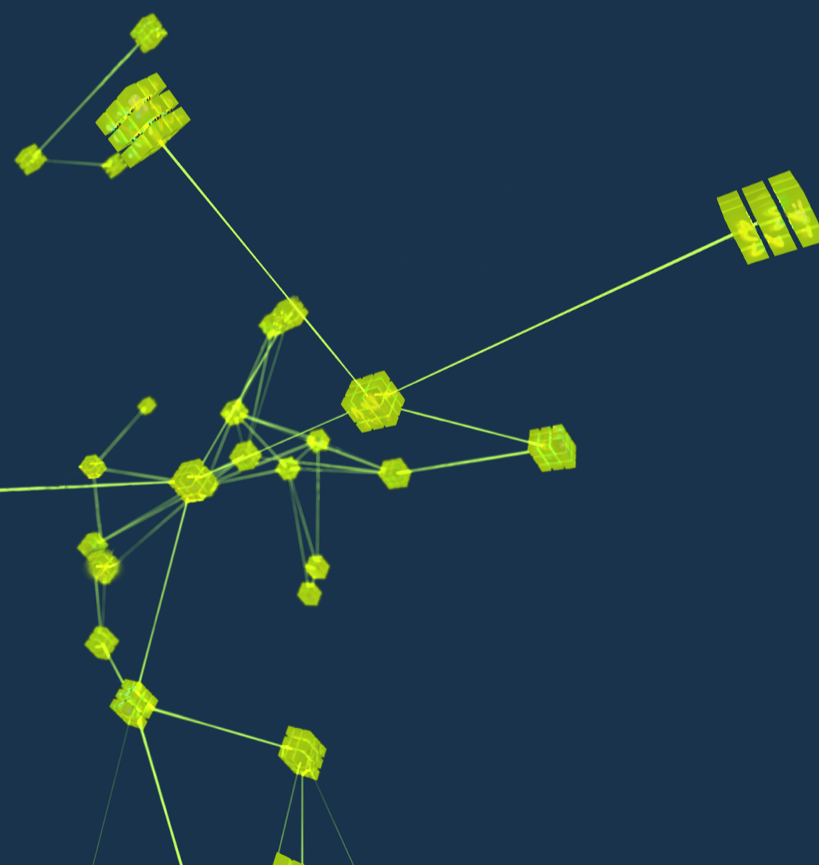


EPICS Collaboration Meeting April 2024

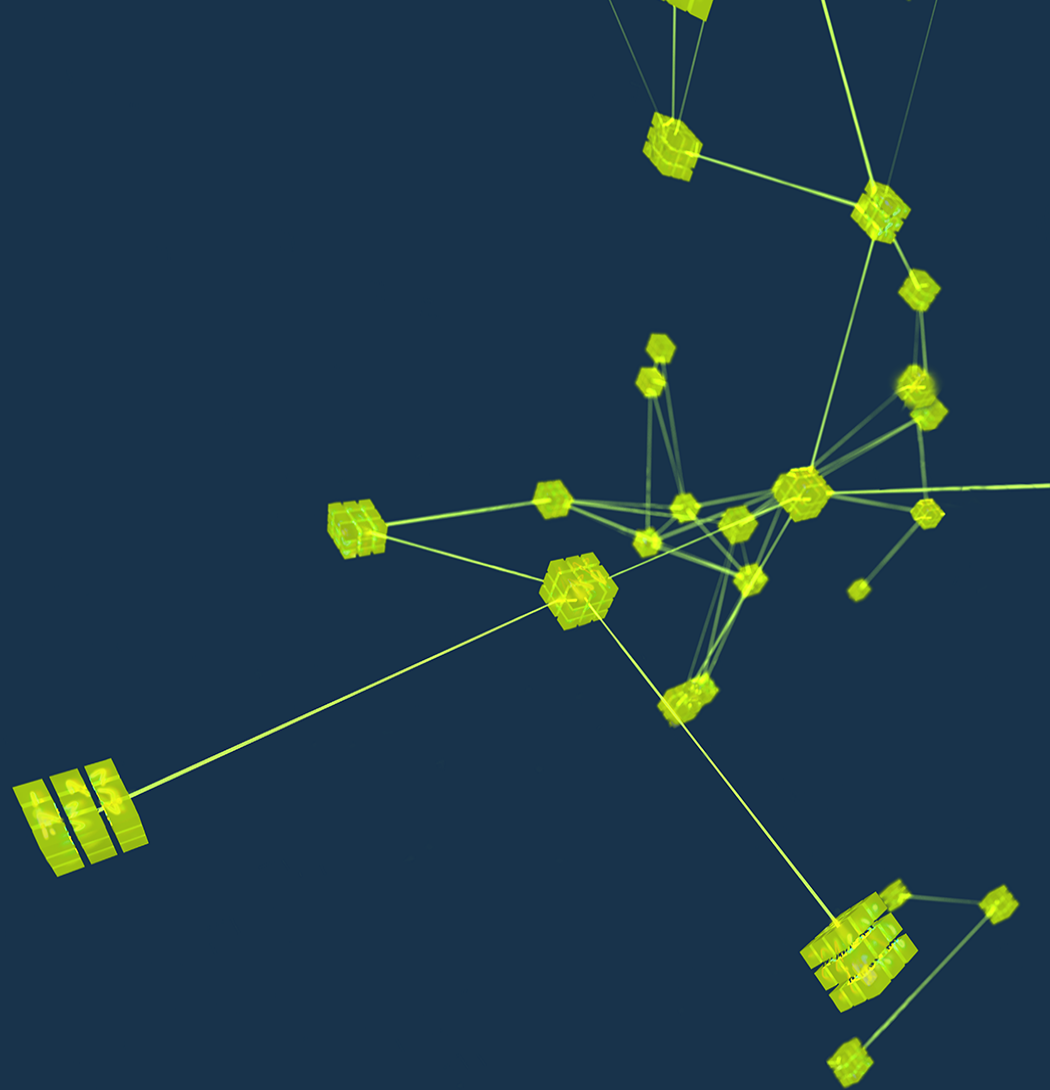


EPICS Core Developments and Plans

Ralph Lange for the EPICS Core Developers

Outline

- Past
 - Changes in Base 7.0.7 (September 2022)
- Present
 - Changes in Base 7.0.8 (December 2023)
- Plans
 - Changes under review or development
 - Future plans and ideas for the IOC
 - Long term plans
- Java
 - Changes in recent releases, under review or development



PAST

FEATURES RELEASED IN EPICS 7.0.7

EPICS Releases since 2021

- EPICS 7.0.6 04 July 2021
- Base 3.15.9 31 August 2021
- EPICS 7.0.6.1 06 October 2021
- EPICS 7.0.7 07 September 2022
- EPICS 7.0.8 15 December 2023

