Python ‘IOC’

Kay Kasemir, Oct. 2018
https://github.com/paulscherrerinstitute/pcaspy

CA Server library for python

cd ~/epics-train/examples/python
python cas1.py
More Examples

- PV supporting put-callback
  
  `python cas2.py`

- CA Client as well as CA server
  
  `python cas3.py`

For more, see pcaspy documentation: Access security, data types, ..
Caveats

• CA server, no records
  – No xxx.RTYP, .EGU, .SCAN, ...

• No Autosave
  – DIY

• No shell with ‘dbl’, ‘dbpr’, ‘casr’
  – DIY
Use Cases

• Number Crunching
  – Numpy instead of CALC record

• Outside Data Source
  – Access web services, relational databases, .. and serve results

• Existing Python code
  – Turn script into IOC
Example: Proposal Database
Example: Computations
Energy Adjustment

<table>
<thead>
<tr>
<th>Desired Incident Energy:</th>
<th>17,000 meV</th>
<th>← Enter on desired energy starts update of motors &amp; choppers!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last successfully set</td>
<td>17,000 meV</td>
<td>Kill Air And Stop</td>
</tr>
<tr>
<td>Fermi Chopper Speed:</td>
<td>300 Hz</td>
<td>Flat Focusing Test Mode</td>
</tr>
<tr>
<td>Fermi speed, flat focusing become active the next time a desired energy is entered. Consider using Test Mode before actual energy update.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2018-10-24 16:00:53 INFO Would set BL14B:Chop:Skf4:EnergyReq = 17
2018-10-24 16:00:53 INFO Adjust positions for PG focus element
2018-10-24 16:00:53 INFO Current values m1pg 20.3678 deg, vm2 40.736 deg, mfp 0.897044 1/m
2018-10-24 16:00:53 INFO Moving to m1pg 19.0826 deg, mfp 0.983566 1/m
2018-10-24 16:00:53 INFO Would set BL14B:Mot:m1pg = 0.983566
2018-10-24 16:00:53 INFO Setting Ei to 17 meV
2018-10-24 16:00:53 INFO Setting T0=30 Hz and Fermi=300 Hz
2018-10-24 16:01:45 INFO T0 is already at requested frequency.
2018-10-24 16:14:50 INFO Fermi frequency is already at requested frequency
2018-10-24 16:14:50 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:14:50 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:14:50 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:22:23 INFO Adjust positions for PG focus element
2018-10-24 16:22:23 INFO Current values m1pg 20.3678 deg, vm2 40.736 deg, mfp 0.897044 1/m
2018-10-24 16:22:40 INFO Evaluating safety of moving drum shield to 38.1653 deg
2018-10-24 16:22:40 INFO Moving to vm2 38.1653 deg
2018-10-24 16:22:40 INFO PG Focus element angle at 19.0826 deg
2018-10-24 16:22:40 INFO PG Focus element focus at 9.98 1/m
2018-10-24 16:40:10 INFO PG Focus element angle at 38.1653 deg
2018-10-24 16:40:10 INFO PG Focus element focus at 9.98 1/m

- Could use sequencer, but had existing python code
What if data is “Table” or Structure?

Channel Access can only handle scalar or arrays of number & string

1. On server side (pcaspy), ‘pickle’ python data into CA byte waveform

2. On client side (CS-Studio), read byte waveform PV in display script, ‘unpickle’, then display in table or …
Handling Table, Structure

Yes, pickled byte waveform is a hack
- Requires scripts
- Only works with python as server & client

pvAccess can handle custom structures
- Better for server side
- Client likely needs more than dump of structure; Will still require script for user-friendly display.
Summary

Python with CA server & client libs can act as IOC

• Great tool to have

• Doesn’t replace all IOCs