Data Collection - J.K. Zhao, zhaoj@ornl.gov

Step 1. Start NewPyDas

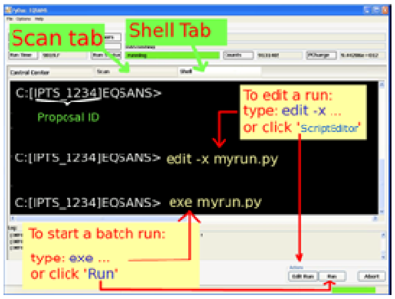
Click this icon on desktop:

Shortcut to NewPyDas.bat

You will be your proposal folder.

If you do not see your proposal ID here, type *gohome()* at the prompt.

PyDas (integrated control)

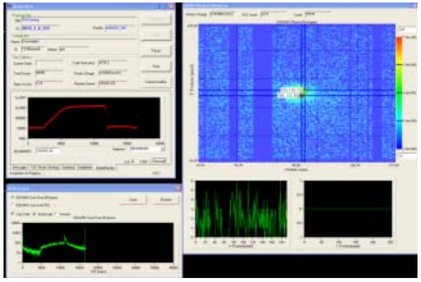


```
C:[IPTS_1234]EQSANS>

C:[IPTS_1234]EQSANS> edit -x myrun.py

C:[IPTS_1234]EQSANS> exe myrun.py
```

To edit a run:
type: edit -x ...
or click 'ScriptEditor'

To start a batch run:
type: exe ...
or click 'Run'

DComClient (data acquisition)



Step 2.

Edit a batch run script (python script):

Type:

edit -x myrun.py

Or click 'ScriptEditor'

'myrun.py' is a file where data collection commands are stored (see below for details)

Step 3. Start a batch Run:

Type:

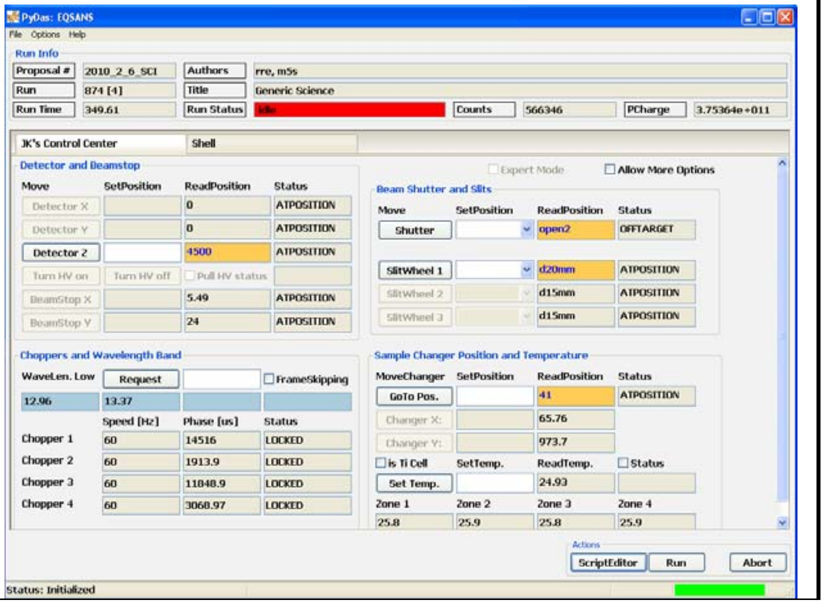
exe myrun.py

Or click: 'Run' and select a script to run.

The following programs are started by *NewPyDas* *as well:*

- **Decomclient (Top Right):** *where data collection takes place. It can be used to manually start/stop data collection*
- **Chopper control**
- **Motor control**
- **Sample environment temperature control**

Most often used instrument parameters can be controlled via the 'control panel' tab:



Run Info

Proposal #	2010_2_6_SCI	Authors	rre, m5s				
Run	874 [4]	Title	Generic Science				
Run Time	349.61	Run Status	idle	Counts	566346	PCharge	3.75364e+011

□ Expert Mode □ Allow More Options

Detector and Beamstop

Move	SetPosition	ReadPosition	Status
Detector X		0	ATPOSITION
Detector Y		0	ATPOSITION
Detector Z		4500	ATPOSITION
Turn HV on	Turn HV off	□ Pull HV status	
BeamStop X		5.49	ATPOSITION
BeamStop Y		24	ATPOSITION

