

HRPT useful commands stacked in approximately chronological order

=====

14/May/2015 -2018... 2025 ...

``clean_nicos_log.pl'`

Standalone script, which periodically reset NICOS logs using
NewExperiment()

Usage: `clean_nicos_log.pl` [options]

Options:

<code>--interval <minutes></code>	Set the time interval between log resets (default: 10 minutes).
<code>--max-time <seconds></code>	Set the maximum allowed time for log reset (default: 300 seconds).
<code>--kill</code>	Kill the running daemon.
<code>--help</code>	Display this help message.

Example:

```
clean_nicos_log.pl --interval 15 --max-time 400
clean_nicos_log.pl --kill
```

``hstartSICS.pl'`hstartNICOS.pl'`

script to switch `_everything_` to SICS/NICOS:
start/stop SICS/NICOS via `hsicos.pl`
and switch perl lib via `switch2sicos_lib.pl`
and change the last file number via `~hrpt/hstarttics`

``hnicos_command.pl'`

script to switch run NICOS command under perl libs
and then return back to SICS
If NICOS is not running, will start it
and execute commands in spirit of perl-based lib under NICOS.

``hsicos.pl'` script to start/stop SICS/NICOS via `marchec`

```
hsicos.pl SICS
hsicos.pl NICOS
```

``hmarchec.pl'` start/stop/anycommand for SICS/NICOS via `marchec`

Usage: `hmarchec.pl` COMMAND [ARGS]...

ARGS is in most cases the name of the service.
Possible services are listed below or can be obtained
with status COMMAND.

Possible COMMANDs:

logs output reload restart start status stop

Examples:

```
hmarchec.pl logs nicos
hmarchec.pl restart nicos.poller
hmarchec.pl output nicos.poller
hmarchec.pl start fixsync
```

``switch2sicos_lib.pl'` a simple script to switch SICS/NICOS perl libs
via git branch

``hsix'` Yet another SICS client. Any SICS command is accepted.

In addition: q - for quit, h - for history

``hsics_help'` help for HRPT control commands under perl-library `Hrpt::.`

``hrpt_help'` will show this file

``hxbu'` run the commands from provided buffer or stop current execution

Use: `hxbu -k`

or

`hxbu buffername &`

The ampersand symbol '&' is to move the execution of 'bufername' into background, so if the terminal window is closed, the execution of the script is continued.

Main options are:

- k to kill the execution and stop sics
- f force to start even if PID-file from the previous hxbu execution had not been deleted or some hxbu are running (trying to kill them first). This might happen in some cases.
- s to stop sics. If some xbu is still running the run will continue. Use this option only if you are sure you want it...

Some extra options are below:

- vV to show what code in the buffer is running now.
V is n,m where n - approximate number of last hxbu-lines before the currently executed one to show,
m - number of lines ahead to show
if V is all the whole buffer script will be shown
- l changes the behaviour of -v, so it shows literally all n previous lines including empty lines, comments, etc.
list of the timing of commands execution is suppressed.
if -v is not specified uses V=5,3
- h the above text

Examples:

```
hxbu
hxbu runme.xbu&
hxbu -k
hxbu -l
hxbu -l -v 5,5
hxbu -lv5,5
hxbu -vall
```

`hix' Yet another interactive SICS client that uses
perl-library \$ENV{'HOME'}/perl/lib/Hrpt::
if used with arg search for library in arg first
hix . will look in ./Hrpt/';

hrun_status Some useful info, like monitors, preset, sample, etc

mkdif.pl

The difference files are created from the files generated by hmake.

Usage:

mkdif.pl template add t1 t2 t3 ...

where

template - file name up to the temperature value. For example: hrpt_Ce5Pb3_2p45n_HI_T_
add - value to add to the difference pattern
t1 - the temperature which is subtracted from the rest of the temperatures t2, t3 ...

