The CSS Scan System

Kay-Uwe Kasemir,

Xihui Chen – SNS/ORNL

Eric Berryman - FRIB/MSU

Oct. 2013

SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy

U.S. DEPARTMENT OF
ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Experiment Components

- Equipment
  - Motors, Temperature Regulators, Detectors, ..
- Supervisory Control
  - Manually control and monitor
- Automation
  - Schedule ‘scan’ of crystal positions, temperatures, …
- Data Acquisition
  - Log data of interest during an experiment ‘run’
- Data Analysis
  - Turn logged data into scientific information

Experiment Components...

- Equipment
- Supervisory Control
- Automation
- Data Acquisition
- Data Analysis

CSS Scan System:

Modular, robust building block for experiment automation.

Interfaces to supervisory control, DAQ, ..., but as a separate component.

Ideally both Flexible and Robust

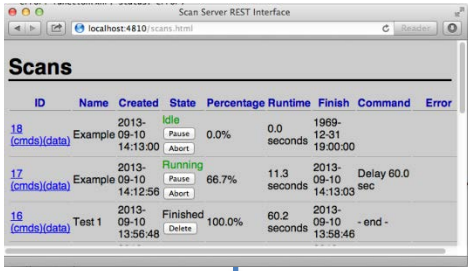
Flexible

- Control any beam line equipment
- Standard tasks:
  - X/Y scan of surface
  - CT-scan sample rotation
  - Temperature range scan
- Ad-hoc tasks:
  - While ramping Temperature, scan X/Y for each T
  - Scan table of T, X, Y

Robust

- Control Scan’s Live-cycle
  - Start/pause/resume/stop
  - Monitor progress
  - Simulate scan
  - Queue multiple scans
- Built-in checks
  - Are devices available?
  - Did device acknowledge?
  - Does read-back match?
  - Did access time out?

Scan Server



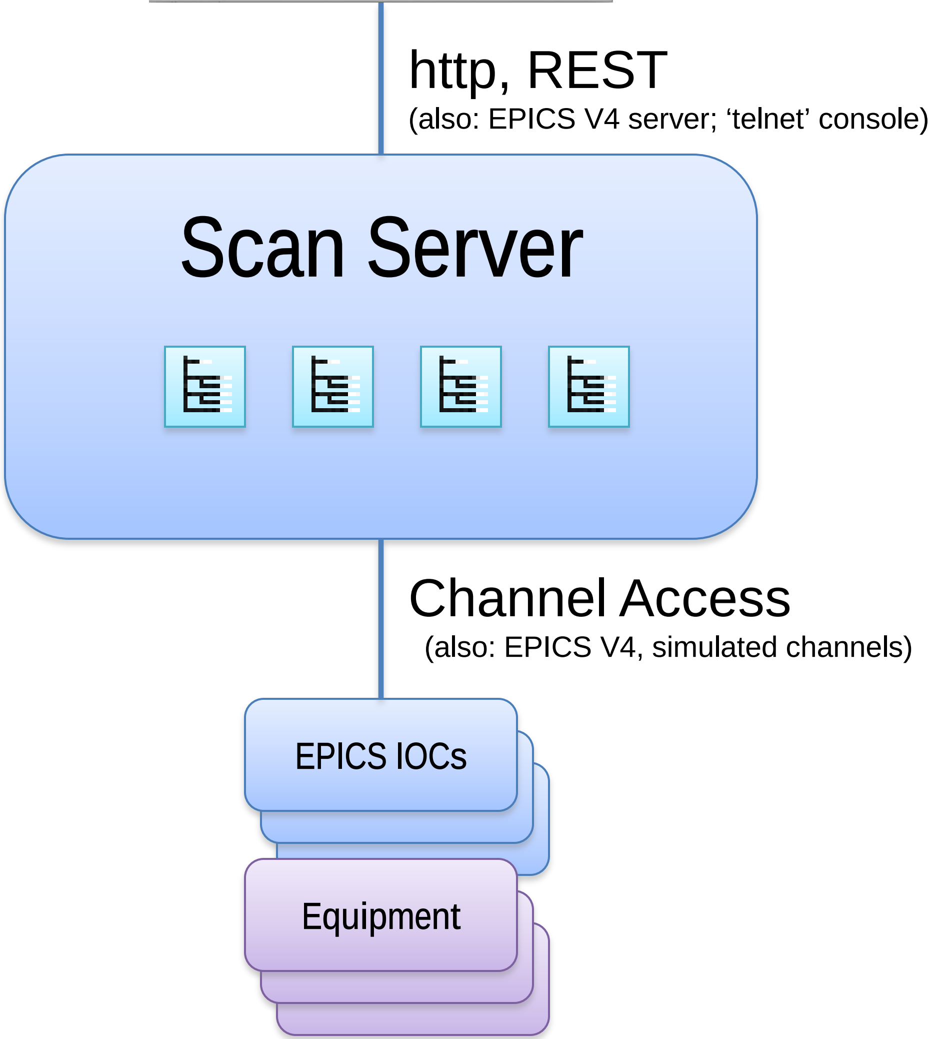
Scan Server REST Interface

localhost:4810/scans.html

Scans

ID	Name	Created	State	Percentage	Runtime	Finish	Command	Error
18 (cmds)(data)	Example	2013-09-10 14:13:00	Idle Pause Abort	0.0%	0.0 seconds	1969-12-31 19:00:00		
17 (cmds)(data)	Example	2013-09-10 14:12:56	Running Pause Abort	66.7%	11.3 seconds	2013-09-10 14:13:03	Delay 60.0 sec	
16 (cmds)(data)	Test 1	2013-09-10 13:56:48	Finished Delete	100.0%	60.2 seconds	2013-09-10 13:58:46	- end -	

Accepts 'Scans', executing them in order of submission



http, REST

(also: EPICS V4 server; 'telnet' console)

Channel Access

(also: EPICS V4, simulated channels)

Scan Commands

- Set device = value

  Optional: wait for completion, verify read-back, with timeout.

- Wait for device == value

  Optional: Wait until a device is ==, >, …, incremented-by a certain value, with timeout.

- Loop device = 1 .. 10, stepping 2

  Optional: completion, read-back and timeout.

- Log device1, device2

  Log data to monitor/analyze scan. No replacement for DAQ!

- Delay for … seconds

  Discouraged. Better: Wait for device to meet condition.

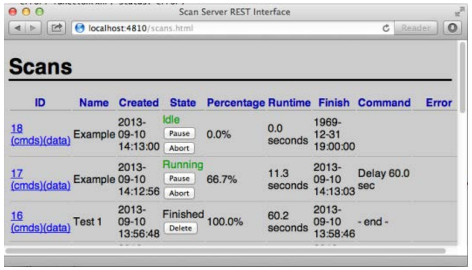
- Script

  –Execute Jython code. *Use with care.*

Custom commands can be added via Eclipse extension points.

