

EPICS Collaboration Meeting in October 2019

Saturday 5 October 2019 - Saturday 5 October 2019

Book of Abstracts

Contents

Display Builder Web Runtime	1
retools: create PV aliases and info tags based on regular expressions	1
Summary of the first EPICS Documentathon	1
RTEMS with EPICS on VME2500 (QorIQ P2020)	2
ADGenICam: areaDetector driver for GenICam cameras	2
areaDetector Data Compression II: HDF5	3
areaDetector Data Compression I: NDPluginCodec, NDPluginPva, ImageJ	3
Just-in-Time EDM converter for Phoebus	4
A User’s Guide for EPICS Lua-Based Data Processing Subscription Update Filters	5
CI for Code and Documentation	5
Transitioning from CS Studio/Eclipse to CS Studio/Phoebus	5
Save-and-restore port to CS Studio/Phoebus	6
High Level Application Architecture Design for APS Upgrade	6
FRIBs diagnostics software development workflow and tools	6
Distributed I/O for Dynamic Equipment	7
Building and Packaging EPICS modules with Conda	7
EPICS driver for health and status monitoring on MTCA.4 Crates	8
Epics 7	8
CS-Studio developers meeting report	9
EPICS Support Module for Efficient UDP Communication With FPGAs	9
Moving EPICS Base to Modern C++	9
EPICS Council Report	10
The Next EPICS Meeting	10

Epics Developers meeting Report	10
Systems Delivery And What Lurks Below: Legacy, Support, Users	10
Celebrating 30 Years of EPICS	11

User Interface tools and techniques / 2**Display Builder Web Runtime****Author:** Kay Kasemir¹¹ ORNL/SNS

The CS-Studio display builder is available in both the older Eclipse-based CS-Studio and the latest “Phoebus” version of CS-Studio that no longer relies on Eclipse.

Recent additions include an MEDM converter that directly translates MEDM *.adl files into the.bob* file format.

The biggest recent addition is a Display Builder Web Runtime which allows viewing most of the display builder screens, including plots and images, on the web.

Track:

User Interface tools and techniques

Lightning Session / 3**retools: create PV aliases and info tags based on regular expressions****Author:** Bruno Martins¹¹ Facility for Rare Isotope Beams

retools is an EPICS module that allows PV aliases and info tags to be created based on specified regular expressions. This talk will briefly go over the use cases that motivated the development of retools.

Track:

Other

EPICS Developements / 4**Summary of the first EPICS Documentathon****Author:** Timo Korhonen¹¹ European Spallation Source ERIC

The first ever EPICS Documentathon was held at ESS in September of this year. A group of 12 people worked on a number of topics to develop, update and improve the EPICS documentation. A new infrastructure for writing and publishing EPICS documentation was put in place. Also the epics-controls website navigation was restructured, to make it easier to use and to integrate with other documentation.

The talk will introduce the documentation toolkit and the workflow, and also summarize the results of the Documentathon.

Track:

EPICS development

Lightning Session / 5**RTEMS with EPICS on VME2500 (QorIQ P2020)****Author:** Heinz Junkes¹¹ *Fritz-Haber-Institut*

There are new CPU cards for the VMEbus available with modern processors. These can replace the proven CPU cards like MVME6100/5500/3100/2100.

I will present the latest developments of EPICS 7 on QorIQ CPU's with SMP capabilities under RTEMS5.

Track:

Status reports

AreaDetector / 6**ADGenICam: areaDetector driver for GenICam cameras****Author:** Mark Rivers¹¹ *University of Chicago*

GenICam is a standard from the European Machine Vision Association that is widely adopted for cameras and other detectors. GenICam cameras contain an XML file that can be downloaded, and which contains a complete description of all of the features that the camera supports and how to access them. GenICam also defines the transport layer API, and supports the GigE Vision, USB3 Vision, CameraLink, and CameraLinkHS standards.