Examples:

```
mkdif.pl hrpt_n_Ce5Pb3_2p45n_HI_T_ 1000 30 1 5
mkdif.pl hrpt_Mn3_OH_2_rdc_2_MOF_V10x35_1p89_HI_T_ 1000 40 1 20
-----
```

`hwatch' watch any sics motor

Use: hwatch [-r -t] space delimited list of motors

- r stay on the same line
- t 10 repeat each 10 seconds, default is 1s
- 1 do it only once
- q put the values in quotes

Examples:

```
hwatch chpos a4 a3
```

or even this
 hwatch 'el737 values'
or that
 hwatch temperature

`sinq' prints the SINC current each 10 seconds

the command is just a perl oneliner that uses `hwatch'.
To see how it works: cat `which sinq`

`sinqlive' plots the SINC current as a function of time

`htime.pl' `hlong`

to show how long the numbers or file was collected and at what
SINC current

`hint4mode.pl`

Will show table of relative intensities normalised by
1p89HI mode: lambda=1.89A, high intensity

Usage:

 hint4mode.pl [mode]

where mode : search pattern for the HRPT-mode
e.g. HI or MR or 1p9

=====

SICS tcl-clients <http://lms00/sinqwiki/Wiki.jsp?page=Main> (M. Koennecke, M. Zolliker)

`sea` X-window client of the SeaServer (temperature and not only).

`six' command line client to SICS-server

=====

Data reduction, etc...

`hmake': automatic generation of the data files from the raw HRPT data

=====

`hmake' creates data files from raw HRPT data performing calibration and spikes
correction similar as `hfc' script does. The title strings supplied by the user
and the sample temperature for the runs are used as the unique identifiers of
the measurement conditions. So in case if you like to identify your measurement
condition better you should put all relevant information in the title. For example,
wavelength, resolution, pressure, etc.
The run numbers with the same experimental conditions can be noncontinuous.
The script will find all matching titles and add the runs together.

Usage:

hmake [-c] [-m value] [-q value] <first_number> <datafile prefix>

 to make datafiles from <first_number> to last number in automatic update mode.

 This is a recommended option. Default <datafile prefix> is 'hrpt_'

or

hmake <range> makes data in the <range> of numbers, e.g 2014:100-200,
 and exits.

or

hmake -- -<number> makes data out of the last <number> of numbers, e.g -50,
 and exits.

The data reduction includes also additional correction on semishadow from
beam reduction at 2theta 160-165 degrees. If -c option is specified no 2theta
correction is done (not recommended for HRPT).

-m value: neutron monitor for normalized _n_ data-files, default 12000000.

-q value : value is the SINC current to show better estimation of statistics

Examples:

 hmake 1000 from number 1000 to last measured

```
and updating each 600 seconds. File names
begin with 'hrpt_'
hmake 1000 ori4_ from number 1000 to last measured
and updating each 600 seconds. File names
begin with 'ori4_'
```

`killhmake': kills the hmake process(es) running on the same computer

Usage:

killhmake

will show the running process(es), but will not kill them. Just proposes
the command(s) to use for the killing.

killhmake -a

will try to kill all hmake process(es) automatically.

`hstat.pl' returns the statistics for the given dat-file that includes neutron
monitor (NM), estimated acquisition time, total number of counts (Sum).

Usage:

hstat.pl filename

To look at all *.dat files in the current directory use:

perl -e 'print "\$_ ", `hstat.pl \$\_` for (<*.dat>)'

`hlog' prints the list of measurements in the `hmake' spirit

Usage: 'hlog <first_number>' to get log from <first_number> to last number

or 'hlog <range>' to get log in the <range>, e.g 2014:100-200

or 'hlog -<number>' to get log of the last <number> of numbers, e.g -50

Example: hlog -50 takes last 50 numbers

hlog 1000 from number 1000 to last measured

`add.pl' generation of data files from range of numbers

Usage:

add.pl instrument/year numor1 numor2 <step> <binmerge> <monitor>

Examples:

add.pl 2011/hrpt 46691 46727

add.pl 2011/hrpt 46691 46727 2 0.2 10000000

`hrpt2xy.pl' converts all dat-files in HRPT/DMC format in current
directory to xysigma type

with temperature inside for funs of winplotr.

The xysigma file names are xxx1.dat xxx2.dat ...

`hrpt2xy.pl' converts hrpt data files into x_y_sigma-format
with correct header, also with temperatures inside,
so it can be read by winplotr. The xy-file names are
xxx1.dat, xxx2.dat, ...