Accessing the Scan Server

- Web Browser



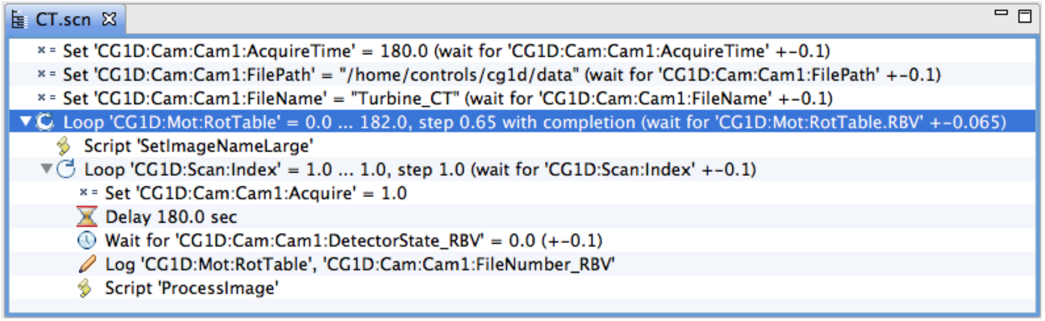
Scan Server REST Interface

localhost:4810/scans.html

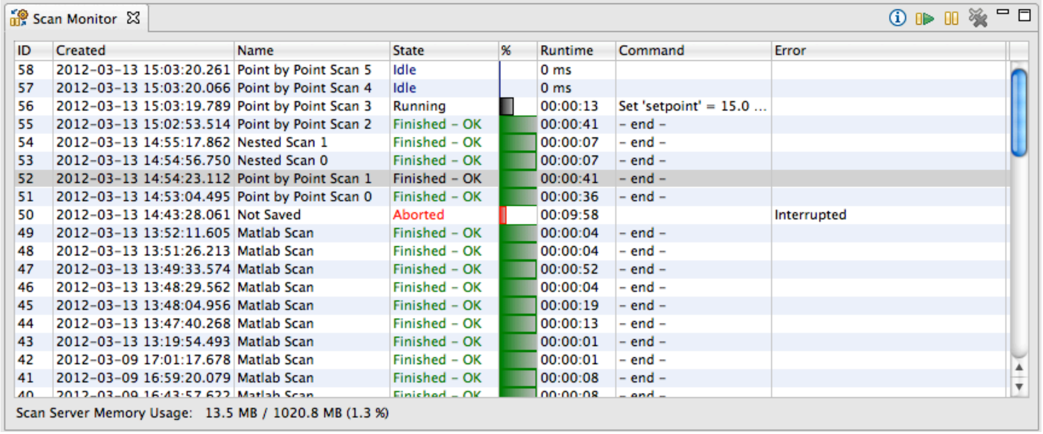
Scans

ID	Name	Created	State	Percentage	Runtime	Finish	Command	Error
18 (cmds)(data)	Example	2013-09-10 14:13:00	Idle Pause Abort	0.0%	0.0 seconds	1969-12-31 19:00:00		
17 (cmds)(data)	Example	2013-09-10 14:12:56	Running Pause Abort	66.7%	11.3 seconds	2013-09-10 14:13:03	Delay 60.0 sec	
16 (cmds)(data)	Test 1	2013-09-10 13:56:48	Finished Delete	100.0%	60.2 seconds	2013-09-10 13:58:46	- end -	

- Java
  - CSS Scan Editor, Scan Monitor
  - Jython (CSS *.opi)
  - Matlab



```
Set 'CG1D:Cam:Cam1:AcquireTime' = 180.0 (wait for 'CG1D:Cam:Cam1:AcquireTime' +-0.1)
Set 'CG1D:Cam:Cam1:FilePath' = "/home/controls/cg1d/data" (wait for 'CG1D:Cam:Cam1:FilePath' +-0.1)
Set 'CG1D:Cam:Cam1:FileName' = "Turbine_CT" (wait for 'CG1D:Cam:Cam1:FileName' +-0.1)
Loop 'CG1D:Mot:RotTable' = 0.0 ... 182.0, step 0.65 with completion (wait for 'CG1D:Mot:RotTable.RBV' +-0.065)
    Script 'SetImageNameLarge'
    Loop 'CG1D:Scan:Index' = 1.0 ... 1.0, step 1.0 (wait for 'CG1D:Scan:Index' +-0.1)
        Set 'CG1D:Cam:Cam1:Acquire' = 1.0
        Delay 180.0 sec
        Wait for 'CG1D:Cam:Cam1:DetectorState_RBV' = 0.0 (+-0.1)
        Log 'CG1D:Mot:RotTable', 'CG1D:Cam:Cam1:FileNumber_RBV'
        Script 'ProcessImage'
```



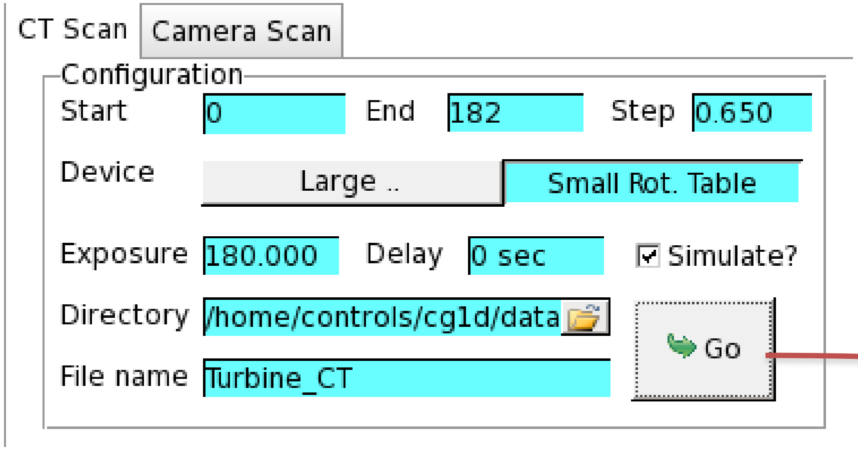
ID	Created	Name	State	%	Runtime	Command	Error
58	2012-03-13 15:03:20.261	Point by Point Scan 5	Idle		0 ms		
57	2012-03-13 15:03:20.066	Point by Point Scan 4	Idle		0 ms		
56	2012-03-13 15:03:19.789	Point by Point Scan 3	Running	▪	00:00:13	Set 'setpoint' = 15.0 ...	
55	2012-03-13 15:02:53.514	Point by Point Scan 2	Finished - OK	🟩	00:00:41	- end -	
54	2012-03-13 14:55:17.862	Nested Scan 1	Finished - OK	🟩	00:00:07	- end -	
53	2012-03-13 14:54:56.750	Nested Scan 0	Finished - OK	🟩	00:00:07	- end -	
52	2012-03-13 14:54:23.112	Point by Point Scan 1	Finished - OK	🟩	00:00:41	- end -	
51	2012-03-13 14:53:04.495	Point by Point Scan 0	Finished - OK	🟩	00:00:36	- end -	
50	2012-03-13 14:43:28.061	Not Saved	Aborted	🟥	00:09:58		Interrupted
49	2012-03-13 13:52:11.605	Matlab Scan	Finished - OK	🟩	00:00:04	- end -	
48	2012-03-13 13:51:26.213	Matlab Scan	Finished - OK	🟩	00:00:04	- end -	
47	2012-03-13 13:49:33.574	Matlab Scan	Finished - OK	🟩	00:00:52	- end -	
46	2012-03-13 13:48:29.562	Matlab Scan	Finished - OK	🟩	00:00:04	- end -	
45	2012-03-13 13:48:04.956	Matlab Scan	Finished - OK	🟩	00:00:19	- end -	
44	2012-03-13 13:47:40.268	Matlab Scan	Finished - OK	🟩	00:00:13	- end -	
43	2012-03-13 13:19:54.493	Matlab Scan	Finished - OK	🟩	00:00:01	- end -	
42	2012-03-09 17:01:17.678	Matlab Scan	Finished - OK	🟩	00:00:01	- end -	
41	2012-03-09 16:59:20.079	Matlab Scan	Finished - OK	🟩	00:00:08	- end -	
40	2012-03-09 16:43:57.622	Matlab Scan	Finished - OK	🟩	00:00:08	- end -	

Scan Server Memory Usage: 13.5 MB / 1020.8 MB (1.3 %)

- Any custom code

  .. with ‘http’ library

CSS User Interfaces for Scans



CT Scan | Camera Scan

Configuration

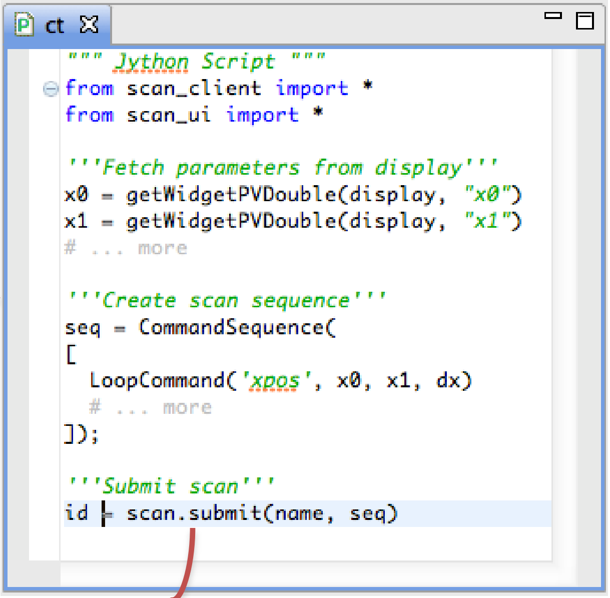
Start 0 End 182 Step 0.650

Device Large .. | Small Rot. Table

Exposure 180.000 Delay 0 sec ☑ Simulate?

