

NOT DEAD YET: RECENT ENHANCEMENTS AND FUTURE PLANS FOR EPICS VERSION 3*

A.N. Johnson[#], J.B. Anderson, ANL, Argonne, IL 60439, USA
 M. Davidsaver, BNL, Upton, NY 11973, USA
 R. Lange, HZB, 12489 Berlin, Germany

Abstract

The EPICS Version 4 development effort [1] is not planning to replace the current Version 3 IOC Database or its use of the Channel Access network protocol in the near future. Interoperability is a key aim of the V4 development, which is building upon the older IOC implementation. EPICS V3 continues to gain new features and functionality on its Version 3.15 development branch, while the Version 3.14 stable branch has been accumulating minor tweaks, bug fixes, and support for new and updated operating systems. This paper describes the main enhancements provided by recent and upcoming releases of EPICS Version 3 for control system applications.

BACKGROUND

The first EPICS (Experimental Physics and Industrial Control System) toolkit was developed at Los Alamos (LANL) and Argonne (ANL) National Laboratories [2] and made available for other organizations to use in the 1990s. Up until 2004 other users of the software had to sign a license agreement with LANL to obtain a copy, even though many people had by then made significant contributions to the code-base. Today most EPICS components can be freely downloaded via the Internet, and the core software is maintained by the community and distributed under an open source license [3].

This paper concentrates on the core EPICS software, a package referred to as EPICS Base, which contains the Input Output Controller (IOC) implementation and its 29 standard record types, the Channel Access (CA) server and client libraries with a set of command-line CA client programs, the EPICS build system, and various support libraries.

Version Control Branches

The EPICS Base source code is managed using the Bazaar version control system [4] to record and publish the code history of the two main branches. New features and other major developments are added on the 3.15 developer branch, while most commits to the 3.14 stable are bug fixes and changes needed to build for new or upgraded operating system versions.

EPICS Version 4 modules have been developed against the 3.14 branch of Base to date, but will be switching to the 3.15 branch after the release of version 4.3.0 in order

to take advantage of several new features that only appear in Base 3.15.

RECENT DEVELOPMENT

Both branches of Base are now rebuilt after every code commit using a Jenkins Continuous Integration [5] server at the APS to compile the code for several different target operating systems. This allows the developers to quickly discover problems with building their latest changes on architectures that they may not normally build or even have access to. Any build failure is immediately sent out by email to the core developers' mailing list, and the output from the build process is publicly accessible [6] from the Jenkins server. The build for the 3.15 branch also runs the built-in self-test programs on the build host, providing additional confidence in the state of that code at all times.

Release History

Table 1 below lists the versions of EPICS Base that are planned or have been released in the last two years.

Table 1: Recent and Planned Base Versions.

| Version | Release Date | Description |
|-----------|--------------|--------------------------|
| 3.14.12.1 | 2011-04-26 | Stable, bug fixes |
| 3.14.12.2 | 2011-12-12 | Stable, bug fixes |
| 3.15.0.1 | 2012-08-01 | Developer, major updates |
| 3.14.12.3 | 2012-12-17 | Stable, bug fixes |
| 3.14.12.4 | Unreleased | Stable, bug fixes |
| 3.15.0.2 | Unreleased | Developer, new features |

3.14 BRANCH

Version 3.14.12.1 fixed a significant bug in the array handling code of the CA server that was introduced with the dynamic array support. A few other bugs were also fixed, largely related to builds for Windows and RTEMS, and the build configuration files for three new VxWorks targets were added.

Version 3.14.12.2 fixed many minor bugs, and updated the build configuration for Apple iOS targets to be able to build universal binaries on new versions of iOS. The IOC now performs sanity checks at start-up to ensure that the menu definitions loaded meet its required standards.

Version 3.14.12.3 fixed several minor bugs and increased the stack space allocated to threads on both Windows and Posix targets. CALC expressions may now

*Work supported by U.S. Department of Energy, Office of Science, under Contract No. DE-AC02-06CH11357.

[#]Andrew Johnson <anj@aps.anl.gov>

use hexadecimal literals (these worked on some target architectures before, but not all), and the CA server library now supports sending DBE_PROPERTY monitor update events. This release also comes with build configuration files to cross-build binaries for Linux running on a Xilinx Microblaze CPU, and to allow an ARM CPU running Linux to be used as a build host.

