EPICS 7

Kay Kasemir, June 2019

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

EPICS 7 =

EPICS 'base'

Records,
Device Support,
Channel Access,
softIoc

Almost like R3.13 from 1994

+

EPICS 'V4'

PVAccess,
softIocPVA

Started ~2010

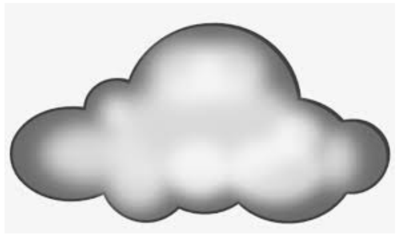
Available since Dec. 2017

EPICS 7 ?

Revolutionizes EPICS

Everything is
- Easier
- Faster
- More colorful
- Service-oriented

Kills EPICS as you know it
- Services replace IOCs
- Channel Access clients no longer connect
- Breaks your device support
- Needs more CPU & Memory



Now What?

1. Use EPICS as before
   - No need to change anything
   - Look at 'RELEASE.local' mechanism
   - Support for 64bit numbers, SMP, locking tweaks
2. Start to use PV Access
   - Images
   - Custom structures
3. Transition everything to PV Access
   - .. Once there's a PVA Gateway, access security, everything "works"

EPICS Base

- Records, device support, databases, sequences, … as before
- Modules encouraged to use RELEASE.local

```
XXX/configure/RELEASE
ASYN=/path/to/asyn1-2-3
EPICS_BASE=…
```

```
YYY/configure/RELEASE
AUTOSAVE=/path/to/auto1-2-3
EPICS_BASE=…
```

```
XXX/configure/RELEASE
-include $(TOP)/../RELEASE.local
```

```
YYY/configure/RELEASE
-include $(TOP)/../RELEASE.local
```

```
RELEASE.local
ASYN=/path/to/asyn1-2-3
AUTOSAVE=/path/to/auto1-2-3
…
EPICS_BASE=…
```

RELEASE.local in Training Setup

```
cd ~/epics-train/tools

ls –d base* seq* asyn*

cat RELEASE.local

cat seq*/configure/RELEASE
```

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

PV Access

- How does it differ from Channel Access?
- How do I…
  - … add PVA to an IOC?
  - … 'caget' with PVA?
  - … use PVA in UI tools (Operator Displays, ..)?
- Area Detector: Any advantage using PVA?
- Custom Data: Any advantage using PVA?
- Some Python Examples

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

History

Channel Access

- Since beginning of EPICS
- DBR_*: Numbers, enums, string, scalar and array, with time, alarm, limits
- Still fully supported

PV Access

- Started as “EPICS V4” development
- PV Data: Arbitrary structures
- Since EPICS 7 (Dec. 2017) included in EPICS base

Review Channel Access

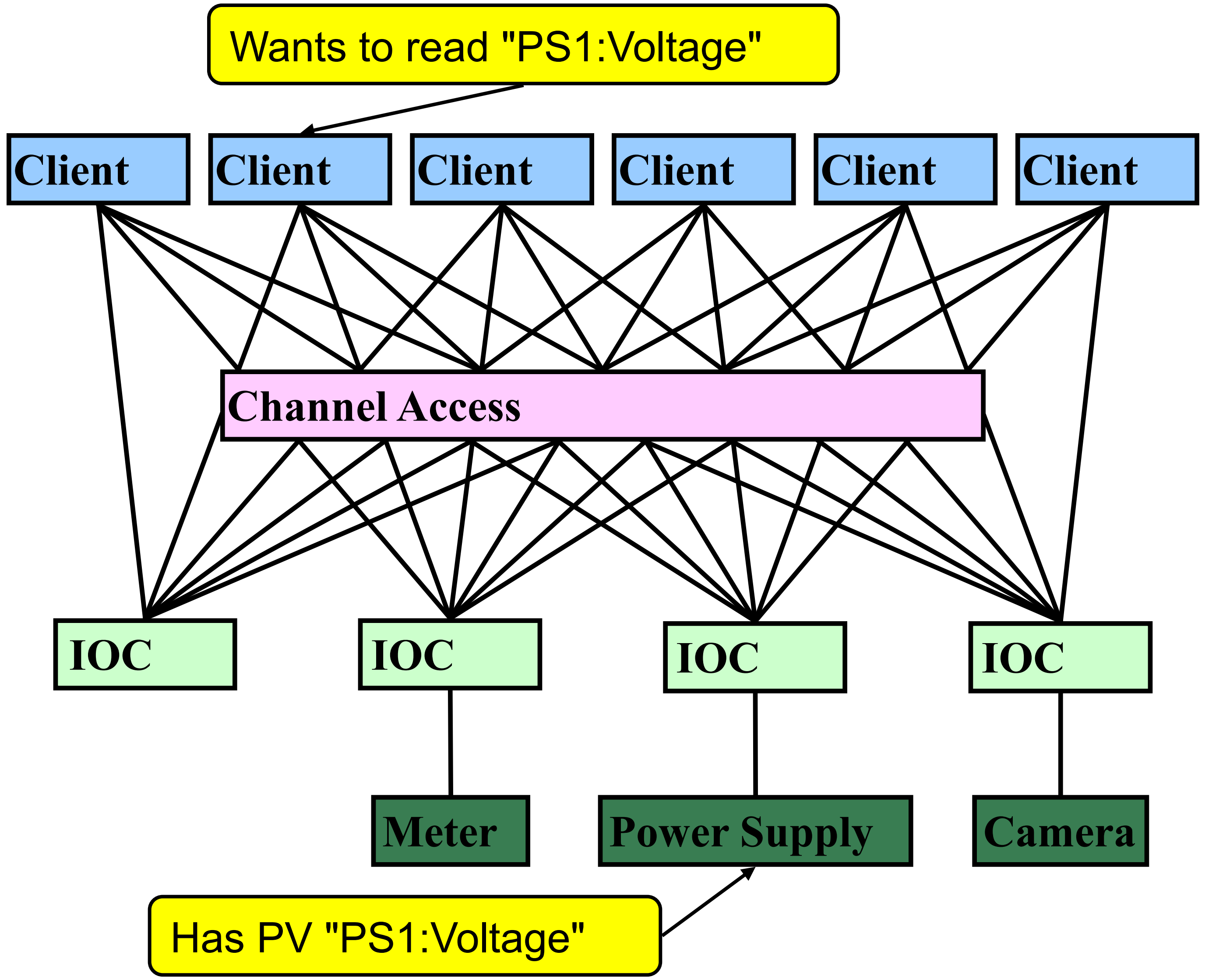
```
cd ~/epics-train/examples/first_steps
cat first.db
softIoc –m S=training –d first.db
```

```
echo $EPICS_CA_ADDR_LIST
caget training:random
camonitor training:random
```

```
cainfo training:random
caget –h
caget –d DBR_CTRL_DOUBLE training:random
```

```
cainfo training:random.SCAN
caget –d DBR_CTRL_ENUM training:random.SCAN
```

How Clients find Channels



OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

Important Environment Variables

- EPICS_CA_ADDR_LIST
  - Determines where to search
  - Is a list (separated by spaces)
    - “123.45.1.255 123.45.2.14 123.45.2.108”
  - Default is broadcast addresses of all interfaces on the host
    - Works when servers are on same subnet as Clients
  - Broadcast address
    - Goes to all servers on a subnet
    - Example: 123.45.1.255
    - Use `ifconfig –a` on UNIX to find it
- EPICS_CA_AUTO_ADDR_LIST
  - YES: Include default addresses above in searches
  - NO: Do not search on default addresses
  - If you set EPICS_CA_ADDR_LIST, usually set this to NO

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

Channel Properties

- Each channel comes with properties:
  - Value
    - of type string or double or int or …
    - Scalar or array
  - Time stamp
    - Up to nanosecond precision
  - Severity code
    - OK, MINOR, MAJOR, or INVALID
  - Status code to qualify the severity
    - OK, READ error, WRITE error, at HIGH limit, …
  - units, suggested display range, control limits, alarm limits.

Client interface to properties

- The available properties are fixed.
  - One cannot add a new 'color' property.
- The request types are fixed.
  - "DBR_…" types.
  - Available:
    a) Just value.
    b) Value with status and severity.
    c) Value with status, severity and time stamp.
    d) Almost Everything: Value, units, status, limits, … but time
  - Not available:
    - Custom combinations like value, precision, units.
  - See `caget –h`

Example: AI record "fred"

- PV "fred" or "fred.VAL"

– value property of channel	= VAL field of record.
• Type double, one element (scalar).	
– time property	= TIME field
– status	= STAT
– Severity	= SEVR
– units	= EGU
– Precision	= PREC
– display limit low, high	= LOPR, HOPR
– control limit low, high	= LOPR, HOPR
– alarm limits	= LOLO, LOW, HIGH, HIHI

- Makes a lot of sense.
  - GUI can display the value together with units, formatted according to the precision, as e.g. "12.37 volts".

Example: AI record "fred"

- PV "fred.SCAN"
  - value property of channel = SCAN field of record.
    - Type enumerated, values: "Passive", "1 second", ...
  - time property = TIME field?
  - status = STAT?
  - Severity = SEVR?
  - control limit low, high = 0, ??

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

How is PV Access different?

```
cd epics-train/examples
cat first.db
softIocPVA -m S=training -d first.db

echo $EPICS_PVA_ADDR_LIST
pvget training:random
pvmonitor training:random
pvget -m training:random


pvinfo training:random
pvinfo training:random.SCAN
```

Channel Access vs. PV Access

Similar command line tools:

`start_iocExample`	
`cainfo training:ai1`	`pvinfo training:ai1`
`caget training:ai1`	`pvget training:ai1`

`camonitor training:ai1`	`pvmonitor training:ai`
	`pvget —m training:ai1`

`caget —d CTRL_DOUBLE training:ai`	`pvget —v -r 'field()' training:ai1`
`caget training:ai1.SCAN`	`pvget training:ai1.SCAN`

PV Access

Fundamentally similar to Channel Access

- Name search via UDP
- Connection for data transfer via TCP
- EPICS_PVA_ADDR_LIST, EPICS_PVA_AUTO_ADDR_LIST

Get, put, monitor

- Plus ‘GetPut’, ’PutGet’, ‘RPC’ type operations

Arbitrary PV Data structures instead of DBR_.. types

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

Arbitrary Data: Great, but then what?

```
structure:
double    value
short     status
short     severity
string    units
time      timeStamp
…
```

```
structure:
short     level
double    data
string    type
time      stamp
…
```

```
structure:
short     level
double    wert
string    typ
long      zeit
…
```

```
structure:
short     info
double    content
string    meta
long      ms
…
```

- Which number to show on a user display?
- What units?
- Is this an alarm?
- Time stamp?