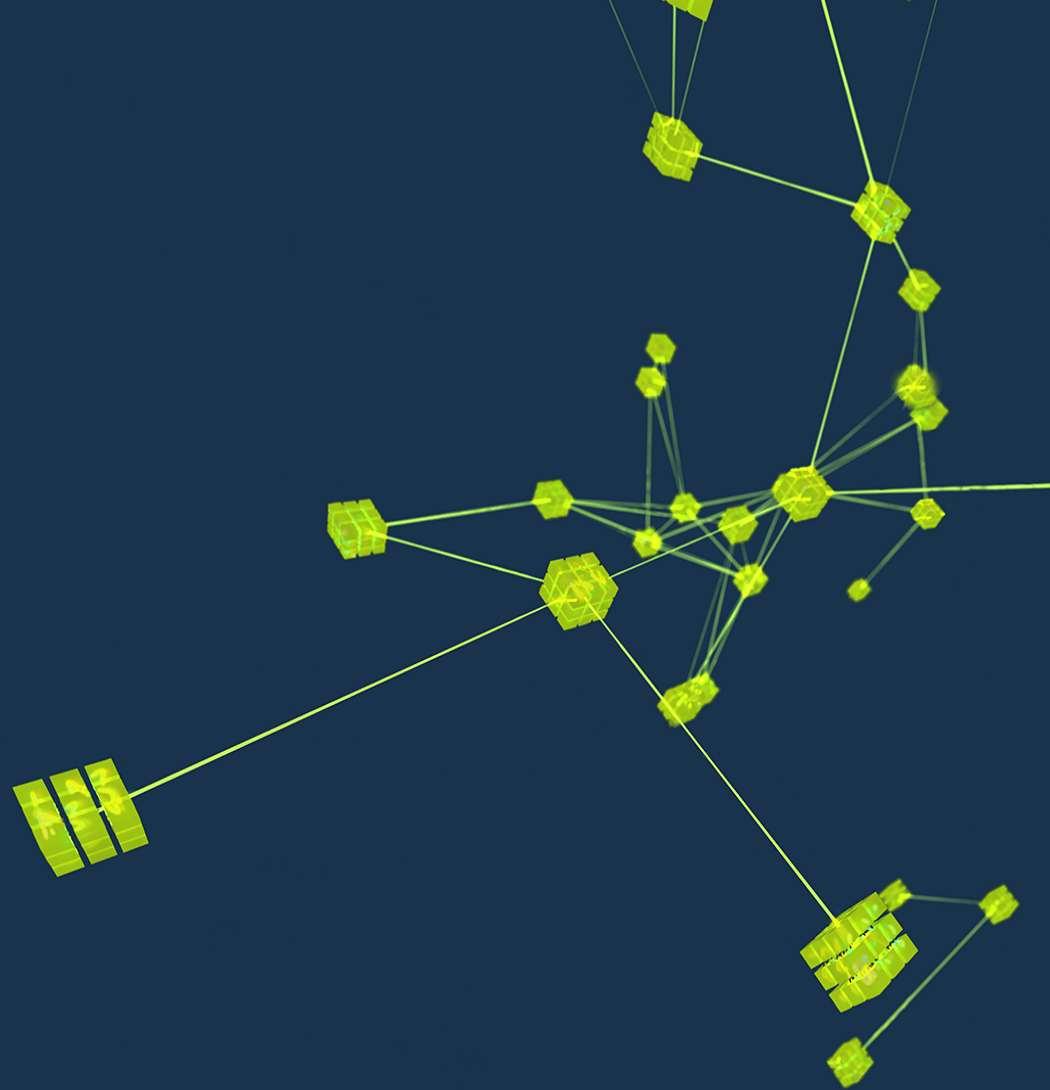
Changes in EPICS 7.0.7

For a full list, see the Release Notes on epics-controls.org

- Made `epicsInt8` signed on all architectures — was previously unsigned where `char` was unsigned (e.g., PowerPC)
- Breaking change to `db_field_log`, affecting IOC channel filters
- Builds with newer compilers can auto-detect GNU Readline
- aao records now have `OMSL` and `DOL`, can copy arrays between records
- IOC's CA server exports `RSRV_SERVER_PORT` (TCP port number)
- Output record types now support RAW Simulation mode (convert `VAL` to `RVAL`, write it through `SIOL`)
- Coloured many IOC warning & error messages, except when logged

More Changes in EPICS 7.0.7

- Accept hex and octal values in hardware links (leading 0 ⇒ octal)
- Fixed time synchronization issues on VxWorks 6.9
- Build system updates to stop a top-level `make uninstall` from triggering the clean target; setting `INSTALL_LOCATION` works again
- pvAccessCPP: bug fixes (enum) and performance improvements
- pva2pva: access security hooks; `timestamp.userTag` from `UTAG` field



PRESENT

FEATURES RELEASED IN EPICS 7.0.8

EPICS
7.0.8

Changes in EPICS 7.0.8

- `pva2pva(QSRV1)`: fix for `dbLoadGroup` (broken in 7.0.7)
- Calc expressions may use the new function `FMOD (NUM, DEN)`
- `iocShell` built with GNU Readline supports tab-completion of commands, filenames, record names, var names and `pdbbase`
- History file (bash-like) for `iocShell`
- ANSI escape sequences are printed on the `stderr` stream (colours!)
- MacOS builds will detect and use GNU Readline from Homebrew or MacPorts if it's available; Apple's `libedit` can't do tab-completion
- Menu fields can hold a value that is not a valid choice. Fetching one as a string now returns the numeric value as a string instead of an error
- The `capr.pl` script has improved significantly since 7.0.6.1

