

# Introduction to EPICS

## For Operators and OPI Designers

# What is EPICS?

- Experimental Physics and Industrial Control System
- A Toolkit for distributed control systems
  - distributed = network
  - toolkit = some assembly required



# Terminology

- EPICS has network servers (IOCs) and clients (OPIs and IOCs)
  - **IOC** = Input Output Controller
    - The server application (sometimes the host computer)
  - **OPI** = Operator Interface
    - Generic term for client application
  - **CA** = Channel Access
    - The network protocol (analogous to HTTP)
  - **PV** = Process Variable
    - The unit/quantum of addressable data with CA protocol

# What is a PV?

- A name identifying some signal value
  - In hardware
    - A temperature or pressure reading
  - In software
    - connection status or error counter
- Meaning is subjective
  - To software it is just a name string which addresses a number

# NSLS2 PV name examples

- **UT:BR-Cu:2{Pmp:1}PD-I**
  - *Booster cooling water skid, copper system #2, pump #1, differential pressure indicator.*
- **LN-RF{KLY:1}Pwr:Fwd-I** 
  - *Linac RF klystron #2 forward power indicator*
- **BR-BI{FCT:1}SumQ-I**
  - *Booster Fast Current Transformer #1 total electron charge*
- **SR:C09-PS:RGB1{PS:CXM1B-ASM:XG-CH1}T:1-I** 
  - *SR cell #9 rack group B1 corrector power supply 1B heat exchanger, channel #1 temperature indicator (chain #1)*

6 of these signals

30,000 of these signals

# Naming Conventions

- PV names are (largely) free form
- The creator of the IOC should pick names which convey meaning.
- All sites have conventions(s)
- Configuring CA clients requires some familiarity with these
- When in doubt ask the IOC engineer

# Basic CA Client Tools

- Always available
- Simple and good for debugging, but not much more
- caput – Change a setting
- caget – Fetch the present value once
- camonitor – Watch the value until interrupted
- cainfo – Fetch diagnostic info

# Basic CA Client Tools (2)

Fetch value

It changes!

Write a new value

See how it changes  
In time

Useful information  
About this PV

```
$ caget testpv
testpv          77
$ caget testpv
testpv          81
$ caput testpv -42
Old : testpv    86
New : testpv    -42
$ caget testpv
testpv          -41
$ caget testpv
testpv          -36
$ camonitor testpv
testpv          2014-08-18 12:29:26.202117 -32
testpv          2014-08-18 12:29:27.202258 -31
testpv          2014-08-18 12:29:28.202319 -30
^C
$ cainfo testpv
testpv
  State:          connected
  Host:           localhost:5064
  Access:         read, write
  Native data type: DBF_DOUBLE
  Request type:   DBR_DOUBLE
  Element count:  1
$
```



# What comes with a PV?

- **Value**: Integer, Float, String
- Absolute **time** of last change
- **Alarm** state (severity and status codes)
- **Limits** (alarm, display, and control)
- **Units** string
- **Precision** (number of decimal digits)
- List of **states** strings (for enumeration)
  - eg. ['invalid','moving','closed','open']

# Investigating a PV

The PV is available

Which we could change

Has a scalar floating point value

Request **GR\_** graphics (aka display) information

Alarm inactive

Value is in megawatts

Two digits precision recommended

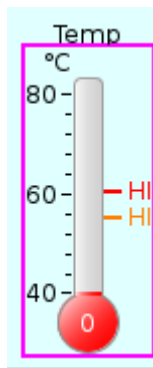
No limits defined

```
$ cainfo LN-RF{KLY:1}Pwr:Fwd-I
LN-RF{KLY:1}Pwr:Fwd-I
  State: connected
  Host: linacioc01.cs.nsls2.local:48449
  Access: read, write
  Native data type: DBF_DOUBLE
  Request type: DBR_DOUBLE
  Element count: 1
$ caget -d GR_DOUBLE LN-RF{KLY:1}Pwr:Fwd-I
LN-RF{KLY:1}Pwr:Fwd-I
  Native data type: DBF_DOUBLE
  Request type: DBR_GR_DOUBLE
  Element count: 1
  Value: 0
  Status: NO_ALARM
  Severity: NO_ALARM
  Units: MW
  Precision: 2
  Lo disp limit: 0
  Hi disp limit: 0
  Lo alarm limit: nan
  Lo warn limit: nan
  Hi warn limit: nan
  Hi alarm limit: nan
$
```

Provided by  
an IOC running on  
this computer

# Investigating a PV (2)

```
$ camonitor LN-RF:PB{Cav}T-I
LN-RF:PB{Cav}T-I          2014-08-06 15:18:32.987066 0 READ INVALID
```



Something is wrong here!

Alarm ranges are defined

Display range: 40 to 80 C

MINOR range: 1 to 55 C

MAJOR range: < 60 C

```
$ caget -d GR_DOUBLE LN-RF:PB{Cav}T-I
LN-RF:PB{Cav}T-I
Native data type: DBF_DOUBLE
Request type:     DBR_GR_DOUBLE
Element count:    1
Value:            0
Status:           READ
Severity:         INVALID
Units:            C
Precision:        1
Lo disp limit:    40
Hi disp limit:    80
Lo alarm limit:   nan
Lo warn limit:    1
Hi warn limit:    55
Hi alarm limit:   60
$
```

# Alarm State

- Severity
  - NO\_ALARM (0) - Normal
  - MINOR (1) – Warning (yellow/orange)
  - MAJOR (2) - Error condition (red)
  - INVALID (3) - Value not meaningful (white/violet)
    - eg. device is powered off or disconnected.
- Status
  - READ, WRITE, ...
  - Knowing status codes isn't as important

# Understanding Alarms

- Alarms are subjective
  - Not all MAJOR alarms are equal
- OPI clients can highlight alarming PVs with a colored border
- Specialized Alarm clients (ALH or Beast) which aggregate large number of alarms.