Usage:

hrpt2xy.pl <filemask>

Examples:

hrpt2xy

All *.dat in current directory will be converted

hrpt2xy *.txt

Files with given mask will be converted

`hradial' status or start stop radial collimator (RC)

Usage:

hradial [-t] start/stop

Examples:

hradial -t 600 start

will start RC if it is stopped each 600 seconds

hradial

will just give the status of RC

`hsave_data_hdf' saves HRPT hdf-data file on the fly without
stopping the current
data acquisition. Use this command only in you exactly
know what you are going to do.

Usage:

hsave_data_hdf <SAVE_NEW_DATA_ON_FLY> <title>
If the second optional argument <title> is provided
its will be prepended to the existent title
if no <title> is given a default string will be added: on_fly_[timestamp]
e.g. on_fly_Wed_Nov_25_13:23:29_2015_

Examples:

hsave_data_hdf SAVE_NEW_DATA_ON_FLY first_cycle_
the string first_cycle_ will be prepended to the existent title

`hplot.pl' continuously plot the points from
hrpt_mot.log or any file given as argument

addsub_strobo

Usage:

addsub_strobo [options] 'filenumber,dimension' [bin_step] [neutron_monitor]

Creates main/master data-files in HRPT-format from the 2-dimensional
NeXus-histogramm collected in stroboscopic mode,
where:

'filenumber,dimension' are the number of NeXus file
and maximal number of histogram you like to extract.
Master files will be created from the histograms
from 1 to 'dimension' with optional
[bin_sep] (default is 1) optionally normalised to the same
[neutron_monitor].

When the options are used performs some ariphmetics with the additional
data saving the results in the master files,
options:

- c number2cancel
Removes duplicated statistics in number2cancel from master
- s number2subtract
Subtract number2subtract from master (has sence if these
are independent measurements)
- a numbers2add
string with additional number, or numbers to be added to the
master. The string can be '2000,2100,2200-2300,2302'...
- t title
By default the template for the file name is generated from
the title. Use this option if you like your own one.
- n
logical option saying that the master files are already present
in the current directory, so the master files are not generated.
Might be useful if one needs some interational ariphmetics.

#30/Apr/2015 11:56 #
#The help below is mainly related to the utilities for the instrument scientists, but #
#of course these utilities can be used by anyone if needed. #

22/Mar/2017, 11:42

hmode

Usage:

hmode [options] filenumber

prints presets for given mode

where:

filenumber are the number(s) of NeXus file

options:

-f file

which contains output of sumvar command shown below

setenv sumvar numor:6,date,title:70,mgpu:7:2,smon,pmon,time,cex2:5.1,momu:7.2 ; sumvar range_of_numbers

>your_file_name

is present in current directory.

-m

print only for

specific mode of operation

-t

do not determine the mode by the

monochromator motors values, but

try to match the mode given by -m

option in the title.

-s

print only summary statistics

-q value

consider only the numors with

SINQ current more than 'value'.

Default is 100 uA.

-o value

'value' can be 'sinq' or 'smon'

creates a XY-file with name hmode_value.txt

with numors SINQ_current or NM_presets

March 11, 2016

`make_cals.pl'

takes the argument, which is string of numbers in spirit of `fit' by Markus

and add an explicit calibration file according to proper `calist.dat'

For example:

'hrpt:2016:1-4,2000:dmc:1-10,hrpt:2010:1-10' is modified to

'hrpt:2016:1[unity.txt],2-4[cal_june1_refined.txt],dmc:2000:1-10[deteff.dat],hrpt:2010:1-10[deteff_rc2_b=0p5_6m
ay10_p.txt]'

=====

`iofch' <file of range of numbers>

prints counts an a function of channel

Usage:

iofch [-m15] [-o] <file or numbs> [output file]

-o prints only ch.no and counts

-m specifies number of channels with max intensities to be
printed at the end

Examples:

iofch -m 3 '5424[cal=0]'

=====

`1hrpt' prints counts for the given channels and summary statistics

usage: 1hrpt [-t] [-m] firstN-lastN chan1,chan2,chan3

or 1hrpt -t 'regexp' firstN-lastN,yet_another_number chanM-chanN [print field length]

-t filters only the records which title matches regexp

-m N merges the number by N

-s V print only summary for the noisy channels which sigma is V-times
larger than sqrt(I). e.g. 1.1, by 10%

-z Z if -s is given look also for the channels counting less than Z

-g reanalyze grand total without noisy channels

-x excluded channels

Examples:

1hrpt -s 2 -m 3 5424-5438,5446-`lastnum` 1-1600

1hrpt -m 3 5424-5438,5446-5448 1-10

=====

`hphs' creates Pulsed Hight Spectrum from HRPT datanumbers

measured with different threshold values.