Beam Shutter and Slits

Move	SetPosition	ReadPosition	Status
Shutter		open2	OFFTARGET
SlitWheel 1		d20mm	ATPOSITION
SlitWheel 2		d15mm	ATPOSITION
SlitWheel 3		d15mm	ATPOSITION

Choppers and Wavelength Band

WaveLen. Low	Request		□ FrameSkipping
12.96	13.37		
	Speed [Hz]	Phase [us]	Status
Chopper 1	60	14516	LOCKED
Chopper 2	60	1913.9	LOCKED
Chopper 3	60	11848.9	LOCKED
Chopper 4	60	3068.97	LOCKED

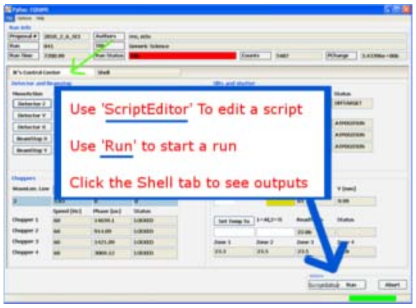
Sample Changer Position and Temperature

MoveChanger	SetPosition	ReadPosition	Status
GoTo Pos.		41	ATPOSITION
Changer X:		65.76	
Changer Y:		973.7	
□ is Ti Cell	SetTemp.	ReadTemp.	□ Status
Set Temp.		24.93	
Zone 1	Zone 2	Zone 3	Zone 4
25.8	25.9	25.8	25.9

Status: Initialized

Edit EQSANS Scan Scripts Using the ScriptEditor

(1) Start the ScriptEditor by clicking the 'ScriptEditor' Button in the PyDas window:



