Bliss fundamental commands

Starting a BLISS session

Pandora's box

In a terminal: `. blissenv [-d]` (note the dot and space)
`bliss -s <session_name>`
`bliss -h`:help about bliss command

To ***detach*** from a session: `ctrl-b d`

To ***exit*** from a BLISS session: `exit ⏎` or `ctrl-d`

Counters

Counters and Measurement Groups (MG)

`lscnt()`: print list of all counters
`lsmg()` : print list of all measurement groups
`ACTIVE_MG` :info about the current measurement group
Measurement groups are created in config file.
`menu(myMG)`: open dialog to edit ***myMG***
`myMG.set_active()`: define ***myMG*** as default M.G.
`ct(0.2, i0diode)` : count for 0.2 second using i0diode counter
`sct(0.2, i0diode)`: `ct` saved as a scan

Axis

`wa()` : display positions (user & dial) of all motors
`wm(mot1,…motN)`: display positions and limits for specified motors
`mv/move(m1, 2.0)`: move motor ***m1*** to 2.0
`umv(mot1,8)` : move + display position during move
`umvr(mot1, 0.1)`: idem with a move of 0.1 unit relative to current position

`m1.position=3.4`: set *user* position (⇒chg. offset)
`m1.velocity=1.5`: set velocity (uu/s)
`m1.acceleration=5.0`: set acceleration (uu/s²)
`m1.`*`offset`*: ≠ between *dial* and *position*
`m1.`*`backlash`*
`m1.dial=10.0`: set *dial* position (⇒change *user* pos.)
`m1.limits=(-10,10)`: set low/high limit to -10, 10
`m1.`*`sign`*: direction of the movement relative to dial
`m1.`*`tolerance`* `m1.`*`steps_per_unit`*
`m1.`*`unit`* `m1.`*`encoder`* `m1.`*`state`*

position=(sign*dial)+offset
uu: user units (mm, µm, deg...)

properties in italic are Read-Only

Standard scans

Common step by step scans

```
ascan(motor, start, stop, intervals,
      count_time, *cnt_args)
```
Perform a step scan from <start> to <stop> counting <intervals> times <count_time> seconds using the given counters or Measurment groups.

`a2scan()..a5scan()`: scan with 2..5 movements
`anscan()`: scan with an arbitrary number of motors
`dscan()..d5scan()`: scan relative to the current pos.
`dnscan()`: relative scan with arbitrary nb. of motors

```
amesh(mot1, start1, stop1, interv1,
      mot2, start2, stop2, interv2,
      count_time, *counters)
```
Absolute 2D scan on a regular grid
`dmesh()`: relative 2D scan
`lineup(....)`: same as dscan then goes to max
`timescan(ctime, *cnt_args)`: endless counts
`loopscan(npoints, ctime, *cnt_args)`
count <npoints> times
`pointscan(mot, pos_list, ctime)`
scan over a positions list
`lookupscan([(m1, <pos_list1>)...], ctime)`
scan over a variable number of motors and positions.

SCAN_SAVING

Data Saving and DATA POLICY

`SCAN_SAVING ⏎`: display saving parameters
`newproposal("mr1234")`
`newsample("kryptonite")`
`newdataset("Zn_inclusion")`

Shutters

Safety shutters and front-end

`bsh1.close()`: close shutter ***bsh1***
`fe.open()`: open front-end
`fe.mode="AUTOMATIC"` / `"MANUAL"`
set front-en d in automatic or manual opening mode

For your favorite commands

Help

Message in a bottle

`help(command)`: print help about <command>
`last_error()`: details about last error occurred
`last_error(-2)`: details about previous error occurred
`prdef(function)`: print the code and location of a function.
BLISS objects have a "in-shell info" feature:
typing its name + ⏎ print details about it.
`lsobj("*diod*")`: list all session's objects with "diod" in their name

Shell functions

`F3`: enter *history mode*
In history mode: `space` to select,
`ctrl-o` : once to validate , second time to execute
History is saved per user
`F2`: ptpython configuration (colors etc.)
`F4`: switch shortcut mode
`F5`: switch to/from scan view
`F6`: paste mode
`F7`: typing helper (de)activation
`F8`: set logbook filling from shell **on/off**
`ctrl-r xx ⇆` : search for commands starting by "xx" in history
`_XX`: reference to the shell output number XX

Plotting

display and play with data

`flint()`: launch flint plotting tool
`plotselect(diode1)` : select diode1 for plotting
`SCAN_DISPLAY`: object to configure plotting
`SCAN_DISPLAY.auto=False`: disable plotting

`cen()/goto_cen()`:display middle of *fwhm* (and *fwhm*) / move scanned motor to this position
`peak()/goto_peak()`: idem for max
`com()/goto_com()`: idem for center of mass