

*Title:*

## **EPICS: RECENT APPLICATIONS AND FUTURE DIRECTIONS**

*Author(s):*

L. Dalesio

*Submitted to:*

<http://lib-www.lanl.gov/la-pubs/00796395.pdf>

# EPICS: RECENT APPLICATIONS AND FUTURE DIRECTIONS\*

L. Dalesio, LANL, Los Alamos, NM 87545, USA

## *Abstract*

The Experimental Physics and Industrial Control System (EPICS) is being used as the platform for the Swiss Light Source (SLS) and the Spallation Neutron Source (SNS). We will look at the use of commercial companies as a way to outsource the labor required during the construction phase of a project. I will discuss the tools that were developed and the issues in having a commercial company provide EPICS applications.

## 1 INTRODUCTION

The Experimental Physics and Industrial Control System (EPICS) is in use at a number of physics projects in North America, Europe and Asia that vary in size, specification, platform and requirements. In both the SNS and the SLS, many developments have been made to the EPICS core software, the configuration tools, and the operator tools. Both projects have also taken advantage of industrial support. In this paper, I discuss the most recent significant developments in the EPICS community and the affect of these changes on the community as well as the use of industry to accomplish laboratory projects.

## 2 RANGE OF PROJECTS USING EPICS

There are a number of projects using EPICS for a broad spectrum of applications. EPICS began as a collaboration between Argonne National Laboratory and Los Alamos National Laboratory in 1991, building on work that was initially done at the Ground Test Accelerator. It is now running on accelerators that have as many as 180 distributed front-end controllers with control rooms with 20 consoles and a gateway to make system parameters available to offices, web sites, and other remote control stations. It is also used at single controller and one workstation systems. Some of the accelerators using EPICS include: Advanced Photon Source, KEKB, CEBAF, BESSY II, SLS, LEDA, and SNS. It is used at several detectors for slow control at D0 at Fermi Lab, Halls A, B, and C at Jefferson Lab, Phenix and Star at RHIC, and Compass at CERN. It is used for telescope control at Keck II, WHT, CFHT, and Gemini. New accelerators planning to use EPICS include: PF-AR, JHF, CLS, and Diamond.

## 3 EPICS SYSTEM ARCHITECTURE

The EPICS architecture is a standard 3-layer architecture [1]. It consists of front-end computers usually running a real-time operating system. The front-end

computers communicate to field I/O through many different field buses. The operator consoles communicate to the front-end computers over Ethernet. In addition to this 3 layer architecture, a gateway is provided to isolate the control network from the facility network yet still provide all control system parameters to computers outside of the control network.

The support for operator consoles and EPICS developers exists on most flavors of UNIX and Windows. Most of the client applications run in both environments. The front-end computers usually run vxWorks and are supported on PowerPC, 68xxx, and x86 architectures. The gateway is currently operating in Solaris and Linux. Until recently, the smallest system possible was one vxWorks system and one console.

Field buses supported from the vxWorks environment are extensive and include: Controlnet, PCI, Can-bus, Industry Pack, VME, VXI, ISA, GPIB, Profibus, Bitbus, Modbus, Modbus+, Yokogawa, Group 3, Allen-Bradley Data Highway, and Ether IP. As many different laboratories develop these drivers, a comprehensive list is difficult to put together. The most complete list is kept at ANL [2].

In EPICS release 3.14, the front-end software was ported into LINUX, Solaris, windows, RTEMS, and L4 (a real-time LINUX)[3]. With this release it is possible to have a one box solution without a license from Wind River Systems. This port was supported by APS, LANL, CLS, and KEK. In these new environments, there are very few drivers written for field bus support. The first support for LINUX is the port of a GPIB driver. Single box EPICS systems are only running in our test labs at the moment.

## 4 EPICS CORE SOFTWARE

In nearly every project, extensions are made to EPICS. Some laboratories develop new hardware interfaces or new client programs. Some develop new configuration tools. It became clear very early that we would need to unbundle all of the EPICS software and give very close management to the communication protocol, Channel Access(CA) and the Process Database. The original design gave very clean interfaces to the protocol, but lacked a clean interface for new hardware support. The engineers at APS completed the task of defining clean interfaces for extending hardware support. They also did a lot of work developing clean installation and build procedures for EPICS. The build procedures, CA protocol and process data make up the Core of EPICS.

EPICS Core is closely controlled and rigorously tested. It consists of the Channel Access Client, the Channel Access Server, the Portable Channel Access Server, the EPICS Database Engine, a standard set of EPICS Database Record Types, and the Gateway. EPICS release numbers distinguish this portion of the code. All other code is considered extensions and it is maintained, released and documented by the authors as they are able.