Version 3.14.12.4 has not been released yet, but will contain more bug fixes, support for VxWorks 6.9, a high-resolution time provider for MacOS, and a Microsoft Windows implementation of the `epicsLoadLibrary()` API.

3.15 BRANCH

The 3.15 developer branch was created in November 2010, immediately after the release of Base 3.14.12. The Bazaar version control software makes it easy to incorporate bug fixes and other changes into the 3.15 code after they have been committed to the 3.14 branch; this is a manual operation that is typically done every few months and usually only takes a few minutes to complete.

Early in the life of the branch, the source tree was reorganized to reduce the number of libraries created and increase the parallelism available to the GNU make program at build time. The new layout arranges the source tree as follows:

- `src/tools`: Build system scripts
- `src/libCom`: Utility routines and OS-independent API
- `src/template`: User application templates
- `src/ca/client`: CA client library and tools
- `src/ca/legacy`: Portable CA server
- `src/ioc`: Core database processing functions
- `src/std`: Standard record types, soft device support and the `softIoc`

The build rules were also modified to further increase the parallelism by separating out the rules that generate dependency files. On Symmetric Multi-Processor (SMP) computer systems the GNU make program can spread the build over many processors and achieve significant speed-ups; some developers see serial builds that take 2 minutes complete in less than 20 seconds using parallel builds. Achieving this level of parallelism does require accurate dependency information in the build instructions though.

EPICS Codeathons

Several of the enhancements in the development branch were developed at EPICS Codeathon meetings [7] held in 2008 at Argonne, 2009 at Brookhaven, 2010 at Diamond and 2011 at Lawrence Berkeley Lab. These meetings are designed to encourage code development on EPICS projects, providing an atmosphere where everyone is working on EPICS code or writing documentation. By taking developers out of their home environment they can be freed up from interruptions or local problems and allowed to concentrate on specific EPICS development tasks. The development work described below is limited to projects that provided code for the 3.15 branch of Base.

One major project from the 2008 Codeathon was to rewrite the IOC's put-notify functionality to make it more general; this allowed Asynchronous Soft Channel device support to be written for input record types that trigger processing of another chain of records and wait for them to complete before reading the value through the input link. The 2008 meeting also saw the implementation of support for Channel Access over TCP alone, thus permitting CA connections to be tunnelled through a Secure Shell (ssh) network session.

During the 2009 Codeathon the open source Yet Another JSON Library [8] was imported into Base to provide a standard JavaScript Object Notation (JSON) [9] parser for EPICS software, and work started on the server-side filtering subsystem that uses JSON for specifying filters in CA channel names [10]. The 2009 meeting also began the development of alarm severity filtering for input record types, which was completed during the 2010 Codeathon.

The 2010 Codeathon saw the conversion of the EVNT field from an 8-bit integer to a string, allowing IOC soft events to be named instead of just numbered. Other developments cleaned up the values of attributes of secondary fields and added PCI-bus support to the EPICS hardware interface library `devLib`.

In 2011 the Codeathon designed and implemented a platform-independent API for atomic operations on SMP architectures, which is needed to optimize EPICS running on such systems. This meeting also marked the start of a thread-pool project to provide a general-purpose multi-threaded work queue subsystem.

Subsequent Development

A major rewrite of the DBD file processing tools into the scripting language Perl was merged several years after it had begun. The intention was to make changes and extensions to the DBD file syntax and semantics easier to incorporate into future releases.

Some work overhauling the internal APIs for numeric conversion resulted in the ability to specify scan rates using units other than seconds; scan menu choices can now specify any of these units after a number: Hertz or Hz, seconds, minutes, or hours.

The Macro Substitution and Include program `msi` has been enhanced and included with Base. Both this new version of `msi` and the IOC's `dbLoadTemplates` command now support setting global macros in substitution files, and `dbLoadTemplates` can now take a list of global macro settings as the second argument on its command line.

The server-side filtering subsystem was finally merged after a long sporadic development period. This lets any CA client program ask an IOC server to modify the data stream delivered to it, without affecting any other clients connected to the same process variable. Four standard filters are provided as standard, but additional filters can be written and installed without making any modifications to the common IOC software.

The OS-independent thread API provided by Base was extended to allow applications to register a routine that

will be called once by every new thread when it first starts up. This allows sites to implement SMP features like CPU affinity without modifying the core software. On Linux systems the EPICS thread name is also set as the Linux light-weight process name at that time.

