Next Up Previous Contents FITSIO Home
**Next:** 6 CFITSIO Error Status **Up:** 5 CFITSIO File Names **Previous:** 5.4.5 Example Row Filters **Contents**

# 5.5 Combined Filtering Examples

**The previous sections described all the individual types of filters that may be applied to the input file. In this section we show examples which combine several different filters at once. These examples all use the `fitscopy` program that is distributed with the CFITSIO code. It simply copies the input file to the output file.**

```
fitscopy rosat.fit out.fit
```

**This trivial example simply makes an identical copy of the input rosat.fit file without any filtering.**

```
fitscopy 'rosat.fit[events][col Time;X;Y][#row < 1000]'
out.fit
```

**The output file contains only the Time, X, and Y columns, and only the first 999 rows from the 'EVENTS' table extension of the input file. All the other HDUs in the input file are copied to the output file without any modification.**

```
fitscopy 'rosat.fit[events][PI < 50][bin (Xdet,Ydet) = 16]'
```

(+)

image.fit

**This creates an output image by binning the Xdet and Ydet columns of the events table with a pixel binning factor of 16. Only the rows which have a PI energy less than 50 are used to construct this image. The output image file contains a primary array image without any extensions.**

```
fitscopy 'rosat.fit[events][gtifilter() &&
regfilter("pow.reg")]' out.fit
```

**The filtering expression in this example uses the `gtifilter` function to test whether the TIME column value in each row is within one of the Good Time Intervals defined in the GTI extension in the same input file, and also uses the `regfilter` function to test if the position associated with each row (derived by default from the values in the X and Y columns of the events table) is located within the area defined in the `pow.reg` text region file (which was previously created with the `fv/POW` image display program). Only the rows which satisfy both tests are copied to the output table.**

```
fitscopy 'r.fit[evt][PI<50]' stdout | fitscopy stdin[evt][col
X,Y] out.fit
```

**In this somewhat convoluted example, fitscopy is used to first select the rows from the evt extension which have PI less than 50 and write the resulting table out to the stdout stream. This is piped to a 2nd instance of fitscopy (with the Unix `Image' pipe command) which reads that filtered FITS file from the stdin stream and copies only the X and Y columns from the evt table to the output file.**

(+)

```
fitscopy 'r.fit[evt][col RAD=sqrt((X-#XCEN)**2+
(Y-#YCEN)**2)][rad<100]' out.fit
```

**This example first creates a new column called RAD which gives the distance between the X,Y coordinate of each event and the coordinate defined by the XCEN and YCEN keywords in the header. Then, only those rows which have a distance less than 100 are copied to the output table. In other words, only the events which are located within 100 pixel units from the (XCEN, YCEN) coordinate are copied to the output table.**

```
fitscopy 'ftp://heasarc.gsfc.nasa.gov/rosat.fit[events][bin
(X,Y)=16]' img.fit
```

**This example bins the X and Y columns of the hypothetical ROSAT file at the HEASARC ftp site to create the output image.**

```
fitscopy 'raw.fit[i512,512][101:110,51:60]' image.fit
```

**This example converts the 512 x 512 pixel raw binary 16-bit integer image to a FITS file and copies a 10 x 10 pixel subimage from it to the output FITS image.**

---

Next Up Previous Contents FITSIO Home
**Next:** 6 CFITSIO Error Status **Up:** 5 CFITSIO File Names **Previous:** 5.4.5 Example Row Filters **Contents**

(+)