

# EPICS 7 Workshop Summary

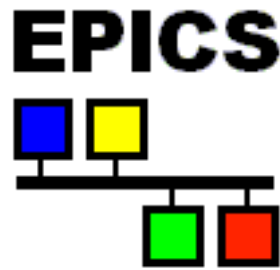
EPICS Collaboration Meeting  
Slovenia, Sept. 2022

Kay Kasemir

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

August 10 “Would you .. PV Access, PVXS, Java Interface, sample client”

Sept. 19 EPICS 7 Workshop



Network Diagram



Network Diagram (new)

“Integrate into EPICS” = “Make accessible on Network”

**EPICS 7** = EPICS with CA and **PVA**

For material,

follow EPICS Meeting web page to

[https://controlssoftware.sns.ornl.gov/training/2022\\_EPICS/](https://controlssoftware.sns.ornl.gov/training/2022_EPICS/)

# EPICS IOCs are now Bi-Lingual!

## Channel Access

- cainfo, caget, caput, camonitor
- **Still fully supported**

## PV Access

- pvinfo, pvget, pvput, pvmonitor
- CS-Studio: 'pva://' prefix
- **Looks very similar**

PV Tree x

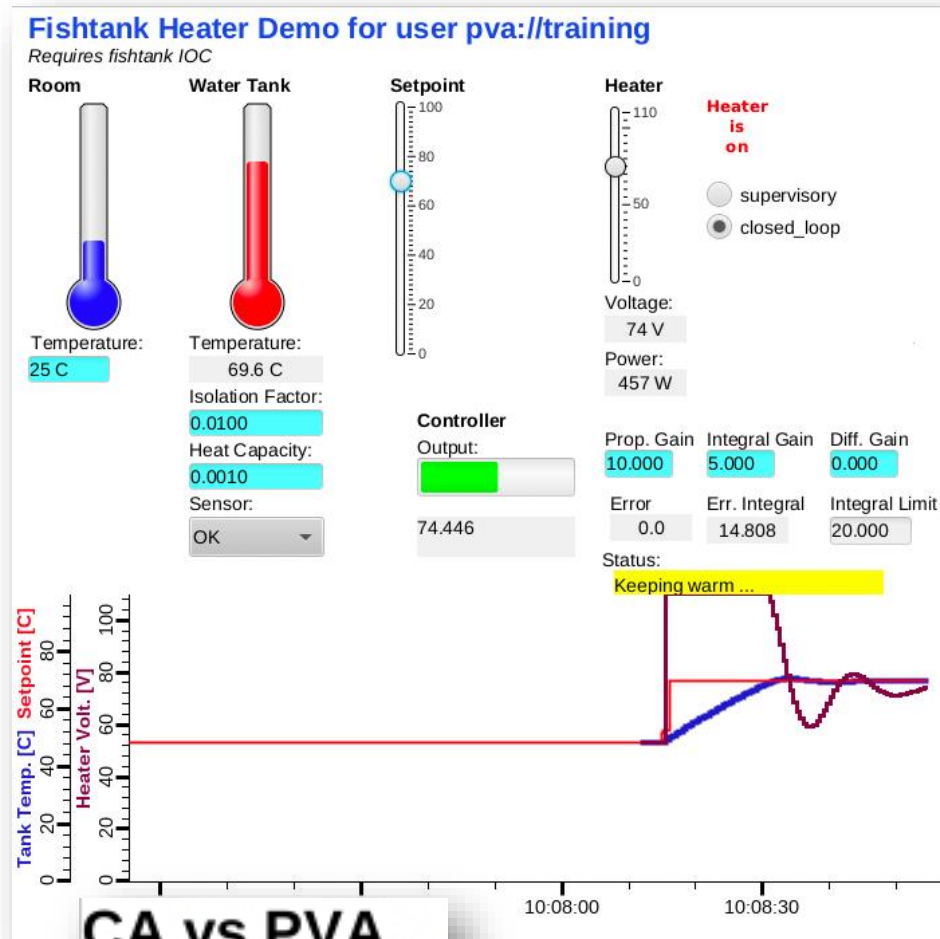
PV: training:tank

- PV 'training:tank' (calc) = 30.0 C
- INPA 'training:sensor' (bi) = OK
- INPB 'training:tank\_clc' (calc) = 30.0000
  - INPA 'training:tank\_clc.VAL' (calc) = 30.0000
    - INPA 'training:tank\_clc.VAL' (calc) = 30.0000
    - INPB 'training:room' (ao) = 25 C
    - INPC 'training:heat\_Pwr' (calc) = 50 W
    - INPD '0.01' [const]
    - INPE '0.001' [const]
  - INPB 'training:room' (ao) = 25 C