Use 'ScriptEditor' To edit a script

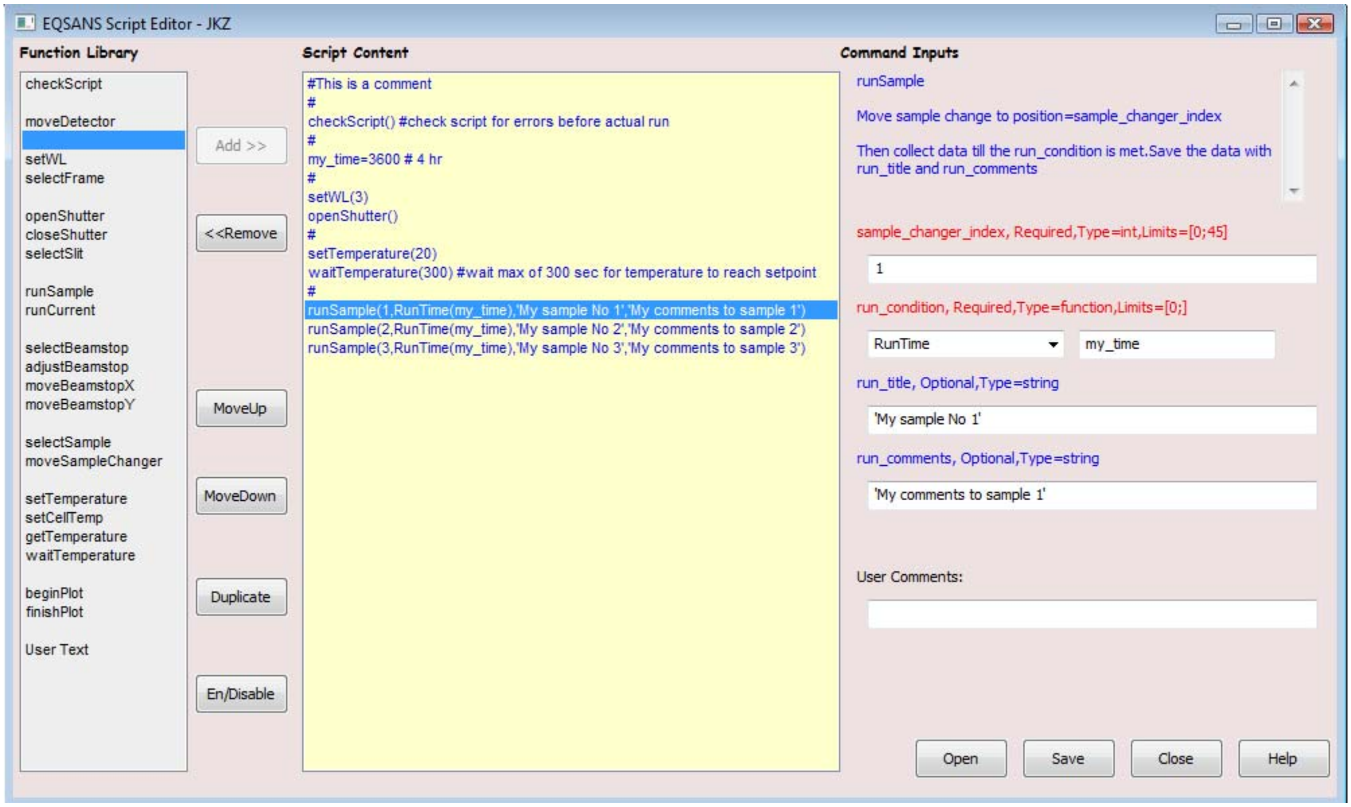
Use 'Run' to start a run

Click the Shell tab to see outputs

(2) Create a script within the script editor.

(3) Save the script to hard disk

(3) Run your script



```
#This is a comment
#
checkScript() #check script for errors before actual run
#
my_time=3600 # 4 hr
#
setWL(3)
openShutter()
#
setTemperature(20)
waitTemperature(300) #wait max of 300 sec for temperature to reach setpoint
#
runSample(1,RunTime(my_time),'My sample No 1','My comments to sample 1')
runSample(2,RunTime(my_time),'My sample No 2','My comments to sample 2')
runSample(3,RunTime(my_time),'My sample No 3','My comments to sample 3')
```

runSample

Move sample change to position=sample_changer_index

Then collect data till the run_condition is met.Save the data with run_title and run_comments

sample_changer_index, Required,Type=int,Limits=[0;45]

1

run_condition, Required,Type=function,Limits=[0;]

RunTime ▾ my_time

run_title, Optional,Type=string

'My sample No 1'

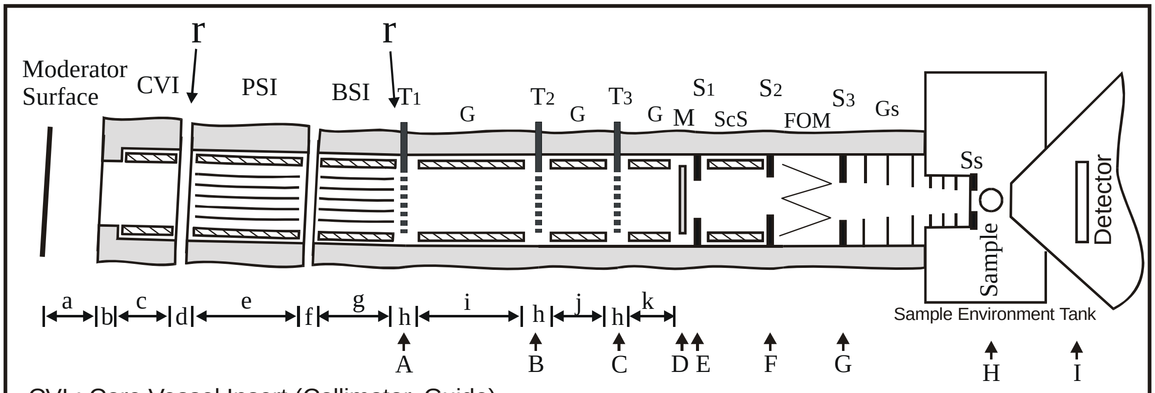
run_comments, Optional,Type=string

'My comments to sample 1'

User Comments:

Appendix

A. EQ-SANS Instrument geometry



CVI : Core Vessel Insert (Collimator, Guide)
PSI : Primary Shutter Insert (Bender)
BSI : Bulk Shield Insert (Bender)
G: Straight Neutron Guides
T1,2,3 : Bandwidth Choppers
S1,2,3 : Variable Beam Slits
M : Beam Monitor
ScS : Secondary Shutter
FOM: Frame Overlap Mirror
Gs: 6 Guard Slits
Ss : Sample Slit

