

## Postprocessing with watcher and merge4k

watcher is a script that automatically converts each incoming extended FITS images from bigccd into a normal (single) FITS image on gerard using merge4k, then displays the newly merged fits image in the ds9 window. After the image is displayed, watcher can also optionally execute any other script that the observer wishes to run.

In order to keep watcher from crashing due to interference from iraf processes in other windows, watcher starts up in a separate watcher subdirectory (/home/bigobs/watcher/) with its own uparm subdirectory.

To start watcher correctly, open a terminal window, and type

```
./watcher_window
```

This pops up a new IRAF window starting in the watcher subdirectory, with the tasks and definitions that watcher needs already loaded. It is also useful to type "cl < mont4k.tasks" in the gerard home directory when you open other iraf windows you may use with your images.

If you aren't running an optional post-processing script, either epar watcher to set extras="" or else type watcher.extra=""

If you're observing light curves, you should set extras = auto\_plot, and epar auto\_plot to customize it for your target. (You can see the results in a separate iraf window with "!tail -f auto\_plot.log")

Before starting watcher, cd to your data directory - this MUST be the same directory where bigccd will send the images - and start watcher:

```
watcher .
```

As each new image is transferred to gerard, watcher will run merge4k on that image. merge4k converts/combines the raw extended FITS images into a normal single FITS image, after first correcting each side separately for the crosstalk and subtracting the two overscan regions. The original extended FITS image is saved in a subdirectory named "raw", and the merged image is renamed with same name in the original directory. After merge4k has finished with the image, watcher displays the combined image in the ds9 window, optionally runs the requested extra script, then starts watching for the next image.

It is useful to reduce the watcher window to a few lines in height and place it under the ds9 image, so that you can see what it's doing as it detects, converts, and displays your images.

If you need to stop watcher, type control-c in the watcher window, followed by a flpr, or better: flpr; flpr

The current version of IRAF has a bug that would cause watcher to crash spectacularly; a memory limit is set too low to allow watcher to indefinitely run the loop that looks for new images. The pragmatic fix is to have watcher stop itself cleanly every 200+ images, after printing out a message telling the user to hit the up-arrow key and restart watcher.

merge4k was written to work properly with images of arbitrary numbers of columns and rows and binning; it reads the image header and works with whatever image size and binning exist.

⇒ The M4K CCD is read out by two amplifiers; there are a few unread columns between the two sides of the CCD which are not read out, so one should either avoid placing targets of interest on the center seam or dither the telescope position with multiple exposures; 1×1, 2×2 and 4×4 binned images don't show the missing columns; 3×3 binned images have one “zero” column between the two sides of the image (although the area of missed sky is somewhat wider than the single displayed empty column).

If gerard is very busy (especially if it's running firefox), watcher will (rarely) miss an image. At the end of the night, check to see whether all of your images were merged by typing “`imhead *.fits | grep FX`”. Any unmerged image will produce an error of the form “`<image_name>:FXF: must specify which FITS extension (<image_name>)`”. Rerun `merge4k` on the image manually in the watcher window (or in any iraf window where `mont4k.tasks` was defined):

```
merge4k <image_name>
```

If you find an image that is damaged or unreadable (again, rare but possible if gerard is running lots of other tasks), check to see if the original image made it to the raw subdirectory:

```
dir raw/<image_name>
```

If the raw image exists, you can usually successfully remerge it by deleting the botched image in the observing directory, moving the raw image back up from the raw subdirectory to the observing directory and rerunning `merge4k` on it:

```
del <image_name>
rename raw/<image_name> .
merge4k <image_name>
```

From:  
<https://lavinia.as.arizona.edu/~tscopewiki/> - MOON

Permanent link:  
[https://lavinia.as.arizona.edu/~tscopewiki/doku.php?id=public:catalinas:bigelow:kuiper\\_61:mont4k:watcher](https://lavinia.as.arizona.edu/~tscopewiki/doku.php?id=public:catalinas:bigelow:kuiper_61:mont4k:watcher)

Last update: 2024/09/19 16:36