PV Tree x

PV: pva://training:tank

- PV 'pva://training:tank' (calc) = 30.0 C
- INPA 'training:sensor' (bi) = OK
- INPB 'training:tank\_clc' (calc) = 30.0000
  - INPA 'training:tank\_clc.VAL' (calc) = 30.0000
    - INPA 'training:tank\_clc.VAL' (calc) = 30.0000
    - INPB 'training:room' (ao) = 25 C
    - INPC 'training:heat\_Pwr' (calc) = 50 W
    - INPD '0.01' [const]
    - INPE '0.001' [const]
  - INPB 'training:room' (ao) = 25 C



## CA vs PVA

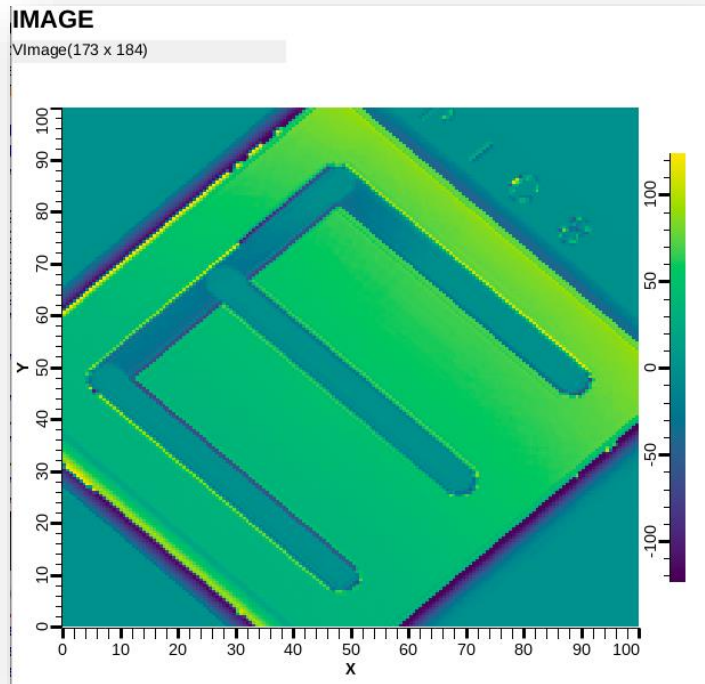
Fishtank

Fishtank (ca://...)

Fishtank (pva://...)

# Advantage PVA: Images

- “NTNDArray” type combines
  - Pixels
  - Color modes
  - Dimensions
  - Compression settings



File Browser Area Detector X simDetector X

### Area Detector Demo

Detector Plugins

1) cd /home/training/epics-train/examples/AreaDetector  
.start\_sim\_ioc.sh  
2) Open Detector page  
3) Press Connect.. "Start"  
4) Open Plugins.. "All" and "Enable" the NDPluginStdArrays

Start Stop Enable

Y

X

Images: 127104 120.00 Hz

Display adapts when image size and data type change

### Simulation Detector

Setup

asyn port SIM1  
EPICS name 135IM1.cam1  
Manufacturer: Simulated detector  
Model: Basic simulator  
Serial number: No serial number  
Firmware version: No firmware  
SDK version: 2.8.0  
Driver version: 2.8.0  
ADCore version: 3.3.2

Connected

Connection: Connect Disconnect

### Plugins

All File Menu ROI Menu

### Readout

	X	Y
Sensor size	1024	1024
Binning	1	1
Region start	0	0
Region size	1024	1024
Reverse	No	No
Image size	1024	1024
Image size (bytes)	2097152	
Gain	1.000	1.000
Data type	Int16	Int16