<u>Distances to Moderator [mm]:</u>

A = 5702.8 T1 Chopper
B = 7802.8, T2 Chopper
C = 9502.8, T3 Chopper
D = 10006, Monitor
E = 10080, Slit S1
F = 11156,Slit S2
G = 12150,Slit S3
H = 14000, Sample
I = 15100-24000, Detector

<u>Dimensions [mm]:</u>

a = 1000, CVI to Moderator
b = 270,   CVI Collimator
c = 1000, CVI Neutron Guide
d = 34, PSI-CVI Guide Gap
e = 1857.6, PSI Bender
f = 31.4, BSI-PSI Guide Gap
g = 1459.8, BSI Bender
h = 100, Chopper Gaps
i = 2000, T1-T2 Guide
j = 1600, T2-T3 Guide
k = 468.1,T3 Guide
r = 95000, Bender Radius

The beam cross section is 40mmx40mm.
Hatched blocks : Glass guide wall.
Shaded blocks : Steel shielding and vacuum boundary.

<u>Glass thickness</u> (all elements use non-borated float glass):
Straight guides : 3mm all around.
Bender top and bottom walls : 5mm; side walls: 1mm; inserts: 0.55mm.

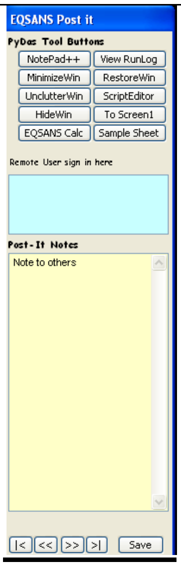
Note: Frame overlap mirror (FOM) is not installed.

B. Helper Applications

PostIt Notepad:

PostIt pad can be used to post note for others (e.g. remote users)

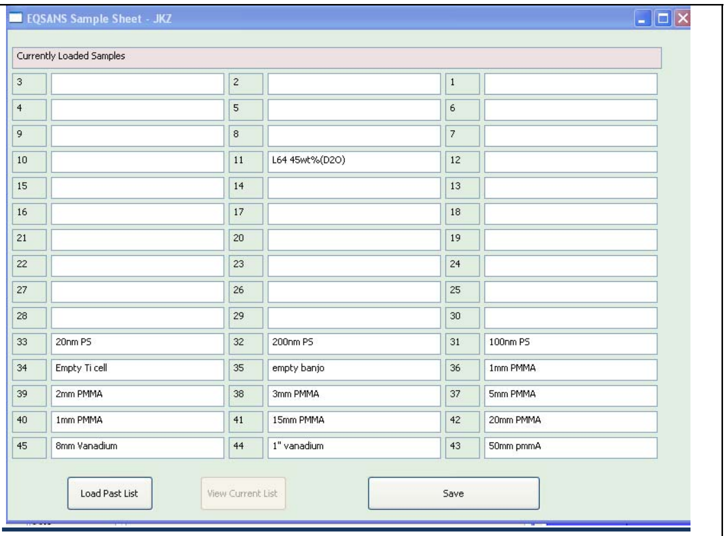
It also has buttons for launching other helper apps