EPICS core releases are produced about every 12 to 18 months. Each release is put out in alpha and beta releases for use on test stands. After several months of operation in this environment, an initial release is made and put onto a limited number of production machines. These are usually machines that require the newest features. After several months in this form, laboratories pick up the new release as their schedule and interest allow. We find few problems after a release is made in this fashion.

## 5 EPICS EXTENSIONS

Since physics applications have a variety of challenging requirements and physicists and programmers have an array of favorite approaches, EPICS provides interfaces for extensions at all levels.

The channel access client interface is the most frequently used. There is access to the EPICS client in several languages: C, C++, Fortran, and Java. EPICS real-time data can be directly accessed in analysis packages like IDL, Mathematica and Matlab. There is channel access client support for windows through active X, DDE, or Visual Basic. Channel access calls can be made from the scripting languages: tcl, PERL and Python. There are two finite state machine languages, SNL and FSQT, supported under vxWorks or UNIX. There are also callable interfaces from the physics codes SAD and SDDS.

A Portable Channel Access Server was developed to support the inclusion of other data stores into EPICS. This library is used to provide connection, read, write and monitor access to existing control systems, LabView systems, or any other data store that is needed by other CA clients.

The EPICS Process Database contains the logic to read data, convert, check alarms, perform interlocks, and provide closed loop – steady state control. A standard set of Record Types are supported for analog I/O, digital I/O, some standard control algorithms, general purpose calculations, C subroutine inclusion, and some data collection. More importantly, this set of record types is easily extended. There are a well defined set of support routines that are required to add a new type – without changing EPICS core. Perhaps a more frequently used interface is the one that allows member sites to add support for new hardware devices. This interface is used as often as the CA client interface.

Two client programs that have used the same approach as EPICS core, are the Channel Archiver[4] and the Extensible Display Manager(EDM) [5]. Both allow extensions through clean interfaces. The Extensible Display Manager provides screen management, file I/O, dynamic colors, and a dynamic data interface. It allows

extensions in two very important areas: new data sources and new display widgets. I expect that this will quickly become the engineer's display platform of choice. The Channel Archiver provides a clean interface for storing archive data and for retrieving archive data. By changing out the data factory object, all channel access and viewing tools can be ported to a new data store. The archiver is currently in use on a proprietary file system, SDDS, and ORACLE.

## 6 EPICS CONFIGURATION TOOLS

Many of the EPICS tools run from configuration files. These configuration files most frequently use a well defined ASCII format. This enables any site to create a configuration tool or use existing technologies to produce and manage configuration files. Some new configuration tools worth mentioning are for configuration of the EPICS process database. Visual DCT, a graphical process database tool[6], was developed at JPI for the Swiss Light Source. It may resolve licensing issues with our other two packages: GDCT which uses Openview and Capfast which is a proprietary package. The ASCII files produced by these graphical configuration tools can be fed into EPICS ORACLE tables and edited from two forms that were developed to support easy field entry[7]. These graphical tools are ideal for understanding complex logic or system hierarchy. The ORACLE tools are good for producing reports or doing data entry. The tools are meant to compliment each other.

## 7 TAKING ADVANTAGE OF THE WEB

There are many laboratories that are developing support for the web to ease their operational procedures and network loads. At Jlab, the Motif Editor and Display Manager was modified to load display configuration files over the web and then use Channel Access to communicate to their control room gateway. This approach allows operational support from home computers with reasonable performance using the exact same control screens that are running in the control room. The Channel Archiver, developed at Los Alamos, provides web based archive management as well as data retrieval. The data retriever allows the user to select channels and time ranges to find interesting data. This data can then be exported to the local machine for analysis in Excel or Matlab. As physics laboratories go to more and more remote support and widely distributed collaborations, this wide area network support will become more critical.

## 8 PLANNED CHANGES TO EPICS CORE

Some fundamental changes are underway in EPICS core. The underlying code for channel access is being replaced with an object oriented (OO) design that is built around a new abstract data object. With initial prototyping complete and performance numbers that show that we will still be able to support high throughput, we are

proceeding with plans to implement this new data object interface. This change will allow us to address several new problems. Our next step is to replace the General Data Descriptor (GDD) in the Gateway with the new Abstract Data Object (ADO) – giving us higher performance and support for all of our data types. We will then replace the embedded Channel Access Server with the Portable Channel Access Server and significantly reduce the amount of effort required for code support. Finally, we will send this new Abstract Data object over the wire. Once this is complete, we should be able to implement several functions with minimal effort.