Directory /home/controls/cg1d/data

File name Turbine_CT



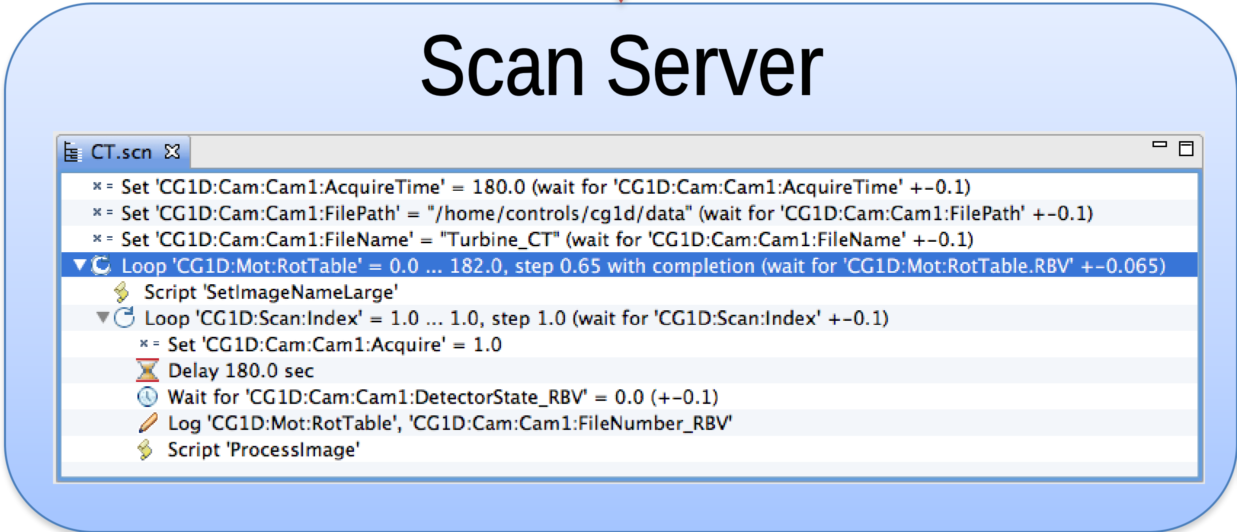
ct

```
""" Jython Script """
from scan_client import *
from scan_ui import *

'''Fetch parameters from display'''
x0 = getWidgetPVDouble(display, "x0")
x1 = getWidgetPVDouble(display, "x1")
# ... more

'''Create scan sequence'''
seq = CommandSequence(
[
  LoopCommand('xpos', x0, x1, dx)
  # ... more
]);

'''Submit scan'''
id = scan.submit(name, seq)
```

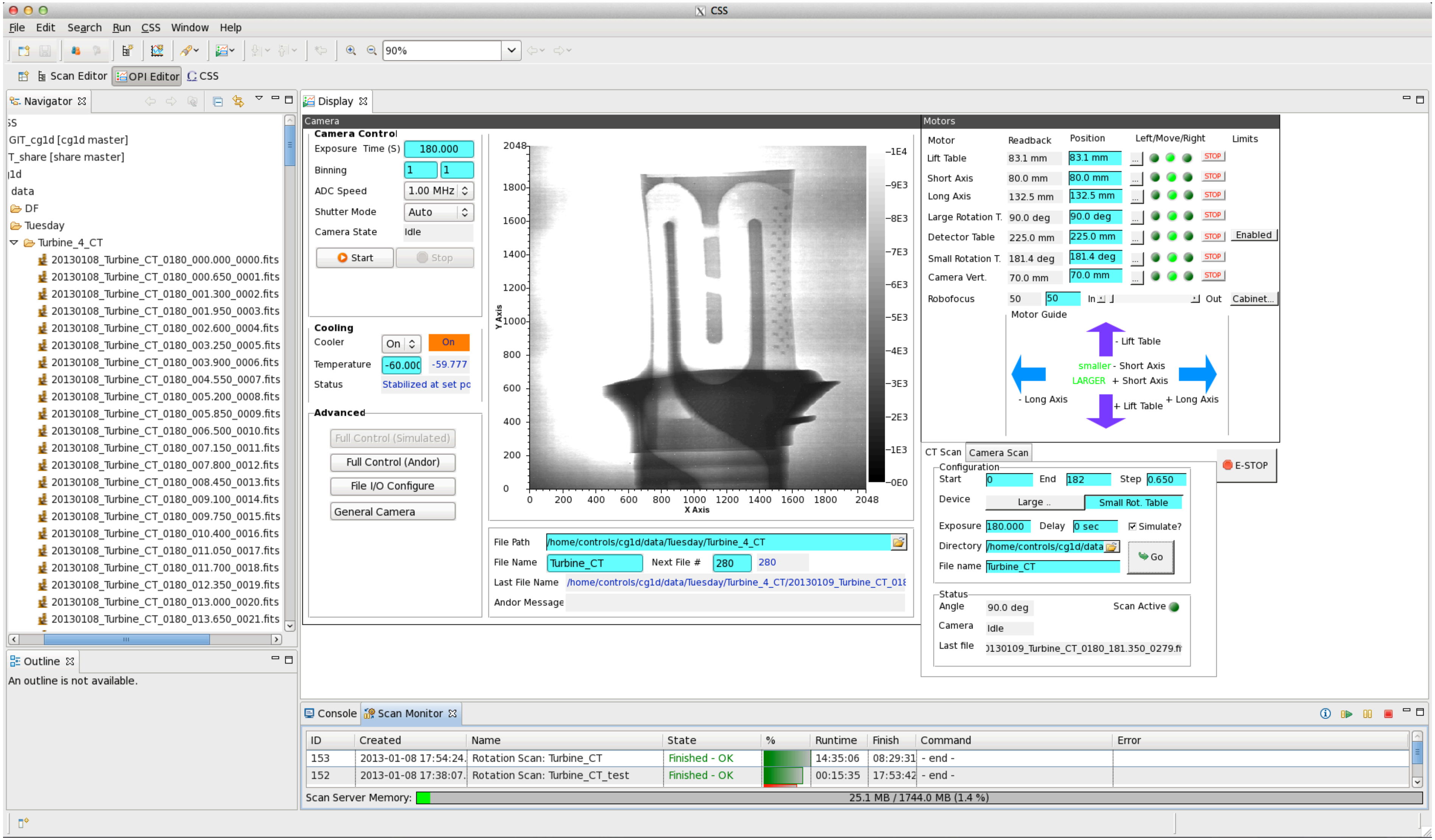


Scan Server

CT.scn

```
  Set 'CG1D:Cam:Cam1:AcquireTime' = 180.0 (wait for 'CG1D:Cam:Cam1:AcquireTime' +-0.1)
  Set 'CG1D:Cam:Cam1:FilePath' = "/home/controls/cg1d/data" (wait for 'CG1D:Cam:Cam1:FilePath' +-0.1)
  Set 'CG1D:Cam:Cam1:FileName' = "Turbine_CT" (wait for 'CG1D:Cam:Cam1:FileName' +-0.1)
▼ Loop 'CG1D:Mot:RotTable' = 0.0 ... 182.0, step 0.65 with completion (wait for 'CG1D:Mot:RotTable.RBV' +-0.065)
    Script 'SetImageNameLarge'
  ▼ Loop 'CG1D:Scan:Index' = 1.0 ... 1.0, step 1.0 (wait for 'CG1D:Scan:Index' +-0.1)
      Set 'CG1D:Cam:Cam1:Acquire' = 1.0
      Delay 180.0 sec
      Wait for 'CG1D:Cam:Cam1:DetectorState_RBV' = 0.0 (+-0.1)
      Log 'CG1D:Mot:RotTable', 'CG1D:Cam:Cam1:FileNumber_RBV'
      Script 'ProcessImage'
```