“Normative Types”

- Channel Access

```
struct  dbr_ctrl_double:
short    status
short    severity
short    precision
char     units[8]
… no timestamp …
double   value
```

```
struct  dbr_time_double:
short     status
short     severity
timestamp stamp
double    value
```

You get what you request
(network always transfers complete struct)

- PV Access

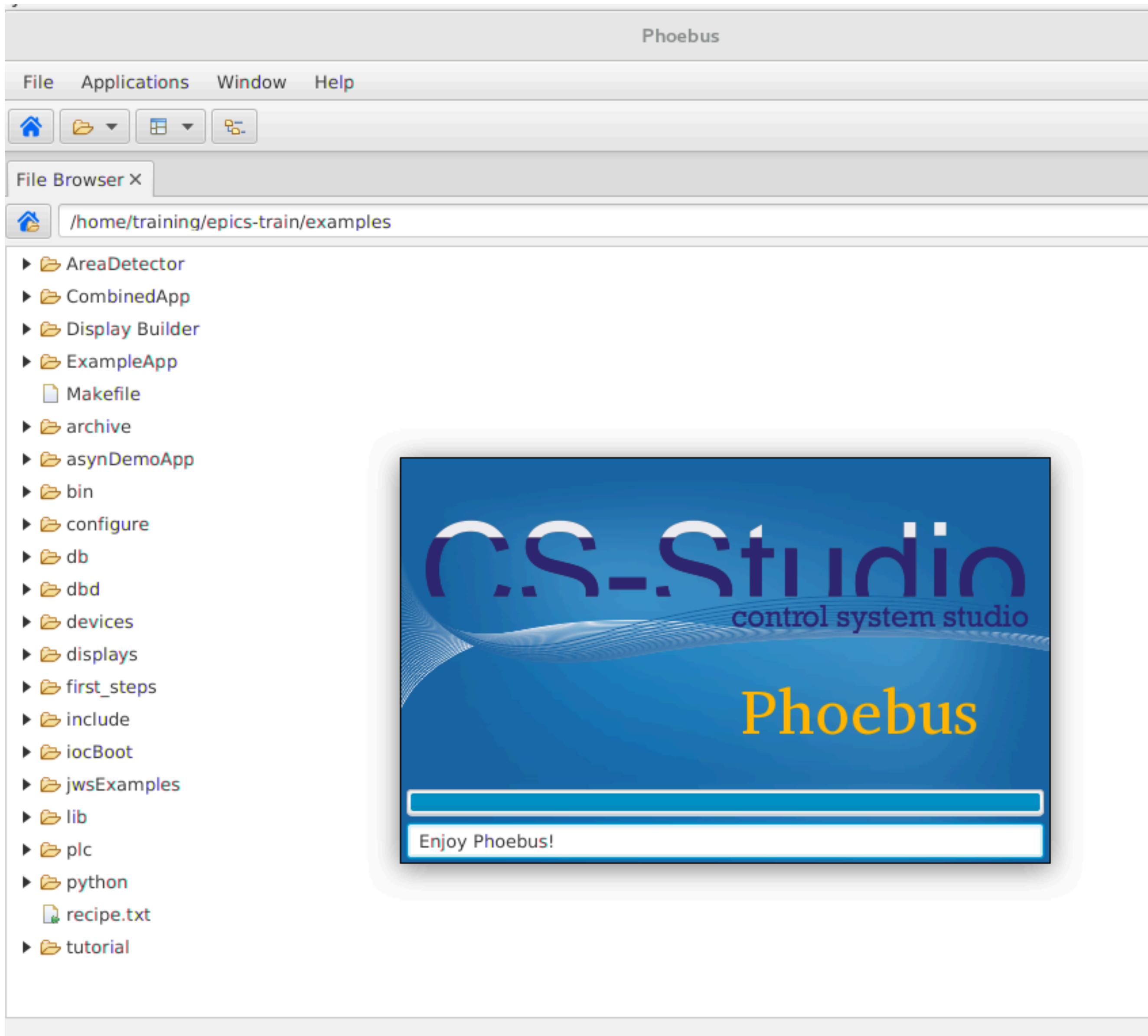
```
epics:nt/NTScalar:
double    value
short     status
short     severity
string    units
time      timeStamp
…
```

You get what you request
(but network only transfers changes)

Channel Access
vs.
PV Access
in
UI Tools

Getting Started with CSS

Start `css`



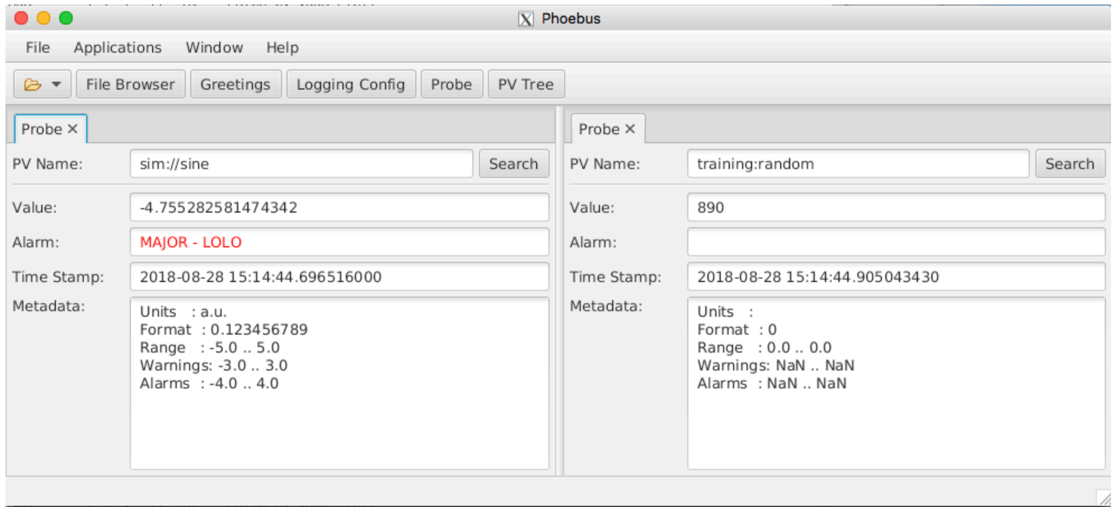
/home/training/epics-train/examples

- AreaDetector
- CombinedApp
- Display Builder
- ExampleApp
- Makefile
- archive
- asynDemoApp
- bin
- configure
- db
- dbd
- devices
- displays
- first_steps
- include
- iocBoot
- jwsExamples
- lib
- plc
- python
- recipe.txt
- tutorial

Enjoy Phoebus!

Probe

- Open Menu *Applications, Display, Probe*
- Enter PV name “sim://sine”
- Open another Probe for “training:random” (or some other PV from your IOC)
- Close Probe
- Open it again
- Note previously used PVs in history as you enter new PV
- Right-click on the “Probe” tab, Select “Split Horizontally”, and move one of the probes to

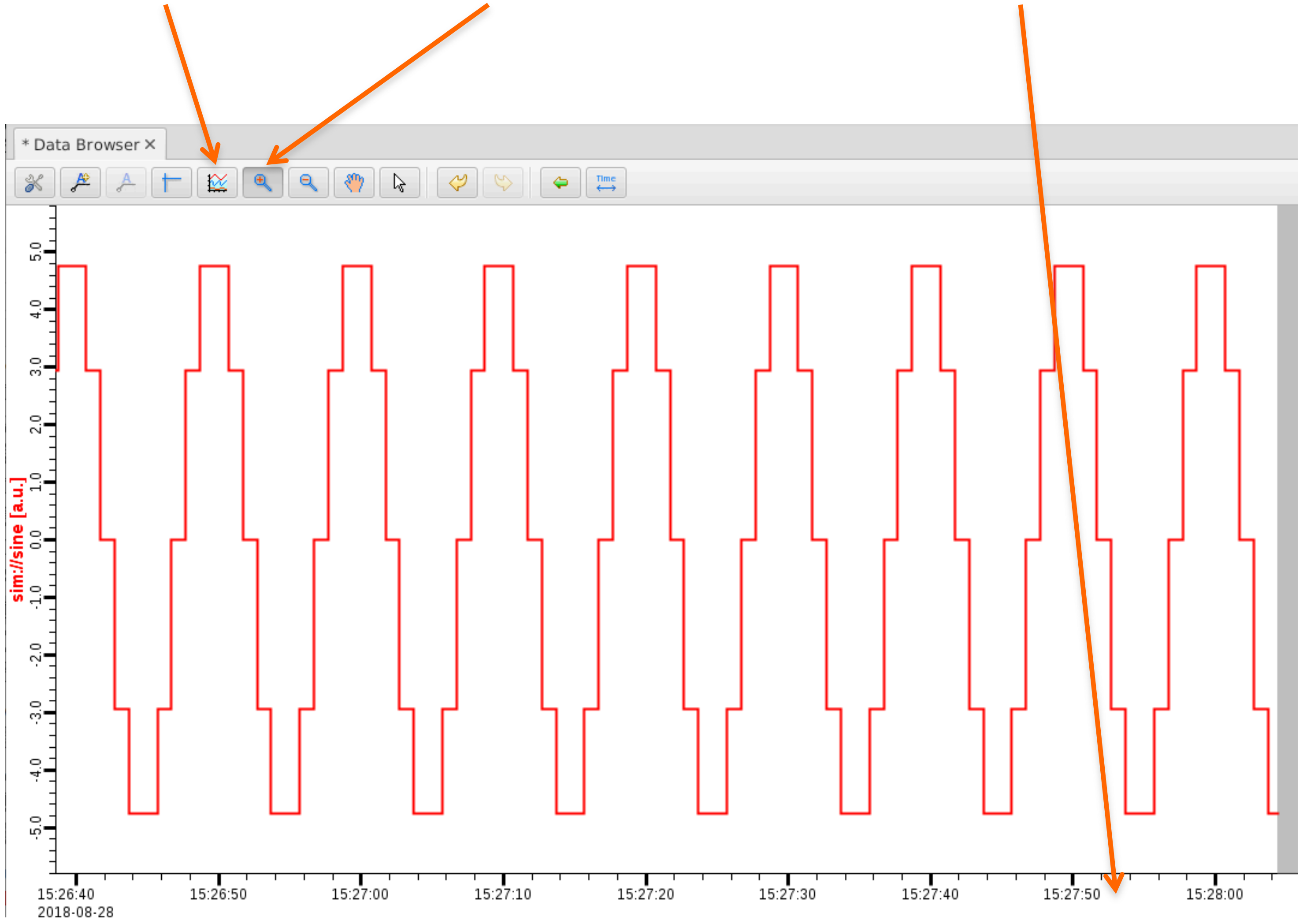


PV Name:	sim://sine
Value:	-4.755282581474342
Alarm:	MAJOR - LOLO
Time Stamp:	2018-08-28 15:14:44.696516000
Metadata:	Units : a.u. Format : 0.123456789 Range : -5.0 .. 5.0 Warnings: -3.0 .. 3.0 Alarms : -4.0 .. 4.0

PV Name:	training:random
Value:	890
Alarm:	
Time Stamp:	2018-08-28 15:14:44.905043430
Metadata:	Units : Format : 0 Range : 0.0 .. 0.0 Warnings: NaN .. NaN Alarms : NaN .. NaN

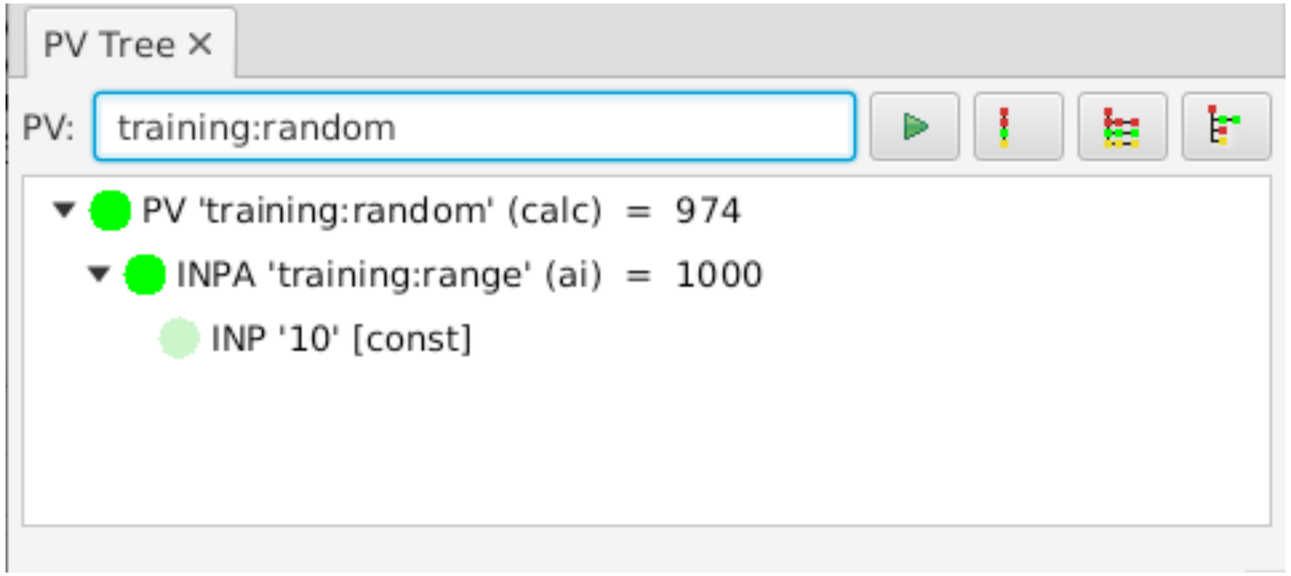
Data Browser

- Menu *Applications, Display, Data Browser*
- Right-click on plot, *Add PV*, "sim://sine"
- Wait a little, press *Stagger* button, then *zoom* and select a region on the time axis



