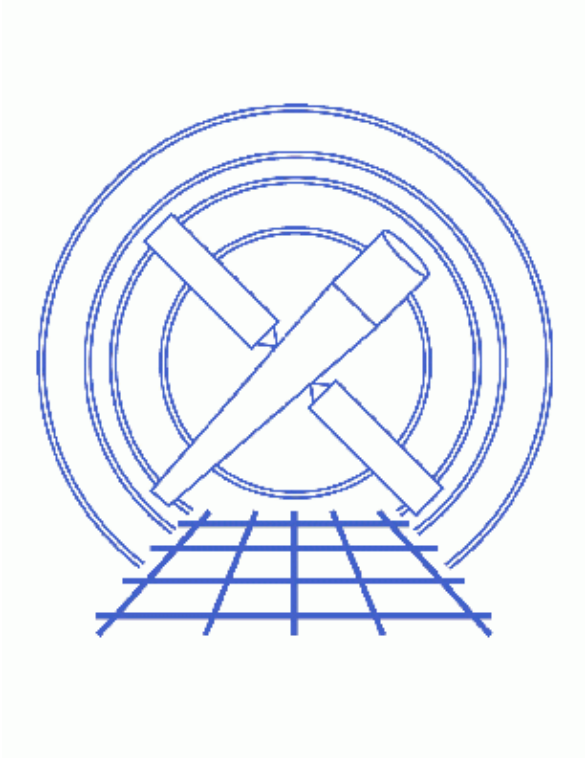
Introduction to the Data Model



CIAO 3.4 Science Threads

Table of Contents

Introduction to the Data Model

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 − updated for CIAO 3.4: updates to screen output in dmgti example

Synopsis:

The CIAO Data Model is a versatile interface used to examine standard format datafiles (FITS, IMH); it also allows powerful filtering of datafiles. The filtered output can either be saved or input directly to an analysis task as a "virtual file." A brief introduction to the Data Model interface and examples on how to use the four core tools are provided here.

Related Links:

- Full details on all the dmtools can be found in their respective ahelp files.

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

Get Started

For illustration, this thread utilizes the ObsID 1843 (ACIS−I, G21.5−0.9) data that was downloaded in the How to Download Chandra Data from the Archive thread.

If this is your first time using CIAO, please read the Starting CIAO thread to ensure that your environment is configured properly.

Data Model Tools

The four most frequently used (aka "core") tools are:

- ***dmlist*** − List contents or structure of a file
- ***dmcopy*** − Filter and bin tables and images
- ***dmextract*** − Make a histogram table file (e.g. PHA or PI file) for spectral fitting or a radial profile for spatial fitting.
- ***dmgti*** − Define custom good time intervals (GTIs)

In addition, there are a number of more specialized tools:

- ***dmappend*** − Append multiple blocks/extensions to an existing output file

- ***dmarfadd*** − Add multiple ARF files together, weighting by exposure time
- ***dmcontour*** − Make contour regions from a 2−D image
- ***dmcoords*** − Convert between Chandra instrumental coordinate systems
- ***dmdiff*** − Compare values in two FITS files
- ***dmfilth*** − Replace pixel values in source regions of an image with interpolated values
- ***dmgroup*** − Group a specified column in a table
- ***dmgroupreg*** − Translates DS9 regions and groups to CIAO format
- ***dmhedit*** − Edit Data Model file headers
- ***dmhistory*** − Extracts processing history from file header
- ***dmimg2jpg*** − Make a color JPEG image from three image files
- ***dmimgcalc*** − Perform arithmetic on images
- ***dmimghist*** − Make histogram of values in a 2−D image
- ***dmimgpick*** − Maps image values to rows in a table
- ***dmimgthresh*** − Set low exposure regions of an image to zero
- ***dmjoin*** − Join two files whose sampling is different
- ***dmkeypar*** − Retrieve information about a keyword from an input file
- ***dmmakepar*** − Write header keywords to a parameter file
- ***dmmakereg*** − Create a FITS region file from an ASCII region description
- ***dmmerge*** − Merge two or more compatible tables into one
- ***dmpaste*** − Add new columns to a table
- ***dmreadpar*** − Add parameters from a `.par` file to a file header
- ***dmregrid*** − Rebin a stack of two−dimensional images
- ***dmsort*** − Sort a table block on a given column
- ***dmstat*** − Compute standard statistics for the column in a table or image
- ***dmtcalc*** − Define new table columns as functions of old ones
- ***dmtype2split*** − Create a type 1 file for specified rows of a type 2 file