ADGenICam is a new areaDetector base class for GenICam cameras. It provides the following features:

- A Python tool to read the XML file from the camera and create an EPICS database file. It creates records for each of the features that the camera supports.
- A Python tool to read the XML file from the camera and create multiple medm adl files containing controls and readbacks for the records controlling the camera features. These adl files can be automatically converted to files for edm, CSS, and caQtDM. The ADGenICam creates the areaDetector parameter library dynamically at iocInit from the drvUser fields passed by each record. It also handles most of the work in reading and writing feature values to the camera.
- New ADAravis driver. This driver uses the aravis library. It works with any GigE, 10 GigE, or USB3 GenICam camera. It runs on most versions of Linux. This driver is designed to replace the aravisGigE driver. It is significantly smaller because much of the code is in ADGenICam.
- New ADSpinnaker driver. This driver uses the FLIR/Point Grey Spinnaker SDK. It works with their GigE, 10 GigE, or USB3 GenICam cameras. It runs on Windows and new versions of Linux, e.g. Ubuntu 18.

- New ADVimba driver. This driver uses the AVT/Prosilica Vimba SDK. It works with their GigE or USB3 GenICam cameras. It runs on Windows, and most versions of Linux, e.g. RHEL7.

Track:

EPICS for data acquisition

AreaDetector / 7**areaDetector Data Compression II: HDF5**

Authors: Mark Rivers¹; Gary Yendell²; Alan Greer³; Xiaoqiang Wang⁴

¹ *University of Chicago*

² *Diamond Light Source*

³ *Observatory Sciences*

⁴ *Paul Scherrer Institut*

This talk will describe new support for data compression in the areaDetector NDPluginHDF5 plugin that writes HDF5 files. These include:

- Support for the JPEG, Blosc, LZ4, and Bitshuffle/LZ4 plugins within NDPluginHDF5. This support receives uncompressed NDArrays and writes them using compression within the HDF5 data pipeline. This has limited performance because it is not multi-threaded, and incurs significant overhead from the pipeline.
- Support for HDF5 Direct Chunk Write. This support receives NDArrays and writes them directly, bypassing much of the HDF5 data pipeline. This works both for writing uncompressed NDArrays, and for writing pre-compressed NDArrays. The latter can come from NDPluginCodec, where they can be compressed in multiple threads, or from the ADEiger driver, which directly receives compressed data from the Eiger detector.
- ADSupport now builds JPEG, Blosc, LZ4, and Bitshuffle/LZ4 codecs as shareable libraries that can be dynamically loaded by the HDF5 library on Linux, Windows, and Mac. This allows existing HDF5 applications (Matlab, IDL, Python, etc.) that are built without support for these codecs to read compressed HDF5 files written by areaDetector. The only requirement is that the application be built with HDF5 1.8.11 or later.
- Performance measurements will be presented. For example, simDetector generating 1350 frames/s of 1024x1024 32-bit images (5.4 GBytes/s) can be compressed by NDPluginCodec using Blosc LZ4+Bytesshuffle and streamed to HDF5 files using the direct chunk write with no dropped frames. This is more than 3.5 times the throughput that can be obtained doing the Blosc compression in the HDF5 plugin itself.

Track:

EPICS for data acquisition

AreaDetector / 8**areaDetector Data Compression I: NDPluginCodec, NDPluginPva, ImageJ**

Authors: Mark Rivers¹; Bruno Martins²; Marty Kraimer³

¹ *University of Chicago*

² *Facility for Rare Isotope Beams*

³ *Advanced Photon Source*

Data compression is highly desirable to increase the speed and decrease the network bandwidth and disk storage required for modern high-performance detectors.

This talk will describe new support for data compression in areaDetector. These include:

