Chandra X-ray Center

AHELP for CIAO 3.4   **dmcols**   Context: dm

Jump to: Description Bugs See Also

Synopsis

Selecting columns in a table

Syntax

```
[cols name1,name2,...]
[cols #1,#4,...]
[cols -name1,-name2]
[cols newname1=name1,newname2=name2,...]
```

Description

Selecting columns

The DM virtual file syntax lets you pick out selected columns from a table.

```
dmcopy "evt.fits[cols time,pha]" tmp1.fits
```
makes a file containing only the columns time and pha, whereas

```
dmstat "evt.fits[cols energy]"
```
passes to dmstat a virtual file containing only the column called "energy".

Renaming columns

```
dmcopy "pi.fits[cols rate=count_rate]" npi.fits
```
renames the column count_rate in pi.fits to be simply rate in the file npi.fits. In this example only the count_rate column is copied; to copy across the other columns you can either specify them individually or use the symbol "*":

```
dmcopy "pi.fits[cols rate=count_rate,*]" npi2.fits
```
Changing the order of columns

The cols directive can be used to reorder the columns in a file, since they will come out in the order you specify. In the previous example – which used
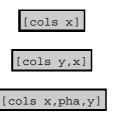
```
cols rate=count_rate,*
```

– the rate column is the first column in the output file, whatever its position in the input file. Note that if you have a vector column sky(x,y), then

```
dmcopy "evt.fits[cols x,y]" table.fits
```

will retain the information that (x,y) is a vector column called sky, but any of

```
[cols x]
```

```
[cols y,x]
```

```
[cols x,pha,y]
```

will not, and the vector–dependent coordinate systems like RA and Dec will be lost.

Using the column number rather than the name

The command

```
dmstat "evt.fits[cols #3]"
```

passes to dmstat a file which contains only the third column in evt.fits (where `column' is used in the DM sense, in which sky(x,y) is a single "vector" column; use

```
dmlist evt.fits cols
```

to find the column numbers).

Excluding columns

You can also exclude columns; both the following commands remove the status column from the file, leaving the other columns intact.

```
dmcopy "evt.fits[cols -status]" tmp1.fits
dmcopy "evt.fits[cols !status]" tmp1.fits
```

When used from the Unix shell the "!" symbol needs to be written as "\!", which makes the first form more convenient.

Accessing virtual columns

A table can contain virtual columns; these are columns whose values are calculated by applying a mathematical transform to an existing column. They are mainly used to represent coordinate systems such as those described in "ahelp coords". For a Chandra event file, the sky(x,y) column contains a coordinate transform to create the eqpos(ra,dec) column – which gives the RA and Dec of each event. These values can be listed by using either of the following two commands:

```
dmlist "evt.fits[cols ra,dec]"  data
dmlist "evt.fits[cols eqpos]"  data
```

You can find the names of the coordinate systems defined for a file by using the 'cols' option to dmlist:

```
dmlist evt.fits cols
```

and then using these coordinate system names in a DM

```
[cols ...]
```

qualifier lets you turn them into actual columns.

It is also possible to filter and bin on virtual columns. For example:

```
dmlist "evt.fits[ra=233.5:233.7,dec=23.4:23.6]" data
dmcopy "evt.fits[bin eqpos=0.01]" img.fits
```

However, filtering and binning do not work correctly with virtual columns derived from non−monotonic coordinate transforms (e.g. MSC(THETA,PHI), or EQPOS near the poles; see 'ahelp coords' for more information on these coordinate systems). Hence, the results of filtering and binning in these cases are not reliable.

Also, world−coordinate region filters on eqpos, such as

```
[eqpos=circle(15:34:55.534,+23:29:57.21,3.9032')]
```

are not handled as one might expect. When the region filter is applied, eqpos is treated as a vector on R^2 (the plane). For a circular region, a point (ra,dec) passes the filter if it satisfies the equation

```
sqrt( (ra − ra0)^2 + (dec − dec0)^2 ) ) < radius
```

However, what one most likely wants in this case is for eqpos to be treated as a vector on S^2 (the sphere), with a point passing the filter if the spherical arc distance between it and (ra0,dec0) is less than the radius. Currently, the best approximation to this can be obtained by applying the region filter to sky rather than eqpos, e.g.

```
[sky=circle(15:34:55.534,+23:29:57.21,3.9032')]
```

Hence, this alternative should be preferred over region filtering on eqpos in most cases.

Bugs

See the bugs page for the Data Model library on the CIAO website for an up−to−date listing of known bugs.

See Also

calibration
      caldb
chandra
      coords, guide, isis, level, pileup, times
chips
      chips
concept
      autoname, parameter, stack, subspace
dm
      dm, dmbinning, dmfiltering, dmimages, dmimfiltering, dmintro, dmopt, dmregions, dmsyntax
gui
      gui
modules
      paramio, pixlib, stackio
slang
      overview, slang, tips
tools
      acisspec, dmappend, dmarfadd, dmcopy, dmextract, dmgroup, dmgti, dmjoin, dmlist, dmmerge, dmpaste, dmsort, dmtcalc, dmtype2split

Bugs