TUSDSC08

Phoebus Tools and Services

K. Shroff, Brookhaven National Laboratory, Upton (NY), USA
R. Lange, ITER Organization, St. Paul lez Durance, France
T. Ford, Lawrence Berkeley National Laboratory, Berkeley (CA), USA
G. Weiss, European Spallation Source ERIC, Lund, Sweden
K. Kasemir, Oak Ridge National Laboratory, Oak Ridge (TN), USA
T. Ashwarya, FRIB, East Lansing (MI), USA

Introduction

Phoebus/Control System Studio (CSS) was developed to streamline workflows for operators, engineers and scientists by offering a suite of integrated applications and a framework that simplifies the development of such applications. For end-users, this translates to a collection of applications that are easy to navigate between, where data can be seamlessly transferred from one application to another, ensuring a consistent user experience.
For developers, CSS provides a framework for creating applications that seamlessly integrate into this ecosystem without requiring tight dependencies on other applications. It also supports modules that offer access to shared and optimized resources, such as connections to EPICS Process Variables (PVs), REST clients, and more.

Phoebus Applications

Phoebus offers a suite of applications designed for interacting with control systems and middle layer services. These applications enhance the functionality and usability of control systems while providing tools for efficient management and monitoring.
Applications:

Display Builder:
Combines an interactive editor and runtime for creating control system displays, offering various display widgets for both monitoring and controlling process variables. It supports easy conversion from other EPICS display tools and uses a minimalistic XML-based file format.

Data Browser:
A trending tool for plotting PV values over time, supporting various archive data providers, including the EPICS Archiver Appliance and RDB-based history stores.

PV Utilities:
Provides tools for introspection of control system PVs, including "Probe" for displaying comprehensive PV information, "PV Table" for displaying value, timestamp, and alarm data for PVs, and "PV Tree" for visualizing input link hierarchies in EPICS records.

Alarm User Interfaces:
Offers a comprehensive set of alarm management tools, including an "Area Panel" for an overview of alarm states, an "Alarm Table" for listing current alarms, an "Annunciator" for voice representation of new alarms, and an "Alarm Tree" for hierarchical alarm configuration.

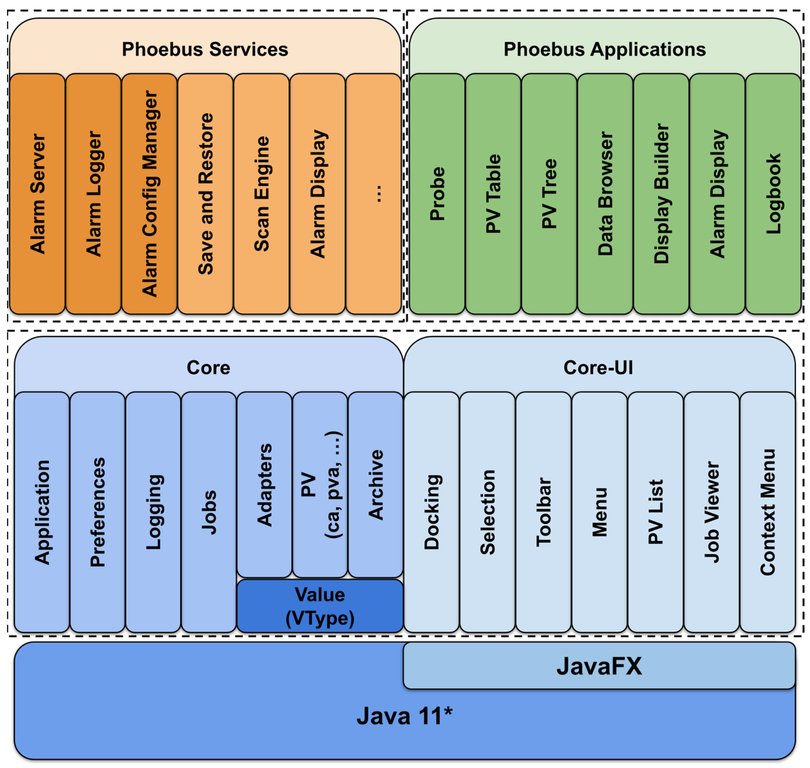
Channel Finder Clients:
Simplifies the search for valid PV names through a case-insensitive substring search, providing additional metadata like IOC name, hostname, address, port, PV status, record type, and IOC developer identity.

Save and Restore User Interface:
A general-purpose tool for saving and restoring user-defined sets of PV values to a remote service. Users can modify PV values before restoring, and multiple snapshots can be combined for broader configurations.

Logbook:
Interfaces with online logbooks to record operational issues and progress. Users can submit screenshots and comments directly from display or alarm panels, enriching logbook entries with metadata relevant to specific logbook implementations.

Phoebus Architecture

The Phoebus architecture provides a comprehensive set of tools and core libraries that empower developers to create applications efficiently. These modules enhance the development process by ensuring consistent UI elements, efficient data processing, responsive multithreading, and flexible data mapping between applications while minimizing dependencies. Phoebus' modular structure also facilitates seamless integration into existing software stacks.



Easily Extendable Functionality through Java Service Provider Interface (SPI)

Phoebus employs Java SPI to modularize its framework. Phoebus SPIs allows applications to register for file extensions, contribute menu or toolbar entries, offer data sources or access to a site-specific services like logbooks, archivers. Adding new applications is straightforward: developers provide an implementation of the app SPI and include it in the classpath, fostering contributions and tailored products.

Phoebus Core Libraries and Services

Phoebus simplifies application development with core modules, providing consistent UI elements, efficient data processing, multithreading, and job scheduling. Core libraries like Core-PV and EPICS VType streamline communication and data type mapping over various protocols.