Beam Line Example: CT



Navigator:

- ;S
- GIT_cg1d [cg1d master]
- T_share [share master]
- ı1d
- data
- DF
- Tuesday
- Turbine_4_CT
  - 20130108_Turbine_CT_0180_000.000_0000.fits
  - 20130108_Turbine_CT_0180_000.650_0001.fits
  - 20130108_Turbine_CT_0180_001.300_0002.fits
  - 20130108_Turbine_CT_0180_001.950_0003.fits
  - 20130108_Turbine_CT_0180_002.600_0004.fits
  - 20130108_Turbine_CT_0180_003.250_0005.fits
  - 20130108_Turbine_CT_0180_003.900_0006.fits
  - 20130108_Turbine_CT_0180_004.550_0007.fits
  - 20130108_Turbine_CT_0180_005.200_0008.fits
  - 20130108_Turbine_CT_0180_005.850_0009.fits
  - 20130108_Turbine_CT_0180_006.500_0010.fits
  - 20130108_Turbine_CT_0180_007.150_0011.fits
  - 20130108_Turbine_CT_0180_007.800_0012.fits
  - 20130108_Turbine_CT_0180_008.450_0013.fits
  - 20130108_Turbine_CT_0180_009.100_0014.fits
  - 20130108_Turbine_CT_0180_009.750_0015.fits
  - 20130108_Turbine_CT_0180_010.400_0016.fits
  - 20130108_Turbine_CT_0180_011.050_0017.fits
  - 20130108_Turbine_CT_0180_011.700_0018.fits
  - 20130108_Turbine_CT_0180_012.350_0019.fits
  - 20130108_Turbine_CT_0180_013.000_0020.fits
  - 20130108_Turbine_CT_0180_013.650_0021.fits

Outline: An outline is not available.

Camera Control

- Exposure Time (S): 180.000
- Binning: 1 1
- ADC Speed: 1.00 MHz
- Shutter Mode: Auto
- Camera State: Idle
- Start / Stop

Cooling

- Cooler: On — On
- Temperature: -60.000 — -59.777
- Status: Stabilized at set po

Advanced

- Full Control (Simulated)
- Full Control (Andor)
- File I/O Configure
- General Camera

- 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_01ε
- Andor Message:

Motors

Motor	Readback	Position	Left/Move/Right	Limits
Lift Table	83.1 mm	83.1 mm	... ● ● ● STOP	
Short Axis	80.0 mm	80.0 mm	... ● ● ● STOP	
Long Axis	132.5 mm	132.5 mm	... ● ● ● STOP	
Large Rotation T.	90.0 deg	90.0 deg	... ● ● ● STOP	
Detector Table	225.0 mm	225.0 mm	... ● ● ● STOP	Enabled
Small Rotation T.	181.4 deg	181.4 deg	... ● ● ● STOP	
Camera Vert.	70.0 mm	70.0 mm	... ● ● ● STOP	
Robofocus	50	50	In — Out	Cabinet...

Motor Guide

CT Scan | Camera Scan

Configuration

- Start: 0 — End: 182 — Step: 0.650
- Device: Large .. / Small Rot. Table
- Exposure: 180.000 — Delay: 0 sec — ☑ Simulate?
- Directory: /home/controls/cg1d/data
- File name: Turbine_CT
- Go

E-STOP

Status

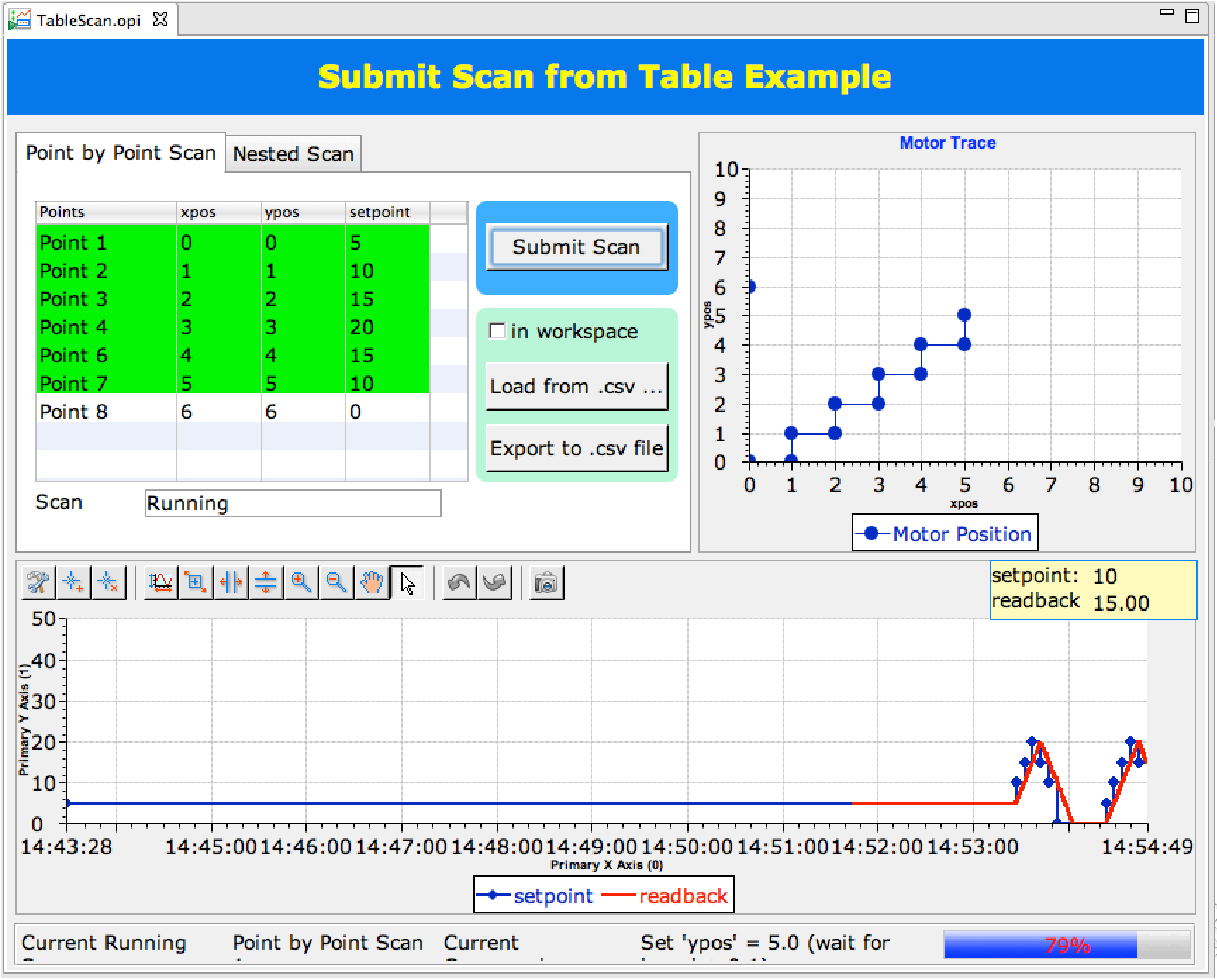
- Angle: 90.0 deg — Scan Active ●
- Camera: Idle
- Last file:)130109_Turbine_CT_0180_181.350_0279.fi