---

Running Data Model Tools

Tools are usually executed from the command−line prompt:

```
unix% dmcopy "acisf01843N001_evt2.fits[EVENTS]" acis_events.fits
```

Alternatively, the parameters can be input to the parameter file before the tool is run:

```
unix% punlearn dmcopy
unix% pset dmcopy infile=acisf01843N001_evt2.fits
unix% pset dmcopy outfile=acis_events.fits
unix% dmcopy
Input dataset/block specification (acisf01843N001_evt2.fits):
Output dataset name (acis_events.fits):
```

The parameter file that was used can be examined with plist dmcopy. Note that the Data Model automatically uses the "interesting" block (e.g. EVENTS for an event file, SPECTRUM for a PHA file) when one is not provided, so we chose not to specify it in the second example.

All of the tool capabilities are also available from a GUI (peg) accessed via the Analysis menu of *Prism* and *filtwin*; see the Introduction to the Analysis Menu and the Introduction to peg threads for more information.

Virtual Files

The Data Model offers an easy and powerful means of filtering data. As mentioned, the filtered file can be directly input to an analysis task without writing it to disk first; this is known as a "virtual file." The virtual file, which can also be referred to as a subspace, is simply a means of defining a subset of interest in the dataset.

The basic syntax of a virtual file has four arguments:

```
filename[block][filter][columns/binning][newblock]
```

- `block` − the section of the file to use
- `filter` − the filter to be applied
- `columns/binning` − either the columns from a table to be included in an output table or the binning
- `newblock` (optional) − the name for the new block in the output file, default is the block used from the input file

See ahelp dmfiltering for more information on filtering syntax.

Examples of virtual files:

- Select the first three columns of the EVENTS block by number:
  ```
  acisf01843N001_evt2.fits[EVENTS][cols #1,#2,#3]
  ```
  or by name:
  ```
  acisf01843N001_evt2.fits[cols time,ccd_id,node_id]
  ```
- Select certain rows from a FITS file:
  ```
  acisf01843N001_evt2.fits[#row=1:4]
  ```
- Bin an event file to create a PI spectrum (using dmextract):
  ```
  acisf01843N001_evt2.fits[EVENTS][bin pi=1:1024:1]
  ```
  or an image (using dmcopy):
  ```
  acisf01843N001_evt2.fits[EVENTS][bin x=3200:4800:4,y=3200:4800:4]
  ```
  More information on binning data is available from ahelp dmbinning.

The Core Tools

These examples show just a few of the uses for each tool − see the ahelp files for more options and examples.

1. dmlist

FITS files generally contain several different blocks, each containing multi−dimensional data. The `dmlist` tool allows the user to inspect all or part of a data file by column, row, etc.

- To list the blocks in a file:
  ```
  unix% dmlist acisf01843N001_evt2.fits blocks

  ---------------------------------------------------------------------------
  Dataset: acisf01843N001_evt2.fits
  ---------------------------------------------------------------------------

  ```

```
     Block Name                          Type         Dimensions
-------------------------------------------------------------------------------
Block    1: PRIMARY                        Null
Block    2: EVENTS                         Table        14 cols x 475869   rows
Block    3: GTI7                           Table         2 cols x 1        rows
Block    4: GTI0                           Table         2 cols x 1        rows
Block    5: GTI1                           Table         2 cols x 1        rows
Block    6: GTI2                           Table         2 cols x 1        rows
Block    7: GTI3                           Table         2 cols x 2        rows
Block    8: GTI6                           Table         2 cols x 1        rows
```