PV Tree

- Menu *Applications, Display, PV Tree*
- Enter a PV from an IOC, like “training:random”



training:random

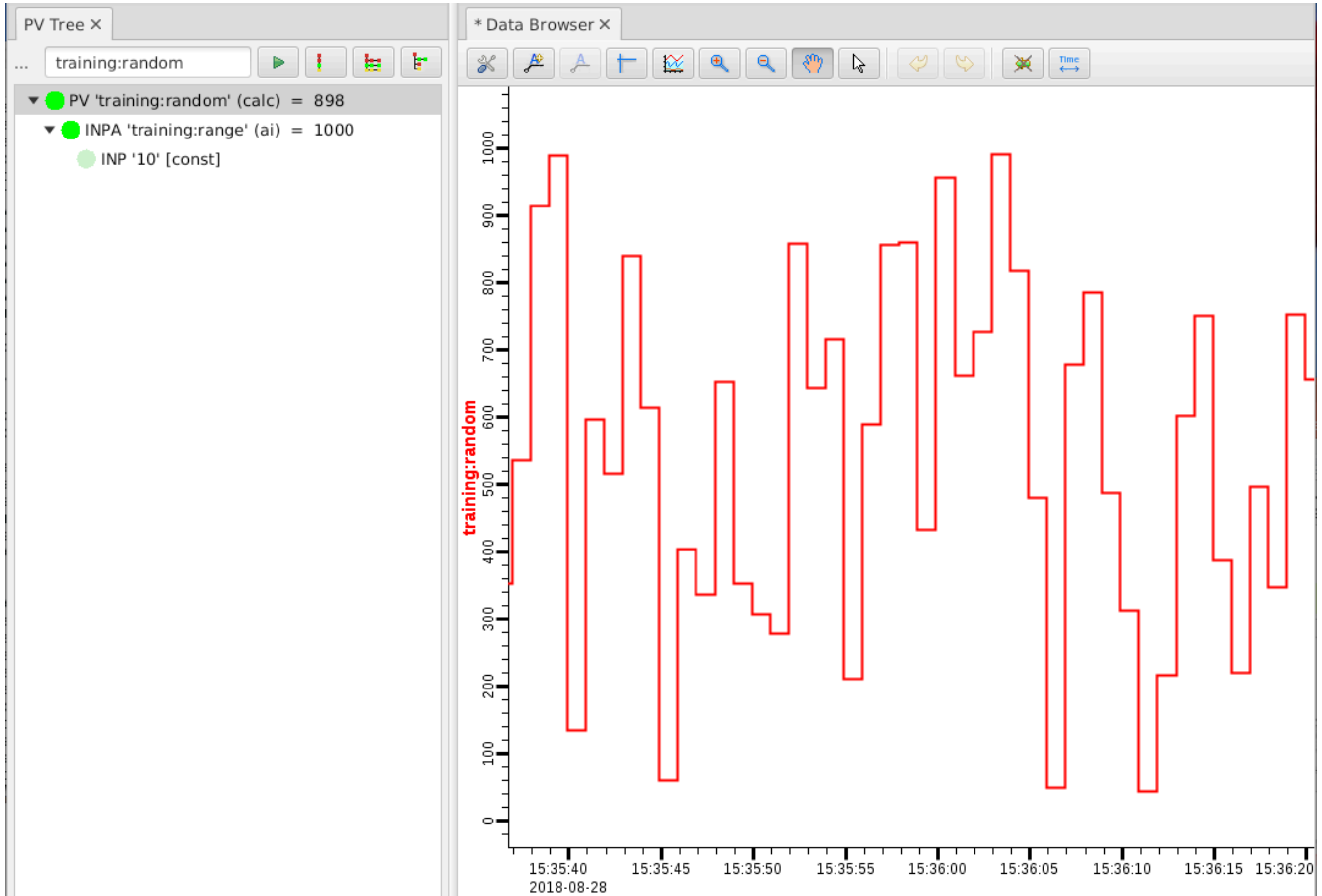
```
▼ ● PV 'training:random' (calc)  =  974
    ▼ ● INPA 'training:range' (ai)  =  1000
        ● INP '10' [const]
```

CSS PV Exchange

- PV in *any* CSS Tool
  ➔ Context Menu ➔ Select other PV Tool

Try:

Right-click on item in PV Tree, select Data Browser

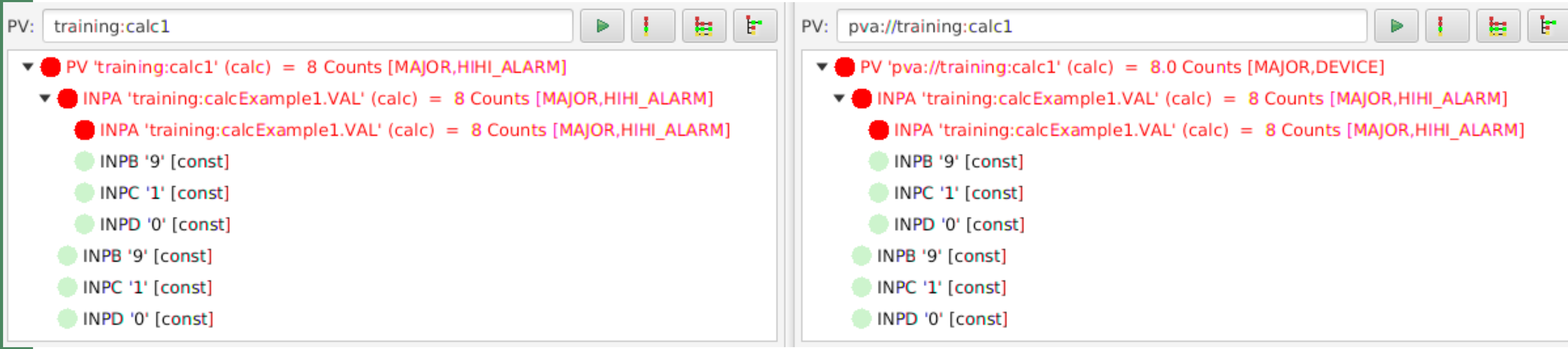

PV Tree

training:random

```
▼ ● PV 'training:random' (calc)  =  898
  ▼ ● INPA 'training:range' (ai)  =  1000
      ● INP '10' [const]
```

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

CS-Studio: Use ‘pva://…’



PV: training:calc1

```
▼ ● PV 'training:calc1' (calc)  =  8 Counts [MAJOR,HIHI_ALARM]
    ▼ ● INPA 'training:calcExample1.VAL' (calc)  =  8 Counts [MAJOR,HIHI_ALARM]
        ● INPA 'training:calcExample1.VAL' (calc)  =  8 Counts [MAJOR,HIHI_ALARM]
        ● INPB '9' [const]
        ● INPC '1' [const]
        ● INPD '0' [const]
    ● INPB '9' [const]
    ● INPC '1' [const]
    ● INPD '0' [const]
```

PV: pva://training:calc1

```
▼ ● PV 'pva://training:calc1' (calc)  =  8.0 Counts [MAJOR,DEVICE]
    ▼ ● INPA 'training:calcExample1.VAL' (calc)  =  8 Counts [MAJOR,HIHI_ALARM]
        ● INPA 'training:calcExample1.VAL' (calc)  =  8 Counts [MAJOR,HIHI_ALARM]
        ● INPB '9' [const]
        ● INPC '1' [const]
        ● INPD '0' [const]
    ● INPB '9' [const]
    ● INPC '1' [const]
    ● INPD '0' [const]
```

For now, just “pvname” is same as “ca://pvname”.

“pva://” selects PV Access.
Eventually, that could become the default.

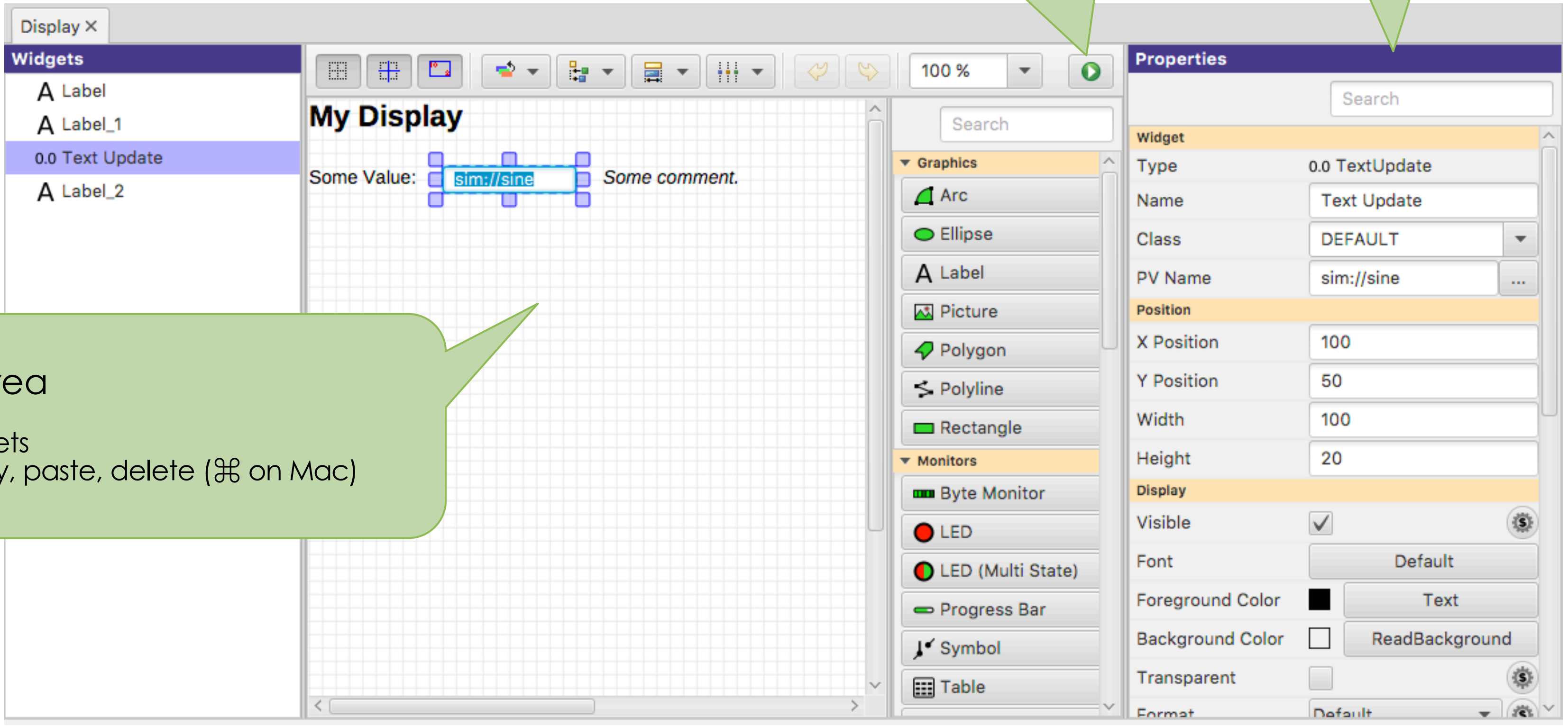
Create New Display

Menu Applications, Display, New Display

– Enter a name with .bob file extension

Save & Execute the Display

Property Panel
Edit properties of selected widgets



Widget	
Type	0.0 TextUpdate
Name	Text Update
Class	DEFAULT
PV Name	sim://sine
Position	
X Position	100
Y Position	50
Width	100
Height	20
Display	
Visible	✓
Font	Default
Foreground Color	■ Text
Background Color	□ ReadBackground
Transparent	
Format	Default