- New NDPluginCodec plugin. This plugin can compress and decompress NDArrays using any of 4 codecs: JPEG, Blosc, LZ4, and Bitshuffle/LZ4. JPEG is lossy while the other codecs are all lossless. Blosc itself supports 6 different codecs, plus bit shuffle and byte shuffle. NDPluginCodec can pass compressed NDArrays to downstream plugins, and can decompress NDArrays from upstream drivers and plugins. NDPluginCodec can run multiple threads to process multiple NDArrays in parallel. The Blosc codec can run multiple internal threads to process a single NDArray faster.
- NDPluginPva can now convert compressed NDArrays to compressed NTNDArrays that are served with pvAccess.
- The ImageJ pvAccess viewer (NTNDArrayViewer) can now receive compressed NTNDArrays, decompress them, and display them. This allows greatly reducing network traffic between the areaDetector IOC and the machine running ImageJ. The decompression in ImageJ is done using the same C/C++ libraries that are used by NDPluginCodec with a thin Java Native Access (JNA) wrapper.
- ADSupport now builds all of the codecs (JPEG, Blosc, LZ4, Bitshuffle/LZ4) on most EPICS architectures. It also builds shareable libraries that can be called dynamically from Java and HDF5.
- ADEiger, the areaDetector driver for the Eiger, can now receive both LZ4 and Bitshuffle/LZ4 compressed data over the ZeroMQ socket. This compressed data can now be sent directly to the HDF5 plugin and to ImageJ without ever decompressing it. It can also be sent to NDPluginCodec to decompress it for other plugins.

Track:

EPICS for data acquisition

User Interface tools and techniques / 9

Just-in-Time EDM converter for Phoebus

Author: Kay Kasemir¹

¹ *ORNL/SNS*

CS-Studio includes an EDM-to-BOY converter. It's being ported to the latest version of CS-Studio, Phoebus, where it turns into an EDM-to-DisplayBuilder converter. It can be used from the UI or in batch mode to convert a bunch of displays. In addition, it offers a new just-in-time mode where it can download EDM displays and convert them when the user first tries to access them.

Track:

User Interface tools and techniques

EPICS Developements / 10**A User's Guide for EPICS Lua-Based Data Processing Subscription Update Filters**

Author: Jeffrey Hill¹

¹ LANL

Multiple previous talks described our embedding of the Lua interpreter in EPICS base for a replacement shell, for a Lua scripting record, and for dynamically filtering and re-configuring subscription update payloads within the Channel Access protocol.

This update talk will provide a end user's tutorial for these production system ready enhancements.

Track:

EPICS development

EPICS Developements / 11**CI for Code and Documentation**

Author: Ralph Lange¹

¹ ITER Organization

At the recent Documentathon, a system to create and host EPICS documentation has been added to the services that EPICS uses for Continuous Integration (CI).

The talk presents the status of the EPICS CI systems for code and documentation, and how these systems can be used for other EPICS modules, e.g. Record, Device and Driver Support modules.

Track:

EPICS services

User Interface tools and techniques / 13**Transitioning from CS Studio/Eclipse to CS Studio/Phoebus**

Authors: Fredrik Söderberg¹; Georg Weiss¹

¹ European Spallation Source, ERIC

The European Spallation Source (ESS) is under construction and commissioning. OPIs created this far are mainly created in Display Builder in CS Studio/Eclipse, and users have become familiar with CS Studio and the tools provided. We intend to deploy CS Studio/Phoebus to the control room during Q4 2019, where it will replace CS Studio/Eclipse. The transition is prepared by investigating how existing OPIs work and how they currently fail in CS Studio/Phoebus. We also try to identify more general features of CS Studio/Eclipse that will not be available in CS Studio/Phoebus, or that require

a different work flow. This talk summarizes rationale for the transition, and findings and conclusions that may be of interest for facilities considering CS Studio/Phoebus.

Track:

User Interface tools and techniques

User Interface tools and techniques / 14**Save-and-restore port to CS Studio/Phoebus**

Author: Georg Weiss¹

¹ *European Spallation Source*

The European Spallation Source (ESS) has contributed to CS Studio/Phoebus with a adaptation of the save-and-restore tool in CS Studio/Eclipse. This adaptation includes both a new service for persistence of save-and-restore data, as well as a UI facelift. This talk briefly summarizes the save-and-restore tool in CS Studio/Phoebus, and discusses limitations and future enhancements.

