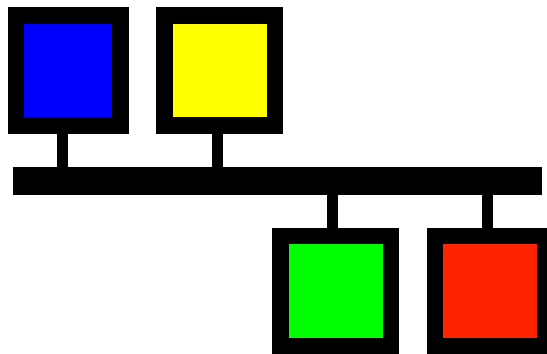# EPICS Collaboration Meeting fall 2011

Monday, 3 October 2011 - Friday, 7 October 2011

Auditorium

Book of Abstracts

Contents

1

The PSI hardware inventory database

Author: Alain Bertrand[1]

Co-authors: Fabian Haemmerli [1]; Hubert Lutz [1]; Renata Krempaska [1]

[1] *PSI*

Corresponding Author: alain.bertrand@psi.ch

Summary:

Has the number of hardware parts required for a control system increase, same goes to the maintenance and handling of those. To keep track of all the control related hardware parts, we developed for the 3rd time an hardware inventory database. In this talk I will explain why a 3rd time, and what do we have exactly as tools / features in this software.

2

Another alarm handler concept

Author: Alain Bertrand[1]

[1] *PSI*

Corresponding Author: alain.bertrand@psi.ch

Summary:

Many if not most of the institute working with EPICS do use the standard Alarm handler, however we come to a point here at PSI, where the standard one doesn't provide the features we required, specially for the beamlines. I will here present our needs talk about our concept and show what we did. A new kind of alarm handler, which runs in 2 or 3 different layers, with a core running all the time on a server and GUIs which can connect to it.

3

Results from the 2011 EPICS Codeathon

Author: Andrew Johnson[1]

[1] *Argonne National Laboratory*

Corresponding Author: anj@aps.anl.gov

Summary:

Report from the code development workshop at LBL in Berkeley, California from August 22-26th.

4

The EPICS v3 development roadmap

Author: Andrew Johnson[1]

[1] *Argonne National Laboratory*

Corresponding Author: anj@aps.anl.gov

Summary:

Recent developments and future plans for EPICS v3.

5

Hard Rock CAFE

Author: Jan Chrin[1]

[1] *Paul Scherrer Institut*

Corresponding Author: jan.chrin@psi.ch

Summary:

A number of channel access (CA) interfaces to various high-level programming languages already exist within the EPICS extensions domain. Many of these are extensions to declarative and 4th generation languages each of which incorporates its own dedicated C/C++ class to establish channel access connectivity. Such CA classes are often embedded with function calls provided by the language's C/C++ API, and are thus, by definition, tied to the programming language at hand. Another approach, however, is to enforce a logical boundary between the channel access components and the specifics of a given application's C/C++ extension framework. Such a methodology then provides the basis for the modular use of a single C++ CA class across a number of C/C++ based programming languages. CAFE (Channel Access interFacE) is a new C++ library, built upon the standard channel access client library, that provides a modern, multi-faceted interface to EPICS. CAFE makes extensive use of templates and containers with multiple STL-compatible access methods that improve efficiency, flexibility, and performance. Stability and robustness, as 'Hard as Rock', are accomplished by ensuring that connectivity to EPICS records remain in a well defined state in every eventuality. CAFE presents the user with a number of options for instigating synchronous and asynchronous operations. In addition to providing basic read/write methods, a further layer of abstraction provides a convenient interface to groups of objects or collections of logical data sets, which can be easily instantiated through an XML-based configuration mechanism. CAFE aims to provide a comprehensive, yet extendable, C++ CA class that can be well tested in-house through its adoption in bindings to various C/C++ based high-level languages, and which, if proven effective, will ultimately serve to reduce the load on maintenance associated with CA client-based software. CAFE's suitability for use within different applications is demonstrated with examples from high performance Qt GUI control widgets, to event processing agents that propagate data through OMG's Data Distribution Service (DDS), and to script-like frameworks such as MATLAB.

