

# Misc reports: EPICS council and Python Working group

Richard Farnsworth  
*Controls program manager*

# Intro to the EPICS council

**This is both Preemptive and presumptive**

**As the Council will meet tomorrow!**

# EPICS Council Charter

At the EPICS Collaboration Meeting hosted by ORNL in September 2016, an exploratory meeting was held to consider the organizational structure of the EPICS Collaboration. Representatives from APS, DLS, ESS, FRIB, ITER, LCLS, NSLS-II, and SNS met to discuss how to better provide long-term support for EPICS development efforts.

A collaborative software project like EPICS requires facilities to make multiyear commitments, investing significant resources in the form of labor and/or procurement to support ongoing development and improvements. The EPICS project has been successful, but has relied on a “best effort” basis by facilities for support. Without a structure for long term planning, development efforts are vulnerable to short term needs and budget availability of individual laboratories. The representatives at the exploratory meeting agreed that an EPICS Council be formed as described below.

-

# The Council at the moment...

<b>Member</b>	<b>Lab/Facility</b>
Henrik Carling	European Spallation Source
Joe Delong	SLAC National Accelerator Laboratory
Richard Farnsworth	Brookhaven National Laboratory
Daniel Flath	SLAC National Accelerator Laboratory
Steven Hartman	Oak Ridge National Laboratory
Mark Heron	Diamond Light Source
Jeff Hill	Los Alamos National Laboratory
Markus Janousch	Paul Scherrer Institute
Guabao? Shen (was J Maclean)	Argonne National Laboratory
Anders Wallander	ITER
Karen White (Chair)* End of Term	Oak Ridge National Laboratory

# Purpose of the Council

- The EPICS Council is formed to optimize the use of resources available at investing facilities to ensure the ongoing viability of EPICS as the control system toolkit of choice for scientific facilities. To that end, the Council will:
- Prioritize major EPICS upgrade projects for Base and Extensions to guide resource allocation decisions at investing facilities.
- Develop a roadmap for future EPICS Core and Extensions development to facilitate planning for all EPICS sites. The roadmap will be developed using technical input from the chairs of relevant working groups (currently EPICS Core Working Group and CS-Studio Working Group).
- Provide support to control system managers in promoting EPICS development efforts to their organization leadership.
- Select semi-annual EPICS Collaboration Meeting sites and dates.
- Ensure that EPICS continues to be an open collaboration and that contributions are open-source.

# Council Composition

- Each EPICS facility, allocating an annual resource of 1 FTE, or more towards EPICS Core and/or Extensions development efforts (identified in the roadmap), may appoint a member to the Council. The appointee should have authority to commit facility resources to support common EPICS development efforts.
- To request appointment of a member to the council, a facility should contact the Council chairperson. The chairperson will document the resource commitment and representative of the facility and put the nomination before the full council for validation. Facilities who, in the opinion of the Council contribute the required minimum resource towards agreed priorities will maintain representation on the Council.

# The “rools” – Council management

The Council meets in person once a year, scheduled in conjunction with an EPICS Collaboration Meeting or similar event that many Council members would already attend. The Council will elect, by simple majority, a chairperson to serve for a period of 2 years with turnover effective May 1 of each odd year beginning 2017. The chairperson will be responsible to arrange annual Council meetings, manage Council representation and facilitate development of Council documents. Documents produced by the Council will be published on-line for use by all EPICS facilities. The chairperson will also ensure sites and dates have been committed for collaboration meetings approximately six month in advance.

# EPICS working groups: New one - Python



# There are several EPICS working groups

- Covering CS-Studio
- EPICS Core
- EPICS V7
- A “database” one (Somewhat deprecated)

And now, recognizing the on the up Python community

- The Python working group

Credit: Much of this is taken from the working paper by Daniel Flathe, Anders Wallander, Tom Caswell, Andrew Johnson et al.

# Python for EPICS Working group

The popularity of Python for scripting and data analysis has steadily increased in the scientific community, and EPICS facilities have been no exception. In the past there have been widely-accepted interface libraries for EPICS information but beyond that different facilities have largely created bespoke Python applications specific to their needs. Unsurprisingly when it comes time to share these tools, facility-specific assumptions are revealed and what should be a shared library is forced to be recreated for the next project.

The mission of the Python for EPICS Working Group is to support the development of Python applications with the broader community in mind. By providing a formal stage for discussion of Python tools, we hope to both encourage developers to create more flexible and recognize facility independent needs that be can addressed through collaboration between facilities.

# Goals - EPICS Clients

- Python applications need well established libraries to exchange data with EPICS IOCs, this includes both [Channel Access](#) as well as [pvAccess](#). For a variety of reasons, facilities may prefer to use one of these connection libraries over another. It is not the role of this working group to demand that one of these modules be used, instead we should ensure that a choice in Python EPICS client does not preclude the use of any higher level application.
- While a specific client will not be specified, it is in the scope of this working group is to define a standard interface for these libraries to employ. Having these strictly defined by representatives from multiple development groups will ensure that the agreed upon interface is fair to all parties. This will benefit downstream developers who can limit the number of interface layers required to support multiple client architectures.
- It is also in the scope of this working group to produce a number of standard bench tests for evaluating the difference in performance between different client libraries. Maintaining a fair and balanced approach to analytically determining the effect of different implementations will better inform facilities choosing between clients, and the developer of those libraries.

# EPICS Tools

There are a number of utilities that are often deployed as part of the EPICS ecosystem. While the Python interfaces to these might not be making calls to EPICS IOCs, the discussion around the clients that interact with these systems is part of the scope of the working group. Example of EPICS tools which may have Python interfaces include, but are not limited to the [EPICS Archiver Appliance](#), operational logbooks, and alarm management systems.

# Device Abstraction Layers

Connections to individual EPICS signals are often not enough to provide a proper API for higher level applications and scripts. Complex yet common IOCs such as AreaDetectors and EpicsMotor required the coordination of many different process variables to perform their job. This working group will discuss the Python modules that have been created to interact with these devices in the control system. Discussion will primarily focus on [Ophyd](#), to represent, introspect, and communicate with EPICS IOCs not as individual PVs but feature-rich Python classes.

# Data Collection and Run Control

Python is often used to coordinate the complex procedures involved in data collection. Multiple axes and detectors are required to move in sync to make meaningful scientific measurements. This working group will discuss Python libraries that assist in the configuration, execution of these experimental plans as well as the organization of the collected data. Focus will be on the [Bluesky](#) library developed at Brookhaven National Laboratory.

# Python Development and Packaging Guidelines

In addition to discussion about the specific goals listed above, the working group will discuss best practices for the distribution and management of Python libraries. Coordination for CONDA and PyPI deployment, centralizing continuous integration procedures, testing routines and benchmarks are all in the scope of discussion.

Collaborators will be encouraged to use the best practices established by the broader scientific Python community[

# Governance

Up to this point the collaboration has successfully managed design discussions through pull-request review on Github, conversation on the Nikea Slack, and phone calls as needed. As the collaboration grows to include more facilities and more people, we recognize the need for a more formal process for establishing roadmaps and resolving disagreements when they arise. We would like to take our time with this process, learning about what models are being used by other collaborations over the next year.