Track:

User Interface tools and techniques

15

High Level Application Architecture Design for APS Upgrade

Author: Guobao Shen¹

¹ *APS*

A modular software platform is under active design and development for high-level applications to meet the requirements of the Advanced Photon Source Upgrade (APS-U) project. The design is based on modern software architecture, which has been used in many other accelerator facilities and has been demonstrated to be effective and stable. At APS-U, we are extending the architecture in order to efficiently commission, operate, and maintain the APS-U. Its open architecture provides good flexibility and scalability. This paper presents the current status of high-level application architecture design, implementation, and progress at APS-U.

Track:

EPICS services

Lightning Session / 16**FRIBs diagnostics software development workflow and tools**

Author: Diego Omitto¹

¹ *Michigan State University*

This presentation will give an overview of the FRIBs diagnostics devices including the software development workflow along with some of the tools created to help the support and maintenance activities.

Track:

EPICS for data acquisition

17

Distributed I/O for Dynamic Equipment

Author: Nathanael Maytan¹

Co-author: Anton Derbenev¹

¹ *Brookhaven National Laboratory*

There are currently two major controller-based systems in use at each beamline at NSLS-II. These are the Equipment Protection System and the Personnel Protection System. As the names imply, these systems work to protect equipment and personnel from hazards which could impact the safety, efficiency, usability, and integrity of the facility.

A third controller-based solution (DIODE) was proposed to further enhance the reliability and maintainability of the beamline subsystems. This additional approach will better decouple the class of equipment which does not serve an equipment or personnel protection purpose, but which nonetheless requires full integration into the beamline control system. We can call this class of devices *Dynamic Equipment*.

Historically, when Dynamic Equipment was integrated into a beamline, it would be lumped onto the EPS. There are two main reasons why the EPS was burdened this way: the EPS is much more mutable than the PPS, which is subject to stricter oversight and regulation, and because the EPS is typically more physically-distributed throughout the beamline.

While using the EPS has been a simple solution to the growing need for control and automation of Dynamic Equipment, it is not without consequences. The inclusion of non-protective and non-protected equipment has added unnecessary complexity within the EPS. This has introduced an undesirable volatility which makes it difficult to keep the EPS fully characterized and documented. Such consequences come with real costs and impact, including an increased difficulty when troubleshooting, extra time or resources needed when integrating new equipment, and the overloading of I/O boxes in terms of amperage and heat dissipation.

With DIODE, these problems can be subverted and even corrected. DIODE reduces the risk that comes from integrating Dynamic Equipment without introducing changes to the EPS. As a result, new devices can be added or modified at the beamline more quickly with less engineering and development overhead.

Track:

Other

Building and Packaging EPICS modules with Conda

Authors: Benjamin Bertrand¹; Joao Paulo Martins¹

¹ ESS

Conda is an open source package, dependency and environment management system. It runs on Windows, macOS and Linux and can package and distribute software for any language (Python, R, Ruby, C/C++...). It's already used in the EPICS community for pvaPy for example. At ESS, we are investigating using Conda to build and package EPICS modules. The modules built are based on PSI concept of dynamically loading EPICS module resources using require. Conda makes it easy to manage dependencies and build different variants of a package. Using anaconda's new compilers, it is possible to build portable (cross-distribution) Linux binaries. This allows developers to easily install the EPICS modules they want.

Track:

EPICS Building, packaging and distributing

Lightning Session / 19

EPICS driver for health and status monitoring on MTCA.4 Crates

Authors: Ursa Rojec¹; Miroslav Pavleski¹

¹ Cosylab

The driver will use Linux i2c driver to communicate with NAT MCH and provide status readings for a MTCA.4 system. These will allow the user to monitor general health of the MTCA.4 chassis such as AMC temperatures, fan speed, etc. Statuses will be exposed as PVs which will allow the user to monitor trends and use them for predictive maintenance (e.g. system temperature is slowly rising) or set alarms that will notify on critical events (e.g. fan speed increase).

Alarms and values can be connected to standard EPICS alarm and archiving tools. They can be displayed on the main operating screens to alert operators, without the need to use the terminal or web-based interfaces that need to be opened in separate windows.

The driver will for now only aim to report status and will not provide configuration options and will be developed and tested with NAT MCH, which is currently the more common used MCH in Scientific installations.