Note to others

Sample sheet

Fill in the sample sheet for loaded samples

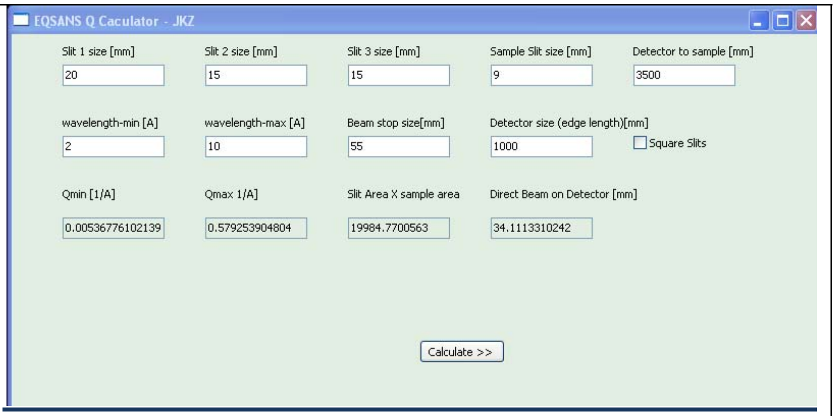


Currently Loaded Samples

3		2		1	
4		5		6	
9		8		7	
10		11	L64 45wt%(D2O)	12	
15		14		13	
16		17		18	
21		20		19	
22		23		24	
27		26		25	
28		29		30	
33	20nm PS	32	200nm PS	31	100nm PS
34	Empty Ti cell	35	empty banjo	36	1mm PMMA
39	2mm PMMA	38	3mm PMMA	37	5mm PMMA
40	1mm PMMA	41	15mm PMMA	42	20mm PMMA
45	8mm Vanadium	44	1" vanadium	43	50mm pmmA

EQSANS calculator.

Simple calculator for helping experimental setup.



Slit 1 size [mm]: 20
Slit 2 size [mm]: 15
Slit 3 size [mm]: 15
Sample Slit size [mm]: 9
Detector to sample [mm]: 3500

wavelength-min [A]: 2
wavelength-max [A]: 10
Beam stop size[mm]: 55
Detector size (edge length)[mm]: 1000
□ Square Slits

Qmin [1/A]: 0.00536776102139
Qmax 1/A]: 0.579253904804
Slit Area X sample area: 19984.7700563
Direct Beam on Detector [mm]: 34.1113310242

C. Data Collection Script - Function Reference

Highlight:	Most Frequently Used.	Frequently Used.	Typically not directly used

Obtaining instrument info:

1. calcBandWidth(chopper_speed1 ,
chopper_phases1 ,
chopper_phases2 ,
chopper_phases3 ,
chopper_phases4):

Calculate the expected wavelength band give chopper speed and phases.
Current detector location is used for calculation

2. **info(),**
3. **getinfo():**
print the current instrument info

Scan/Run:

4. **runSample(** *sample_id,*
run_condition=ProtonCharge(1e12),
title=None,
*comments=None***):**

Move sample to position, start a run, stop the run when run condition is met, and save the data

sample_id	*: Sample cell position on sample changer*	
run_condition	: finish run when condition is met	
example:	ProtonCharge(1e12)	=> run till proton charge >=1e12
	RunTime(1000)	=> run till time >=1000 sec
	DectectorCounts(1e5)	=> run till detector counts >=1e5
	RoiCounts(1e5),	=> run till counts >=1e5
	MonitorCounts(1e6),	=> run till monitor counts>=1e6
title (optional):	title of the run	(a string within quotes)
comments (optional):	comments	(within quotes)

Example:
runSample(10,RunTime(3600*2),'My Sample','Run no 1') #=> run sample at position 10 for 2 hours

Note: If decomclient is already collecting data, runSample() will not move the sample changer and will wait for the current run to finish (i.e. to meet our new run condition)

5. **runCurrent**(*run_condition=ProtonCharge(1e12), title=None,comments=None*)**:**
6. **runNow*(***run_condition=ProtonCharge(1e12), title=None,comments=None***):***

   Same as runSample(), but without moving the sample changer.
   Use these when sample changer is not used.

   Example:
   runCurrent(RunTime(3600*2),'My Sample','Run no 1') #=> run for 2 hours

7. **waitRun(run_condition=ProtonCharge(1e12), title=None,comments=None):**

   Wait for the current run to finish and save it.

Sample changer:

8. **whereIsSample():**

   Print the current sample cell location

9. **setSampleLocation(location) :**

   Sample to moderator distance in mm (now overwritten by value from DAS). Used for bandwidth calculation only.

10. **loadSample(wait=None):**

    Move the sample changer to loading position.
    If wait == 1, then wait until the sample changer is in position before returns.

11. **selectSample(sample_id):**

    Move sample cell no: sample_id into the beam
    Example:
    selectSample(1) #=> move sample changer to pos. 1

12. **moveSampleChanger(x=0,y=0),**
13. **moveSampleChangerBy(dx=0,dy=0):**

    Move the sample change to a absolute coordintate

```
Sample changer mapping:
# dy per row = 24
# dx per column= 41
# row 1   y = 898
# row 15 y = 1964
# column 1 x = 116
# column 3 x = 14
#
#         pos     3              2              1
#(x,y)=           (14,1964)      (65,1964)      (116,1964)
#
#                 ...
```

```
#
#          pos     45            44            43
#(x,y)=            (14,989)      (65,989)      (116,989)
```

14. **holdChanger() :**

    Hold the sample changer and prevent it from moving within python control

    Useful when sample changer is not used for experiment.