Usage: hphs [-c 20] [-b0.1] [-m 1800] <range of data numbers in sumvar spirit> [prefix]
 where -b specifies optional binning in Volts,
 -c 20, e.g. means no spikes correction, but disregards
 channels with counts more then 20.
 Usefull if there are some channels with 0 counts...
 Note that the patterns are normalized to monitor
 36000/1000000 for counning by time/NM
 -m 1800 will override the above default monitor value

=====

Making data files with `hmake': technical info

31/Mar/2016, 14:46: All redone. The previous versions as explained between #####

are git'ed at

dab7f01 (HEAD, refs/heads/new_hmake) before modifications in hfind etc, to unify hf and hfc...

Now everything w and w/ 2theta correctionis in hfind -> hfind_.pl

#####

hmake -> /afs/psi.ch/user/p/pomjakushin/bin/upd.pl -> hfindc_.pl -> 9 Jul 2014 ahnc_.pl -> 12 Nov 11:35 ahc
-> ahc.pl

finally ahc_.pl uses the calibration named as `calib_c.txt' and then performs correction with brcorr.pl

Making data files with `hf'. Added Jan'15 binwidth as the last param. But ahn_.pl that does the sum job should
be modified, as shown below.

`hf' is simply `hfind -100 \$@', which in turn is

/afs/psi.ch/user/p/pomjakushin/bin/hfind -> hfind_.pl

`ah\_ \$toadd \$ARGV[2] \$ARGV[4]`; print "done.";

to get merging option. Not always work due to limitations

#print "=====ah2_ \$toadd \$ARGV[2] \$ARGV[4] \$binwidth =====\n";

`ah2\_ \$toadd \$ARGV[2] \$ARGV[4] \$binwidth`; print "done.";

#####

=====

calib.pl

Usage:

calib.pl filename

making HRPT calibration file from the file prepared by

makecalib.pl with correction for beam reduction from 1566 channel with a4=5.

The correction table is taken from /home/hrptlinsg/pombin/brcorr.pl:

lines /BEGIN ACTUAL CORRECTION CODE/ ../END ACTUAL CORRECTION CODE/.

This script substitutes the matlab script (<2017).

The input file name must have extention dat.

The output names are similar, but with extention txt.

=====

makecalib.pl

Makes the calibration file (HRPT format) from raw V- and BG-data numbers

or datafiles and optionally creates the calibrations (in deteff format) by

means of `calib.pl' both native and corrected to semi-shadow from br-slits

Usage:

makecalib.pl [-c -f -m -r -w -l] V-numbs/V-filename BG-numbs/BG-filename

-c calibname , e.g. -c unity.txt

default is 0, i.e. uncalibrated and no spikes removal

-f rms_for_spikes_removal, e.g -f 10

default is 5

-m -r -w are [muR] [Vradius] [fwhm of RC], respectively

defaults for these optional parameters are 0.31 4 7

-l linkname will also create calibration files in deteff format

(the file names might be quite long) and symbolic

links to them in HRPT calibration spirit: linkname.txt for native

calibration and linkname_c.txt for proper calibration with correction

for semi-shadow from br-slits.

Examples: makecalib.pl V.dat BG.dat

makecalib.pl -w14 47048-47061 47043-47047

=====

05/Mar/2014 10:38

`xynac' script determines the position of NAC in detector coordinate system from fullprof fit of sysin and sycos from data @1.9A. Then the values of stx and sty corresponding to the position of the sample in the rotation center O_S are calculated as one can see in '/afs/psi.ch/user/p/pomjakushin/bin/nac_off/tomatlab.pl'. If pcr-file is given, no fullprof fit is performed. Just sy[sin,cos] are taken from .out-file, and stx,sty from .dat-file that should be in current directory.