Scan Monitor

ID	Created	Name	State	%	Runtime	Finish	Command	Error
153	2013-01-08 17:54:24.	Rotation Scan: Turbine_CT	Finished - OK		14:35:06	08:29:31	- end -	
152	2013-01-08 17:38:07.	Rotation Scan: Turbine_CT_test	Finished - OK		00:15:35	17:53:42	- end -	

Scan Server Memory: 25.1 MB / 1744.0 MB (1.4 %)

- ORNL HFIR CG-1D



TableScan.opi

Submit Scan from Table Example

Point by Point Scan | Nested Scan

Points	xpos	ypos	setpoint
Point 1	0	0	5
Point 2	1	1	10
Point 3	2	2	15
Point 4	3	3	20
Point 6	4	4	15
Point 7	5	5	10
Point 8	6	6	0

Scan: Running

Submit Scan

□ in workspace

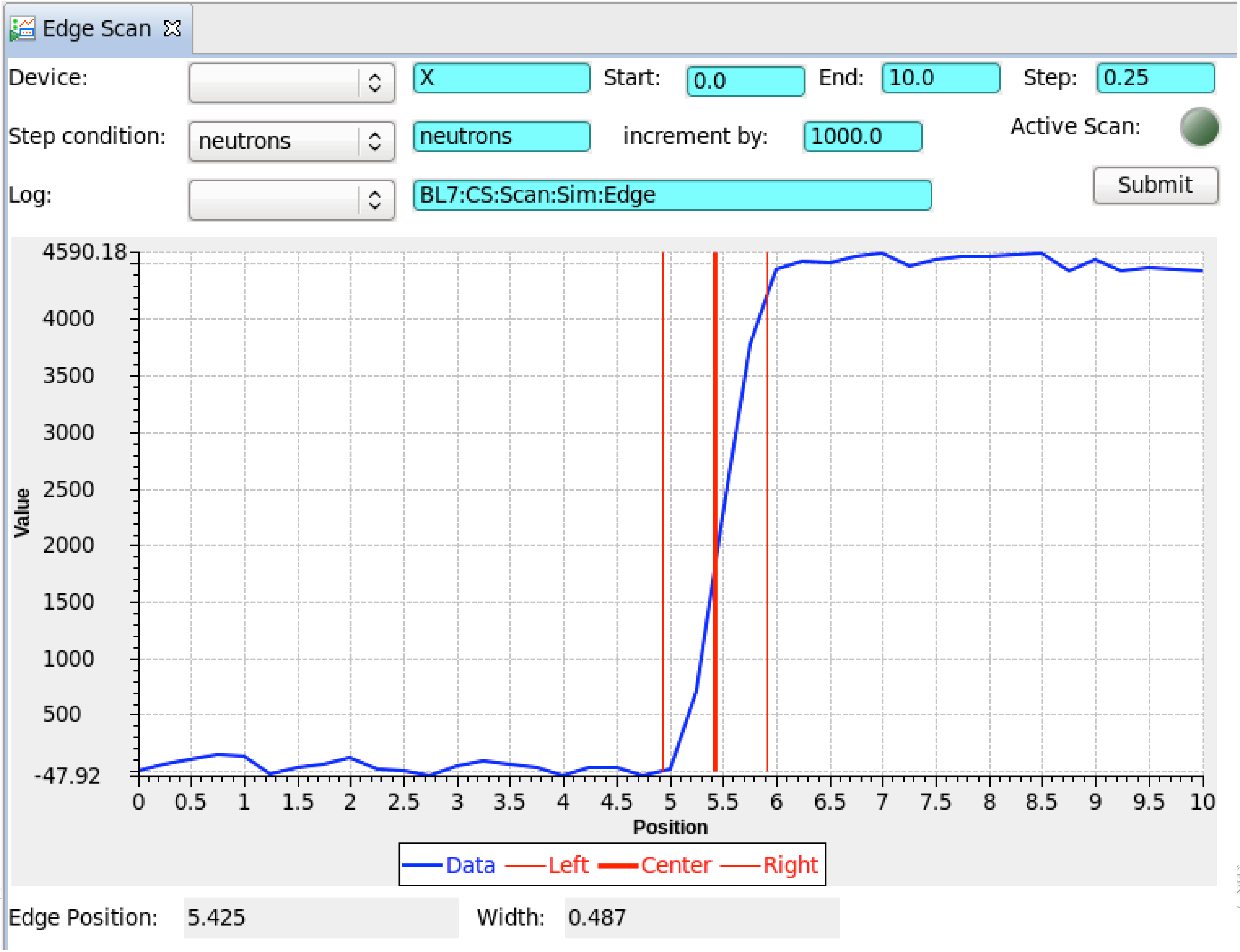
Load from .csv ...

Export to .csv file

setpoint: 10
readback 15.00

Current Running | Point by Point Scan | Current | Set 'ypos' = 5.0 (wait for | 79%

"Range" Scan with Edge Detection



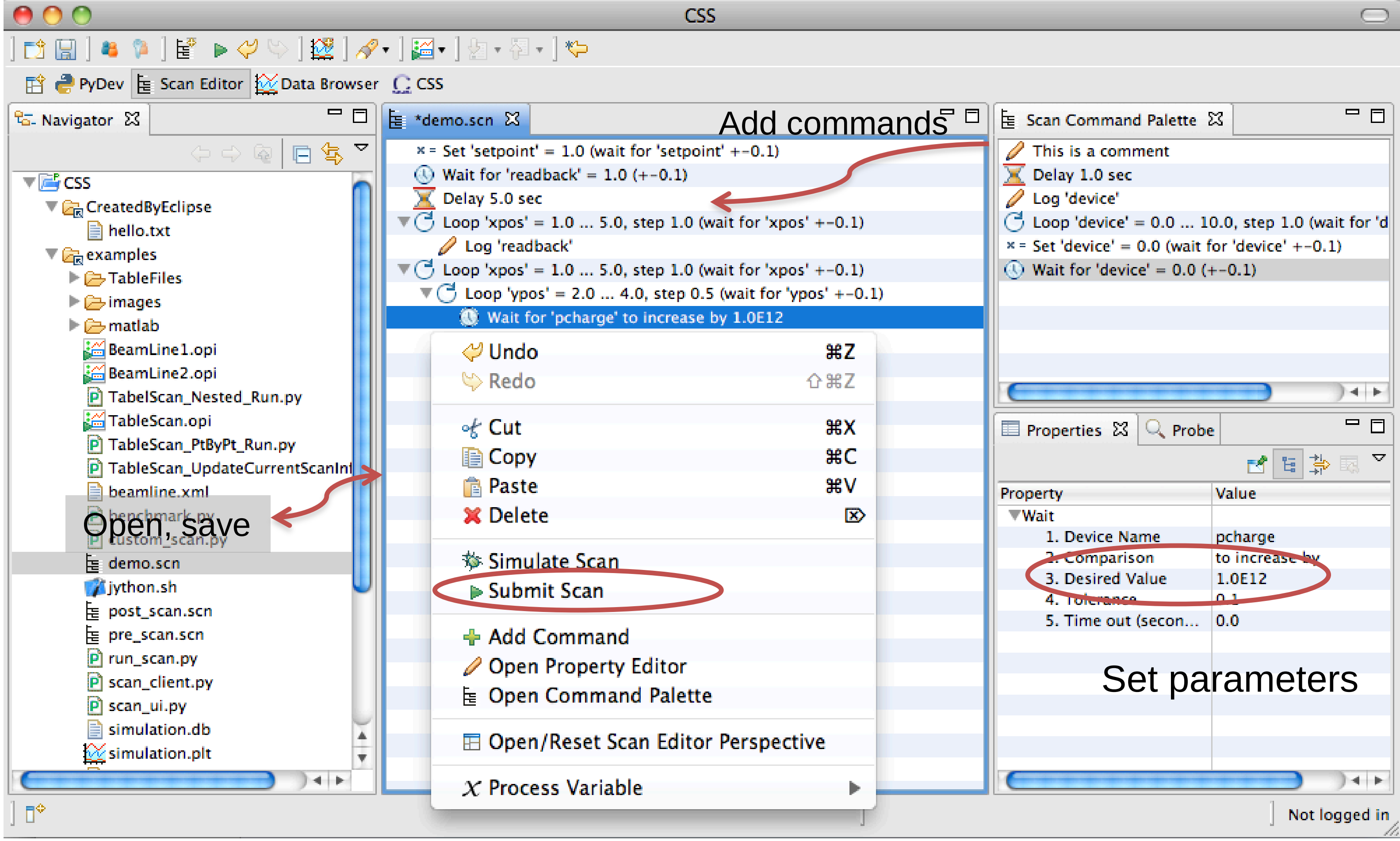
Device: X Start: 0.0 End: 10.0 Step: 0.25