6

Introducton to EPICS Version 4 and its use use at BNL

Author: Leo Dalesio[1]

Co-authors: Greg White [2]; Guobao Shen [1]; Marty Kraimer [3]; Matej Sekoranja [4]; Nikolay Malitsky [1]

[1] *Brookhaven National Laboratory*

[2] *SLAC*

[3] *None*

[4] *CosyLab*

Corresponding Author: dalesio@bnl.gov

Summary:

EPICS version 4 has been under development for the last 5 years. The project started as a platform for developing hierarchical EPICS records. The first practical application does not use the database, but instead takes advantage of the ability to expand the data types available over Channel Access. This paper discusses the overall architecture that employs EPICS version 4 and the new dbr_types developed to support the new services for high level applications.

7

ChannelFinder: operational experience at BNL

Author: Gabriele Carcassi[1]

Co-author: Kunal Shroff [1]

[1] *Brookhaven National Laboratory*

Corresponding Author: carcassi@bnl.gov

Summary:

ChannelFinder has been in full production at BNL since June 2010. We will go through the local setup, and how is being used by our physics group.

9

EPICS Sequencer 2.1

Author: Ben Franksen[1]

[1] *HZB*

Corresponding Author: benjamin.franksen@helmholtz-berlin.de

Summary:

The EPICS State Notation Language (SNL) and Sequencer is a central building block of many EPICS applications. A new version 2.1 was published in July 2011. It features a number of language extensions and enhancements, many bug-fixes, a cleaned up implementation, and an automated test suite. I present an overview of these developments, with an emphasis on the user visible changes, as well as plans for future developments.

20

EPICS for beginners (Lecture)

Corresponding Author: elke.zimoch@psi.ch

Summary:

Getting started, installing EPICS, my first IOC, client tools MEDM, StripTool, AlarmHandler)

21

Running user C-code on IOCs

Corresponding Author: dirk.zimoch@psi.ch

22

Using the StreamDevice driver

Corresponding Author: dirk.zimoch@psi.ch

23

State Notiation Language

Corresponding Author: ralph.lange@helmholtz-berlin.de

26

Epics for beginners (Hands-on)

Corresponding Author: elke.zimoch@psi.ch

27

iPhone Apps for EPICS

Author: Matthias Clausen[1]

Co-author: Tom Pelaia [2]

[1] *DESY*

[2] *SNS*

Corresponding Author: matthias.clausen@desy.de

Summary:

SNS (Tom Pelaia) and DESY/ Univ. of Hamburg have jointly developed Apps for the iPhone. This includes Probe, an Archive Viewer and an Alarm Table. We will demonstrate how these Apps can directly interact with web pages and (alarm) SMS messages. Of course we will also discuss future plans.

28

Beam-synchronous DAQ at SwissFEL test injector

Author: Babak Kalantari[1]

[1] *Paul Scherrer Institut*

Corresponding Author: babak.kalantari@psi.ch

Summary:

This talk gives an overview of the beam-synchronous data acquisition at SwissFEL injector test facility which is required for on-demand data collection to do correlation studies. I will show how we have achieved synchronous collection of the acquired data (EPICS channels) across separated IOCs with asistance of the event timing system and by using "only" the statndard EPICS records (no custom record/device support). The mechansim has been built into a generic IOC software application package which can be installed on any IOC equipped with an event receiver and and allows runtime configuration of most of parameters including channels to be collected etc.

29

Welcome

30

Goodbye

Corresponding Author: dirk.zimoch@psi.ch

32

ChannelFinder & Control System Studio

Author: Kunal Shroff[1]

[1] *Brookhaven National Lab*

Corresponding Author: shroffk@bnl.gov

Summary:

A Look at various client applications and utility plugins using ChannelFinder in CSS.

