CHANGING HARDWARE LIMITS IS A NO-GO.

Orion Command List

Please report errors/recommendations to Jurg.Schefer@psi.ch
Similar instrument ZEBRA: https://zebra.intranet.psi.ch

Experiment User and Sample Information

Unix-user:	orionlnsg@orion.psi.ch (don't login as orion@orion.psi.ch)
Password:	17lns1
Computer:	orion.psi.ch

After starting six -m, type reset

(all soft-zeropoints will be set to 0, reset of software limits, all motors enabled, default user, default title, default sample, default UB, default cell)

User
	stores the name of user in each file

sample [sample name]
	stores the name of the sample in each file

title [scan title]
	gives the scan the title

projectdir /home/orionlnsg/your-account
(mandatory, otherwise ubrefine may not work)

Motors and Angles

som (or om)
	returns the value of the sample table

stt
	returns the position of the detector

phi (or ph)
	returns the value of the rotation phi

chi (or ch)
	returns the value of the tilt chi

four
	returns all four current positions of the angles stt, om, chi and phi

Driving and Counting

mv [motor] [value]
    moves by the increment [value]

dr [motor] [value]   (dr and drive are equivalent)
    drives [motor] to position [value]

dr [motor1] [value1] [motor 2] [value 2]
    drives motors 1 and 2 in parallel, can be om, stt, ch, ph

dr hkl [h k l]
drives to the position of reflection hkl as calculated from the present ub

stop
    stops all motors and scans

co [mn or ti] [counts or seconds]
    count for `counts' of monitor counts or `seconds' of time

home
    drives cradle angles to 0 and detector to 90

Batch files

projectdir [pathName])
    prints or (updates) the file path to `pathName'
    all batch files are first searched in this directory afterwards

batchrun [fileName]
    runs the file, found in the path with the name `fileName'
    call the batch-file in the batch-file is possible (indefinite loops)

exe [fileName]
    equivalent to batchrun

Normal scans

cscan [motor] [pos] [step size] [# of steps each side of pos] [count time]
sscan [motor] [start pos] [end pos] [# of steps + 1] [count time]

sscan [motor1] [start pos1] [end pos1] [motor2] [start pos2]
[end pos2] [# of steps + 1] [count time]

fastscan on / fastscan off: starts the counter before driving in all scans (default)

Cone scans:

coneconf [xxxx] [h2 k2 l2] [qscale]
configures the conescan
[xxxx] is the ID of the reference reflection in reflist
[h2 k2 l2] are the indices of the reflection to be found.
qscale: optional value to change the length of the q-vector.

sscan cone [start angle] [end angle] [# of steps+1] [count time]

Centering

peak
Finds the center of your last scan (but does not move)

center
Moves to the center of your last scan previously calculated with peak

centerref [count time]
Centers the reflection in all of the angles

Crystal calculations

cell [a b c α β γ]
prints or (updates) the cell parameters
spgrp [A a a a])
prints or (updates) the space group

calcstt h k l
calculate two theta for the given `h k l'

hkltoang h k l
calculate all angles for the given `h k l'

angtohkl [stt som ch ph]
calculate h k l for the current or (given) angles

ub [x11 x12 x13 x21 x22 x23 x31 x32 x33]
prints or (updates) the ub matrix

ubreco
recovers previous ub matrix

ubcalc [xxxx] [xxxx] [xxxx]
calculates a ub matrix using the reflections `xxxx'
from the reflection list, see with command reflist
2 or (3) reflections can be used

initaux [h1 k1 l1] [h2 k2 l2]
generates auxillary UB matrix:
h1 k1 l1 is parallel with the cradle = perpendicular to the beam
h2 k2 l2 lies in the scattering plane.

Reference reflections

reflist
prints the current list of reflections

refclear
clears the list of reflections

refdel [xxxx]
delete the reflection with the ID 'xxxx' from list (4 digits)

refhkl [xxxx] [h k l]
change the hkl values of reflection `xxxx' to `h k l'

refadd idx [h k l]
add a reflection to the list with index `h k l'

refadd ang [stt som chi phi]
adds the current or (given) motor positions

refadd idxang h k l
adds the current motor positions for reflection
with index `h k l'

refadd idxang h k l stt som ch ph
adds the given motor positions for reflection
with index `h k l'

refang [xxxx] [stt om chi phi]
adds the current or (given) motor positions
at position `xxxx'

addauxref [h k l]
adds an auxillary reflection to allow generation
of a ub matrix
stt and om are for the given cell
chi is assumed to be zero
phi is zero when there are no other reflections in the list
else phi is calculated from the first reflection in the list

refsave [fileName]
saves the reflection list to `fileName'

refload [fileName]
loads a stored reflection list from `fileName'

refindex
index the current reflection list with
the current ub matrix where indices
are rounded to the next integer

ubrefine refines the ub from the current reflection list
(can also be done by rafin >! rafin.lis out-of-sics using rafin.dat as input)
ubload loads the ub