Version 3.15.0.1 was released at this point, the first one issued from the development branch. The zero minor version number indicated that the code in this release was not ready for production use in operational control systems; the first fully tested production-quality release from this branch will be given the version number 3.15.1.

Since the 3.15.0.1 release various record types have had enhancements made to them and anachronistic behaviours removed. The fanout and sequence records now both have 16 links instead of their original 6 and 10 links, respectively. Both the multi-bit binary direct record types now support non-contiguous bit-masks and behave sensibly when the OMSL field is changed.

The aai, aao, and waveform record types now support replacing their array buffer pointer instead of always having to copy data into a single fixed buffer, which offers the potential for better performance from IOCs using large arrays.

A new field named UDFS, which is common to all record types, was added to set the alarm severity of a record in an undefined alarm state; this defaults to Invalid to provide backwards compatibility.

Three new record types have been written to better support long string data. The lsi (long string input) and lso (long string output) record types have a VAL field that is a character array; the printf record type does too, but provides the ability to convert data read from up to 10 input links into a printable string. All three record types come with the usual device support, and stdout, stderr, and errlog support for the lso and printf types.

Continuing the work to better support SMP systems, a platform-independent spin-locks API has been added for situations where the overhead of taking and releasing a mutex would be too high. The precise semantics required of the individual platform implementations are still being debated between the developers.

Future Plans

Several development projects are currently at different stages of completion and will get merged into the 3.15 branch after successful review. Releases of new 3.15.0.*n* versions will occur as the core developers complete new features and decide to expose the current state of the code to the wider EPICS community.

The thread-pool and spin-locks features that were described above should be completed soon. Additional work on optimizing EPICS for SMP systems will include the ability to run multiple Callback subsystem threads at the same priority, allowing more callback operations to be run in parallel, so spreading the tasks across the available CPUs more efficiently.

With DBD files now being processed mostly using Perl scripts, an effort has begun to incorporate the reference documentation text for the built-in record types and other

modular code into the actual DBD file that defines them, and to convert this text into HTML documentation at compile time. There are two major aspects to this project, implementing the conversion process in Perl, and then importing the existing record documents into the DBD files. The first part is mostly complete, but the current record reference documentation needs rewriting to bring it up to date, which is likely to take some time to complete.

The IOC database lock-set implementation is likely to undergo changes to allow monitor updates to provide coherent sets of data, allowing values to be fetched simultaneously from multiple records that may be in different lock-sets. This will provide some functionality needed for the EPICS V4 development effort.

Record links are not currently modular, so adding a new kind of link is not an easy task. A desire to rectify this limitation has triggered a number of minor changes to date, and more work along these lines is planned.

CONCLUSIONS

Development of EPICS Version 3 continues alongside the Version 4 development effort, adding new features to the toolkit and extending the capabilities made available to its users, who are the developers of EPICS-based control systems at facilities and companies around the world. Without significant dedicated developer effort progress has often been slow, but the software is a long way from dying [11]. The EPICS collaboration continues to attract new projects and users, most of whom contribute back to the community in one way or another.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of all EPICS core developers and past EPICS Codeathon participants who have worked on the continued development of EPICS Base.

REFERENCES

- [1] T. Korhonen et al., "EPICS Version 4 Progress Report," TUCOCB04, these proceedings.
- [2] L.R. Dalesio et al., "EPICS Architecture," proc. of ICALEPCS'91, pp. 278-282 (1991).
- [3] EPICS Open License; <http://www.aps.anl.gov/epics/license/open.php>
- [4] Bazaar version control system; <http://bazaar-vcs.org/>
- [5] Jenkins continuous integration server software; <http://jenkins-ci.org/>
- [6] EPICS builds at the APS Jenkins server; <https://jenkins.aps.anl.gov/view/EPICS%20Jobs/>
- [7] EPICS Codeathons web-page; <http://www.aps.anl.gov/epics/meetings/codeathon.php>
- [8] Lloyd Hilaiel, Yet Another JSON Library; <http://lloyd.github.io/yajl/>
- [9] JavaScript Object Notation; <http://json.org/>
- [10] A. Johnson and R. Lange, "Evolutionary plans for EPICS version 3," proc. of ICALEPCS 2009, WEA003, p. 364 (2009).
- [11] "Monty Python and the Holy Grail" Scene 2, Python (Monty) Pictures, 1974.