33

Connecting EPICS with Easily Reconfigurable I/O Hardware

Authors: Arun Veeramani[1]; Thierry Debelle[1]

[1] *National Instruments*

Corresponding Author: arun.veeramani@ni.com

Summary:

This presentation covers recent developments aimed at facilitating the integration of Reconfigurable I/O (FPGA-based) hardware with EPICS. The Linux driver available for National Instruments Reconfigurable I/O (NI-RIO) hardware provides the interface between the FPGA firmware and the host controller. The FPGA programming of these devices is accomplished using LabVIEW and can include custom VHDL code if needed. Support is currently available for basic record types, including waveform, and we will discuss briefly the implementation details as well as future possible improvements.

34

Using COTS Hardware with EPICS Through LabVIEW - A Status Report

Authors: Arun Veeramani[1]; Thierry Debelle[1]

[1] *National Instruments*

Corresponding Author: arun.veeramani@ni.com

Summary:

Many commercial-off-the-shelf (COTS) hardware need custom driver development for use with EPICS. With LabVIEW having drivers for many of the COTS hardware, the research community has been implementing interfaces for EPICS and LabVIEW over the years. This presentation is a status report of the development work done by National Instruments and covers different methods of interfacing EPICS with LabVIEW. Both configuration-based and programmatic method of implementation will be covered. The different operating systems that the options cover along with advantages and limitation will also be discussed.

35

ITER tools and drivers

Author: Franck Di Maio[1]

Co-author: lana abadie [1]

[1] *ITER*

Corresponding Author: franck.dimaio@iter.org

Summary:

ITER specific tools:
- Self-Description Data tools: Eclipse-based editor, web browser + translators and parsers to generate and parse EPICS and Control System Studio files.
- Maven plug-ins for compilation, test, packaging
- Linux drivers and device support for N.I PXI/PXIe I/O boards.
Will be briefly presented (updated status & short-term plans).

38

Marshalling data using the concat record

Author: andrew starritt[1]

[1] *australian synchrotron*

Corresponding Author: andrew.starritt@synchrotron.org.au

Summary:

At the Australian Synchrotron, we has a need to scan and save over 300 PVs from out 100-element XIA detector, but the sscan record is limited to 70 detectors. We decided to group the 100 fast peak values, 100 slow peak values and 100 ROI values into three 100-element waveforms and use the "scanH"record to capture this data.

The concat record was developed as a means to group an arbitrary set of scalar PVs into a single waveform record.

39

pvManager cookbook

Author: Gabriele Carcassi[1]

Co-author: Ralph Lange [2]

[1] *Brookhaven National Laboratory*

[2] *BESSY II*

Corresponding Author: carcassi@bnl.gov

Summary:

This talk will briefly show the motivation behind the development of pvManager, to then concentrate on a series of examples that show the power of the approach.

40

EPICS V4 Normative Data Types

Author: Gabriele Carcassi[1]

Co-author: Leo Dalesio [1]

[1] *Brookhaven National Laboratory*

Corresponding Authors: dalesio@bnl.gov, carcassi@bnl.gov

Summary:

This talk will concentrate on the data type definition, and show how they are being "exercised" using pvManager and other client tools.

41

ChannelFinder Service

Author: Ralph Lange[1]

[1] *Helmholtz-Zentrum Berlin für Materialien und Energie / BESSY II*

Corresponding Author: ralph.lange@helmholtz-berlin.de

Summary:

ChannelFinder is a directory server for control systems, implemented as a REST style web service.