Our xy detector coordinate axes

^ y_D
|
| x_D
|<=====incoming beam
|
|
| \ //
2theta \ //
| //
to detector

Usage:

xynac numors/filename [2theta zero] [1/0 var flag] [1/0 vary a flag]

Examples:

xynac hrpt:2012:25457-25472 -0.1 1 1

xynac filename.dat -0.1 1

xynac filename.dat

xynac filename.pcr

10/May/2013 12:47

/afs/psi.ch/user/p/pomjakushin/fits/hrpt/2012/fitfit/plot_sysincos.m

latest m-file to plot the stx,sty(xy_D)

xynac updated with varriables \$O_Sx, \$O_Sy

09/May/2013 11:03 from now on the latest news are on the top with the time stamp

`hf': automatic generation of the data files from the raw HRPT data.

It is a successor of `ah'.

`hf' creates data files from raw HRPT data performing calibration and spikes correction similar as `ah' script does. The title strings supplied by the user and the sample temperature for the runs are used as the unique identifiers of the measurement conditions. So in case if you like to identify your measurement condition better you should put all relevant information in the title. For example, wavelength, resolution, pressure, etc.

The run numbers with the same experimantal conditions can be noncontinuous.

The script will find all matching titles and add the runs together.

Usage:

hf NumbersForSumvar <filename_template> <neutron_monitor>

where

NumbersForSumvar:

numbers' list acceptable by `sumvar', e.g. 1-10 or 1-100,200,202-300

filename_template:

optional filename to be used as a prefix for the datafile names.

If empty, no datafiles are produced, but script shows what has been found.

neutron_monitor:

optional parameter. If present, all data files will be normalized by the same neutron_monitor. It is useful if the data are later used by the software that does not recognize the monitor to compare, e.g., different patterns collected at different temperatures with different monitors.

Examples:

hf hrpt:2008:131067-132115

hf hrpt:2008:131067-132115 set1_

hf hrpt:2008:131067-132115 set1_ 10000000

=====

```

=====
'statmon.pl' print daq threshold, total counts from files with prefix
statmon.pl cosmic
-----
find_chmot.pl a3 1-100
Find changing motor in the sequence of measured
numers.
Usage:
    find_chmot.pl 1-100 a3
-----
`henfill.pl' shows he or n2 refills in current year
henfill.pl 2012
henfill.pl 2012 n2
-----
ah, ad:
scripts for adding HRPT, DMC files
ah2: additional 3rd argument is the binning step
-----
statch:
Calculate average, dispersion and expected dispersion for all 1600
wires over the given datafile numbers
-----
lastnum, lastnumd: last written file number for HRPT and DMC
-----
hlast, dlast: last <argument> numbers table with `sumvar'
-----
ahs:
to subtract an sdditional pattern with the spike background with time-calibration
-----
detteff2dat:
detteff.dat to .dat-file for fit
-----
fl:
first and last field
-----
fcomma:
all spaces to commas
-----
hrpt_det_p.pl, hrpt_det_t.pl:
make a table(time) with HRPT detector normalized gas pressure, temperature of electronics
`hrpt_det_t2alarm.pl' prints alarms
`hrpt_det_t2.pl' time in string format
`hrpt_det_t2_watch.pl' same as above but continuously monitors
-----
makesplist.pl:
table of file no. and channel no. with spikes. To change something edit the file
-----
makehist_spikes.pl:
creates the histogram of spiky channels from the output of `makesplist.pl'.
-----
To get softlimits from the previous year. copy/paste to six:
    [hrpt@hrpt ~/hrpt_sics]$ cat hrptdf.log | perl -lane 'if (/d1./ && /soft/) {s/^.+://; print}'
=====
Obsolete stuff, but still possible to use...
-----
Procedure of determining sysin and sycos (since 2011 we use fullprof to determine them):

Latest dir: /afs/psi.ch/user/p/pomjakushin/fits/hrpt/2009/dec_nac_rc/fitfit
            /afs/psi.ch/user/p/pomjakushin/fits/hrpt/2010/ori4_sch/fitfit

pomjakusx5: work/HRPT_RandD/summary_sycosis_pot_smpch_etc_2009/nac_oct/fp_fits/

* nac 6mm powder is collected at several a3-positions with 1.9A.

* There is global fit when all peak shifts for all a3-values are fitted together to the formula  $y(2t) = y_0 + y_1 \cos(2t) + y_2 \sin(2t)$ .
** step 1. nac.perl (DOIT_g subroutine, file doit_pos_g.pm) script fits all the patterns indicated in the

```

