

# PVXS in your IOC

Michael Davidsaver  
George McIntyre

<https://mdavidsaver.github.io/pvxs/>

# What is PVXS?

*PV access in eXcesS ?*

- Library providing PV Access network client and server APIs
  - Like pvDataCPP+pvAccessCPP
  - **Inter-operates** with ...
    - And other PVA clients/servers
  - Does **not** depend on or use ...
  - Does not conflict with ...

<https://mdavidsaver.github.io/pvxs/>

# Release History

- 1.2.0 ??? ???? ↗
- 1.1.4 Apr 2023
- 1.1.3 Mar 2023
- 1.1.2 Feb 2023
- 1.1.1 Dec 2022
- 1.1.0 Nov 2022
- 1.0.1 Oct 2022
- 1.0.0 Sep 2022
- 0.3.1 Jun 2022 ↖
- 0.3.0 May 2022 ↖
- 0.2.2 Jan 2022
- 0.2.1 Oct 2021
- 0.2.0 Jul 2021
- 0.1.5 May 2021
- 0.1.4 Apr 2021
- 0.1.3 Feb 2021
- 0.1.2 Feb 2021
- 0.1.1 Jan 2021
- 0.1.0 Dec 2020 ↘

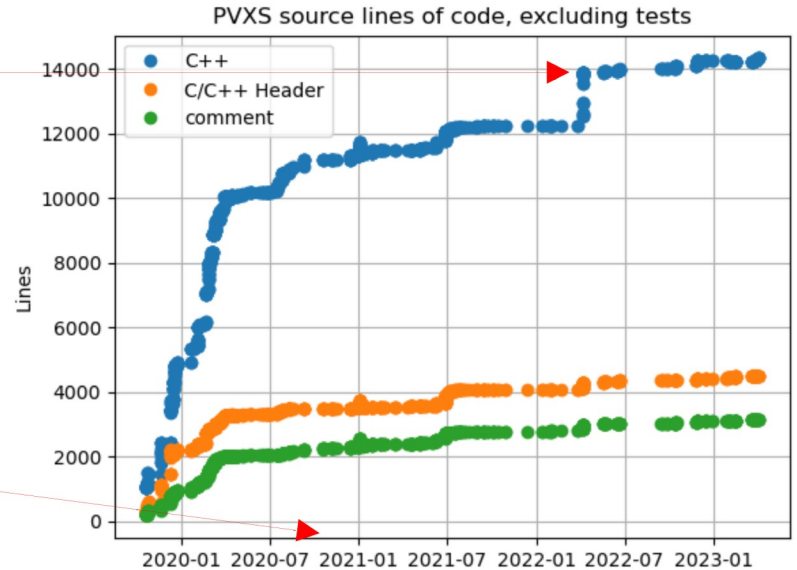
IOC Integration

18 point releases  
13 without (known) ABI change

IPv6 Support

*P4P 4.0.0 (pvagw + python)*

Last presentation...



<https://mdavidsaver.github.io/pvxs/>



# Goals / Choices

- Standalone impl. of PVA server and client
- Robust
  - Clear threading/locking
  - Less error prone API
- C++11 required
  - GCC  $\geq 4.8$ 
    - Yes, RHEL7
    - Also clang, msvc
- Use libevent 2.x
  - <http://libevent.org/>
- Base  $\geq 3.15$ 
  - Still works w/ 3.14\*

# API Design goals

- End user API is first class citizen
- Safety
  - Avoid possible \*NULL
  - API enforce required ordering
  - Clear lifetime wrt. cancellation
    - Reference loops still possible w/ callbacks
- Synchronous (blocking) and Asynchronous (callbacks)
- No global ctor/dtor

# Ex: Sync. Client GET

```
#include <iostream>
```

```
#include <pvxs/client.h>
```

Uses `$EPICS_PVA_*`

```
auto ctxt(pvxs::client::Config::fromEnv().build());
```

```
auto result(ctxt.get("pv:name")
```

```
    .exec()
```

```
    ->wait(5.0);
```

```
// wait() throws on timeout
```

```
std::cout<<result["value"];
```

# IPv4/6 configuration

<https://mdavidsaver.github.io/pvxs/netconfig.html>

- EPICS\_PVA\_ADDR\_LIST
- EPICS\_PVAS\_INTF\_ADDR\_LIST
- Address forms
  - 10.1.1.1:5076
  - [2600:1234::42]
  - 224.0.2.3,255@192.168.1.1
  - [ff02::42:1]@br0
  - [ff02::42:1],1@br0

```
<ip4-or-host>[:<port#>][,TTL#][@ifacename]  
"["<ip6-or-host>"][:<port#>][,TTL#][@ifacename]
```

<https://mdavidsaver.github.io/pvxs/>

# ~~Going Forward~~ (1)

Looking backwards

- Initial Development
  - Q4 2019
- Public Beta
  - Q1 2020 .....▶ Announce on Tech-talk list
- Pre-production 0.X
  - **Q4 2020?** .....▶ Begin release notes. eg. on API changes  
*Dec. 2020*
- Stable 1.X
  - **2021?** .....▶ 1.X API “freeze”  
*Sept 2022*  
*aka. Incompatible change → 2.X*

<https://mdavidsaver.github.io/pvxs/>



# Going Forward (2)

Looking backwards

pvDataCPP/pvAccessCPP

PVXS

- Staged deprecation
  - End of feature development
    - Q1 2020
  - Critical fixes only
    - 2021?
  - Removal (from Base releases)
    - ~~2022?~~
- v1.0 ?
- Expand use of PVXS
  - Gateway ✓
  - IOC integration (aka. QSRV 2)
    - **Server** (coming soon)
    - PVA Links (pending?)
  - CLI tools
  - Language bindings
    - P4P (python) ✓*

<https://mdavidsaver.github.io/pvxs/>



# QSRV 2

- Adding IOC database integration to libpvxsloc.so
  - **Alpha** status. **Testable**, missing features
  - Testers wanted! <https://github.com/mdavidsaver/pvxs/pull/37>
- Status
  - Single + Group PV working
  - No PVA Links (yet)

*Opt-in at runtime*  
export PVXS\_QSRV\_ENABLE=YES

Work by  
George McIntyre

<https://mdavidsaver.github.io/pvxs/>

<https://mdavidsaver.github.io/pvxs/>



# Test Coverage (gcov)

## GCC Code Coverage Report

By PVXS unittest suite

Directory: <code>../..</code>	Exec	Total	Coverage
Date: <b>2020-10-20 12:42:34</b>	Lines: 5187	7351	70.6 %
Legend: low: < 75.0 % medium: >= 75.0 % high: >= 90.0 %	Branches: 4351	11605	37.5 %

Directory: <code>./</code>	Exec	Total	Coverage
Date: <b>2023-04-14 10:25:36</b>	Lines: 6710	9057	74.1 %
Legend: low: < 75.0 % medium: >= 75.0 % high: >= 90.0 %	Branches: 5751	14516	39.6 %