Main Editor Area
Select Widgets
Move, resize widgets
Ctrl-C, V, X to copy, paste, delete (⌘ on Mac)

Editing a Display

Selecting Widgets

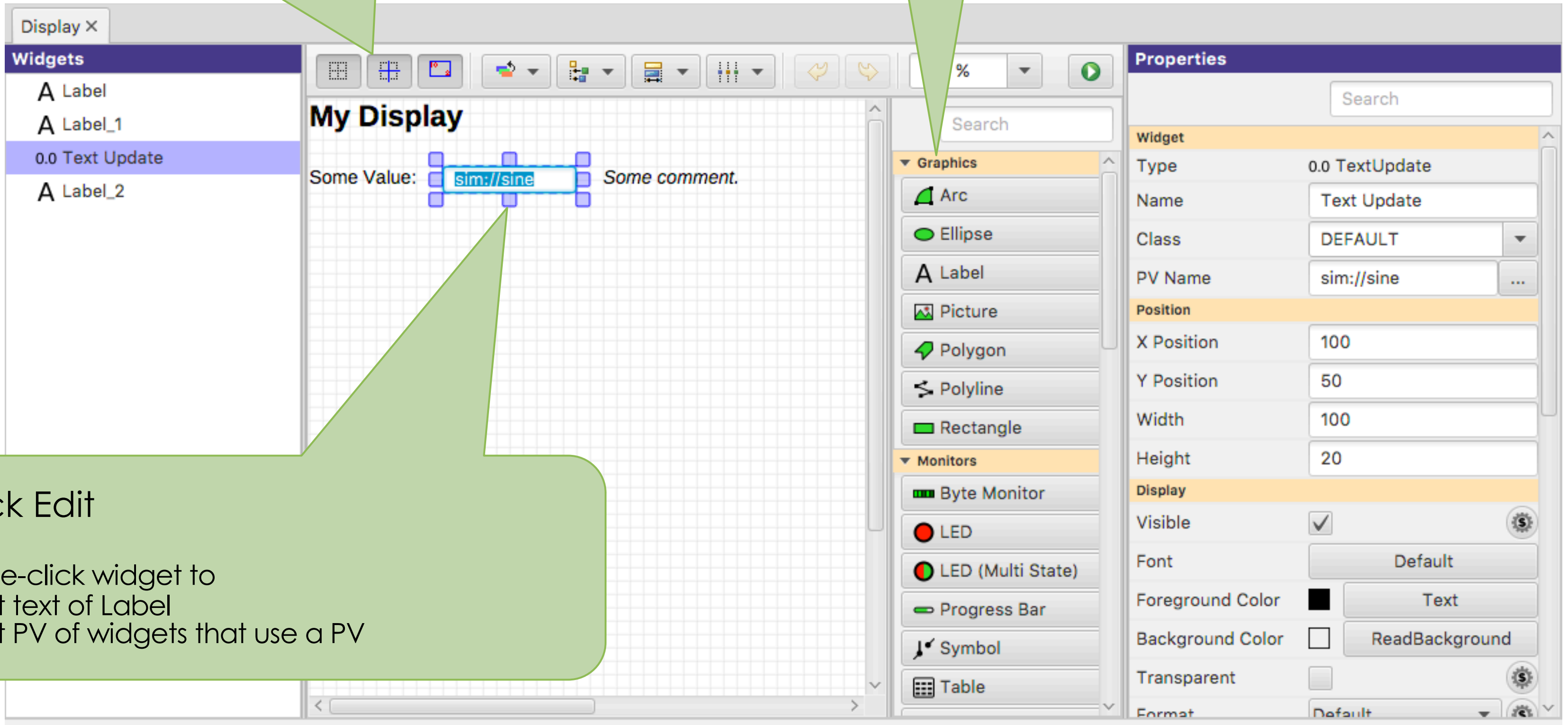
a) Click single widget
b) Ctrl-click to add widget (⌘ on Mac)
c) Drag 'rubberband' around widgets
d) Click or Ctrl/ ⌘ click in widget list

Widget Palette

Drag widget into editor

- or -

1) Select Widget Type
2) Draw rectangular area in display



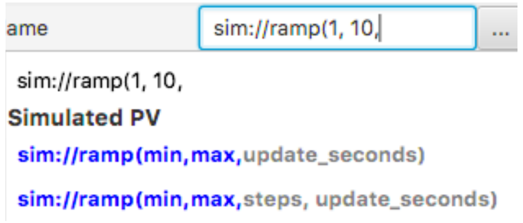
Widget	
Type	0.0 TextUpdate
Name	Text Update
Class	DEFAULT
PV Name	sim://sine
Position	
X Position	100
Y Position	50
Width	100
Height	20
Display	
Visible	✓
Font	Default
Foreground Color	Text
Background Color	ReadBackground
Transparent	
Format	Default

Quick Edit

Double-click widget to
a) Edit text of Label
b) Edit PV of widgets that use a PV

Extend the First Display

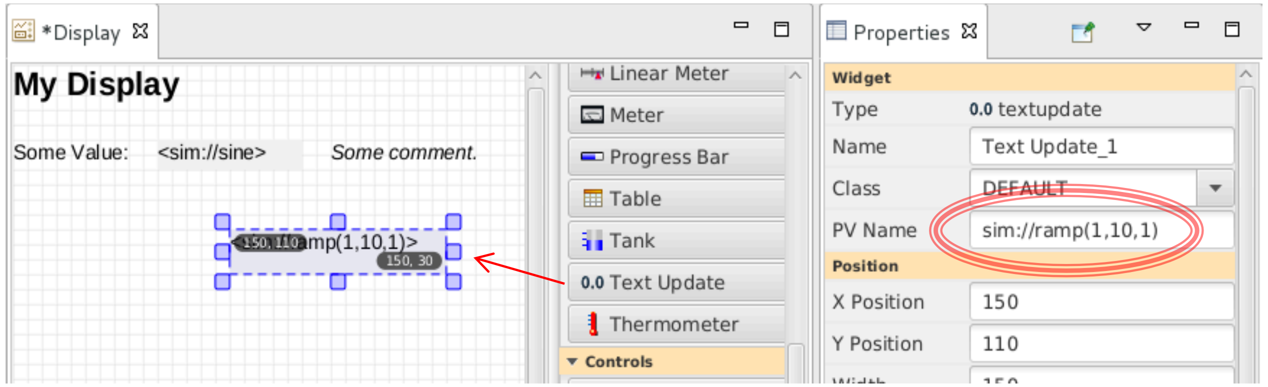
- Drag a “Text Update” from the palette
  - Enter PV name “sim://ramp(1, 10, 1)”. Note PV name auto-completion popup.



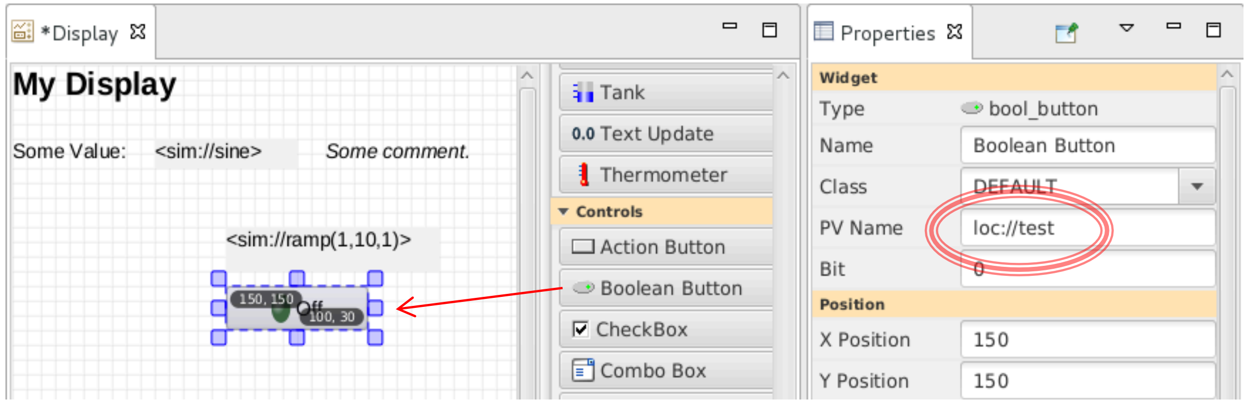
ame	sim://ramp(1, 10,

sim://ramp(1, 10,
Simulated PV
sim://ramp(min,max,update_seconds)
sim://ramp(min,max,steps, update_seconds)

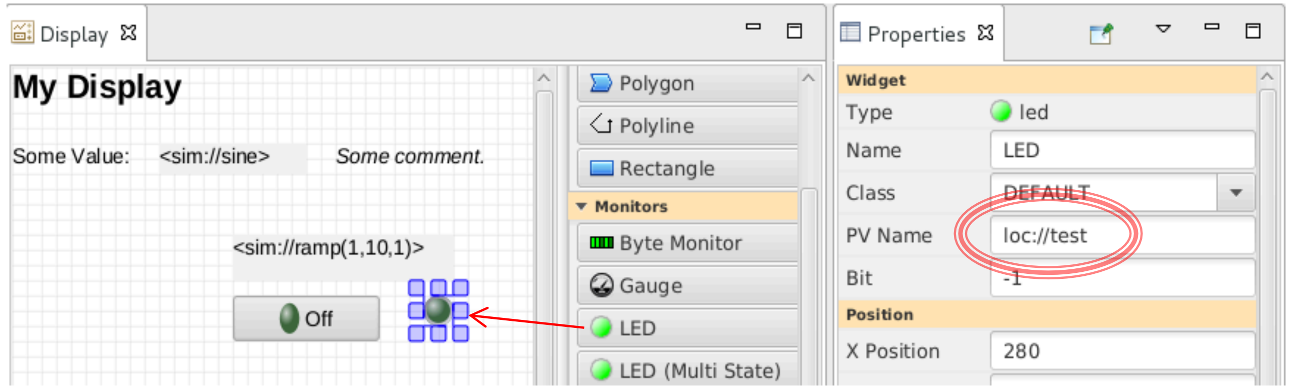
- Add “Boolean Button”
  - PV name “loc://test”
- Add “LED”
  - PV name “loc://test”. Note name in PV History.
- Execute the display
  - Toolbar Button or Context Menu



Widget	
Type	0.0 textupdate
Name	Text Update_1
Class	DEFAULT
PV Name	sim://ramp(1,10,1)
Position	
X Position	150
Y Position	110



Widget	
Type	bool_button
Name	Boolean Button
Class	DEFAULT
PV Name	loc://test
Bit	0
Position	
X Position	150
Y Position	150