Track:

EPICS for data acquisition

EPICS Developements / 21

Epics 7

Author: Andrew Johnson ¹

¹ Argonne National Laboratory

An introduction to EPICS 7 and a report on the latest developments in the EPICS 7 core packages.

Track:

EPICS development

User Interface tools and techniques / 22

CS-Studio developers meeting report

Author: Kunal Shroff¹

¹ *Brookhaven National Lab*

A report of the discussions and developments from the CS-Studio core developers meeting.

Track:

Lightning Session / 24

EPICS Support Module for Efficient UDP Communication With FPGAs

Authors: Martin Konrad¹; Enrique Bernal^{None}; Mark Davis^{None}

¹ *Facility for Rare Isotope Beams*

The driver linac of the Facility of Rare Isotope Beams (FRIB) contains 332 cavities which are controlled by individual FPGA-based low-level RF controllers. Due to limited hardware resources the EPICS IOCs cannot be embedded in the low-level RF controllers but are running on virtual machines communicating with the devices over Ethernet. An EPICS support module communicating with the devices over UDP has been developed based on the Asyn library. It supports efficient read and write access for both scalar and array data as well as support for triggering actions on the device. Device-related parameters like register addresses and data types are configurable in the EPICS record database making the support module independent of the hardware and the application. This also allows engineers to keep up with evolving firmware without recompiling the support library. The implementation of the support module leverages modern C++ features and relies on timers for periodic communication, timeouts, and detection of communication problems. The latter allows the communication code to be tested separately from the timers keeping the run time of the unit tests short.

Track:

EPICS Developements / 25

Moving EPICS Base to Modern C++

Author: Martin Konrad¹

¹ *Facility for Rare Isotope Beams*

During the last decade C++ has evolved rapidly providing new language features, better standard-library facilities and more sophisticated compilers. This presentation provides an overview of these new features and motivates why we might want to leverage them for EPICS.

Track:

26

EPICS Council Report

Author: Henrik Carling¹

Co-author: Karen White²

¹ *ESS*

² *ORNL*

A report from the epics council

Track:

27

The Next EPICS Meeting

Details on the next EPICS collaboration meeting

EPICS Developments / 28

Epics Developers meeting Report

Author: Ralph Lange¹

¹ *ITER Organization*

A report on the discussions and developments from the EPICS core developers meeting.

Lightning Session / 29

Systems Delivery And What Lurks Below: Legacy, Support, Users

Author: Anton Derbenev¹

¹ *Brookhaven National Laboratory*

The NSLS-II Control System has workstations and servers standardized to the usage of Debian OS. With exceptions like RTEMS and Windows systems where software is built and delivered by hand, all hosts have EPICS development software installed from an internally-hosted and externally-mirrored Debian package repository. The sysv-rc-softioc toolkit is used to manage deployments locally. Configured by Puppet, machines have a similar environment with EPICS base, modules, libraries, and binaries. The repository is populated from epicsdeb, a collaboration-maintained project collection on GitHub.

Currently, packages are available for Debian 8 and 9 with legacy support being provided for Debian 6 and 7. Since packaging creates overhead on how quickly software updates can be available, keeping production systems on track with development is a challenging task. Software is often customized and built manually to get recent features, e.g. for AreaDetector. Another challenge is services like GPFS which underperform or do not work on Debian, or older hardware, or both.

Different facilities utilize different approaches to enable longer-term system flexibility and accommodation of continuous update needs. When considering larger time frames, it becomes very important that standards, practices, and approaches remain consistent with existing system solutions, and vice versa. As the system grows and ages, tending to new requirements and introducing new solutions, e.g. for deployment, is even more challenging, especially when it involves bringing new technologies and tools. The talk is dedicated to major challenges identified in that process: Legacy, for ensuring consistency of approaches for existing and new cases; Support, for tools and infrastructure which enable the new solution; and Users, for developers' knowledge and compliance.

Track:

EPICS Building, packaging and distributing

30

Celebrating 30 Years of EPICS

Authors: Henrik Carling¹; Richard Farnsworth²

¹ ESS

² NSLS-II

Keeping things in control for 30 years.