

EPICS Base Status and Plans

Ralph Lange
for the EPICS Core Developers

Available EPICS Base release series

- **3.14: maintenance**
 - Used in many production systems
 - Gets bugfixes and security patches
 - Intended for continued support and maintenance of legacy systems
- **3.15: stable**
 - Used in many production systems
 - Gets patches plus carefully tested, compatible improvements
 - Intended for minor developments that are aimed to go in production
- **3.16: current**
 - Gets patches plus re-implementations, backward compatible improvements and new features
 - Some new things may still change
 - Intended for all new developments

New¹ in the 3.14 (maintenance) series

- Dynamic (variable length) array support
- Fixes and optimizations for Windows targets
 - Whole program optimization for Microsoft compilers
 - MinGW cross builds from Linux
 - Better Cygwin support
- Support for hex literals in string parsing
 - E.g. `field(ZRVL, "0xE4FFC")`
- Support for `DBE_PROPERTY` event type
- Lots of bug fixes

¹ since 2011

New¹ in the 3.15 (stable) series

All of the above, plus

- Device support for “getenv”
- Base builds set build info in environment variables
- Improved re-implementation of `promptgroup/gui_group`
- IOC server (RSRV) can bind to subset of available interfaces
- IPv4 multicast for name resolution and beacons
- Assembling files from numbered snippets
 - E.g. `st.cmd` can be created from application snippets
- Max length for INP/OUT raised to 255 characters
- Improved performance when loading large databases

¹ since 2015

New in the 3.16 (current) series

All of the above, plus

- IOC database support for 64bit integers
- Link support (extensible link types) *see next slide*
- Support for valgrind instrumentation
- Database locking re-implemented (`dbScanLockMany()`)
- compress Record buffering order selectable: FIFO or LIFO
- Removed support for CapFast and dbst tool

Link Support in 3.16 (structured link types)

- JSON syntax to define structured link types
- `calc` link type can use CALC expressions to generate link
- `const` link type allows e.g. strings and arrays for initialization

```
field(INP, {calc:{expr:"A+B+1",  
                args:[5,{const:6}]  }})
```

```
record(stringin, "const:string") {  
  field(INP, ["Not-a-PV-name"])  
}  
record(waveform, "const:longs") {  
  field(FTVL, LONG)  
  field(NELM, 10)  
  field(INP, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])  
}
```

New in EPICS V4 (4.6.0)

- pvData / pvAccess implementations for both C++ and Java are complete
- pvaPy Python APIs are complete
- Improved documentation
- All command line tools support Channel Access and pvAccess
- All example code moved into a single example module

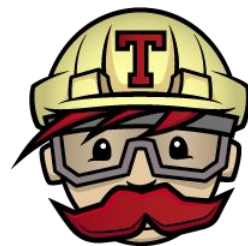
Build and test infrastructure

- Number of unit tests in EPICS Base is increasing:
3.14: **2.6k** → 3.15: **8.5k** → 3.16: **9.2k**
- Jenkins instance on APS (<https://jenkins.aps.anl.gov/>)
 - C++ builds
 - Only master branch
 - Host builds: Linux, MacOS, Solaris, Windows
 - Cross builds to many targets
- Jenkins instance on CloudBees (<https://openepics.ci.cloudbees.com/>)
 - C++ and Java
 - Linux host
 - Release builds
 - Pull request builds



Build and test infrastructure

- Travis (<https://travis-ci.org/epics-base/epics-base>)
 - Host builds on Linux (gcc, clang), WINE
 - Cross builds for RTEMS, Windows
- Appveyor (<https://ci.appveyor.com/project/epics-base/epics-base>)
 - Host builds on Windows
 - Compilers: VS 2008, 2010, 2012, 2013, 2015, Cygwin, MinGW
 - 32bit or 64bit architecture
 - DLL or static build
 - Debug or optimized build
 - Full matrix: 56 builds, taking ~6.5 hours



The big merge – EPICS 7

- Existing Base (V3) sources will be modularized (split)
 - core build system, libCom, tools
 - database db access and processing, record types, device support, filters
 - ca channel access client and server
- Main V4 modules will be added
 - pvData structured data, helper libraries
 - pvAccess network protocol for pvData, client and server
 - normativeTypes standard containers
 - pvaClient simplified user-level libraries
 - Examples example code
- First release of EPICS 7 scheduled for October 2017

Source code repositories

- Base sources on Launchpad have been converted from Bazaar to Git
- Existing V3 code will continue to stay in Git@Launchpad
 - We rely on Launchpad's issue tracker that provides unique features
- Existing V4 and new modules will stay in Git@GitHub
 - Issue referencing for bugs will be across sites
- Git submodule feature will pull modules together for build

Conclusion

- V3 branches are in good shape, maintenance and update processes are clear
- EPICS 7 merge is a major organizational effort, but only a minor technical risk
- In a growing collaboration, the core development team is getting smaller
- We need help, especially for reviewing and testing

Please contribute!