15. **releaseChanger() :**

    Release the sample changer and allow it to move with python

16. **saveChangerHolderState() :**

    Save the current sample changer holding state to a file
    Default file: C:\\eqsans_runs\\logs\\runlog\\ sample_changer_holding_status.txt

17. **readChangerHolderState() :**

    Read in the current sample changer holding state from the default file

18. **setTemp(new_t):**
19. **setTemperature(new_t):**

    Set the sample changer temperature setpoint in C

    Example:
    setTemp(20) #=> set the sample changer temperature set point to 20C

20. **getTemp():**
21. **getTemperature():**

    Get the average sample changer temperature in C

    Example:
    getTemp() #=> get the current average sample temperature

22. **waitTemp(timeout=None):**
23. **waitTemperature(timeout=None):**

    Wait sample to reach its temperature of a maximum of 'timeout' seconds.
    Default: if timeout is not given, wait until temperature is in range

24. **calcTiCellTemp(cell_no=None,set_temp=None) :**
25. **calcAlCellTemp(cell_no=None,set_temp=None) :**
26. **calcCellTemp(cell_type,cell_no=None,set_temp=None):**

    Calculate the expected sample (liquid temperature) for a given cell at the set_temp for Titanium or Aluminum cells

    Note that Al and Ti cells have different behaviors cell_no = 1 - 45

If cell_no < 1 or not given, calculate for all cells
If set_temp not given, use current set value from DAS

cell_type = 'Al' for aluminum cells. 'Ti' for titanium cells

Example:
calcCellTemp('Al',2) #=> calculate the estimated *sample* temperature in cell 2.

27. setTiCellTemp(cell_no,set_temp):
28. setAlCellTemp(cell_no,set_temp):
29. setCellTemp(cell_type,cell_no,set_temp):

Set the temperate such that cell cell_no will reach set_temp for a Titanium or Aluminum cell.

cell_type = 'Al' for aluminum cells. 'Ti' for titanium cells

Example:
setCellTemp('Al',3,25) #=> set the changer temperature such that cell 3 reaches 25C.

Detector:

30. whereIsDetector():
Print detector location

31. moveDetector(z_value_in_mm,
y_value_in_mm=None,
x_value_in_mm=None) :

Move the detector to new location.
Range of motion:
z_value_in_mm = 1210. to 10100.
x_value_in_mm = -20. to 20.
y_value_in_mm = -20. to 20.
(Hardware range: z = 1205. to 10177.,x = -25. to 25., y = -25. to 25.)

Note: the sequence for moving the detector and selecting a bandwidth:

1. moveDetector(...)
2. setWL(...) or setFrame(...)
3. adjustBeamstop(...) (optional)

Example:
moveDetector(4000) #=> move the detector to 4000mm from sample.

!!! Note that when detector is moved beyond 5000mm, smaller slits have to be selected to prevent the direct beam hitting the detector. !!!

32. setBeamstop(type='c',start_wavelength=None) :

33. **selectBeamstop(type='c',start_wavelength=None) :**

Select a beam stop and move it to beam center according to the current detector location and selected wavelength band

Type ='c' (default) for circular beamstop, otherwise for square.

Example:

selectBeamStop('c') #=> select the circular beamstop, adjust its position according to current detector location and selected wavelength bands.

34. **moveBeamstopX(x) :**
35. **moveBeamstopY(y) :**

Move the beamstop.

Approx. range:
-100 <=x <=100 (in mm)
0 <=y <=600 (in mm)

36. **adjustBeamstop(start_wavelength=None) :**

Adjust the beamstop position according to detector location and selected wavelength.

Note: the sequence for moving the detector and selecting a bandwidth:

1. moveDetector(...)
2. setWL(...) or setFrame(...)
3. adjustBeamstop(...) (optional)

Example:

adjustBeamstop() #=> Adjust the position of the current beamstop according to current detector location and selected wavelength bands.