- To list the columns of the events block in a file, use the `cols` option (note the required quotes):

```
unix% dmlist "acisf01843N001_evt2.fits[events]" cols

-------------------------------------------------------------------------------
Columns for Table Block EVENTS
-------------------------------------------------------------------------------

ColNo  Name                 Unit        Type            Range
   1   time                 s            Real8          84270902.1682000011: 84280645.8185659945
   2   ccd_id                            Int2           0:9                  CCD reporting event
   3   node_id                           Int2           0:3                  CCD serial readout a
   4   expno                             Int4           0:2147483647         Exposure number of C
   5   chip(chipx,chipy)    pixel        Int2           1:1024               Chip coords
   6   tdet(tdetx,tdety)    pixel        Int2           1:8192               ACIS tiled detector
   7   det(detx,dety)       pixel        Real4          0.50:     8192.50    ACIS detector coordi
   8   sky(x,y)             pixel        Real4          0.50:     8192.50    sky coordinates
   9   pha                  adu          Int4           0:36855              total pulse height c
  10   energy               eV           Real4          0:  1000000.0        nominal energy of ev
  11   pi                   chan         Int4           1:1024               pulse invariant ener
  12   fltgrade                          Int2           0:255                event grade, flight
  13   grade                             Int2           0:7                  binned event grade
  14   status[4]                         Bit(4)                              event status bits

-------------------------------------------------------------------------------
4:
-------------------------------------------------------------------------------

ColNo    Name
5:    CPC(CPCX) = (+0) +(+0.0240)* (chip(chipx)−(+0.50))
         (CPCY)   (+0)  (+0.0240)  (    (chipy) (+0.50))
7:    MSC(THETA) = (+0) +TAN−P[(+0.000136667)* (det(detx)−(+4096.50))]
         (PHI  )   (+0)        (+0.000136667)  (   (dety) (+4096.50))
8:    EQPOS(RA ) = (+278.0474) +TAN[(−0.000136667)* (sky(x)−(+4096.50))]
           (DEC)   (−10.5711 )      (+0.000136667)  (   (y) (+4096.50))
```

- To list the first 5 events in a file, use the `data` option and `rows` parameter together:

```
unix% dmlist acisf01843N001_evt2.fits data row=1:5

-------------------------------------------------------------------------------
Data for Table Block EVENTS
-------------------------------------------------------------------------------

ROW    time                 ccd_id node_id expno
        chip(chipx,chipy) tdet(tdetx,tdety) det(detx,dety)           sky(x,y)
        pha        energy       pi     fltgrade grade status[4]



     1  84272488.5504292250    6    3          3    (861,148)
        (3736,1850) (     3697.3767089844,     6789.5180664062) (
        1439.8631591797,     3556.3112792969)       3601     14079.6406250
        965      0    0
        00000000000000000000000000000000

     2  84272488.5504292250    6    3          3    (962,609)
        (3837,2311) (     3798.8344726562,     6328.2709960938) (
```

```
       1894.8839111328,     3682.7961425781)       3737     15603.13281250
       1024    208    6
       00000000000000000000000000000000

    3  84272488.5914692283    7    2          3     (524,10)
       (4441,1712) (     4404.6499023438,     6927.2255859375) (
       1263.7524414062,     4255.0126953125)       2625     12244.93750
       839     10    6
       00000000000000000000000000000000

    4  84272488.5914692283    7    3          3     (807,52)
       (4724,1754) (     4687.8652343750,     6885.5917968750) (
       1289.8640136719,     4540.0781250)       3509     15991.6308593750
       1024     16    4
       00000000000000000000000000000000

    5  84272488.5914692283    7    1          3    (448,135)
       (4365,1837) (     4329.1870117188,     6802.6430664062) (
       1392.2685546875,     4186.4628906250)       1712      8132.1030273438
       557     64    2
       00000000000000000000000000000000
```