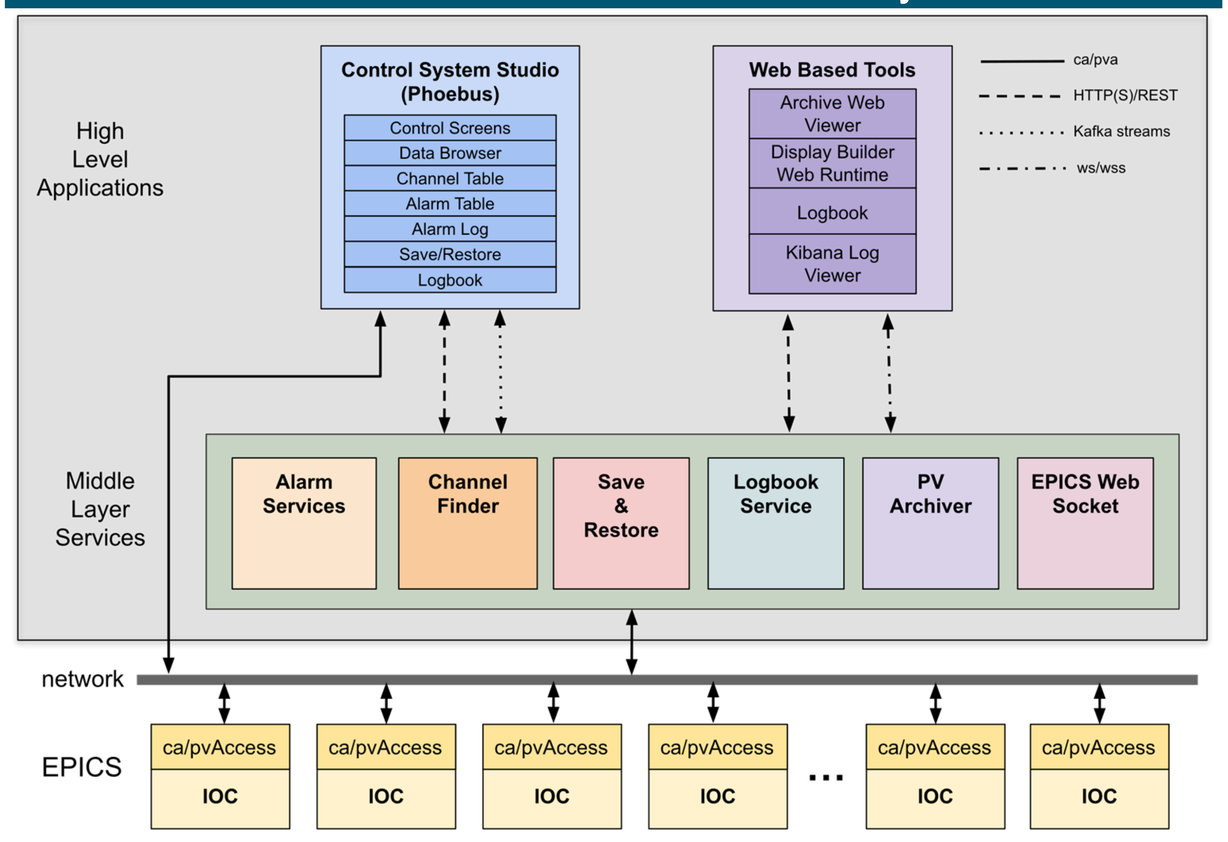
Formula Functions and Data Processing

Phoebus includes formula functions for processing data efficiently, enhancing UI responsiveness. It also offers services for scheduling and managing jobs.

Selection Provider and Adapter Framework

Phoebus features a Selection Provider and Adapter Framework for flexible data type mapping between applications. Developers can define mappings without introducing rigid dependencies, facilitating seamless data exchange.

Phoebus Tools and Services Software Ecosystem



Phoebus Middle Layer Services

Phoebus Middle Layer Services adhere to the principles of the microservices architecture, emphasizing modularity with each service having a well-defined scope and service APIs. This approach facilitates the creation of lightweight, specialized services, eliminating the need for complex, resource-intensive ones. Within the Phoebus framework, clients can seamlessly utilize this integrated ecosystem, offering users a smoother and more streamlined experience.

Alarm Server:
Provides core alarm system functionality, monitoring a configurable list of PVs (Process Variables) and notifying operators via the alarm user interface of new, pending, and acknowledged alarms.

Alarm Logger:
Functions as a historical timeline of alarms and operator actions, such as acknowledgements, while also serving as a foundation for generating valuable statistics

Alarm Configuration Manager:
Keeps a history of changes to the alarm configuration

Phoebus Save and Restore:
Persists snapshots and other save-and-restore data objects, offering a REST API over HTTP(S) and leveraging Elasticsearch for powerful and efficient search capabilities. Users can locate snapshots based on metadata such as snapshot name, date, user identity, and user-defined tags.

Phoebus Olog:
An online logbook built on Elasticsearch for log entry persistence and MongoDB for attachments.

Channel Finder:
A directory service for EPICS PV's. designed to address the challenge of organizing and accessing channel names within the flat namespace of the EPICS Channel Access protocol.

Compatibility and Adaptability

The Phoebus tools and services embrace both backward and forward compatibility in response to evolving software technology and computer graphics.
Ensures smooth transitions by allowing the import of older EPICS display tool file formats, preserving legacy content.
Introduces minimalistic and generic file formats, designed to facilitate the adoption of future tools while maintaining compatibility.
Middle layer services are built on technology-agnostic APIs, ensuring longevity of client support even during significant backend changes.

Learn more

https://github.com/ControlSystemStudio/phoebus