More Changes in EPICS 7.0.8

- The new `OOPT` field in the longout record controls whether the output value gets written to device support, similar to the calcout record
- bi record “Raw Soft” support uses `MASK` field
- Scan and callback threads are stopped on normal IOC shutdown
- Improved sub record behaviour on bad `INP*` links

Changes in EPICS 7.0.8 from the Codeathon 2023

Results of the Codeathon at Diamond in March 2023 have been merged:

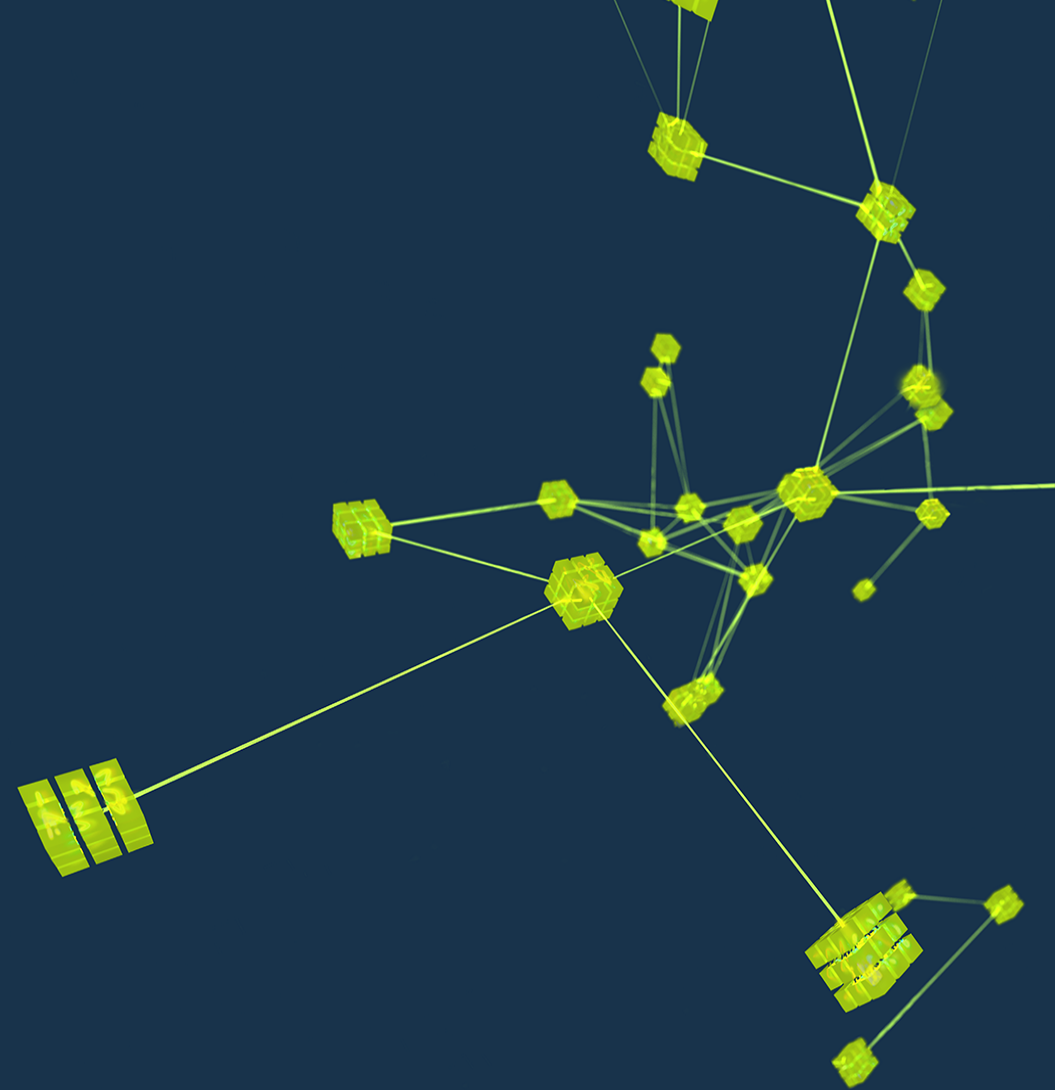
- Setting the new compress record `PBUF` field to `YES` enables partially filled buffers to be processed with the N-to-1 algorithms (introduced a bug that will be fixed shortly)
- A subroutine record with an invalid `INP*` link no longer fails silently
- Many C & C++ compiler and UBSan warnings have been cleaned up
- The code now has a `.editorconfig` file, format checked by CI builds
- The “ts” server-side filter can now fetch a record’s timestamp with a choice of formats. The updated filter documentation has details
- A new `dbCreateAlias` `iocShell` command allows record aliases to be defined (before `iocInit`) without loading a separate `.db` file
- More detailed and better-formatted `iocShell` “help” command output