- To list the file header:

```
unix% dmlist acisf01843N001_evt2.fits header

--------------------------------------------------------------------------------
Header keys for block EVENTS
--------------------------------------------------------------------------------

 --  COMMENT               This FITS file may contain long string keyword values that are
 --  COMMENT               continued over multiple keywords.  The HEASARC convention uses
 --  COMMENT               character at the end of each substring which is then continued
 --  COMMENT               on the next keyword which has the name CONTINUE.
0001 HDUCLASS             OGIP                            String
0002 HDUCLAS1             EVENTS                          String
0003 HDUCLAS2             ALL                             String
0004 ORIGIN               ASC                             String       Source of FITS file
0005 CREATOR              cxc − Version CIAO 2.0b         String       tool that created th
0006 REVISION             1                               Int4
0007 ASCDSVER             R4CU5UPD8.2                     String       ASCDS version number
0008 CHECKSUM             kiJFmhIFkhIFkhIF                String       HDU checksum
0009 DATASUM              2822987099                      String       data unit checksum
0010 DATE                 2000−09−04T05:09:46             String       Date and time of fil
0011 DATE−OBS             2000−09−02T08:35:02             String       Date and time of obs
0012 DATE−END             2000−09−02T11:17:25             String       Date and time of obs
.
. (etc.)
.
```

- All of the above examples may be combined into one command that yields the greatest possible amount of information:

```
unix% dmlist acisf01843N001_evt2.fits full
```

which is shorthand for:

```
unix% dmlist "acisf01843N001_evt2.fits[events]" blocks,header,cols,subspace,data
```

- To save the output to a text file:

```
unix% dmlist acisf01843N001_evt2.fits blocks outfile=blocks.txt
unix% more blocks.txt

--------------------------------------------------------------------------------
Dataset: acisf01843N001_evt2.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
```

```
Block    1: PRIMARY                        Null
Block    2: EVENTS                         Table        14 cols x 475869   rows
Block    3: GTI7                           Table         2 cols x 1        rows
Block    4: GTI0                           Table         2 cols x 1        rows
Block    5: GTI1                           Table         2 cols x 1        rows
Block    6: GTI2                           Table         2 cols x 1        rows
Block    7: GTI3                           Table         2 cols x 2        rows
Block    8: GTI6                           Table         2 cols x 1        rows
```

2. dmcopy

The dmcopy tool can be used to manipulate data. Unlike dmlist, which produces text output, this tool produces a new data file in one of the supported formats.

- Copy the events from the central region of a file into a new FITS file:

```
unix% dmcopy "acisf01843N001_evt2.fits[events][x=3600:4000,y=3800:4200]" \
      acis_center.fits
```

- Generate a blocked image:

  A convenient way to display the full field of view of an event file is to bin it into an image. In this example, the full range of sky coordinates is blocked by a factor of 4:

```
unix% dmcopy "acisf01843N001_evt2.fits[events][bin x=::4,y=::4][IMAGE]" \
      acis_img.fits
```

  Since the image file is a binned version of the event file, only the selected variable (i.e. sky coordinates) is retained. All other information (photon arrival times, energy, etc.) is lost.

- Filter using a region defined in sky coordinates:

```
unix% dmcopy "acisf01843N001_evt2.fits[sky=ellipse(1628,4116,92,160,0)]" \
      source.fits
```

  Note that the name of the column in the filter must match the name within the file itself; in Chandra data, "sky" is shorthand for "(x, y)."

3. dmextract

The dmextract tool is similar to dmcopy. It is used to bin tables into images, but writes the binned data to a table instead of creating an image.

- Make a histogram from a table column:

```
unix% dmextract "acisf01843N001_evt2.fits[bin pha=1:2048:2]" histogram.pha
```

- Extract the PI spectrum of a source in sky coordinates:

```
unix% dmextract "acisf01843N001_evt2.fits[sky=region(ds9.reg)][bin pi]" \
      spectrum.fits
```

  where

```
unix% more ds9.reg
# Region file format: CIAO version 1.0
ellipse(1628,4116,92,172,0)
```

  More examples of region syntax are available from ahelp dmregions.