Collecting a dataset on a *.ccl file (as on ZEBRA)

Details: https://zebra.intranet.psi.ch/zebra-experiment.htm

hkllimit hmin kmin lmin hmax kmax lmax sttlow stthigh (sets limits for hkl-generation)
hklgen (generats the list)
indsave list.hkl (saves the list) (seems to have changed the name)
loadx list.hkl (loads the list)
ubload (loads the last ub-matrix from rafin.lis, using the projectdir directory)
exe table.res (loads the resultion)
collect (starts the collection on a *.ccl file)
collect append (starts the collection and continuous with the last ccl-file)

file table.res: contains all information for the data collection

tabadd stt om omstep np monpreset
(up to stt collect makes om-steps with omstep, np points and a montitor of the value monpreset, several lines may be added for different stt values)

Instrument control commands

Motor list
    gives all info and commands associated with a motor
om list
stt list
ch list
ph list

motor enable (0 / 1)
    prints or (disables / enables) the given motor (motor=stt,om,ch,ph)

om refrun
    To re-zero motor om after a restart (only for som, not for stt, chi and phi)

Orientation routines

to scan omega at different ph and ch:
    (works also with ph and ch when cradle is on ????)

localsearch [omStart-omStop]
    scans omega, between omStart and omStop,
    at incremental values of ch and ph
    sgu, sgl and omega scans are controlled with
    peakconf parameters: tiltstep, ntilt, a3step

to find the orientation of the crystal:

orient [h k l] [h k l] [count time]

    searches for the two given reflections
    the omega range and values of ch and ph
    can be controlled using peakconf
    the cell parameters need to be updated:

to find the direction of a face:

facescan [h k l] [omStart-omStop]   [count time]

    fixes theta and two theta
    within the given omega range
    for different values of ch and ph
    these can be controlled using peakconf
    returns the angles of the first peak found

to find crystallites:

qualityscan   [count time]

    scans omega, ch and ph at a fixed two theta value

to set the parameters:

peakconf

    shows a list of configurable parameters
    for the search routines

peakconf [parName] [value]

    sets new parameter values
    for the following search routines
    Edit the step width and other parameters in
    /home/orion/orion_sics/tassearch.tcl until this works nicely.

After a restart of SICS or a restart of the motor controller (MCU):

Do not forget to check, if all motors are enabled in the MCU.
Normally, this is done automatically.

```
stt enable 1
om enable 1
ph enable 1
```

ch enable 1

eventually, a refrun for the angles will be necessary (if angles seem to be wrong)

optical check: omega=0: cradle is vertical to the incoming beam (marker)
stt=0: detector is in the direct beam (marker, to be made)
ch=0: cradle mount is on lower side horizontal (marker)
ch=90: cradle mount is on the side to FOCUS
ph=0: (marker, to be made)

Useful Programs:

Index:
Finds possible combinations of reflections based on different peaks found
Input file: index.dat (example in /home/orionlnsg/examples)
Start of program: index n (where n is the number of peaks to be indexed)
Output: index.out

```
LiCoPO4
10.2021 5.9227 4.700 90 90 90 2.2149 1.5
30.447    5.432  176.394  167.984 .2
26.735    5.232  183.980  233.156 .2
0

Line 1:   Title
Line 2:   cell wavelenght angle-acceptance (1-3 degrees normally)
Line 3ff: stt om chi phi error-stt
Line 4:   zero to end
```

Rafin:
Calculates the UB from index files. The cell constants may be adjusted (which is not possible when doing it from inside six)
Code for refined parameters: 1 in front of the parameter, 2 if dependent
Input file: rafin.dat (may be created by typing ubrefine inside of six)
Start: rafin >! rafin.lis (creates always the same output file rafin.lis
Loading the ub: type ubload inside six (loads also the new cell parameters)
Be aware: you have to be in the directory projectdir defined inside six

```
MgCr2O4
2 1 0 0 45 3 4 1 .5 0
0  2.2149
0 .0 0 .0 0 .0
0 8.908 0 8.908 0 8.908 0 90 0 90 0 90
   2.0000    2.0000    0.0000   41.178   20.589   -0.113   89.391
   1.0000    1.0000   -3.0000   48.707   24.353    0.876  154.140
   2.0000    2.0000   -2.0000   51.021   25.510    0.536  124.690
   2.0000    2.0000    2.0000   51.022   25.511   -0.503   54.206

-1

Line1:   Title
Line2:   2 1 0 0 minimum-angle 3 4 1 max-error-angle 0
```

```
Line4:   0 wavelength                         (1 instead 0: refine)
Line4:   0 sttzero 0 omegazero 0 chizero      (1 instead 0: refine)
Line5:   0 a 0 b 0 c 0 alpha 0 beta 0 gamma   (1 instead 0: refine)
Line6ff: h k l stt omega chi phi
Line7:   -1
```