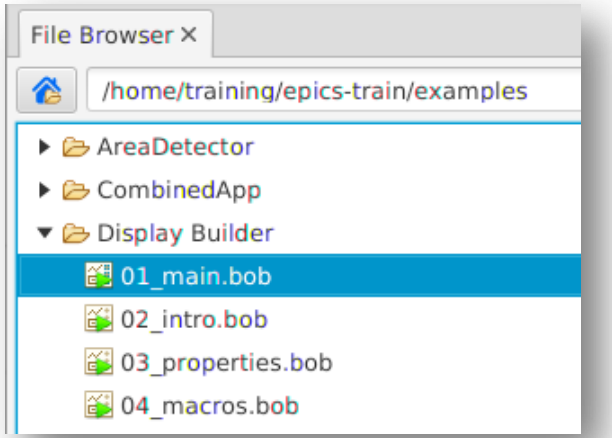


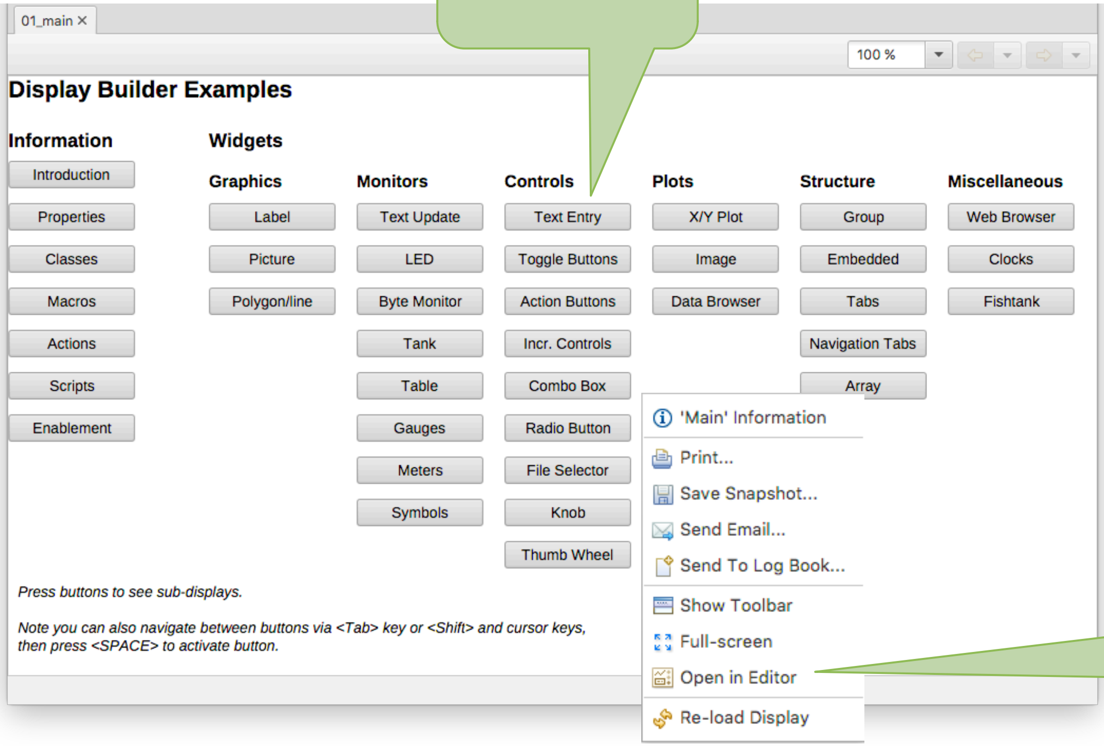
Widget	
Type	led
Name	LED
Class	DEFAULT
PV Name	loc://test
Bit	-1
Position	
X Position	280

Browse the Display Examples

Training setup: Open /home/training/epics-train/examples/Display Builder/01_main.bob

- Fresh CS-Studio Setup: Applications, Display, Examples, Install Example Displays





Press buttons to see sub-displays.

Note you can also navigate between buttons via <Tab> key or <Shift> and cursor keys, then press <SPACE> to activate button.

Try all of them..

Context Menu:
Open in Editor to inspect how it's done

How is PV Access different?

Images!

```
start_imagedemo
pvinfo IMAGE
# CSS displays/PVA_Image.bob
```

Area Detector

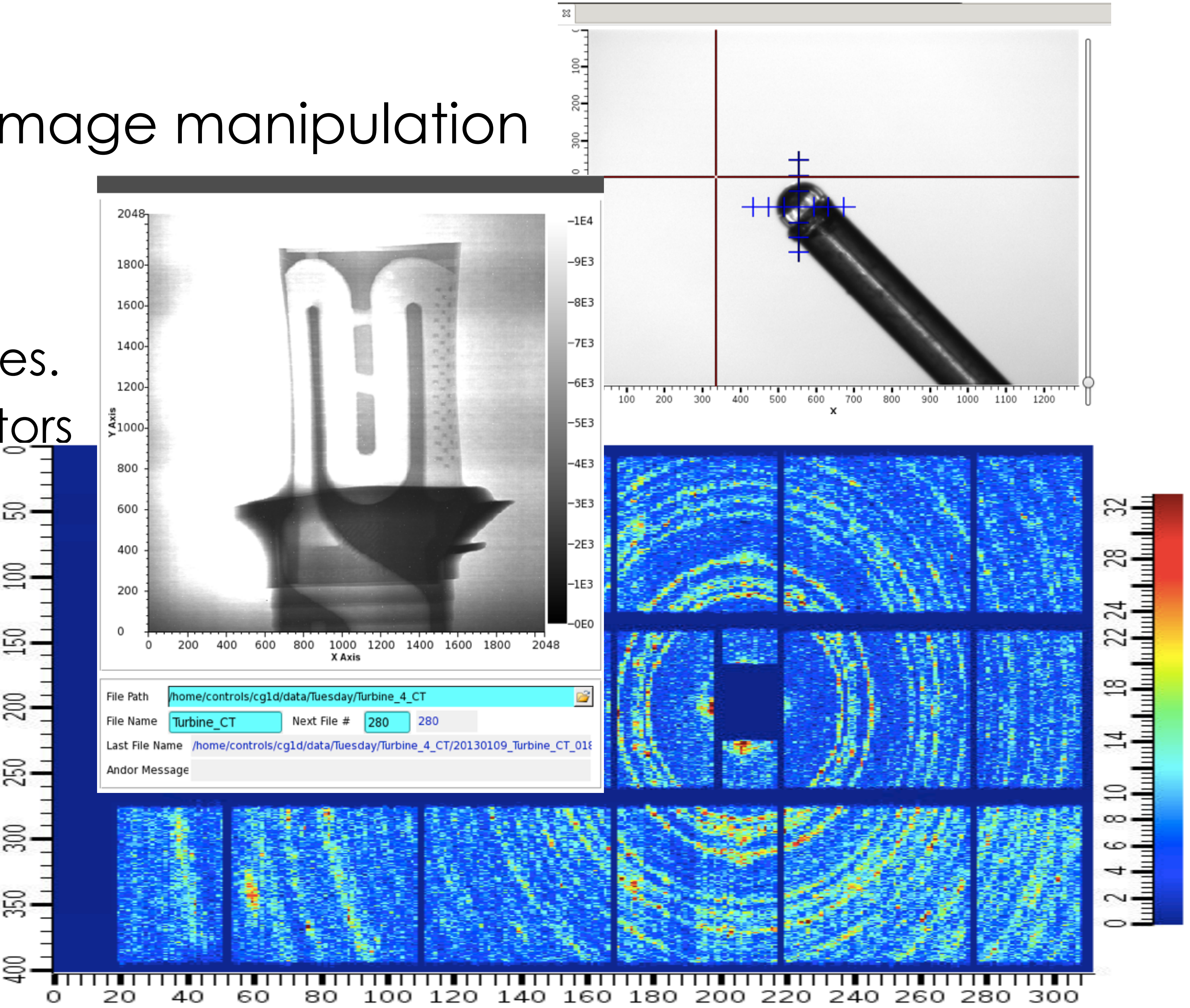
Disclaimer:

This will only scratch the surface.

EPICS web site has several days of training material if you are serious about using the A.D.

Area Detector

- EPICS framework for image manipulation
- Detectors/Cameras
  - Cheap "Web Cam"
  - $$$ high speed, high res.
  - Neutron, X-Ray detectors
- Plugins collection
  - ROI
  - Transform
  - ColorConvert
  - Etc.



File Path /home/controls/cg1d/data/Tuesday/Turbine_4_CT
File Name Turbine_CT Next File # 280 280
Last File Name /home/controls/cg1d/data/Tuesday/Turbine_4_CT/20130109_Turbine_CT_01
Andor Message

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

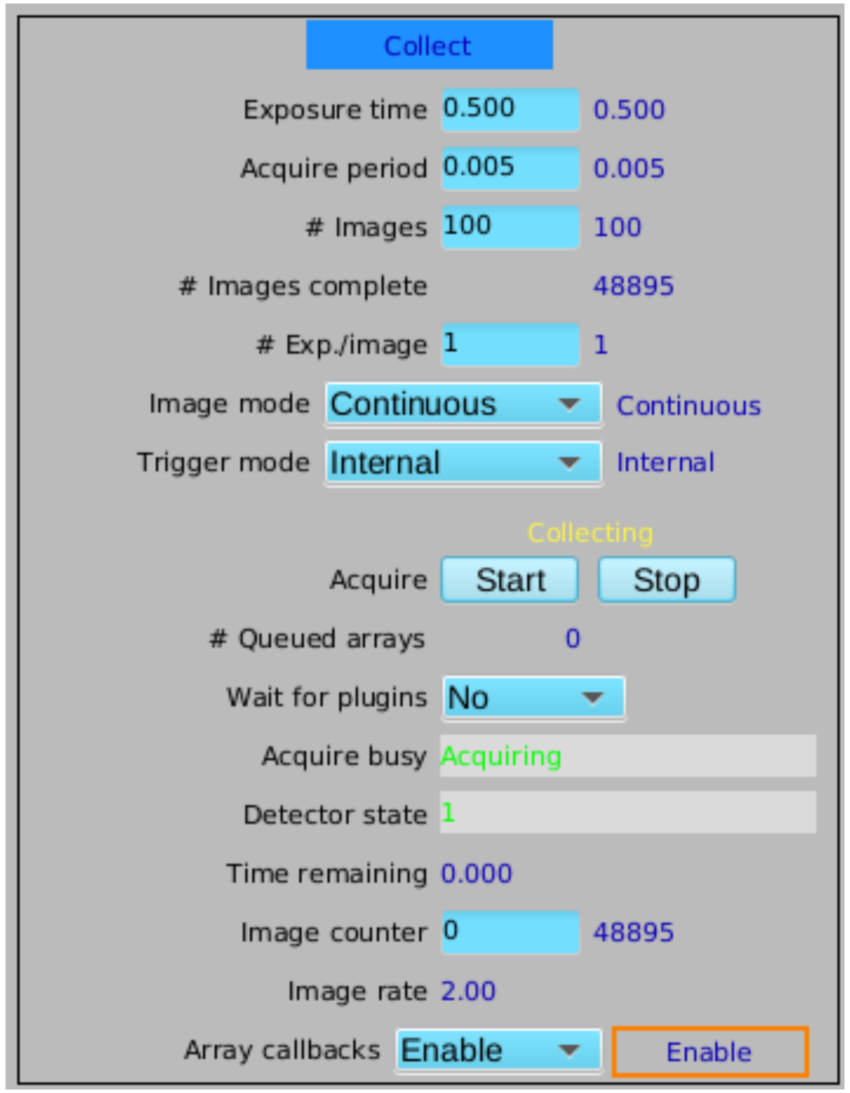
ADSimDetector

- Simulated images

```
cd ~/epics-train/examples/AreaDetector
./start_sim_ioc.sh
```

- Open the AreaDetectorDemo.bob
  - On "Detector" page, "Start" the SIM1 detector

