Linux `.bashrc` shell configuration file. I started the EPICS build process and observed the progress in the Linux terminal window. After ~ 10 minutes, the build failed with the nondescript errors “Error 1, Error 2” (FIG. 1).

```

./././././configure/RULES_BUILD:220: caServer.o] Error 1
directory '/home/bonneau/Downloads/baseR3.14.12.8/base-3.14.12.8/src/cas/build/
././././configure/RULES_ARCHS:61: install.linux-x86_64] Error 2
directory '/home/bonneau/Downloads/baseR3.14.12.8/base-3.14.12.8/src/cas/build'
./././configure/RULES_DIRS:83: build.install] Error 2
    
```

FIG.1. EPICS base build errors

The errors occurred during build for `caServer`. Written in C++, the EPICS CA server is a critical part of the EPICS base and I have successfully built systems using base version 3.14 and have not encountered this error. However, the systems were built on older versions of the operating system and compilers.

I reviewed the EPICS source code for the CA server, investigated the root cause of the build error, and researched if other EPICS system managers had also encountered this issue during the build of EPICS base 3.14. My research revealed that other EPICS system managers had also encountered an build error on base 3.14 on computer systems which had up to date operating systems and program compilers. The error was caused by rule changes in the compiler and the EPICS 3.14 source code was not in compliance with the new compiler rule changes. To fix this error, the EPICS source code is altered to comply with the rule changes.

I searched through the CA server EPICS source code and found and corrected the location of the compiler rule violation in the `ioBlocked.h` file. The build successfully completed after the code correction and I verified the build was error free by reviewing the 1918 lines of the build log file.

I plan to develop an EPICS IOC as the next step in the development of the Phoebus alarm system.

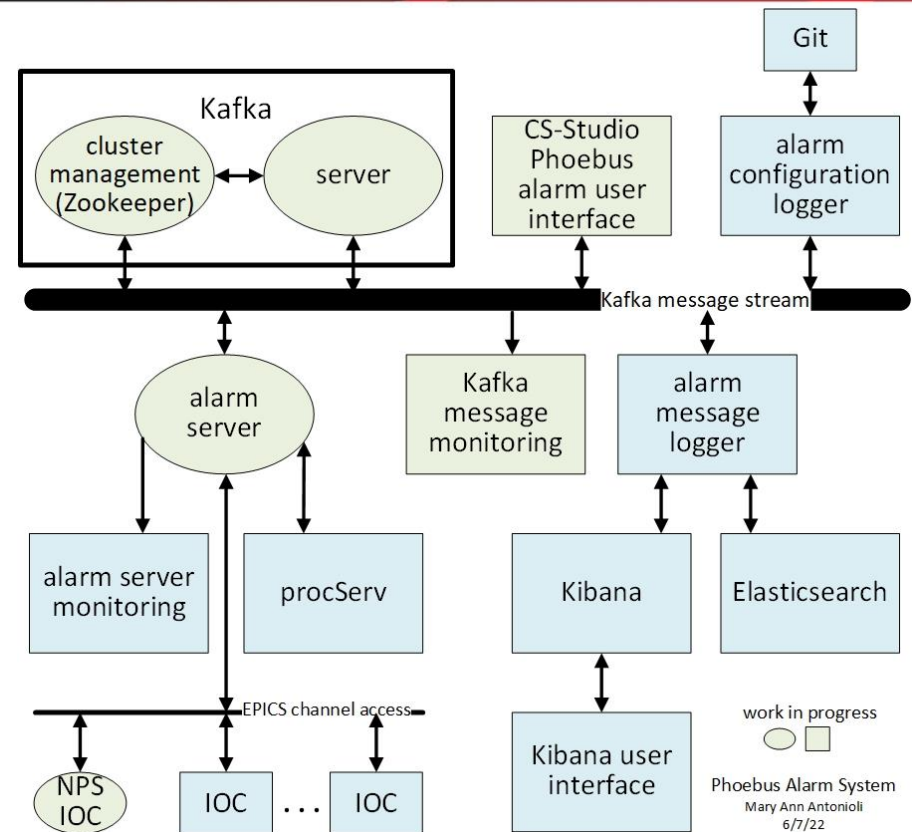


FIG.2. Phoebus Alarm System Programs

work in progress
 ○ □
 Phoebus Alarm System
 Mary Ann Antonioli
 6/7/22