

# Specifications for the “rowfluxcorr” Module

F. Masci, 11/18/2003

## I. Command-line specifications

The proposed command-line specifications are as follows:

```
rowfluxcorr  -n <namelist>                (Optional)
                -i <Input_FITS_Filename>      (Required)
                -u <InputUncert_FITS_Filename>  (Optional)
                -d <DMaskImage_FITS_Filename>   (Optional)
                -df <Fatal_DMaskbit_Template>   (Optional)
                -ds <DMaskbit_To_Set>           (Optional)
                -k <Multiplicative_Constant>    (Optional, Default = 1)
                -c1 <FluxLevel_Cal_FITS_Filename> (Required)
                -c2 <FluxCorr_Cal_FITS_Filename> (Required)
                -c3 <UncertFluxCorr_Cal_FITS_Filename> (Optional)
                -o <Output_rowfluxcorr_FITS_Filename> (Required)
                -ou <Output_Uncert_FITS_Filename> (Optional)
```

where:

- n <namelist> Should allow specification of three configuration parameters corresponding to “-k”, “-df” and “-ds”: respectively the input image multiplicative constant, the fatal dmask bit pixels to ignore and the dmask bits to set if anything went wrong in processing (see below).
- i <Input\_FITS\_Filename> Input FITS image filename to apply corrections (e.g. science image).
- u <InputUncert\_FITS\_Filename> Input uncertainty FITS image filename corresponding to the above. Ensure NAXIS1, NAXIS2 and BUNIT equal those in the input image. Abort gracefully otherwise.
- d <DMaskImage\_FITS\_Filename> Input accompanying DMask FITS image (16-bit).
- df <Fatal\_DMaskbit\_Template> If specified, “&” this with input DMask and if result is true, do not apply the correction to these pixels and set a bit specified by “-ds” (see below).
- ds <DMaskbit\_To\_Set> If specified, set this bit in the input DMask if anything goes wrong in processing (see below) or if the pixel is ignored due to a fatal DMask bit being satisfied.
- k <Multiplicative\_Constant> Multiply input image pixel data by this optional constant (float) to get in the same units as the input calibration file:  
“FluxLevel\_Cal\_FITS\_Filename”; see below)
- c1 <FluxLevel\_Cal\_FITS\_Filename> Contains flux levels. This is a 128(col) x  $N$  (row) array where 128 refers to the number of rows in the input image (for mips-24) and  $N$  is the number of available flux correction factors (bins) for that row.

- c2 <FluxCorr\_Cal\_FITS\_Filename> Contains dimensionless flux correction factors corresponding to the “FluxLevel\_Cal\_FITS\_Filename”. This is also a 128 (col) x  $N$  (row) array. Please ensure that NAXIS1 and NAXIS2 of this image are the same as that defining “FluxLevel\_Cal\_FITS\_Filename”, otherwise, abort gracefully.
- c3 <UncertFluxCorr\_Cal\_FITS\_Filename> Uncertainty image corresponding to “FluxCorr\_Cal\_FITS\_Filename”. Ensure the NAXIS1, NAXIS2 keyword values are consistent.
- o <Output\_rowfluxcorr\_FITS\_Filename> Output FITS representation containing flux-corrected pixels in same units as input image. Actually, just copy input header to output header and write a HISTORY keyword with module name/execution time.
- ou <Output\_Uncert\_FITS\_Filename> Output FITS representation containing flux-corrected pixels in same units as output/input image.

## II. Algorithm

1. Read-in the flux level of pixel  $i, j$  [col, row]. Skip pixel if it is flagged as fatal in the fatal template specification if it was specified. Report such fatal pixels in the DMask by setting bit specified by “-ds” if it was specified.
2. Multiply input pixel value by constant specified by “-k” (default is one). It will be up to the user to ensure the correct constant is chosen such that input and calibration images: “FluxLevel\_Cal\_FITS\_Filename” have the same units.
3. Given the row number  $j$  of this pixel, find the appropriate flux bins which bracket this flux from the “FluxLevel\_Cal\_FITS\_Filename”.
4. Do a linear-interpolation (or cubic spline if the data varies wildly – Frank’s suggestion) between the adjoining flux bins in the corresponding “FluxCorr\_Cal\_FITS\_Filename” to compute the applicable correction factor.
5. The Instrument Team asks to also linearly-interpolate the corresponding uncertainty but please choose a method which you think is correct and robust.
6. If the input flux is higher or lower than the highest or lowest flux bin, set the correction factor to one (i.e. no correction) and the corresponding uncertainty to zero. Indicate that this has occurred in the DMask by setting bit specified by “-ds” if it was specified. Also write verbose warning message to standard output.
7. Multiply the input pixel flux by the interpolated correction factor to get the output pixel flux.
8. Combine the computed correction factor uncertainty and the input uncertainty pixel value to get the output uncertainty pixel value. Note: the correction factor and input flux may be correlated.
9. Produce two output images: “Output\_rowfluxcorr\_FITS\_Filename” and “Output\_Uncert\_FITS\_Filename” with updated DMask if either input pixels satisfied the fatal condition (1) or processing found cases satisfying (5).