By itself, this creates an
Area Detector port "SIM1".
To see it, need to <u>publish</u> via CA or PVA

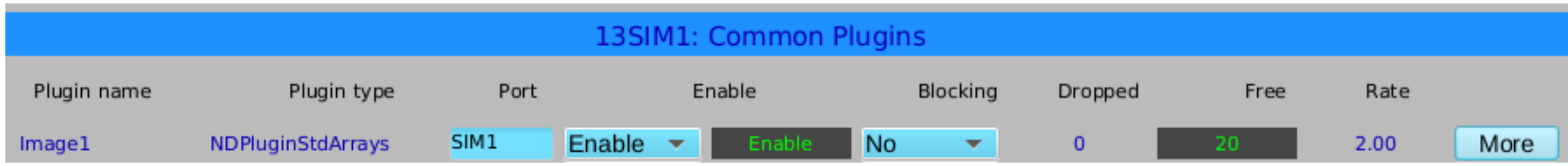


Collect

Exposure time	0.500	0.500
Acquire period	0.005	0.005
# Images	100	100
# Images complete		48895
# Exp./image	1	1
Image mode	Continuous	Continuous
Trigger mode	Internal	Internal
		Collecting
Acquire	Start	Stop
# Queued arrays		0
Wait for plugins	No	
Acquire busy	Acquiring	
Detector state	1	
Time remaining	0.000	
Image counter	0	48895
Image rate	2.00	
Array callbacks	Enable	Enable

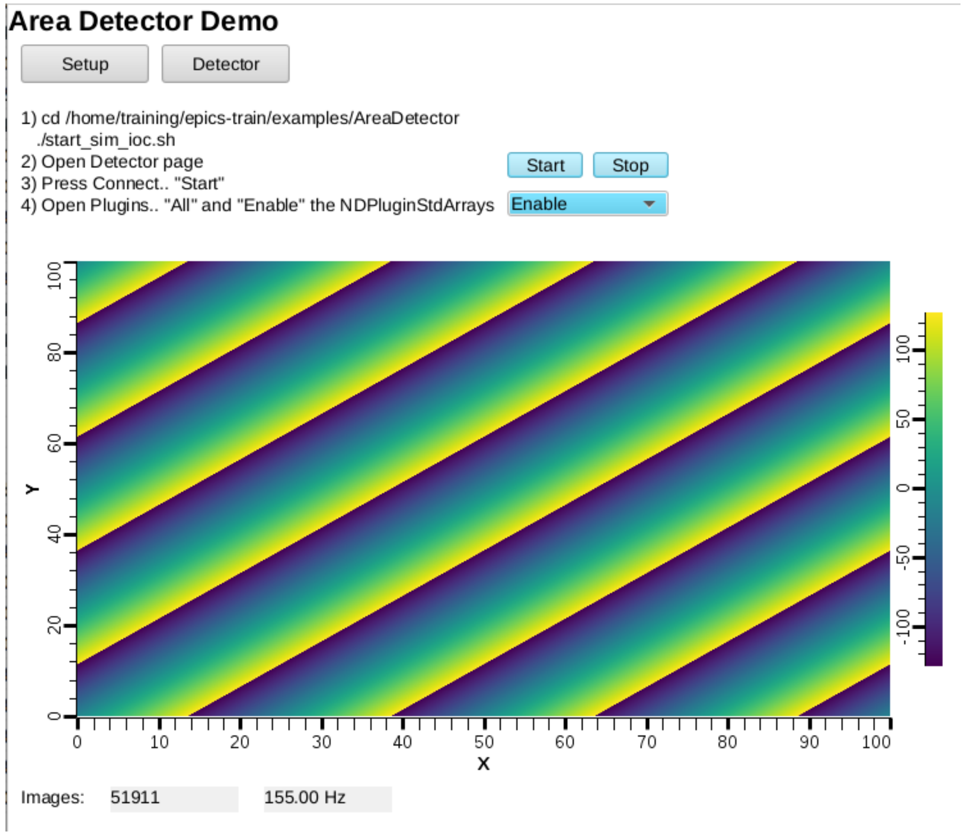
NDPluginStdArrays

- Serves image as Channel Access waveform
- On Detector, Plugins, All, find NDPluginStdArrays
  - Port = “SIM1”
  - Enable
- AreaDetectorDemo.bob shows image
  - PV: 13SIM1:image1:ArrayData
  - Width x Height: 1024 x 1024
  - Unsigned



13SIM1: Common Plugins

Plugin name	Plugin type	Port	Enable		Blocking	Dropped	Free	Rate	
Image1	NDPluginStdArrays	SIM1	Enable	Enable	No	0	20	2.00	More



Area Detector Demo

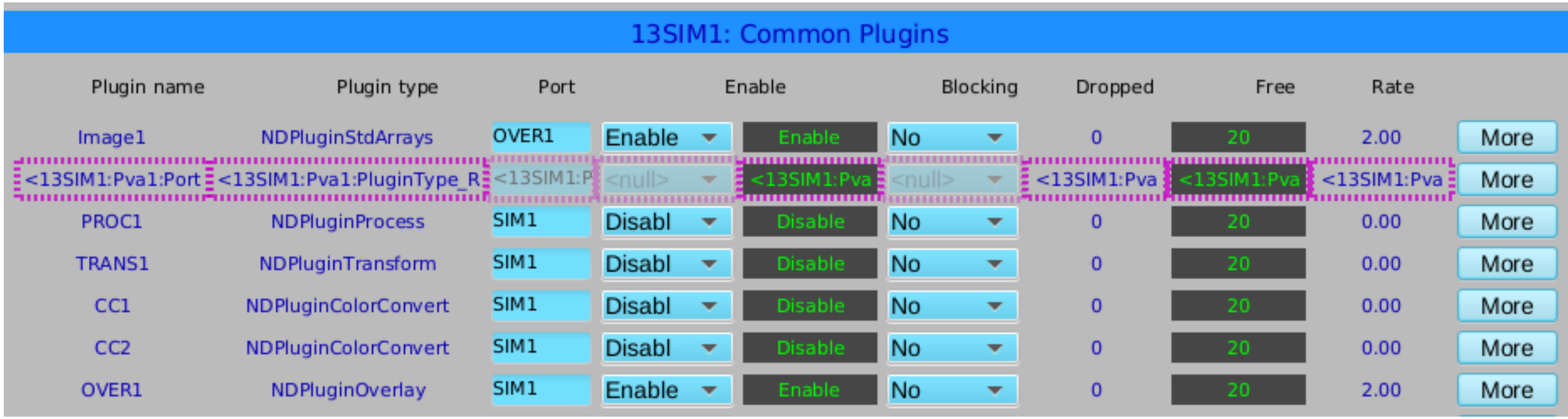
1) cd /home/training/epics-train/examples/AreaDetector
   ./start_sim_ioc.sh
2) Open Detector page
3) Press Connect.. "Start"
4) Open Plugins.. "All" and "Enable" the NDPluginStdArrays

Images: 51911 155.00 Hz

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

NDPluginOverlay

- Adds rectangles, text etc. to image
- On Detector, Plugins, All, find NDPluginOverlay "OVER1"
  - Set its Port to "SIM1", Enable
  - Change NDPluginStdArrays's Port to "OVER1"



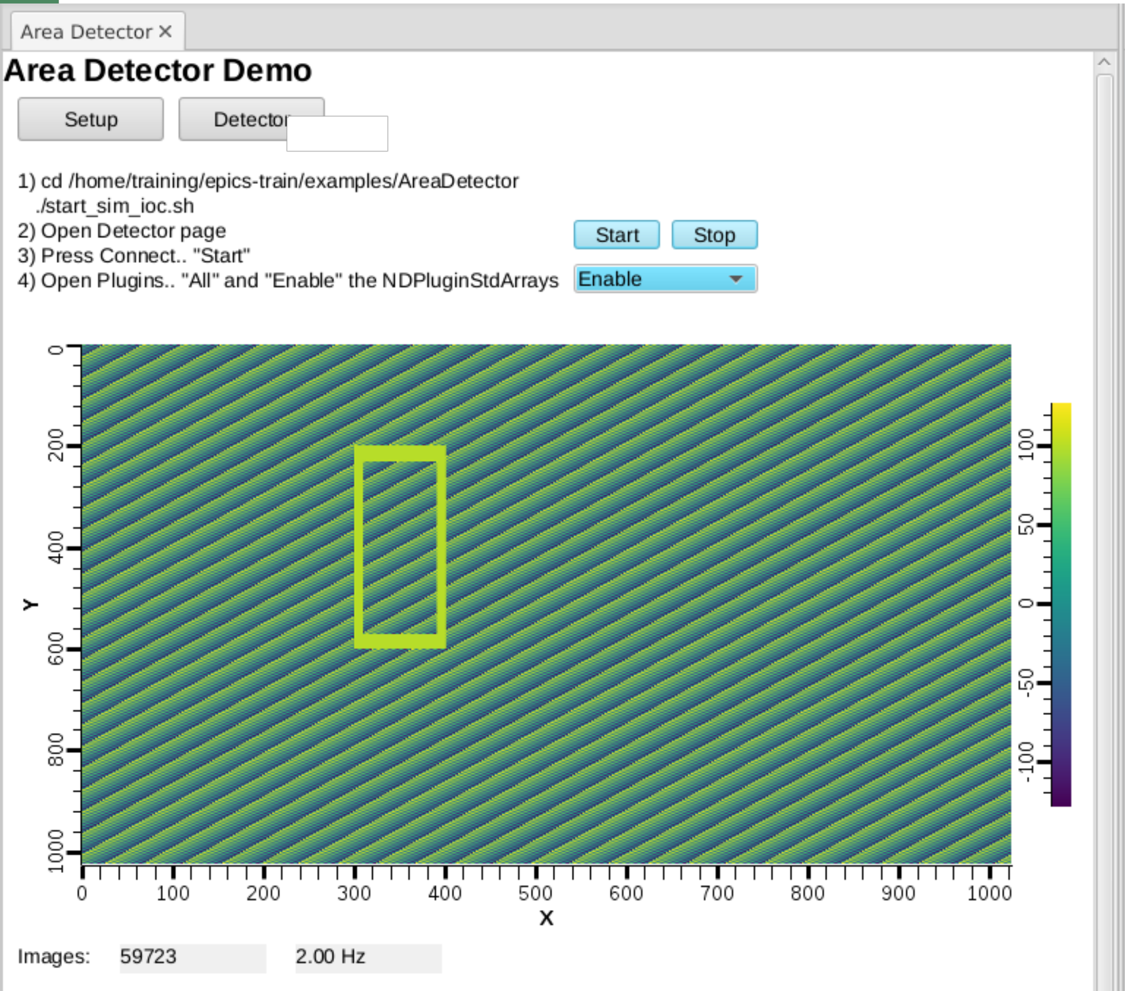
13SIM1: Common Plugins

Plugin name	Plugin type	Port	Enable		Blocking	Dropped	Free	Rate	
Image1	NDPluginStdArrays	OVER1	Enable	Enable	No	0	20	2.00	More
<13SIM1:Pva1:Port	<13SIM1:Pva1:PluginType_R	<13SIM1:P	<null>	<13SIM1:Pva	<null>	<13SIM1:Pva	<13SIM1:Pva	<13SIM1:Pva	More
PROC1	NDPluginProcess	SIM1	Disabl	Disable	No	0	20	0.00	More
TRANS1	NDPluginTransform	SIM1	Disabl	Disable	No	0	20	0.00	More
CC1	NDPluginColorConvert	SIM1	Disabl	Disable	No	0	20	0.00	More
CC2	NDPluginColorConvert	SIM1	Disabl	Disable	No	0	20	0.00	More
OVER1	NDPluginOverlay	SIM1	Enable	Enable	No	0	20	2.00	More