Chopper and wavelength selection:

37. **setChopperSpeedsAndPhases(speed_for_all_choppers,
chopper_1_phase_in_microsec,
chopper_2_phase_in_microsec,
chopper_3_phase_in_microsec,
chopper_4_phase_in_microsec):**

Change chopper setting

38. **setChopperByStartingWavelength(start_wavelenth,
chopper_speed_in_Hz,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None):**

Set chopper speeds and phases.
Detector to sample and sample to detector distance are obtained from DAS.

39. setChopperByStartingFrameNumber(**start_frame,**
chopper_speed_in_Hz,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None):

Set chopper speeds and phases to a selected frame.
Detector to sample and sample to detector distance are obtained from DAS.

40. **selectFrame(frame_no,**
is_frame_skipping=None,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None) :

41. **setFrame(frame_no,**
is_frame_skipping=None,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None) :

Select a frame by phasing the choppers.
Detector to sample and sample to detector distance are obtained from DAS.

42. **selectStartingWavelength(wavelength,**
is_frame_skipping=None,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None) :

43. **setWL(wavelength,**
is_frame_skipping=None,
sample_to_detector_in_mm=None,
sample_to_moderator_in_mm=None) :

Set chopper speeds and phases to start at wavelength (in Angstrom).
Detector to sample and sample to detector distance are obtained from DAS.

Note: the sequence for moving the detector and selecting a bandwidth:

1. moveDetector(...)
2. setWL(...) or setFrame(...)
3. adjustBeamstop(...) (optional)

Example:

selectFrame(2) #=> Select Frame No. 2 at the current detector location. No frame skipping.
setWL(2.5,'skipping') #=> Select start wavelength =2.5 at the current detector location in frame skipping mode.

Collimation Slits and Shutter

44. **setBeamSlit(slit_no,wheel_no=1):**
45. **selectSlit(slit_no,wheel_no=1):**
46. **setSlit(slit_no,wheel_no=1):**
47. **changeSlit(slit_no,wheel_no=1):**

Move to slit to selected position on selected slit wheel

wheel_no = 1,2,3 (1 being the most upstream one)

Possible value for slit_no:
For wheel no 1:
sllit_no='closed', d10mm','10x10mm','d15mm','15x15mm','d20mm','20x20mm','open'
For wheel no 2:
sllit_no=open1', d10mm','10x10mm','d15mm','15x15mm','d20mm','20x20mm','open'
For wheel no 3:
sllit_no=open1', d10mm','10x10mm','d15mm','15x15mm','d20mm','20x20mm','open'

Example:
selectSlit('d10mm') #=> select slit d10mm on wheel 1

48. **setSecondaryShutter(position) :**

Set secondary shutter to position 1-8

49. **openShutter(position=2) :**

Open the secondary shutter, position =1,2,3,4. Default =2

50. **closeShutter(position=2) :**

Close the secondary shutter, position =1,2,3,4. Default =2.

Example:
openShutter()

Plotting:

51. **clearPlotData():**
Clear stored plot data

52. **storePlotData():**
Store data for plotting

53. **plot() :**
plot stored data (default: detector counts vs run number) using PyDas's plotting routine

54. **plotROI() :**

plot stored ROI data using PyDas's plotting routine

55. **plotMonitor() :**

plot monitor data using PyDas's plotting routine

56. **beginPlot(type=None):**
Get ready for plotting data. Scans afterwards will be plotted

Type = 'roi', 'ROI','Roi','r','R' for ROI plot
Type = 'monitor','m','M','Monitor','MONITOR' for monitor plot
Default: detector counts

57. **finishPlot():**

Finish plotting. Scans afterwards will not be plotted

High Voltage routines :

58. **HVStatus():**
The status of the detector High Voltage supply

59. **isHVOn() :**
True if all three H.V. are on

60. **isHVOff() :**
True if all three H.V. are off

61. **setHVOn() :**
Turn H.V. on

62. **setHVOff() :**
Turn H.V. off

Testing and others:

63. **setTesting(yes=None) :**
Set a testing flag (1 or 0). No actual command will be sent to the instrument

64. **checkScript() :**
Check script for errors

65. **setVerbose(yes=None) :**
Verbose 1 or 0. (not yet consistently implemented)

66. **checkHelperApps ():**
Check motor and chopper apps are running

67. **gohome(home=None):**

Goto home folder. Default : folder that corresponds to current proposal ID.