

Automatic Startup and Initialization of the Phoebus Alarm System Software Packages

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

March 28, 2024

This note describes the software developed to automate startup initialization of the Phoebus alarm system software packages for the electron collider’s (EIC’s) detection of internally reflected Cerenkov light (DIRC) detector.

Software packages [1] developed for the alarm applications [2] of the EIC DIRC laser interlock system [3] were initialized manually in a unique sequence during hardware and software integration [4, 5, 6]. Each application program was launched by its own Linux terminal window to aid development, testing, and debugging of the alarm system—displaying the application programs’ statuses and providing command line access.

Table I shows that after initializing the Linux kernel, the Linux system service manager systemd is started.

Program name	Function summary
Linux kernel	Initialization of Linux kernel
Linux system server manager (systemd)	Linux service manager
Linux network service	Starts Linux network service
Kafka system management (Zookeeper.service)	Kafka management for alarm system inter-program communication
Kafka server (DIRC_Kafka_server.service)	Hosts the three DIRC alarm system message communication streams
procServ (DIRC_softIOC.service)	Launches DIRC EPICS softIOC
DIRC EPICS SoftIOC	Hosts DIRC process variables
procServ (DIRC_alarm_server.service)	Launches DIRC Phoebus alarm server
DIRC Phoebus Alarm Server	Monitors DIRC EPICS PVs for alarm conditions

TABLE 1. Automatic startup sequencing.

systemd has a file with the extension `.service`, with specific program configurations and startup commands that manage program initialization, automate software package startup for booting the alarm system’s Linux computers, user processes, network connections, device management, and user logins.

Next, the Linux network service used by all the alarm system programs is started.

Once the above three steps are completed, the Kafka system management, Zookeeper, is the first alarm program started by the Linux system service manager for management of the alarm system’s inter-program communication—a generic step for automatic startups of all Phoebus alarm systems [7].

The Kafka server `DIRC_Kafka_server.service` hosts the three DIRC alarm system message streams required for communication between DIRC alarm programs. Upon initialization, the alarm state, command, and configuration streams are active and available to service alarm system programs.

`procServ DIRC_softIOC.service`—an interactive command wrapper that enables remote access to the target program’s command console via Telnet while the program is running and keeps a log of the status messages transmitted by the program—launches the target program DIRC EPICS softIOC, an EPICS server that hosts the DIRC process variables (PVs), stores the alarm limits for the PVs, and raises an alarm if the PV value exceeds the limits [8].

Lastly, the `procServ DIRC_alarm_server.service` file starts the DIRC Phoebus alarm server. The Phoebus alarm server monitors DIRC PVs for alarm conditions via EPICS channel access. The server stores alarm configuration settings for each PV. If a monitored PV is in an alarm state, the PV value is latched and the alarm time is recorded. The alarm is displayed on the user interface and an audible alarm is given by the annunciator [8].

To conclude, a service manager sequence has been developed to automate the startup and initialization of the software packages at the boot of the EIC DIRC alarm system’s Linux computer. The software package developed initializes the alarm packages in the correct sequence.

- [1] P. Bonneau, et al. *Development and Implementation of the Phoebus Alarm System Software Packages for the Laser Interlock System*, DSG Note 2023-49, 2023.
- [2] P. Bonneau, et al. *Debugging the Phoebus Alarm System Annunciator*, DSG Note 2023-18, 2023.
- [3] P. Bonneau, et al. *Development of the CS-Studio Phoebus Alarm System for the EIC DIRC Laser Interlocks*, DSG Note 2023-43, 2023.
- [4] P. Bonneau, et al. *Signal Simulator for Developing and Testing Phoebus Alarm System Software Packages for the Interlocks of the EIC DIRC Quartz Bar Laser Testing Lab*, DSG Note 2023-56, 2023.
- [5] P. Bonneau, et al. *EPICS Alarm System in Phoebus*, DSG Note 2024-03, 2024.
- [6] P. Bonneau, et al. *Distributed Phoebus Alarm System for the Electron-Ion Collider’s Detection of Internally Reflected Cerenkov-Light Detector*, DSG Note 2024-07, 2024.
- [7] P. Bonneau, et al. *Automated Startup and Sequencing of*

the CS-Studio Phoebus Alarm System Core Programs,
DSG Note 2022-16, 2022.

- [8] P. Bonneau, et al. *Testing of the CS-Studio Phoebus Applications and Alarm System Core Programs,* DSG Note 2023-06, 2023.