RECENT ADVANCEMENTS AND DEPLOYMENTS OF EPICS VERSION 4

Greg White, Murali Shankar (SLAC, Menlo Park, California), Andrew Nicholas Johnson, Sinisa Veseli (ANL, Argonne, Ilinois), Arman Arkilic, Leo Bob Dalesio, Michael Davidsaver Martin Richard Kraimer, Nikolay Malitsky, Bruno Seiva Martins (BNL, Upton, Long Island, New York), Matej Sekoranja (Cosylab, Ljubljana), David Gareth Hickin (DLS, Oxfordshire), Timo Korhonen (ESS, Lund), Guobao Shen (FRIB, East Lansing, Michigan), Ralph Lange (ITER Organization, St. Paul lez Durance), Steven M. Hartman, Kay-Uwe Kasemir (ORNL, Oak Ridge, Tennessee)

Abstract

EPICS version 4 is a set of software modules that add to the base of the EPICS toolkit for advanced control systems. Version 4 adds the possibility of process variable values of structured data, an introspection interface for dynamic typing plus some standard types, high-performance streaming, and a new front-end processing database for managing complex data I/O. A synchronous RPC-style facility has also been added so that the EPICS environment supports service-oriented architecture. We introduce EPICS and the new features of version 4. Then we describe selected deployments, particularly for high-throughput experiment data transport, experiment data management, beam dynamics and infrastructure data.

EPICS BASE AND EPICS VERSION 4

The Experimental Physics and Industrial Control System (EPICS), is a software framework for high-performance distributed control and data acquisition in large scientific instruments, such as accelerators and telescopes.

The "base" of EPICS is software for supervisory control, closed loop control, archiving, alarm management, timing and other aspects of front-end processors and device facing hardware. Typically hosted in an embedded system processor such as RTEMS or VME, this software and its host are collectively known as the IOC (Input/Output Controller) in an EPICS control system. IOCs are optimized for low-latency I/O. They control and/or monitor a collection of devices such as actuators and measurement diagnostics. Each IOC node contains a memory resident real-time database.

The IOC database is a set of "smart" records, which are interconnected in a *data flow* pattern. They're smart in that their field values may come directly from hardware, or a result of processing that was dependent on the type of record. The records may contain "device support" code, to interface the processing to physical devices through device drivers. Much more information can be found on the EPICS base at [1].

EPICS base and the software extensions built on top of it have proven very successful for the control aspects of scientific instruments, providing excellent low level I/O, DAQ and optimal control.

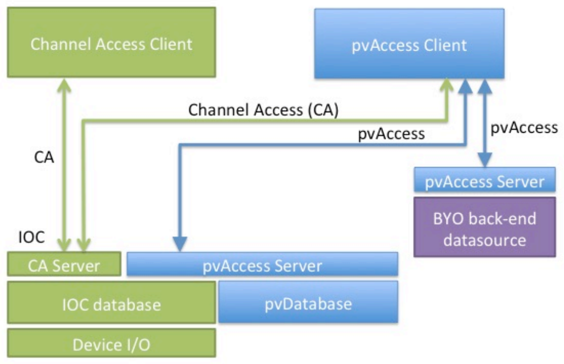


Figure 1: The basic architecture of EPICS version 4, showing the classic "base" components of EPICS in green, and components added by version 4 in blue. A new protocol, pvAccess, transports potentially complex data types encoded by pvData. pvAccess connects clients to IOCs, which may include a new database type, pvDatabase to support complex data processing services (and is the basis of the high-performance detector streaming applications described here), and to middleware services that can themselves connect to enterprise data stores, the web, etc. Note that hardware device I/O remains under the IOC database.

EPICS Version 4 for New Controls Problems

Modern control systems of large instruments call for more science to be done in the control system itself, than is possible with the EPICS base software alone. High output detectors call for pipelined data processing; physics applications deal with systems of process variables and their values rather than one process variable (PV) at a time; process variables may be structured, multi-modal, or require significant metadata associated with values in order to be understandable (such as images, or timing related metadata to give context when reviewing archives).

EPICS version 4 is a system of software modules that extend EPICS base to address these emerging requirements.