More Changes in EPICS 7.0.8 from the Codeathon 2023

- Environment variable `EPICS_CLI_TIMEOUT` to set default timeout for catools (`caget` etc.)
- Generate error on any attempt to alias a record.field
- Unit tests created for standard record types (ai, bi, bo, printf)
- Add CI cross-build tests for aarch64 and linux-arm (BE and LE) targets
- More Doxygen annotations for header files in libCom and ca
- New `EPICS_PRINTF_FMT(format_string)` annotation macro to let Microsoft compilers check printf-like declarations
- Support modules may call `errSymbolAdd()` after early initialization
- Fix timestamps on array-record `NORD` field updates
- Additional work on the RTEMS target code, including RTEMS-6

FUTURE

WHAT NEXT?



Under Review or Development

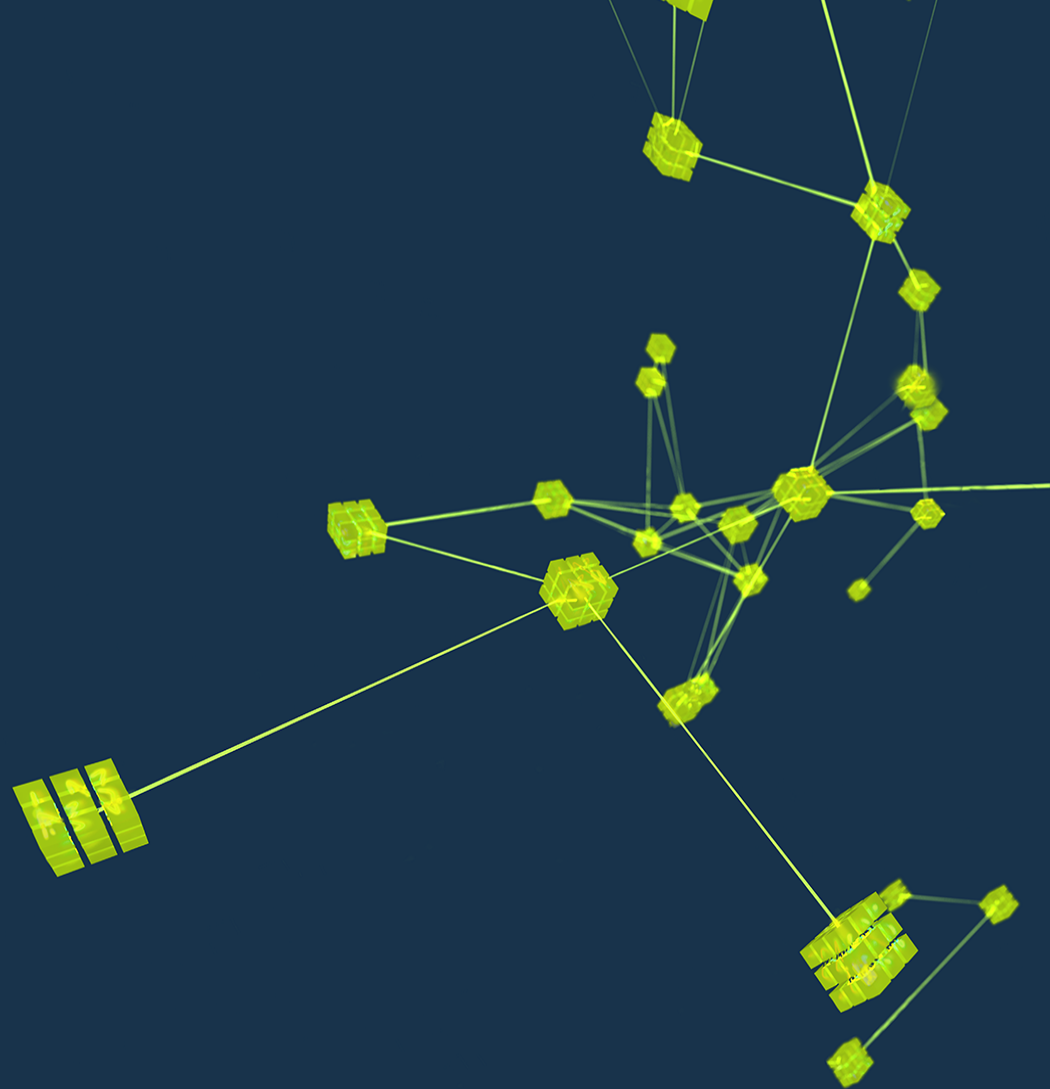
- Server-side filter to read record `info()` item strings
- Explicit `INT` and `EXT` link modifiers to help detect typos in PV names
 - `INT` means link target must be local to the IOC (`EXT` overrides)
 - Additional DB-file syntax can set default to one of these
- Add `OOPT` to more output record types (only write “On Change” etc.)
- Link type that can do file I/O
- Make the next generation user-level API library `PVXS` part of Base
- Rebase `QSRV` (the IOC PVA server) on top of `PVXS`: `QSRV2`