- Use a stack input:

  This tool can take a stack as input, output, or both. To combine both of the previous examples into one command:

```
unix% more infiles.lis
acisf01843N001_evt2.fits[bin pha=1:2048:2]
```

```
acisf01843N001_evt2.fits[sky=ellipse(1628,4116,92,172,0)][bin pi]

unix% more outfiles.lis
histogram_lis.pha
spectrum_lis.fits

unix% dmextract infile=@infiles.lis outfile=@outfiles.lis
```

4. dmgti

Pipeline processing of Chandra data uses `dmgti` to calculate good time intervals (GTIs) based on input MTL files. Although this is what the tool was designed for, dmgti will work equally well on any GTI filter as long as the first column of the input file is time.

A look at a lightcurve file (created with `dmextract` and these parameters) shows background flares where the count rate reaches values much higher than the mean (several rows were omitted for the sake of space):

```
unix% dmlist "bkg_lc.fits[cols time,count_rate,stat_err]" data

--------------------------------------------------------------------------------
Data for Table Block LIGHTCURVE
--------------------------------------------------------------------------------

ROW    TIME                 COUNT_RATE           STAT_ERR

     1  84270903.7887200117                   0                   0
     2  84270907.0297600031                   0                   0
     3  84270910.2707999945                   0                   0
.
.
   490  84272488.6572799981         7.8124999849                 5.0
   491  84272491.8983199894         6.8749999867         4.6904157598
   492  84272495.1393600106        10.6249999794         5.8309518948
   493  84272498.3804000020         7.8124999849                 5.0
.
.
   742  84273305.3993600011        83.7499998377        16.3707055437
   743  84273308.6403999925       121.8749997639        19.7484176581
   744  84273311.8814400136        84.0624998371        16.4012194669
   745  84273315.1224800050        97.4999998111        17.6635217327
   746  84273318.3635199964        65.6249998728        14.4913767462
.
.
  3005  84280639.8728800118                   0                   0
  3006  84280643.1139200032                   0                   0
  3007  84280646.3549599946                   0                   0
```

The tool `dmsort` is used to sort the `count_rate` column in descending order, to show more clearly the times of high background:

```
unix% dmsort bkg_lc.fits sorted_bkg_lc.fits keys=-count_rate copyall=yes

unix% dmlist "sorted_bkg_lc.fits[cols time,count_rate,stat_err]" data | more

--------------------------------------------------------------------------------
Data for Table Block LIGHTCURVE
--------------------------------------------------------------------------------

ROW    TIME                 COUNT_RATE           STAT_ERR

     1  84274261.5061599910       179.6874996518        23.9791576166
```

```
     2  84274245.3009600043       177.4999996561        23.8327505756
     3  84274248.5419999957       168.4374996736        23.2163735325
     4  84274235.5778400004       164.9999996803        22.9782505862
.
.
   453  84280056.4856800139        77.1874998504        15.7162336455
   454   84280046.762560010        77.1874998504        15.7162336455
   455  84279926.8440800011        77.1874998504        15.7162336455
.
.
  2322  84272702.5659199953         8.1249999843         5.0990195136
  2323  84272634.5040799975         8.1249999843         5.0990195136
  2324  84272997.5005600154         7.8124999849                  5.0
.
.
  3005  84270910.2707999945                    0                    0
  3006  84270907.0297600031                    0                    0
  3007  84270903.7887200117                    0                    0
```