Core Modules of EPICS Version 4

Apart from EPICS base, the two core modules of EPICS version 4 are "pvData" and "pvAccess."

The pvData module is the high-performance structured data backbone of EPICS version 4. It provides an introspection interface for the dynamic creation and management of structured data types and arrays. Although types can be created on the fly, EPICS version 4 also defines a standard set of data types oriented towards scientific controls data (called "Normative Types" [3]). The types there defined include matrices, key-value sets, tables, histograms, continua, and images, among others both generic and application specific.

pvAccess is the network protocol of EPICS v4. Like the equivalent protocol in version 3, "Channel Access", named endpoints (PVs) are discovered by broadcast, while data I/O is by TCP. In addition to the expected get, put, and put/get methods one would expect, pvAccess supplies a Remote Procedure Call (RPC) method, and a method to deal efficiently with exchanging only the changed elements of large arrays. Working in concert with pvData, the two include memory management and efficient encoding and deserialization, to minimise copy and wire transactions for even large complex data I/O. For instance, in the case of a PV defined by a complex structure, a client can subscribe only to the fields of the structure of interest to that client [4].

```
$ eget -s XCOR:IN20:491:RMAT -a b BPMS:IN20:525
0.669544 0.694654 0        0        0        0
-0.57085 0.901274 0        0        0        0
0        0        1.333434 0.966896 0        0
0        0        0.358411 1.009578 0        0
0        0        0        0        1        0
0        0        0        0        0        1

$ eget
pva://mccas0.slac.stanford.edu/XCOR:IN20:491:RM
AT?b=BPMS:IN20:525

$ eget QUAD:LI24:900:TWISS
    energy 5.00512
    psix 37.7625
    alphax 13.6562
    betax -2.78671
    etax -0.00698294
    etaxp 0.00107115
    psiy 31.9488
    alphay 116.762
    betay 5.2592
    etay 0
    etayp 0
    z 2438.72
```

Figure 2: Examples of three *normative types* used with the new pvAccess command *eget*. The first shows an RPC request for the response matrix between two accelerator devices (sometimes called "rmat A to B"), in LCLS. The PV is "XCOR:IN20:491:RMAT". An argument named "b" whose value is the name of the second device was also given. Since the returned data self identified as normative type *NTMatrix*, eget displayed the data appropriately. The second example is of eget being given the same request in URL form. In fact, in both the first and second examples, the server received its request in the form of the normative type *NTURI*. The last example shows use of *NTNamedValue* to get and display the Courant-Snyder parameters of a quadrupole.

The RPC channel method enables a client to pass arguments along with a PV value request, in which case the value is computed with respect to the arguments.

Notably, the same PV may be accessed by either an asynchronous get method, or an RPC method. In this way, a get-value operation on one PV, can be either the familiar "monitor", or the "compute value subject to given parameters and return result." In either case, the processing path (and so potentially the PV's value) would be different. This RPC facility is being used extensively at SLAC, and is being adopted also at ESS, for beam dynamics (see Figure 2), infrastructure, and directory service data.

RECENT ADDITIONS TO EPICS VERSION 4

The pvAccess system is now codec based; the communication layer is decoupled from the transport layer, allowing alternative transport layer protocols such as one using zeroMQ. A codec for the former transport layer of pvAccess is provided.

Array handling has been greatly enhanced in the C++ implementations of the core modules. Reference counted shared vectors help minimize copy operations. Fixed and bounded arrays have been added in order to more easily facilitate high-performance array management. These complement the existing unbounded arrays. Stride length can be specified in put and get operations.

The Normative Types have been extended to include full dynamic typing (essentially an "Any" in the CORBA sense).