Last update: 2.11.2015
PC 9909: OrionCradleCommands.docx
J. Schefer

Orion Default Parameters (never change, not for users)

ph sign	= -1 (Drehrichtung von om und phi für chi=0 muss gleich sein) wir benötigen wegen der UB-Matrix ein rechtshändiges System, Busing-Levy Definition der Winkel
ph softzero	= -360 (bringt input von 0-360 bei ph sign=-1 in hardware limits)
ph hardlowerlim	= -0.999834
ph hardupperlim	= 360.94
ph softlowerlim	= 0
ph softupperlim	= 720
ph fixed	= -1
ph interruptmode	= 0
ph precision	= 0.1
ph accesscode	= 2
ph failafter	= 3
ph maxretry	= 3
ph ignorefault	= 0
ph movecount	= 10
ph staticoffset	= 0
ph status	= idle
ph error	= None
ph interest	= FUNCTION
ph interest/name	= UNKNOWN
ph numinmcu	= 8
ph enable	= 1
ph scale_factor	= 0.0004882
ph maxspeed	= 6
ph commandspeed	= 5
ph maxaccel	= 0.1
ph commandedaccel	= 0.01
ph invert	= 0
ph offset	= 0
ph axisstatus	= 0
ph axiserror	= 0
ph poshwlimitactive	= 0
ph neghwlimitactive	= 0
ph liftaircushion	= 0

```
ph halt              = FUNCTION
ph refrun             = FUNCTION
ph badencoder          = 0
```

```
ch sign                 = 1
ch softzero             = 0
ch hardlowerlim         = -33.0134
ch hardupperlim         = 131.053
ch softlowerlim         = - 33
ch softupperlim         = 120
ch fixed                = -1
ch interruptmode        = 0
ch precision            = 0.1
ch accesscode           = 2
ch failafter            = 3
ch maxretry             = 3
ch ignorefault          = 0
ch movecount            = 10
ch staticoffset         = 0
ch status               = idle
ch error                = None
ch interest             = FUNCTION
ch interest/name        = UNKNOWN
ch numinmcu             = 7
ch enable               = 1
ch scale_factor         = 0.00024424
ch maxspeed             = 3
ch commandspeed         = 2
ch maxaccel             = 0.1
ch commandedaccel       = 0.01
ch invert               = 0
ch offset               = 0
ch axisstatus           = 0
ch axiserror            = 0
ch poshwlimitactive     = 0
ch neghwlimitactive     = 0
ch liftaircushion       = 0
ch halt                 = FUNCTION
ch refrun               = FUNCTION
ch badencoder           = 0
```

```
stt sign                 = 1
stt softzero             = 0.57
stt hardlowerlim         = -46.9999
stt hardupperlim         = 151
stt softlowerlim         = -46.4
stt softupperlim         = 150.6
stt fixed                = -1
stt interruptmode        = 0
stt precision            = 0.1
stt accesscode           = 2
stt failafter            = 3
stt maxretry             = 3
stt ignorefault          = 0
stt movecount            = 10
stt staticoffset         = 0
stt status               = idle
stt error                = None
stt interest             = FUNCTION
stt interest/name        = UNKNOWN
stt numinmcu             = 4
stt enable               = 1
stt scale_factor         = 0.00024414
stt maxspeed             = 1.63
stt commandspeed         = 1.54
stt maxaccel             = 0.0042
stt commandedaccel       = 0.0034
stt invert               = 0
stt offset               = 0
stt axisstatus           = 0
stt axiserror            = 0
stt poshwlimitactive     = 0
stt neghwlimitactive     = 0
stt liftaircushion       = 0
stt halt                 = FUNCTION
stt refrun               = FUNCTION
stt badencoder           = 0
```

mom1 or mom2

returns the values of the monochromators. DO NOT ADJUST.

```
om sign                 = 1
om softzero             = 0
om hardlowerlim         = -27
om hardupperlim         = 51
om softlowerlim         = -26.9
om softupperlim         = 50.9
om fixed                = -1
om interruptmode        = 0
om precision            = 0.1
om accesscode           = 2
om failafter            = 3
om maxretry             = 5
om ignorefault          = 0
om movecount            = 10
om staticoffset         = 0
om status               = idle
om error                = None
om interest             = FUNCTION
om interest/name        = UNKNOWN
om numinmcu             = 3
om enable               = 1
om scale_factor         = 0.0005
om maxspeed             = 1.65
om commandspeed         = 1.65
om maxaccel             = 0.0025
om commandedaccel       = 0.0025
om invert               = 0
om offset               = 0
om axisstatus           = 0
om axiserror            = 0
om poshwlimitactive     = 0
om neghwlimitactive     = 0
om liftaircushion       = 0
om halt                 = FUNCTION
om refrun               = FUNCTION
om badencoder           = 0
```

Drehrichtungen von omega und phi müssen sich für chi=0 (Halter unten) aufheben:
Bsp: mv om +10 ph   -10
        Reflex/Instrument muss am gleichen Ort sein

(gleiche Konfiguration wie Zebra, nur ist dort normalerweise chi=180,
  dh. das Spiel ist umgekehrt)