Step condition: neutrons neutrons increment by: 1000.0 Active Scan: ●

Log: BL7:CS:Scan:Sim:Edge

Edge Position: 5.425 Width: 0.487

Scan Editor



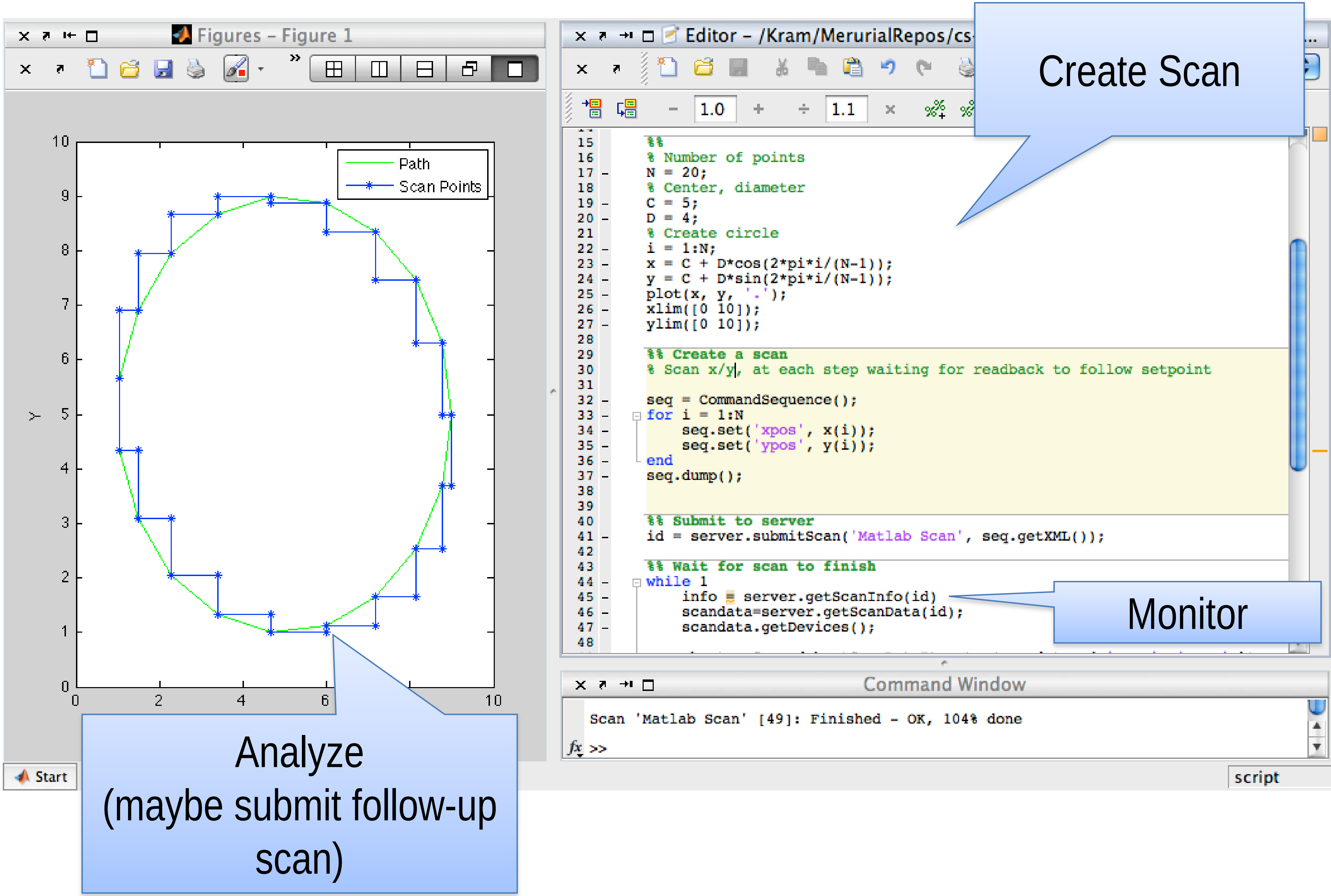
Set 'setpoint' = 1.0 (wait for 'setpoint' +-0.1)
Wait for 'readback' = 1.0 (+-0.1)
Delay 5.0 sec
Loop 'xpos' = 1.0 ... 5.0, step 1.0 (wait for 'xpos' +-0.1)
  Log 'readback'
Loop 'xpos' = 1.0 ... 5.0, step 1.0 (wait for 'xpos' +-0.1)
  Loop 'ypos' = 2.0 ... 4.0, step 0.5 (wait for 'ypos' +-0.1)
    Wait for 'pcharge' to increase by 1.0E12

This is a comment
Delay 1.0 sec
Log 'device'
Loop 'device' = 0.0 ... 10.0, step 1.0 (wait for 'd
Set 'device' = 0.0 (wait for 'device' +-0.1)
Wait for 'device' = 0.0 (+-0.1)

Property	Value
Wait	
1. Device Name	pcharge
2. Comparison	to increase by
3. Desired Value	1.0E12
4. Tolerance	0.1
5. Time out (secon...	0.0

- “Undo”
- Drag/drop commands or PV names (also as XML text)
- Device PVs (or alias) can be picked from beamline-specific configuration

Matlab



```
%%
% Number of points
N = 20;
% Center, diameter
C = 5;
D = 4;
% Create circle
i = 1:N;
x = C + D*cos(2*pi*i/(N-1));
y = C + D*sin(2*pi*i/(N-1));
plot(x, y, '.');
xlim([0 10]);
ylim([0 10]);

%% Create a scan
% Scan x/y, at each step waiting for readback to follow setpoint

seq = CommandSequence();
for i = 1:N
    seq.set('xpos', x(i));
    seq.set('ypos', y(i));
end
seq.dump();


%% Submit to server
id = server.submitScan('Matlab Scan', seq.getXML());

%% Wait for scan to finish
while 1
    info = server.getScanInfo(id)
    scandata=server.getScanData(id);
    scandata.getDevices();
```

```
Scan 'Matlab Scan' [49]: Finished - OK, 104% done
>>
```

Create Scan

Monitor

Analyze
(maybe submit follow-up scan)