CSS Workshop / 42

CSS Intro

Authors: Kay Kasemir[None]; Kunal Shroff[1]

[1] *Brookhaven National Lab*

Corresponding Author: shroffk@bnl.gov

CSS Workshop / 43

CSS application: BOY

Authors: Kay Kasemir[None]; Xihui Chen[None]

Corresponding Author: ralph.lange@helmholtz-berlin.de

CSS Workshop / 44

CSS application: DataBrowser

Author: Leo Dalesio[1]

[1] *Brookhaven National Laboratory*

Corresponding Author: dalesio@bnl.gov

45

CSS Product

CSS Workshop / 46

CSS application: Channel Finder Based application

Author: Kunal Shroff[1]

[1] *Brookhaven National Lab*

Corresponding Author: shroffk@bnl.gov

CSS Workshop / 47

CSS Product: How to put together your own Site specific CSS product

Author: Kunal Shroff[1]

[1] *Brookhaven National Lab*

Corresponding Author: shroffk@bnl.gov

CSS Workshop / 48

CSS Infrastucture: the build and update mechanism, future developments

Author: Kunal Shroff[1]

[1] *Brookhaven National Lab*

Corresponding Author: shroffk@bnl.gov

49

EDM Screen Display using Python

Author: Glen Wright[1]

[1] *Canadian Light Source, Inc*

Corresponding Author: glen.wright@lightsource.ca

Summary:

To learn how to use the Python language, I started a project to display EDM's .edl files using python and QT. This presentation will discuss the progress to date, the advantages and disadvantages of this project as compared to EDM, and some of the other EPICS/Python/Qt projects underway at the Canadian Light Source.

50

Prototyping the Next EPICS Archiver

Author: Nikolay Malitsky[1]

[1] *BNL*

Corresponding Author: malitsky@bnl.gov

Summary:

The report presents analysis of the EPICS Channel Archiver and a proposal of the next version based on the SciDB array-oriented data management and analytics platform. The resulting system aims to support millions of control process variables and streaming rate of one million events per second.

51

Portable Channel Access Server in Python

Author: Xiaoqiang Wang[1]

[1] *Paul Scherrer Institut*

Corresponding Author: xiaoqiang.wang@psi.ch

Summary:

Portable Channel Access Server (PCAS) library exists along with database channel access server (RSRV) in EPICS base. PCAS provides several C++ classes (server tool), making use of abstract callback methods, to let server application respond to channel access clients requests. Due to the intrinsic complexity of C++ and channel access request/data handling, this is not widely used among EPICS developers.

Python is a language easy enough to get in quickly while possessing rich standard libraries and numerous extensions. This makes it very attractive to prototype a PCAS application in Python.

52

A IOC based on PXI and virtualization technology

Authors: Bruno C. Yenikomochian[1]; Diego O. Omitto[2]; Harry Westfahl Jr.[2]; James R. Piton[2]; Marcio P. Donadio[2]; Marco A. Raulik[2]

[1] *National Instruments Brasil*

[2] *Brazilian Synchrotron Light Laboratory-LNLS*

Corresponding Author: james@lnls.br

Summary:

The Brazilian Synchrotron Light Laboratory (LNLS) has a 1.37 GeV machine open to scientific community since 1997. More recently, the control system of its beamlines, originally designed within a proprietary Delphi/Windows platform, is going through an upgrade to the open source EPICS/Linux platform used in many other synchrotrons. Within this upgrade strategy, the use of off-the-shelf hardware was also considered an alternative to the original in-house developed equipment, while keeping the EPICS/Linux compatibility. As a possible solution, a PXI chassis and its modules were made available to EPICS through the NI Real-Time Hypervisor virtualization system that allows running simultaneously EPICS/Linux and LabVIEW RT in the same PXI controller, sharing a common memory block for communication. A data exchange protocol was developed to implement motor, scaler and binary in/out EPICS records and channel access in the Linux layer, leaving the low-level hardware control to the LabView

RT layer. The data exchange was done using optimized libraries of the virtualization system that allows a robust yet flexible operation of IOCs. This solution was tested to fully control an X-ray absorption spectroscopy beamline, showing a substantial reduction of counting deadtime, stability and software development time for integrating new hardware.

