

Notes on SHARP observing scripts within IRC client

D. Dowell, 2006 Oct 27

Brief Description of Scripts

- * SHARP_Standard – HWP/nodding sequence with each HWP in a different data file
Single_Nod – Take data for a single nod with no HWP motion
- * SHARP_Single_File – HWP/nodding sequence with each HWP in the same data file
SHARP_Sweep – HWP sequence with each HWP in a different data file; no nodding
- * Coarse Dither – sequence of 4 data files, each with a SHARP_Single_File-like nature
- * Fine Dither – same as Coarse Dither
- Sky Dip – I don't use this.

How to Run a Script

In the “IRC – Instrument Remote Control” window, find the “Commanding” window in the upper left. Double click on “Observing Procedures”. That should bring up the list of scripts. These scripts reside both on the client host `irc/` directory and also in the compressed client code which is downloaded from Goddard when there are new updates. I believe the ones in `irc/` are first in the search path, but Troy would have to check to be sure.

Updating Scripts

To make sure that new changes to the script and parameter files are actually used, exit and restart the IRC client. I don't see why the IRC server would need to be restarted, but you could try that, too (before restarting the client).

Troubleshooting

If the telescope doesn't nod:

- 1) Make sure the “Chop Throw” parameter is > 0 .
- 2) Maybe the IRC->UIP link is down. Here are Hiro's notes for restarting it:
Log on to alpha1 as SYSTEM, and see if the service is already running by:

```
$ show system
```


The process you are looking for is named "UIP Daemon." If it is not there, then start a new one by:

```
$ @ bigdisk: [hiro.uip.exe] uipd.com
```

If it looks like old script definitions are being used:

- 1) Exit and restart the client. Maybe even the server.
- 2) Maybe you have to be running `sharcClientTest`, not `sharcClient`.

IRC Name: SHARP_Single_FILE

script: `kilauea% ~sharc/irc/polarimeterSingleFile.py`

parameter file: `kilauea%`

```
~sharc/.irc/sharc_v1_5/resources/sharc/xml/default_command_procedures.xml
```

The timing calculation within the script is as follows (all in seconds):

`duration = “Time per HWP”`; default 90, but adjustable

overhead = 20; predicted time before samples start writing to disk
slewtime = 5; time for telescope to settle after *starting* a nod command
hwptime = 5; time for HWP to settle after *starting* a move command
totalScans = number of HWP angles; default 4, but adjustable

totaltime = (duration + hwptime)*totalScans + overhead -
hwptime = 395 (default)
beamtime = (duration - 2*slewtime)/4 = 20 (default)

The observation sequence is:

- 1) Start totaltime second scan (first ~20 seconds not written to disk).
- 2) Start move to left beam; start HWP move.
- 3) Wait for overhead for first HWP, otherwise wait for hwptime.
- 4) Wait for beamtime in first left.
- 5) Start move to right beam.
- 6) Wait for (slewtime+beamtime*2) in right.
- 7) Start move to left beam.
- 8) Wait for (slewtime+beamtime) in second left.
- 9) Return to step (2) if doing more HWP angles.

IRC Name: Coarse Dither, Fine Dither

script: kilauea% ~sharc/irc/FourPointDither.py

parameter file: kilauea%

~sharc/.irc/sharc_v1_5/resources/sharc/xml/default_comman
d_procedures.xml

The operation of this script is straightforward. Four SHARP_Single_File's are run at four different dither positions. The dithering is done in bolometer coordinates:

xstart, xstep, ystart, ystep are user adjustable.

```
rotZero = 1.7  
theta = (elev - rotZero)*3.141592653589/180.0  
arcsecperpix = 4.63
```

```
dx = (xstart + {0, xstep})/10.0*arcsecperpix  
dy = (ystart + {0, ystep})/10.0*arcsecperpix
```

```
daz = dx*cos(theta) + dy*sin(theta)  
dza = -dx*sin(theta) + dy*cos(theta)
```

```
UIP> AZO daz  
UIP> ZAO dza
```

One more thing is worth mentioning. The default setting of "Zen. Ang." (zenith angle) is -1, meaning that the script reads from the antenna computer. However, I have seen intermittent failure of this communication. The user can override the value by entering a positive number. Check the IRC/Java log to see what value the script ended up using.