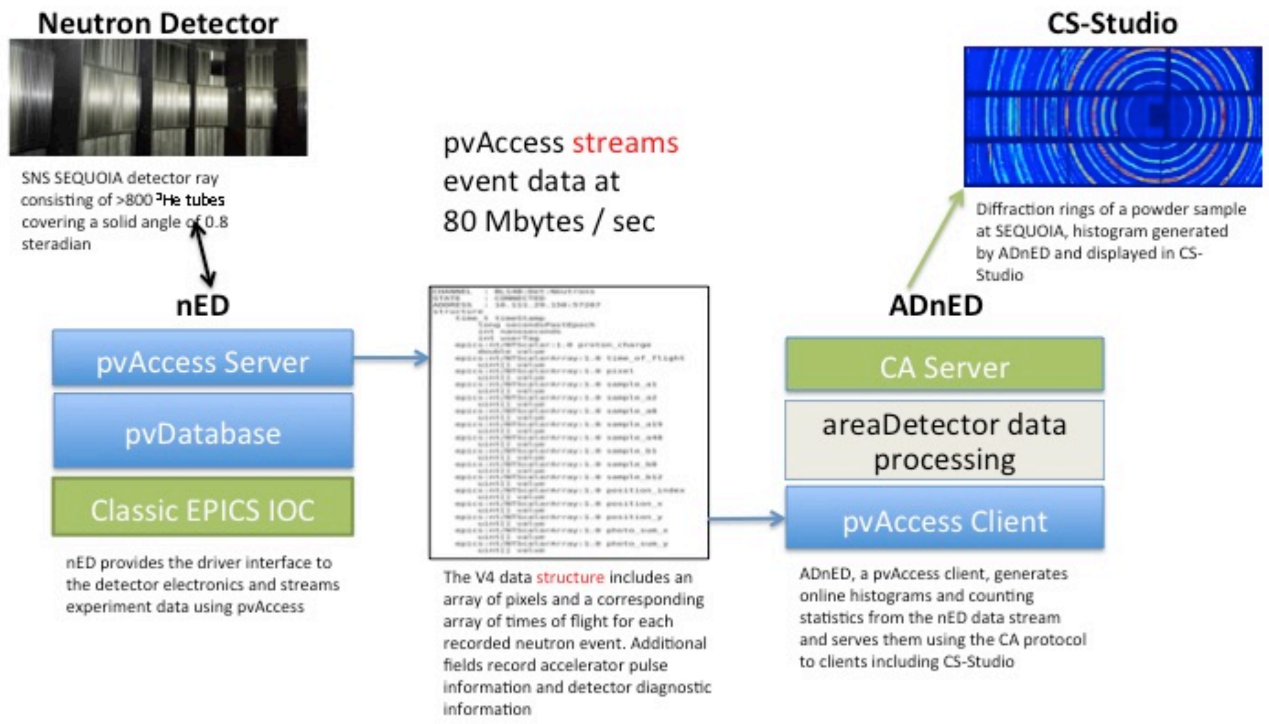
A plug-in architecture for implementing security schemes has been added, along with the plugin for Channel Access security.

The command line tools have been extended to better support time stamp, alarms, and correct display of enumerated types. All of the command line tools now support both pvAccess and Channel Access operations. A new tool, *pvlist* gives information about pvAccess servers on the network, including lists of the PVs they host.

Extensive API changes have been made to give greater flexibility for channel callbacks and converting from one introspection data type to another.

New Data Processing Database

The new "pvDatabase" module of EPICS v4 implements a framework for a memory resident database of records defined in terms of pvData structures. Like the IOC database of classic EPICS, the records of pvDatabase can process on I/O events; unlike the IOC the records may be of any structure the engineer wishes, and may pull in data from any pvAccess-ible data source, plus Channel Access. pvDatabase images may be standalone, or hosted within an IOC, where they might interface directly to base records, asynchronous device driver support (asynDriver), or detector control (areaDetector). pvDatabase is then useful for complex optimal control tasks, data assembly, and preprocessing. Combined with pvAccess streaming, it can be used as the basis of a data processing pipeline.