If we wanted to exclude the times when the count rate was less than 40.0 and greater than 100.0:

```
unix% dmgti infile=bkg_lc.fits outfile=acis_gti.fits \
      userlimit="(count_rate>40.0)&&(count_rate<100.0)"

unix% dmlist "acis_gti.fits[gti]" data
--------------------------------------------------------------------------------
Data for Table Block GTI
--------------------------------------------------------------------------------

ROW    START                STOP

     1  84273206.5476399958  84273209.7886800170
     2  84273213.0297200084  84273232.4759600163
     3  84273242.1990800202  84273307.0198799968
.
.
    72  84279795.5819600224  84279798.8230000138
    73  84279818.2692400217  84279857.1617200077
    74  84279860.4027599990  84280443.7899599969
```

The event file may now be filtered on the newly calculated GTI:

```
unix% dmcopy "acisf01843N001_evt2.fits[@acis_gti.fits]" acis_filtered_evt2.fits
```

Doing a dmlist on each of the files shows how the number of events and the GTI blocks are affected by this filter:

```
unix% dmlist acisf01843N001_evt2.fits blocks

--------------------------------------------------------------------------------
Dataset: acisf01843N001_evt2.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
Block    1: PRIMARY                        Null
Block    2: EVENTS                         Table        14 cols x 475869   rows
Block    3: GTI7                           Table         2 cols x 1        rows
Block    4: GTI0                           Table         2 cols x 1        rows
Block    5: GTI1                           Table         2 cols x 1        rows
Block    6: GTI2                           Table         2 cols x 1        rows
Block    7: GTI3                           Table         2 cols x 2        rows
Block    8: GTI6                           Table         2 cols x 1        rows

unix% dmlist acis_filtered_evt2.fits blocks
```

```
--------------------------------------------------------------------------------
Dataset: acis_filtered_evt2.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
Block    1: PRIMARY                        Null
Block    2: EVENTS                         Table        14 cols x 407991   rows
Block    3: GTI7                           Table         2 cols x 74       rows
Block    4: GTI0                           Table         2 cols x 74       rows
Block    5: GTI1                           Table         2 cols x 74       rows
Block    6: GTI2                           Table         2 cols x 74       rows
Block    7: GTI3                           Table         2 cols x 75       rows
Block    8: GTI6                           Table         2 cols x 74       rows
```

```
Parameters for /home/username/cxcds_param/dmcopy.par


        infile = acisf01843N001_evt2.fits Input dataset/block specification
       outfile = acis_events.fits Output dataset name
       (kernel = default)         Output file format type
       (option = )                Option − force output type
      (verbose = 0)               Debug Level
      (clobber = no)              Clobber existing file
         (mode = ql)
```

```
Parameters for /home/username/cxcds_param/dmextract.par


        infile = acisf01843N001_evt2.fits[(x,y)=field()−ellipse(1628,4116,100,140,0)][bin time=::
       outfile = bkg_lc.fits       Enter output file name
          (bkg = )                 Background region file or fixed background (counts/pixel/s) sub
       (error = gaussian)          Method for error determination(gaussian|gehrels|<variance file>
    (bkgerror = gaussian)          Method for background error determination(gaussian|gehrels|<var
      (bkgnorm = 1.0)                Background normalization
          (exp = )                 Exposure map image file
       (bkgexp = )                 Background exposure map image file
      (sys_err = 0)                Fixed systematic error value for SYS_ERR keyword
          (opt = ltc1)             Output file type
    (defaults = ${ASCDS_CALIB}/cxo.mdb −> /soft/ciao/data/cxo.mdb) Instrument defaults file
          (wmap = )                WMAP filter/binning (e.g. det=8 or default)
      (clobber = no)               OK to overwrite existing output file(s)?
      (verbose = 0)                Verbosity level
         (mode = ql)
```

History

23 Dec 2004	reviewed for CIAO 3.2: no changes
01 Dec 2005	updated for CIAO 3.3: default value of `dmextract` `error` and `bkgerror` parameters is "gaussian"; updates to screen output in dmgti example
01 Dec 2006	updated for CIAO 3.4: updates to screen output in dmgti example

URL: http://cxc.harvard.edu/ciao/threads/dm_intro/ Last modified: 1 Dec 2006