Future Plans and Ideas for the IOC

- Features being considered:
 - Server-side filtering of put operations (semantics tricky)
 - How to support complex structures as database fields
 - JSON link addresses for device support

Long Term Plans

- Next major release – EPICS 7.1
 - No specific timetable for this yet
 - Will require C99 and C++11 compilers (GCC 4.8.1)
 - No support for VxWorks 6.x or RTEMS 4.x
- IPv6 is supported by the next-gen PVA libraries (PVXS, core-pva)
Should Channel Access also get it??



JAVA

EPICS JAVA AND SERVICES DEVELOPMENTS

EPICS
ATLAS

EPICS Core Java and core-pva

epicsCoreJava (old PVA libraries) and **JCA** (CA library) are broadly stable

- Replace internal JCA data structures with standard Java equivalents
- More JCA configuration properties (e.g., socket timeouts)
- Ensure support with newer Java versions (16, 17 LTS, ...)

core-pva (2nd gen PVA library) is stable

- Used in Phoebus for years... core-pva is the default PVA library
- All EPICS middle layer services (alarm, archive appliance, save and restore, ...) have been migrated to core-pva

cfnameserver: Early prototype of a channelfinder-based nameserver for pvAccess

Middle Layer Services

All middle layer services are being standardized to use the same structure/solution for Authentication/Authorization (AA)

ChannelFinder Directory Service

- Expanded testing and code coverage (CI / Sonar)
- PR: Migration to JDK 17 (maintaining compatibility)

Phoebus Olog

- Support for non-destructive edit / edit history
- Expanding testing and code coverage (CI / Sonar)
- PR: Migration to JDK 17 (maintaining compatibility)

Middle Layer Services

Save and Restore

- Migrated to use core-pva
- New release with an expanded REST API

Alarm Services

- Migrated to use core-pva
- WIP: Expanding the annunciator, possibly making it a standalone service for better integration with various notification interfaces
- WIP: Upgrading Kafka 2.2 to 3.x – aiming at better catching and reporting errors to improve the reliability of Alarm Services

Middle Layer Services

Archive Appliance

- Moved to a new dedicated GitHub organization
<https://github.com/archiver-appliance>
- Migrated to use core-pva
- New build system (Gradle) simplifies updating dependencies – potentially migrating to Maven at the next Codeathon (June 24)
- Use Google protobuf 3
(with protobuf 2 syntax to ensure compatibility)

Thank You for Your Attention!

Comments or Questions?