SNS SEQUOIA detector ray consisting of >800 ^{3}He tubes covering a solid angle of 0.8 steradian

nED provides the driver interface to the detector electronics and streams experiment data using pvAccess

The V4 data structure includes an array of pixels and a corresponding array of times of flight for each recorded neutron event. Additional fields record accelerator pulse information and detector diagnostic information

Diffraction rings of a powder sample at SEQUOIA, histogram generated by ADnED and displayed in CS-Studio

ADnED, a pvAccess client, generates online histograms and counting statistics from the nED data stream and serves them using the CA protocol to clients including CS-Studio

Figure 3: Outline dataflow for neutron detector data transport at SNS. An EPICS version 4 IOC (*nED*) takes data from *areaDetector* – a well-known detector data system – assembles it into a structured package using pvData, and transports it to EPICS v4 clients over pvAccess. One client in particular, *ADnED*, disassembles the data into classic PVs for consumption by existing display tools such as CS Studio. *Thanks to S. Hartman and K. Kasemir, ORNL*

SELECTED DEPLOYMENTS

The primary uses of EPICS version 4 have so far been found to be in high-performance detector data fan out, and in middleware data services.

EPICS v4 is used at SNS in data acquisition and processing of neutron scattering experiments [4], see figure 3. Neutron scattering is a technique complementary to X-ray scattering used in the understanding of the structure and properties of materials. At SNS, detector image pixel data from an experiment is combined with meta-data such as probe pulse id and charge, into a pvData structure, which then fully describes an experiment observation event. The event data is then streamed by pvAccess to clients for processing and analysis. Clients may subscribe to the fields of the event structure of interest to them. The primary client – effectively an EPICS v4 plugin for areaDetector – includes statistical processing and publication of selected fields as histograms by Channel Access, for consumption by existing GUI tools.

Diamond and BNL have also developed an areaDetector pipeline based on EPICS version 4, and the BNL framework is being distributed with areaDetector [5][6]. Figure 4 shows a CS-Studio image from an Eiger detector image processed with the BNL pipeline.

Both BNL and SNS have thoroughly investigated EPICS v4 network performance. Their findings are that pvAccess' network efficiency is constant, well-behaved, and capable of delivering at or near 95% of the nominal maximum bit rate on 1 Gb/s or 10 Gb/s Ethernet, with no CPU saturation [4][5].

BNL have also developed an interesting experimental beam-line data management system on EPICS version 4 [7]. The raw experiment data is dissociated from the metadata for storage and management, so that different and appropriate technologies can be used for the two different data categories. In particular, since the experiment data may itself be very differently structured from one experiment to the next, MongoDB is used for the store, and EPICS v4's flexible data typing can be used for high-performance transport. One particular EPICS v4 service, the "databroker," can interface to all experiment and metadata to provide a single interface to all.

FRIB and BNL have been using a PV configuration management tool named *MASAR* for some time. MASAR uses pvAccess to deliver sets of CA PV values to clients.

A collaboration of SLAC and ESS, are using EPICS version 4 for beam dynamics modelling services and device infrastructure data. Now EPICS (base and v4) provides all the data required for commissioning applications – both device PV values, plus optics, response matrices, device type lists etc. Services for beam synchronous data and mass archive data are planned.

CONCLUSIONS

EPICS version 4 can be easily integrated into an EPICS installation to supplement existing device support, DCS and SCADA. It effectively extends EPICS to support structured scientific data and complex processing. Early users have found it particularly useful for high-throughput detector data and for integration of beam dynamics and accelerator enterprise data, into the controls system.

THANKS

The authors wish to thank all users and testers of EPICS version 4 for their excellent feedback and guidance. Questions and feedback can be given on EPICS *tech-talk*.

REFERENCES

[1] EPICS Version 4 project website: http://epics-pvdata.sourceforge.net. [2] EPICS version 4 source code repositories, https://github.com/epics-base.

[3] G. White, L. Dalesio, M. Rivers, M. Kramer, D. Hickin, EPICS V4 Normative Types, http://tinyurl.com/l3pypbc

[4] K. U. Kasemir, G. S. Guyotte, M. Pearson, EPICS V4 Evaluation for SNS Neutron Data, WEPGF105, these proceedings, ICALEPCS 2015, Melbourne Australia.

[5] B. Martins, M. Davidsaver, areaDetector EPICSv4 modules, EPICS Meeting spring 2015, Michigan State, Michigan, U.S.A., http://tinyurl.com/os29bla.