Ongoing Work

Separation of Scan Execution and GUI

✓ Good for stability

– Added work

“Start”: GUI → Script to submit scan → Scan Server

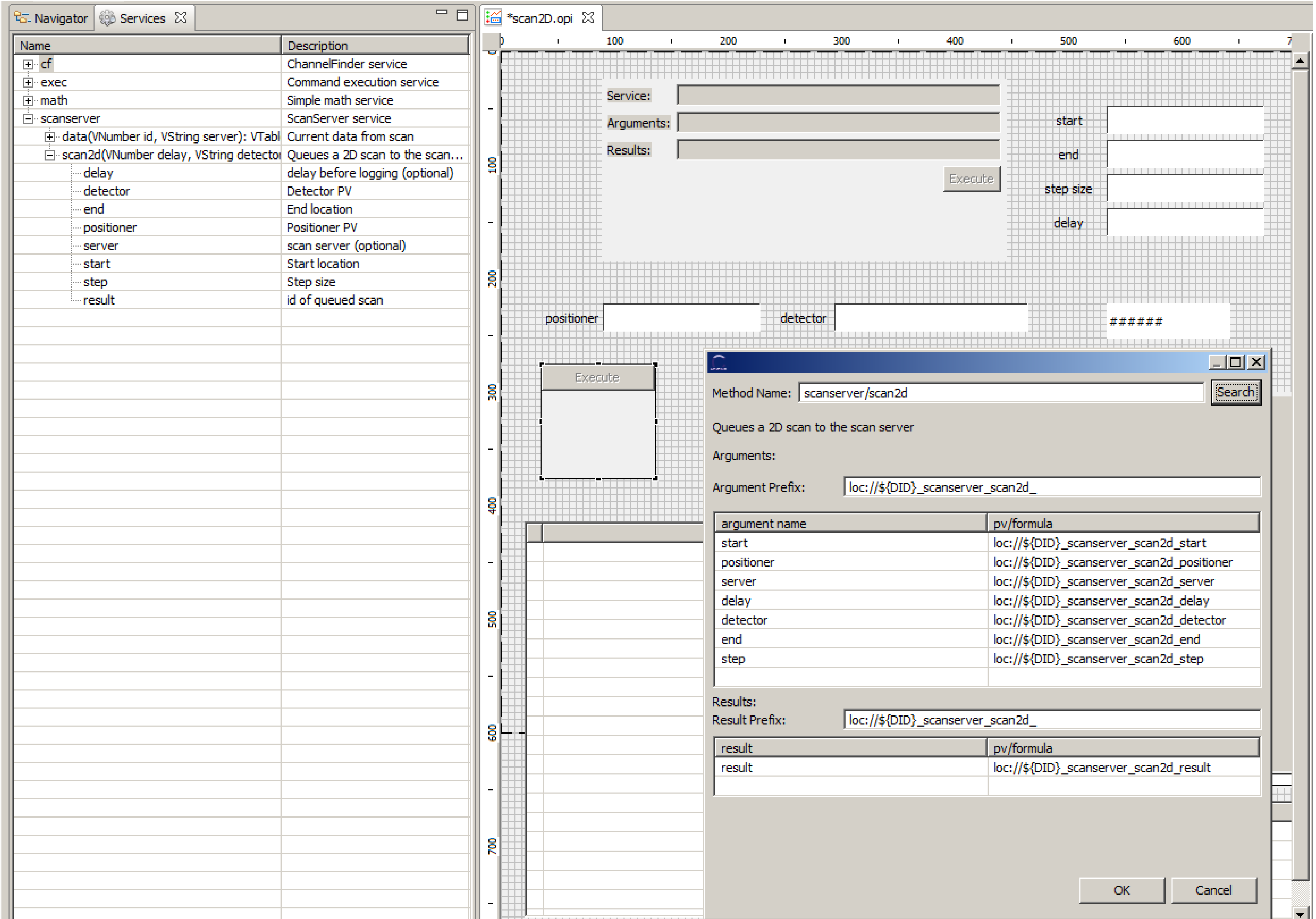
“Plot”: Scan writes to PVs → BOY displays PVs

Simpler interfacing via PVManager and V4 Data Types

✓ “Start”: Invoke pvmanager ‘service’ for 1D-scan

✓ “Plot”: Read scan log as V4 Table ‘pva:scanData?id=47’

Submit Scan via PVManager “Service”



Name	Description
cf	ChannelFinder service
exec	Command execution service
math	Simple math service
scanserver	ScanServer service
data(VNumber id, VString server): VTabl	Current data from scan
scan2d(VNumber delay, VString detecto	Queues a 2D scan to the scan…
delay	delay before logging (optional)
detector	Detector PV
end	End location
positioner	Positioner PV
server	scan server (optional)
start	Start location
step	Step size
result	id of queued scan

Service:

Arguments:

Results:

Execute

start

end

step size

delay

positioner

detector

######

Execute

Method Name: scanserver/scan2d Search

Queues a 2D scan to the scan server

Arguments:

Argument Prefix: loc://${DID}_scanserver_scan2d_

argument name	pv/formula
start	loc://${DID}_scanserver_scan2d_start
positioner	loc://${DID}_scanserver_scan2d_positioner
server	loc://${DID}_scanserver_scan2d_server
delay	loc://${DID}_scanserver_scan2d_delay
detector	loc://${DID}_scanserver_scan2d_detector
end	loc://${DID}_scanserver_scan2d_end
step	loc://${DID}_scanserver_scan2d_step

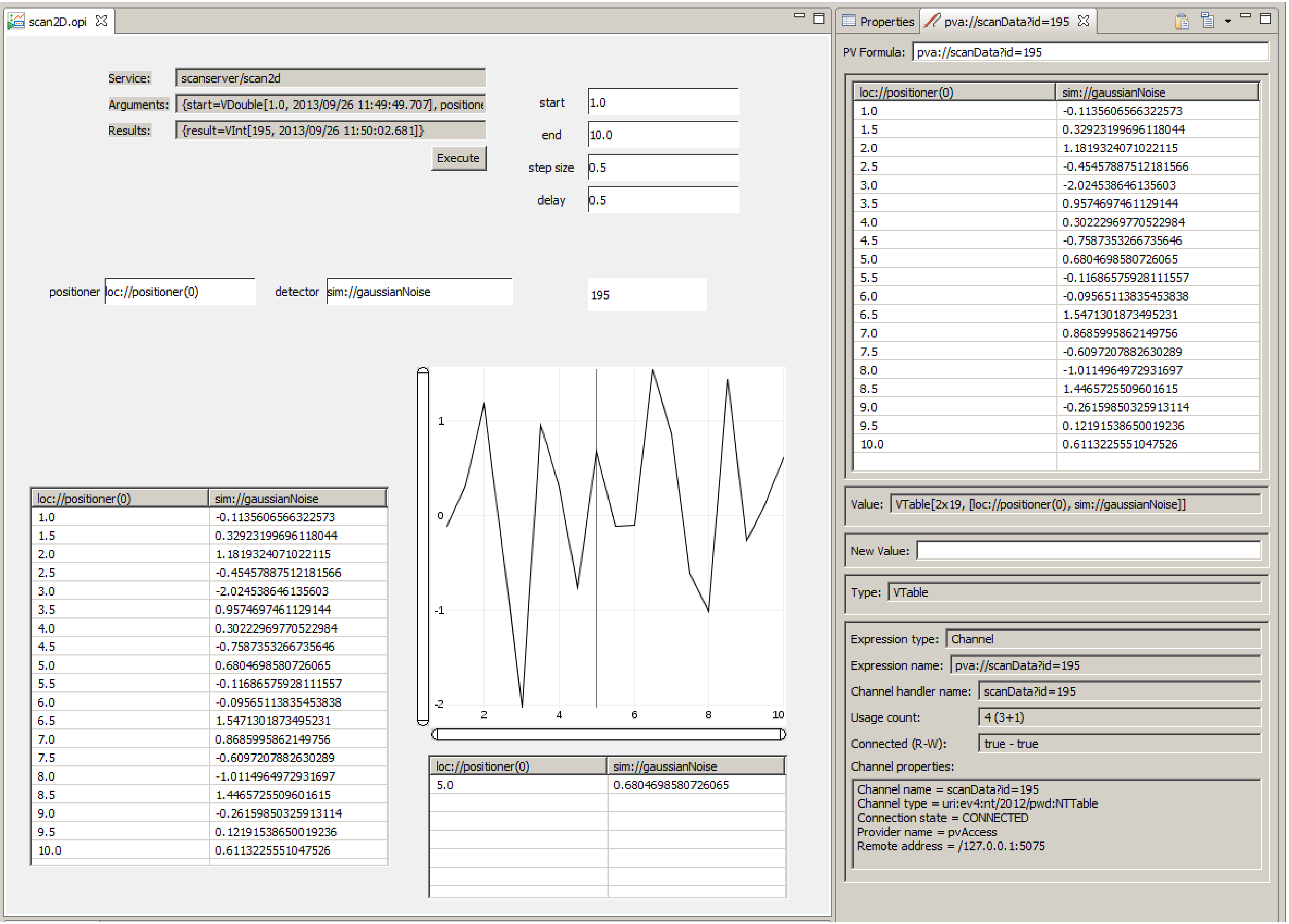
Results:

