**BMM Command Cheat Sheet**

|  |  |
| --- | --- |
| shb.open() / shb.close() | Open / close the photon shutter |
|  |  |
| tu() / td() | Tune the mono 2nd crystal |
| TUNE\_STEP=0.004 | Tuning step size – 0.004 is good for the Si(111), 0.002 for the Si(311) |
|  |  |
| RE(mv(dcm.energy, <value>)) | Move **TO** an energy value |
| RE(mvr(dcm.energy, <value>)) | Move **BY** an energy step |
|  |  |
| RE(mv(xafs\_liny, <value>)) | Move motor **TO** a position |
| RE(mvr(xafs\_liny, <value>)) | Move motor **BY** an amount |
|  |  |
| dcm.wh() | where's the mono? |
| slits3.wh() | where are the slits? |
| m2.wh() | where's mirror 2? (focusing) |
| m3.wh() | where's mirror 3? (flat, harmonic rejection) |
| xafs\_table.wh() | where's the XAFS table? |
|  |  |
| RE(change\_mode('X')) | set mirror mode, see table below |
| RE(change\_xtals('h11')) | set monochromator, Si(111) or Si(311), h=1 or h=3 |
| RE(rocking\_curve()) | optimize pitch of 2nd DCM crystal (moves automagically) |
| RE(slit\_height()) | explore position of slits3 (then you move dm3\_bct) |

To stop a scan or a move

**Ctrl-C twice**

then

RE.stop()

|  |  |  |
| --- | --- | --- |
| **Mode** | **focused** | **energy range** |
| A | **4** | 8keV and up |
| B | **4** | 6keV to 8keV |
| C | **4** | below 6 keV |
| D | 7 | 8keV and up |
| E | 7 | 6keV to 8keV |
| F | 7 | below 6 keV |

**Slits3 coordinated motions:**

* slits3.hsize (nominally 8 mm) and slits3.hcenter
* slits3.vsize (nominally 1.4 mm) and slits3.vcenter

**Common motors:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Motor** |  | **description** |  | **xafs\_linxs** | **foil position** |
| xafs\_linx | x | sample stage, X direction |  | 90 | bottom |
| xafs\_liny | y | sample stage, Y direction |  | 45 | 2nd from bottom |
| xafs\_pitch | p | tilt stage, roll |  | 0 | middle |
| xafs\_roll | r | tilt stage, pitch |  | -45 | 2nd from top |
| xafs\_linxs | xs | reference foil stage |  | -90 | top |

Where is a sample motor? xafs\_<motor>.wh()

What are the soft limits? xafs\_<motor>.hlm.value / xafs\_<motor>.llm.value

Set a soft limit: xafs\_<motor>.hlm.put(<value>) / xafs\_<motor>.llm.put(<value>)

**Line scan:**

RE(linescan(<detector>, <motor>, <start>, <stop>, <N>))

where

<detector> is one of: 'it', 'if', or 'i0'

<motor> is one of: 'x', 'y', 'roll', 'pitch', or a motor name

<start>,<stop>,<N> are the boundaries and the number of steps.

The plot will be determined from the values of <motor> and <detector>

This is a **relative** scan.

After prompt, *single click the left button*  after a linescan to move to a position.

RE(pluck())

to repeat that on the current plot. RE(pluck()) only works on *most recent* plot.

**Energy scan:**

RE(xafs('myscan.ini'))

In the INI file, set “mode” to transmission, fluorescence, reference, or both to control the in-scan plotting display (both = show transmission *and* fluorescence)

**Experiment log:**

Log entries are made for each scan. System and beamtime-specific logs are maintained. To insert a comment in the log, do:

BMM\_log\_info(“This is my log entry”)