A major gain of this new infrastructure will be the ability to create new data structures. This will allow us to add new data types that could include: statistical data that is produced by processing a large data set into small statistical packages before network transmission, or physics type data such as orbit, or historical data that includes time. These new data types, if carefully defined, could provide the basis for developing network aware and portable high level physics applications.

Another major gain for the EPICS community will be the ability to add new event types. Currently, data requested on change can only respond to channel type events. New events could be created to send data from many channels when a single channel changed. We could also add events for notifying clients when metadata changes – like display limits. The new event types will provide channel access client programmers with the ability to define their information very accurately and thus limit the amount of traffic going over the network.

## 9 EPICS AND INDUSTRY

Laboratories face tighter cost controls and a shortage of technical expertise, so large budget peaks that occur during construction are managed with more industrial support. Several examples of complete industrial involvement are seen throughout the collaboration. SLS had their LINAC and RF systems delivered from industry with EPICS control systems. PSI trained their vendors in the use of EPICS. The same approach is being taken at the SNS for the conventional facilities. A less integrated approach is one taken at LEDA, where a subsystems are delivered with PLC control is supplied by subsystem vendors. The integration into EPICS was done by the LANL control group. Good communication at the beginning of the project made the integration task very easy for the vacuum subsystem. For the SNS RF control, we learned to communicate better and earlier than we did at LEDA. The best example of industrial involvement is at KEK where Mitsubishi was trained to use EPICS and delivered the entire control system. They continue to operate the control system under control and supervision of the KEK physicists. Recently, EPICS has been put into the public domain. This makes it possible for any laboratory or industrial concern to use EPICS for any purpose. The intent is to encourage industrial participation in providing support for laboratory projects.

## 10 EPICS – NO FREE LUNCH

There are several issues with getting started with EPICS. Most successful sites have two things in common, they received some EPICS training and they have a real-time systems expert that can help debug hardware and low level system problems. If you are trying to do something new and complex, you will need to have experienced engineers to do the work. If you have a simple, well-defined problem that is not likely to expand – LabView may be appropriate and it is much easier to learn.

*Caution:* Programs written by non-programmers tend to lack good structure in any language.

Other issues with EPICS are that there are many tools to do the same job. It is a bit difficult for beginners to decide which to use. Again, training helps and collaboration meetings are also useful. The EPICS collaboration is supportive through web based documentation[8][9] and an email exploder – but training is really the best first step.

## 11 ACKNOWLEDGEMENTS

The work reported was accomplished by members of the EPICS collaboration. The success of any of the members in this community was built on preceding work. These projects then made contributions for their successors.

## 12 CONCLUSIONS

The modification that are being made to the channel access protocol will provide options to the collaboration for passing user defined data structures under an extendible set of events. The public domain availability of EPICS will make it easier to use commercial support for our projects. The collaboration continues to provide support for new hardware and software technologies by distributing the work throughout our member laboratories. Web based technology is being supported to provide more efficient operations. New tools are being developed to reduce the cost of creating and managing our applications. The collaboration is an active an ongoing enterprise that has been involved in many successful control system implementations.

[1] Dalesio, L. R., Kraimer, M.R., Kozubal, A. J., "EPICS Architecture," in *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems*, C. O. Pac, S. Kurokawa and T. Katoh, Eds. (ICALEPCS, KEK, Tsukuba, Japan, 1991), pp. 278-282.

[2] "Hardware Support by Bus", <http://www.aps.anl.gov/epics/modules/bus.php>

[3] M. Kraimer, "EPICS: Porting IOC Core to Other Operating Systems," ICALEPCS, Trieste, October 1999.

[4] K. Kasemir, "Data Archiving in EPICS", ICALEPCS 99, WC1P-61, Trieste, October 1999.

[5] K. Kasemir and J. Sinclair, "EDM", <http://mesa53.lanl.gov/lansce8/epics/Epics2001SL/kay/edm.ppt>

[6] M. Sekoranja, "Visual Database Configuration Tool", <http://lansce.lanl.gov/lansce8/epics/Epics2001SL/matej/VisualDCT.ppt>

[7] K. Kasemir, "SNS EPICS Configuration Database", [http://lansce.lanl.gov/lansce8/epics/Epics2001SL/kay/EPICS\\_RDB.ppt](http://lansce.lanl.gov/lansce8/epics/Epics2001SL/kay/EPICS_RDB.ppt)

[8] S.A.Lewis "Recommended EPICS Documentation", <http://csg.lbl.gov/EPICS/RecommendedDocs.html>

[9] L.R.Dalesio, "Guide to EPICS", <http://mesa53.lanl.gov/lansce8/Epics/epicsguide.html>