LabView and EPICS / 53

LabView Channel Access Server

- Features and demonstration

- Hands-on exercise

- Programmatic configuration for large deployment

54

LabVIEW Channel Access Client

- Features and demonstration

- Hands-on exercise

cRIO and EPICS hands-on / 55

CompactRIO and EPICS IOC

56

EPICS IOC running on CompactRIO

- Hands-on exercise

57

EPICS on the Altera NIOS2 FPGA Softcore Processor

Author: Jeffrey Hill[1]

[1] *LANL*

Corresponding Author: johill@lanl.gov

Summary:

My progress porting EPICS/RTEMS to the Altera NIOS2 FPGA Softcore Processor. I will also include some slides on my EPICS core work.

Timing Workshop / 58

Timing goes Express

Corresponding Author: jukka.pietarinen@mrf.fi

Timing Workshop / 59

”mrfioc2” driver status

Corresponding Author: mdavidsaver@bnl.gov

Timing Workshop / 60

Challenges of Timing at high repetition

Corresponding Author: babak.kalantari@psi.ch

Timing Workshop / 61

Event generator sequencer programming

Corresponding Author: mdavidsaver@bnl.gov

Timing Workshop / 62

Timing requirements of SwissFEL plus open discussion

Corresponding Author: babak.kalantari@psi.ch

63

EPICS for beginners (hands-on)

Corresponding Author: elke.zimoch@psi.ch

64

EPICS for beginners (Lecture)

Corresponding Author: elke.zimoch@psi.ch

65

Writing CA Clients

Corresponding Author: dirk.zimoch@psi.ch

66

Using the StreamDevice driver

Corresponding Author: dirk.zimoch@psi.ch

67

State Notation Language

Corresponding Author: ralph.lange@helmholtz-berlin.de

68

The PSI web interface to the EPICS channel archiver

Author: Gaudenz Jud[1]

[1] *Paul Scherrer Institut*

Corresponding Author: gaudenz.jud@psi.ch

Summary:

The EPICS channel archiver is a powerful tool to collect control system data of thousands of EPICS process variables with rates of many Hertz each to an archive for later retrieval.
Within the package of the channel archiver version 2 you get a Java application for graphical data retrieval and a command line tool for data extraction into different file formats. For the Paul Scherrer Institute we wanted a possibility to retrieve the archived data from a web interface. It was desired to have flexible retrieval functions and to allow to interchange data references by e-mail. This web interface has been implemented by the PSI controls group and has now been in operation for several years. This presentation will highlight the special features of this PSI web interface to the EPICS channel archiver.

.

69

Running user C-Code on IOCs

Corresponding Author: dirk.zimoch@psi.ch

70

pvAccess, pvData, pvIoc, pvService overview and status

Corresponding Author: mrkraimer@comcaste.net

71

Interoperation with EPICS V3

Corresponding Author: mrkraimer@comcaste.net

72

EPICS V4 Discussions and Feedback

73

EPICS V4 Services

Corresponding Author: greg@slac.stanford.edu

74

EPICS V4 roadmap and charter

Author: Greg White[1]

[1] *SLAC*

Corresponding Author: greg@slac.stanford.edu

75

Welcome

Corresponding Author: dirk.zimoch@psi.ch

76

Server-Side Plug-In Framework

Author: Ralph Lange[1]

[1] *Helmholtz-Zentrum Berlin für Materialien und Energie / BESSY II*

Corresponding Author: ralph.lange@helmholtz-berlin.de

Summary:

EPICS 3.15 will add a framework for server-side plug-ins.
Clients can add JSON modifiers to channel names, which will instantiate and configure plug-ins on the server, that are inserted into the client's Channel Access connection.
Plug-ins may change update data and meta-data (time stamps, data type, array sizes etc.), and are a very powerful tools to allow client-defined subscriptions.

77

Feedback from LabView Users to NI

78

Virtual Servers for Controls Applications

Author: Rene Kapeller[1]

[1] *Paul Scherrer Institut*

Corresponding Author: rene.kapeller@psi.ch

Motion Systems Workshop / 81

Motion systems at PSI current status

Corresponding Author: claude.pradervand@psi.ch