# Custom Data

- Create from IOC, python, java, C++
- Use in python, java, CS-Studio, C++

```
pva = Context('pva')

def check(value):
    x = value['x']['value']
    y = value['y']['value']
    print("%.4f %.4f -> %.8f" % (x, y, sqrt(x*x + y*y)))

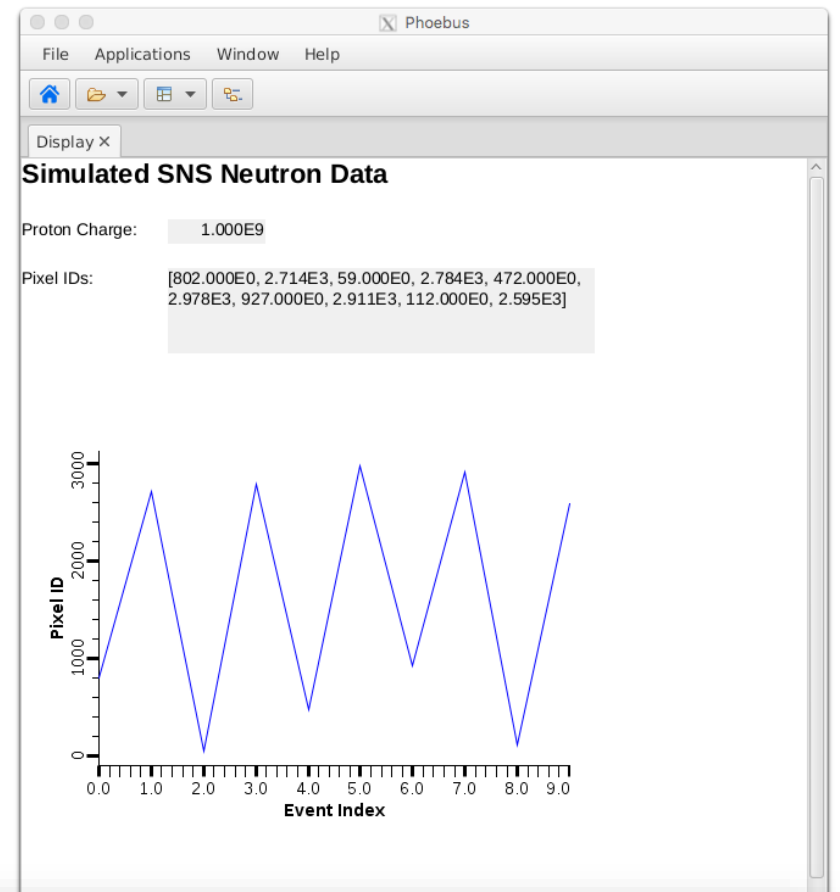
with pva.monitor('training:circle', check):
    sleep(1000)
```

```
training@VB:/ics/examples/24_pvaccess$ pvinfo neutrons
neutrons
Server: 10.0.2.15:5075
Type:
structure
  time_t timeStamp
    long secondsPastEpoch
    int nanoseconds
    int userTag
  epics:nt/NTScalarArray:1.0 time_of_flight
    uint[] value
  epics:nt/NTScalarArray:1.0 pixel
    uint[] value
  epics:nt/NTScalar:1.0 proton_charge
    double value
```

```
try
{
    PVAServer server = new PVAServer()
}

// Construct a data structure with value and time, claim it's an NTScalar, with some extra stuff
PVADouble value = new PVADouble("value", 3.14);
PVATimeStamp time = new PVATimeStamp("timeStamp");
PVAStructure data = new PVAStructure("data", "epics:nt/NTScalar:1.0",
    value,
    time,
    new PVAStrng("owner", "Fred"),
    new PVAStrng("color", "pale enticing turquoise"));

// Create PV
ServerPV pv = server.createPV("demo", data);
```



# State of PV Access, Sept. 2022

## Done, operational

- Server and client libraries for C++, Java, Python
- Area Detector image transfer
  - Used to distribute processing from camera hosts
- Custom data servers and clients
  - SNS: neutron data
  - APS: services

## Done, to be tested

- PVA server for records in IOC
  - All record types
  - 'Description'
  - Full 'units'
  - Full 64 bits for 'int64in', 'int64out'
  - No enum state limit
  - Supports changing metadata
- CS-Studio client
- Gateway

## To do

- IOC links
  - Default to CA.  
  
Initial support for  
`field(INP, {pva:{pv:"tgt"}})`
  - Channel Access Get/put callback → ??
- How to best combine data from records into custom PVA data?

# Summary: PV Access is ..

- Alternative to Channel Access
  - Both can be used in parallel
- Similar, but supports custom data types
  - Already useful for images and site-specific cases
- Since EPICS 7 included in base IOC