Result Prefix: loc://${DID}_scanserver_scan2d_

result	pv/formula
result	loc://${DID}_scanserver_scan2d_result

OK Cancel

PVAccess & PVManager for Status & Log



Service: scanserver/scan2d

Arguments: {start=VDouble[1.0, 2013/09/26 11:49:49.707], positione

Results: {result=VInt[195, 2013/09/26 11:50:02.681]}

start 1.0

end 10.0

step size 0.5

delay 0.5

positioner loc://positioner(0)

detector sim://gaussianNoise

195

loc://positioner(0)	sim://gaussianNoise
1.0	-0.1135606566322573
1.5	0.3292319969618044
2.0	1.1819324071022115
2.5	-0.45457887512181566
3.0	-2.024538646135603
3.5	0.9574697461129144
4.0	0.30222969770522984
4.5	-0.7587353266735646
5.0	0.6804698580726065
5.5	-0.11686575928111557
6.0	-0.09565113835453838
6.5	1.5471301873495231
7.0	0.8685995862149756
7.5	-0.6097207882630289
8.0	-1.0114964972931697
8.5	1.4465725509601615
9.0	-0.26159850325913114
9.5	0.12191538650019236
10.0	0.6113225551047526

loc://positioner(0)	sim://gaussianNoise
5.0	0.6804698580726065

PV Formula: pva://scanData?id=195

loc://positioner(0)	sim://gaussianNoise
1.0	-0.1135606566322573
1.5	0.3292319969618044
2.0	1.1819324071022115
2.5	-0.45457887512181566
3.0	-2.024538646135603
3.5	0.9574697461129144
4.0	0.30222969770522984
4.5	-0.7587353266735646
5.0	0.6804698580726065
5.5	-0.11686575928111557
6.0	-0.09565113835453838
6.5	1.5471301873495231
7.0	0.8685995862149756
7.5	-0.6097207882630289
8.0	-1.0114964972931697
8.5	1.4465725509601615
9.0	-0.26159850325913114
9.5	0.12191538650019236
10.0	0.6113225551047526

Value: VTable[2x19, [loc://positioner(0), sim://gaussianNoise]]

New Value:

Type: VTable

Expression type: Channel

Expression name: pva://scanData?id=195

Channel handler name: scanData?id=195

Usage count: 4 (3+1)

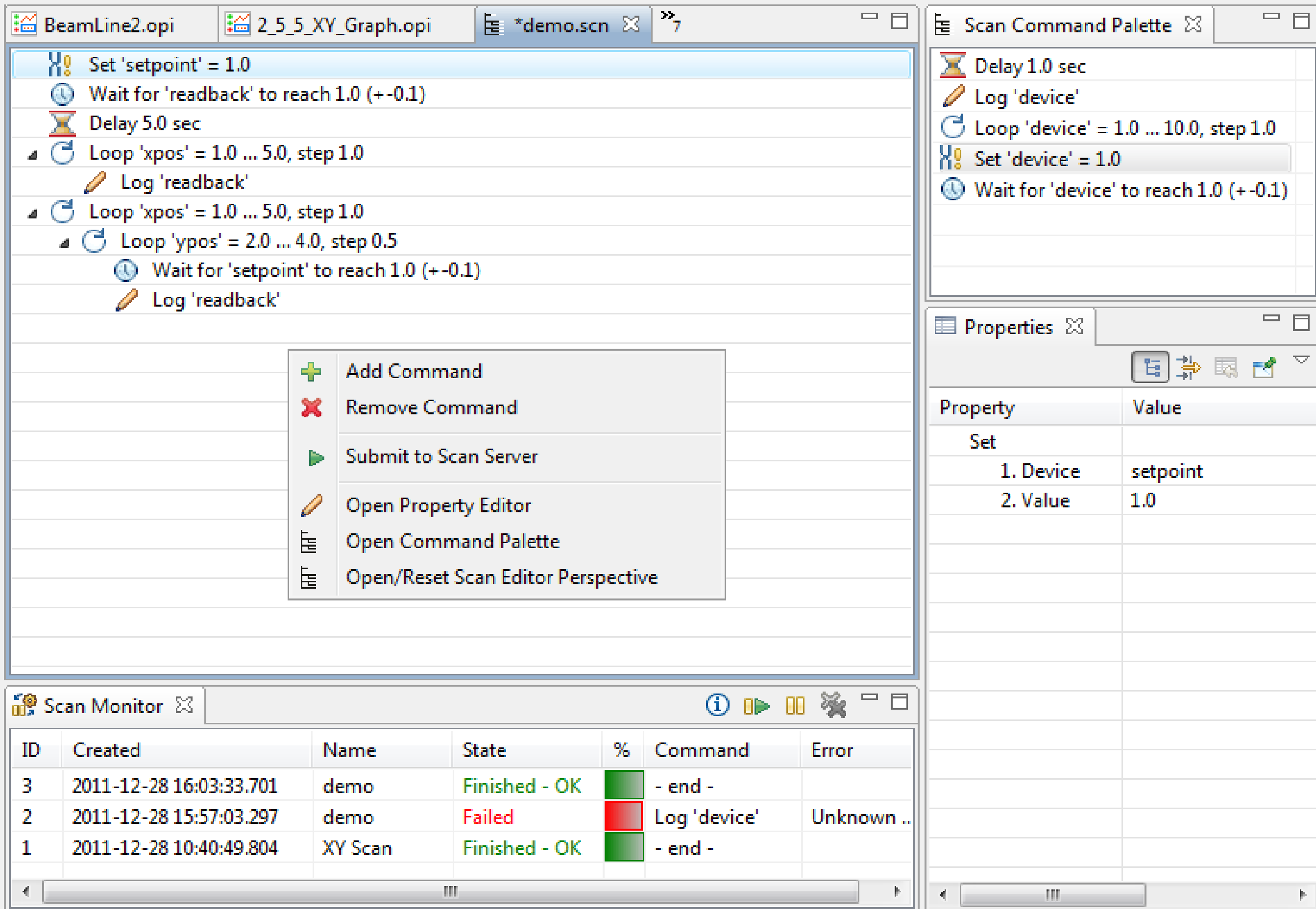
Connected (R-W): true - true

Channel properties:

Channel name = scanData?id=195
Channel type = uri:ev4:nt/2012/pwd:NTTable
Connection state = CONNECTED
Provider name = pvAccess
Remote address = /127.0.0.1:5075

Summary: CSS Scan System

Modular, robust building block for experiment automation.



```
Set 'setpoint' = 1.0
Wait for 'readback' to reach 1.0 (+-0.1)
Delay 5.0 sec
Loop 'xpos' = 1.0 ... 5.0, step 1.0
    Log 'readback'
Loop 'xpos' = 1.0 ... 5.0, step 1.0
    Loop 'ypos' = 2.0 ... 4.0, step 0.5
        Wait for 'setpoint' to reach 1.0 (+-0.1)
        Log 'readback'
```

```
Delay 1.0 sec
Log 'device'
Loop 'device' = 1.0 ... 10.0, step 1.0
Set 'device' = 1.0
Wait for 'device' to reach 1.0 (+-0.1)
```

Property	Value
Set	
1. Device	setpoint
2. Value	1.0

ID	Created	Name	State	%	Command	Error
3	2011-12-28 16:03:33.701	demo	Finished - OK	🟩	- end -	
2	2011-12-28 15:57:03.297	demo	Failed	🟥	Log 'device'	Unknown ..
1	2011-12-28 10:40:49.804	XY Scan	Finished - OK	🟩	- end -	