[6] D. Hickin, EPICS V4/ areaDetector Integration, areaDetector, Diamond, D.L.S., Didcot, Oxfordshire, U.K. (2014), http://tinyurl.com/pngcfb3.

[7] A. Arkilic, NSLS-II Data management Framework, EPICS Meeting spring 2015, Michigan State, Michigan, U.S.A., http://tinyurl.com/os29bla

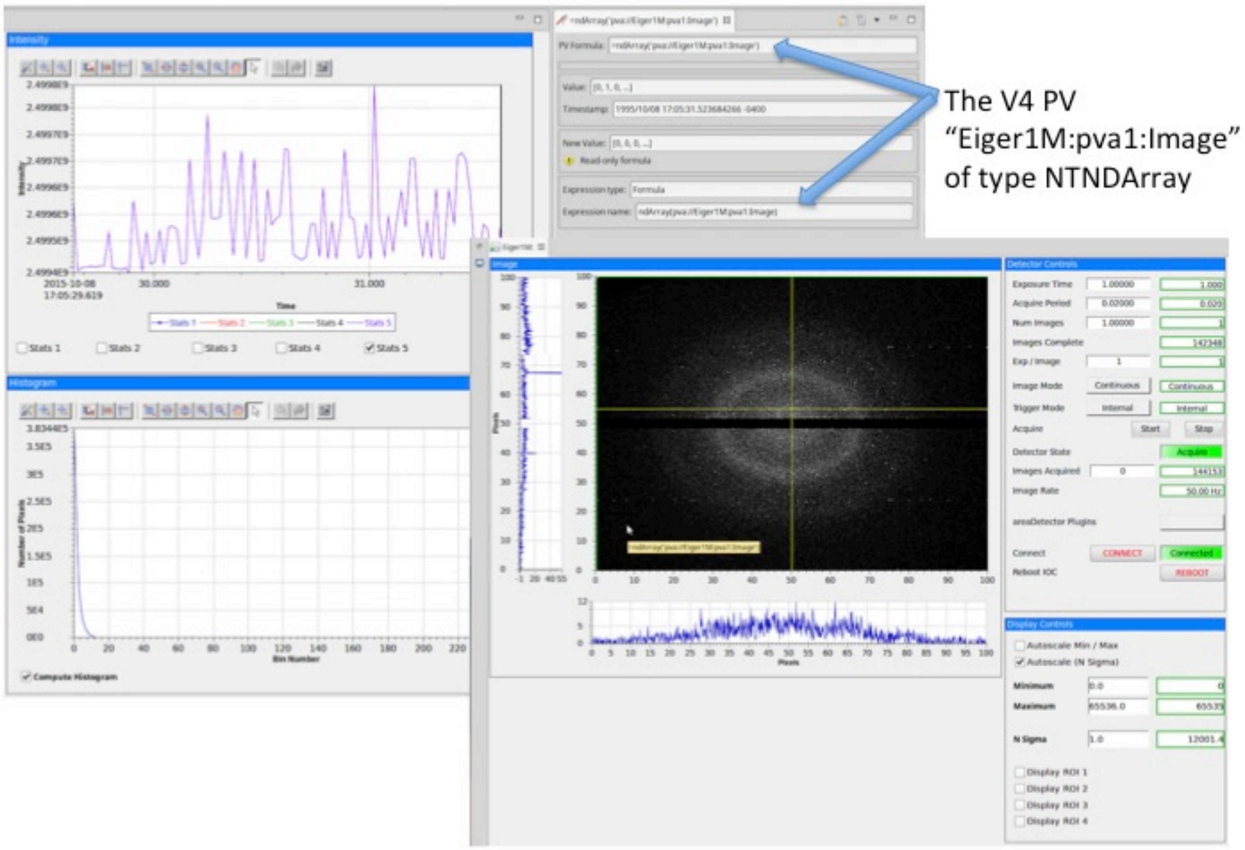


Figure 4: CS-Studio screenshots from NSLS-II, showing an EPICS v4 PV of the Normative Type for areaDetector images (NTNDArray) displayed using a CS-Studio “formula.” *Thanks to Bruno Martins, BNL.*