  - Press "More", select first of the "Individual Overlays"

NDPluginOverlay.. Overlay #1

Set Use: Yes, Shape: Rectangle, set X and Y as shown



Area Detector ×

Area Detector Demo

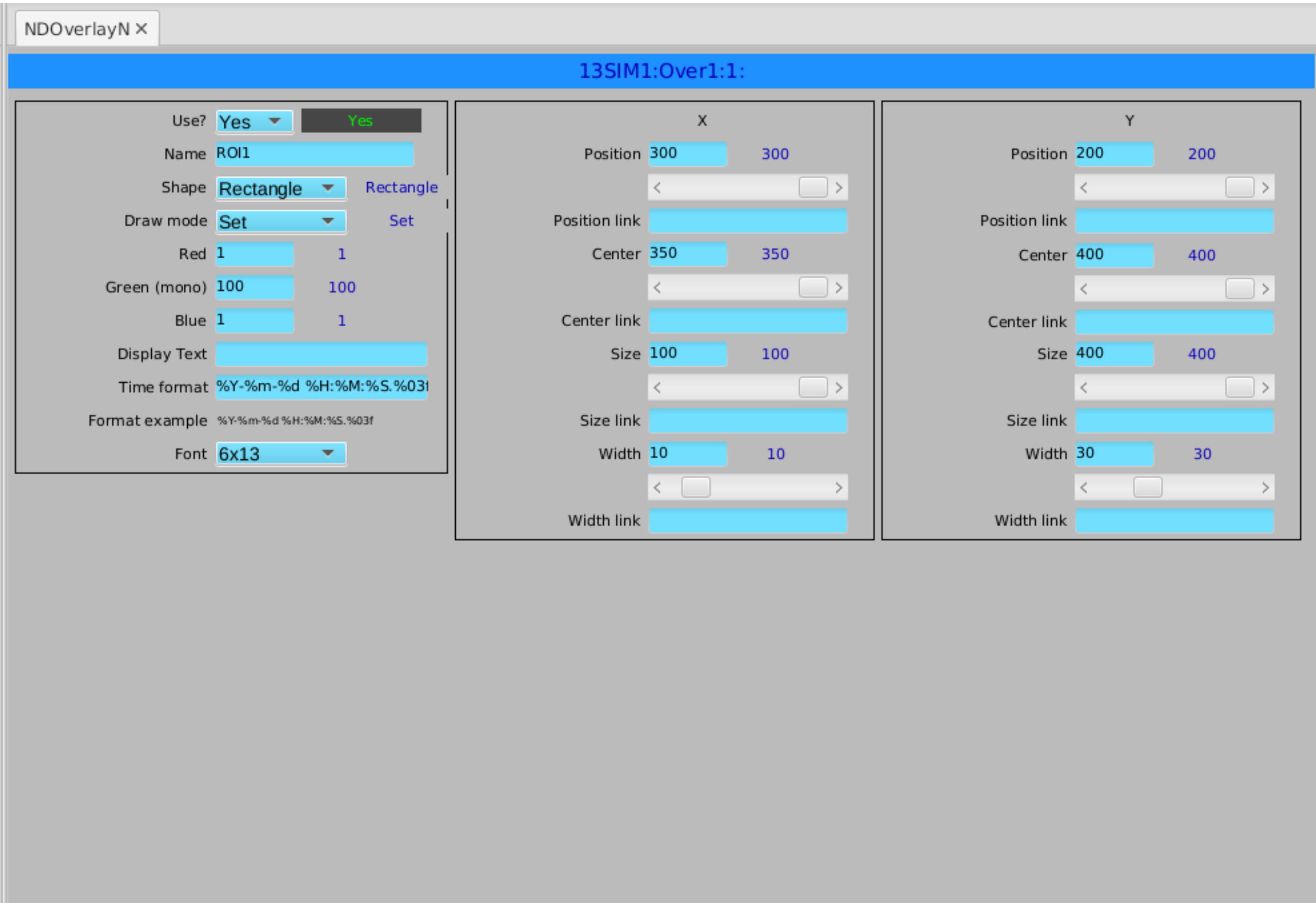
Setup | Detector

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

Images: 59723 2.00 Hz



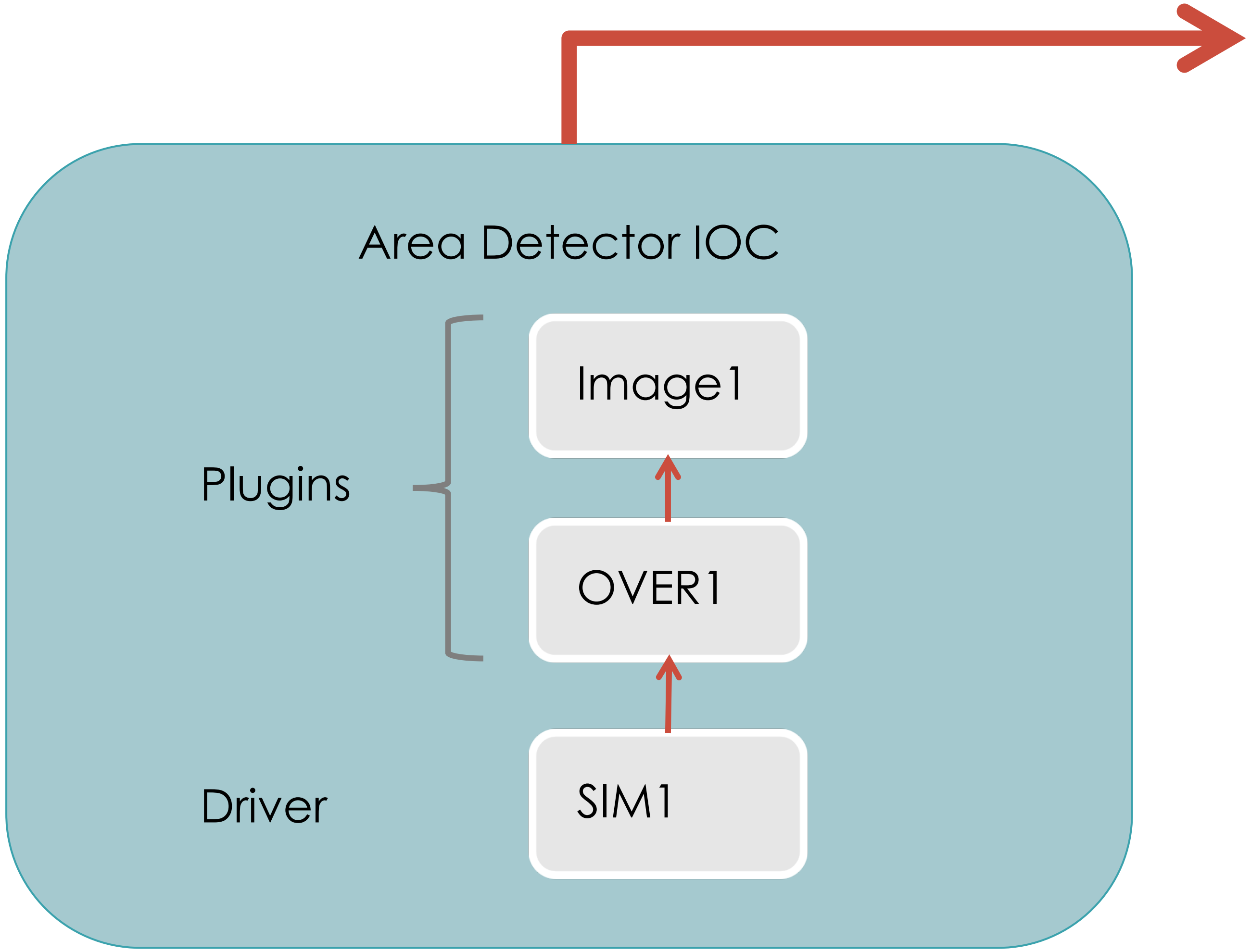
NDOverlayN ×

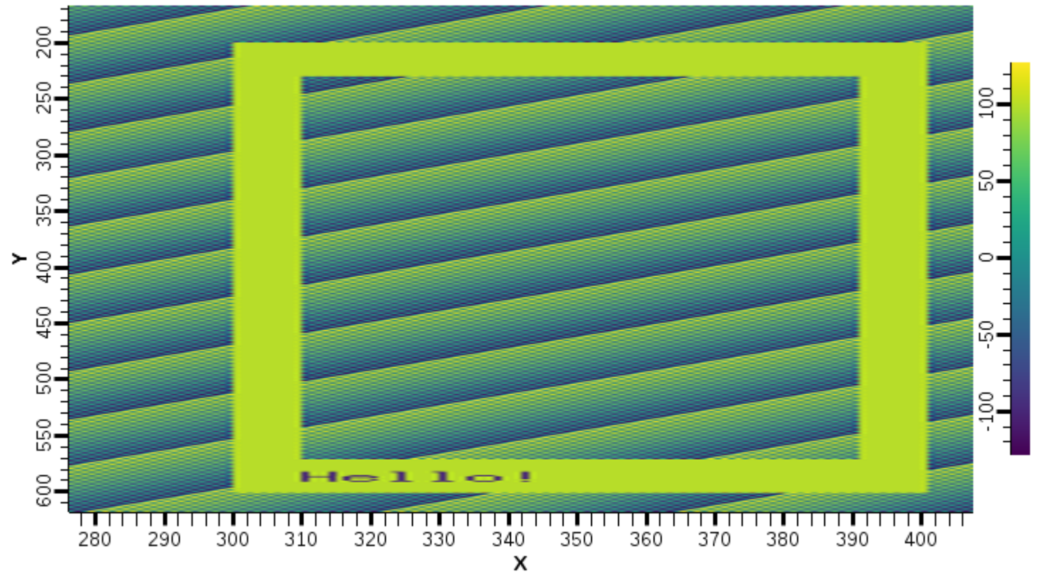
13SIM1:Over1:1:

Use?	Yes	Yes
Name	ROI1	
Shape	Rectangle	Rectangle
Draw mode	Set	Set
Red	1	1
Green (mono)	100	100
Blue	1	1
Display Text		
Time format	%Y-%m-%d %H:%M:%S.%03	
Format example	%Y-%m-%d %H:%M:%S.%03f	
Font	6x13	

	X		Y	
Position	300	300	200	200
Position link				
Center	350	350	400	400
Center link				
Size	100	100	400	400
Size link				
Width	10	10	30	30
Width link				

What we did





OVER1 offers 8 overlays:
1) Rectangle
2) Text “Hello”
3) …

Many More Plugins…

- Process
  - Background subtraction, clipping, recursive averaging over N images, ..
- Saving images in various formats
  - Adding data from PVs as “Attributes”
  - PNG, JPEG, TIFF, HDF5, …
- Serving NDArray via PVA
  - For displays: No need to configure size, data type, …
  - For ADPvAccess Driver: Process data on different hosts

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

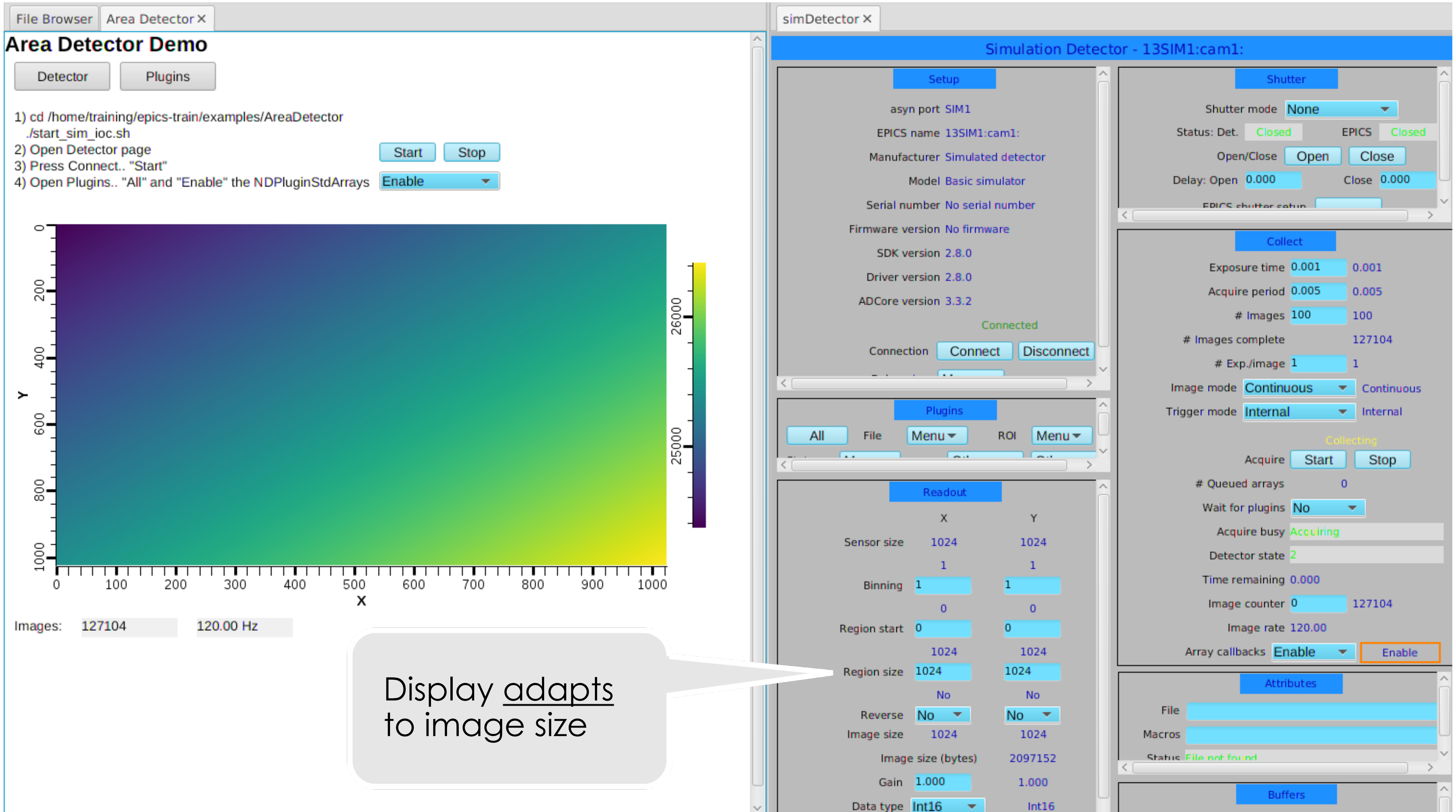
NDPluginPVA – Serve PVA 'Image'

- In Plugins, "PVA1"
  - Set its Port to "SIM1" or "OVER1", Enable

- PVAccess Tests
  - pvinfo 13SIM1:Pva1:Image
  - pvget -r 'dimension' 13SIM1:Pva1:Image

- In Display
  - Use "Image" widget
  - Set PV
  - No need to configure data size, data type

NDPluginPVA – Serve PVA ‘Image’



File Browser | Area Detector ✕

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

Images: 127104 | 120.00 Hz

simDetector ✕

Simulation Detector - 13SIM1:cam1:

Setup

asyn port	SIM1
EPICS name	13SIM1: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
	1	1
Binning	1	1
	0	0
Region start	0	0
	1024	1024
Region size	1024	1024
	No	No
Reverse	No	No
Image size	1024	1024
Image size (bytes)		2097152
Gain	1.000	1.000
Data type	Int16	Int16

Shutter

Shutter mode	None		
Status: Det.	Closed	EPICS	Closed
Open/Close	Open	Close	
Delay: Open	0.000	Close	0.000

Collect

Exposure time	0.001	0.001
Acquire period	0.005	0.005
# Images	100	100
# Images complete		127104
# Exp./image	1	1
Image mode	Continuous	Continuous
Trigger mode	Internal	Internal
		Collecting
Acquire	Start	Stop
# Queued arrays		0
Wait for plugins	No	
Acquire busy	Acquiring	
Detector state	2	
Time remaining	0.000	
Image counter	0	127104
Image rate	120.00	
Array callbacks	Enable	Enable

Attributes

File	
Macros	
Status	File not found

Buffers

Display adapts to image size

How is PV Access different?

Custom Data!

Custom PV Data

SNS Beam Lines started to use this in ~2014

```
start_neutrondemo
pvinfo neutrons
```

Allows fetching just what's needed:

```
# For detector pixel display
pvget -r 'field(pixel)' neutrons
pvget –m -r 'field(timeStamp, pixel)' neutrons

# For energy displays
pvget –m -r 'field(time_of_flight, pixel)' neutrons
```

Custom PV Data in CS-Studio

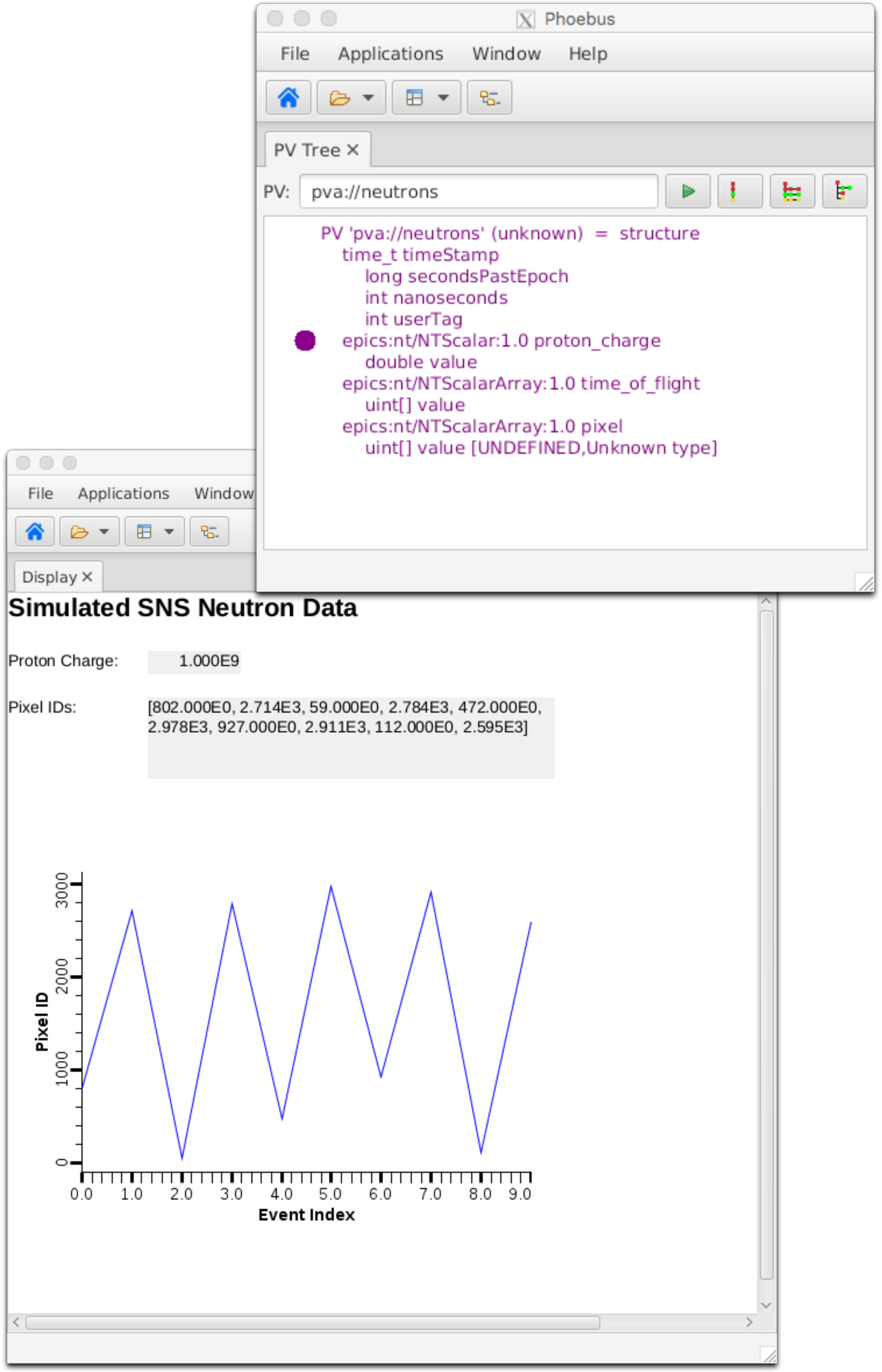
Cannot handle *arbitrary structure*

```
pva://neutrons
```

Can handle fields which are *scalar* or *array*

```
pva://neutrons/proton_charge
```

```
pva://neutrons/pixel
```



```
PV 'pva://neutrons' (unknown) = structure
  time_t timeStamp
    long secondsPastEpoch
    int nanoseconds
    int userTag
● epics:nt/NTScalar:1.0 proton_charge
    double value
  epics:nt/NTScalarArray:1.0 time_of_flight
    uint[] value
  epics:nt/NTScalarArray:1.0 pixel
    uint[] value [UNDEFINED,Unknown type]
```

Simulated SNS Neutron Data

Proton Charge:	1.000E9
Pixel IDs:	[802.000E0, 2.714E3, 59.000E0, 2.784E3, 472.000E0, 2.978E3, 927.000E0, 2.911E3, 112.000E0, 2.595E3]

Custom PV Data from IOC Records

`makeBaseApp.pl –t example` includes “group”, see `~/epics-train/examples/ExampleApp/Db/circle.db`

Calc records ..:circle:x & ..:circle:y compute (x, y) coordinate on circle

info() annotations create PV ”training:circle” PV as struct `{ angle, x, y }`

PVA “training:circle” updates atomically

`camonitor training:circle:x training:circle:y` separate x, y updates

`pvget –m training:circle` will always see `sqrt(x²+y²)==1`

```
cd ~/epics-train/examples/python
python circle.py
```

Python

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

PV Access and Python

```
start_iocExample
```

Basic ‘get’

```
cd ~/epics-train/examples/python/
python example1.py
```

‘monitor’

```
python example2.py
```

Custom PV Data in Python Client

Python receives data as dictionary, access to any element

```
python neutrons.py
```

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

Custom PV Data from Python Server

```
# Server
python server.py


# Client
pvinfo pair
pvget -m -r "x, y" pair
```

Surprisingly easy:

```
pv = PvObject({'x': INT, 'y' : INT})

server = PvaServer('pair', pv)

x = 1
while True:
    pv['x'] = x
    pv['y'] = 2*x
    server.update(pv)
    sleep(1)
    x = x + 1
```

More Examples

Display Builder `pva_server_ramp`

Python code that serves ‘pva://ramp’ with alarm, prec, timestamp, …

Display Builder `table_server`

Python code that serves ‘pva://table’ as “NTTable”

→Impractical to replace all regular IOCs with python,
but useful when custom data is needed

OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

Ongoing Work

- PVA Gateway
- Access Security
- Normative Type details: ‘format’, precision, …
- Database: Support PVA links.

```
field(INP, “pva://other_record”)
```

EPICS 7

EPICS ‘base’

Records,
Device Support,
Channel Access,
softIoc

+

EPICS ‘V4’

PVAccess,
softIocPVA

- No need to worry about existing R3.x setups
- You may start using PVAccess
  - ☑ Images
  - ☑ Custom Data
- Good Python support
- CS-Studio is one of the early ‘bilingual’ tools