



Introduction to CIAO (Chandra Interactive Analysis of Observations)



CIAO: Chandra's data analysis system
Fruscione et al. 2006, SPIE Proc. 6270,
62701V, D.R. Silvia & R.E. Doxsey, eds.



From "ahelp ciao"....

- A powerful data analysis system originally written for the needs of users of the Chandra X-ray Observatory.
- Mission independent (a part from a few instrument specific tools).
- Built to handle *N-dimensional* data without concern about which particular axes are being analyzed: Chandra is the first mission with 4-dimensional data (2 spatial, time, energy) in which each dimension has many independent elements.



- Provides users with the ability to filter down and project the 4-D Chandra event data to manageable sizes and convenient array: all CIAO tools take a ‘*filtering and binning*’ specification on the command line, making use of a general purpose ‘regions’ syntax: “**ahelp dm**” for information on the Data Model that makes all this possible.
- To keep track of how the data had been filtered and binned CIAO relies on the ‘data subspace’. Tools keep track of this subspace automatically and allow users to review previous data processing: see “**ahelp subspace**”.



- The CIAO design allows close interconnection of tools. For example, the output of any of source detection program can be fed into *dmextract* to create a spectrum which can then be fit in *Sherpa*.
- The modeling and fitting tool *Sherpa* is central to the CIAO system. *Sherpa* performs forward fitting of models to data in N-dimensions. *Sherpa* (and ChIPS!) includes the “**Python**” (and “**S-Lang**”) languages which can be used for scripting and data manipulation. See “**ahelp sherpa**.”



http://cxc.harvard.edu/ciao : it all begins here!

Chandra X-ray Observatory

CXC Home Proposer Archive Data Analysis

Instruments & Calibration NASA Archives and Centers

Last modified: 19 January 2010

Search the CIAO website or contact the CXC Helpdesk

Look in here

Search here

Quick links

Navigate here

WHAT'S NEW! WATCH OUT

Quick CIAO links: [ChIPS](#) | [Sherpa](#) | [FITS with ChIP](#) | [CALDB](#) | [Chandra Source Catalog](#) | [TTCat](#) | [Chandra Guest Observer Catalog](#)

Download CIAO 4.2 and CALDB 4.2.0: the newest versions of the Chandra Interactive Analysis of Observations software and the Chandra Calibration Database, released on 15 December 2009.

Read the [CIAO 4.2 release notes](#) for detailed information on this release.

- The Sherpa modeling and fitting package has a number of enhancements including a new algorithm for calculating confidence, simulations to probe model parameter space and calculate flux uncertainties, full support for wavelength analysis, and a "save_all" command to save the session in ASCII format. It also has capabilities to run parallel processing. Refer to the [Sherpa Latest Updates page](#) for more information.
- The ChIPS plotting application includes beta support for displaying images and colorbars, as well as the ability to create an editable ASCII save file.
- Two new tools - `create_obs_map` and `delregions` - enable users to create spatial background maps equivalent to those in the Chandra Source Catalog and useful for source detection and for detection sensitivity maps.
- `axis_build_badpix` allows users more control over customizing the bad pixel files.
- `inc_process_events`, combined with files released in CALDB 4.2.0, can apply time-variable gain calibrations. Using the gain-corrected PI values can reduce the background.
- `tg_create_mask` has improved the custom mask settings.

The CIAO 4.2 software requires CALDB 4.2.0 to work correctly. Information on the CALDB 4.2.0 release is available from the [CALDB 4.2.0 Release Notes](#) and [How CIAO 4.2 and CALDB 4.2.0 Affect Your Analysis](#).

Transitioning to ChIPS and Sherpa in CIAO 4

The CXC is committed to helping CIAO users transition to the new [ChIPS](#) and [Sherpa](#) syntax as smoothly as possible. If you have existing scripts or save files from CIAO 3, submit them to us via the [CXC Helpdesk](#), and we will provide the CIAO 4 equivalent.

Please be sure to tell us if you are working in S-Lang or Python, and place the relevant data files on the [FTP \(FTP Instructions\)](#).

Ever never used CIAO before, Where should I begin?

Welcome to CIAO
[Introduction to the Tools & Applications](#)

[Platform Support](#)

[Download CIAO 4.2](#)
[Installing CIAO 4.2 Binaries](#)

[Introductory Science Tutorials](#)



Chandra X-ray Observatory

CXC Home | Proposer | Archive | Data Analysis

Instruments & Calibration | NASA Archives and Centers

See modified: 20 January 2010

Search the Sherpa website or contact the CXC Helpdesk

Sherpa

CIAO's modeling and fitting package

WHAT'S NEW | WATCH LIST

Home | About | Installation | User's Guide | FAQ | Support

Sherpa is the CIAO modeling and fitting application. It enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistical and optimization methods (see [Gallery of Examples](#)). For the most up-to-date changes and additions to Sherpa functionality, see the [Latest Update page](#).

Sherpa lets you:

- fit 1-D data sets (simultaneously or individually) including spectra, surface brightness profiles, light curves, general AMOS arrays;
- fit 2-D images/regions in the Photon/Counts regime;
- access the internal data arrays;
- build complex model expressions;
- import and use your own models;
- choose appropriate statistics for modeling Photon or Counts data;
- import raw statistics, with priors if required by analysis;
- visualize a parameter space with simulations or using 1-D/2-D views of the parameter space;
- calculate confidence limits on the best-fit model parameters;
- choose a robust optimization method for the fit: Levenberg-Marquardt, Nelder-Mead Simplex or Simplex-Cover Differential Evolution;
- use Python to create complex analysis and modeling functions, build the Sherpa model analysis or extend the provided functionality to meet the required needs.

Chandra X-ray Observatory

CXC Home | Proposer | Archive | Data

Instruments & Calibration | NASA Archives and

See modified: 21 July 2009

Search the ChART website or contact the CXC

Chandra Ray Tracer (ChART)

The Chandra Ray Tracer (ChART) - the Chandra PSF simulator - is a web interface to the SAOAc raytrace code which was developed by the CXC for calibration purposes. ChART traces rays through the [Chandra X-ray optics](#) to produce a collection of rays. The rays are then projected onto the detector (via [SARAD](#)), taking into account any detector effects. The result is an event file from which an image of the point spread function may be created.

Since ChART uses the same code that is used internally at the CXC for calibration, it gives the best available 1800A PSF for a point source at any off-axis angle and for any energy in the spectral range available from the ChART description page.

Why should you use ChART instead of the standard PSF libraries? The [ChART Compared to PSF Libraries](#) page explains why ChART is the recommended method.

Citing ChART in a Publication

If you are writing a paper and would like to cite ChART, we recommend the following paper:

ChART: The Chandra Ray Tracer
C. Corbett et al.
ASAS 500 ASP Conference Series, Vol. 286, 2863, p.477

The specific version of CIAO and CALDB (if applicable) used for the analysis should be mentioned as well.

See modified: 21 July 2009

CXC Home | Chandra Profile | Astronomers' Guide | CXC/CXC web | Helpdesk

The Chandra X-ray Center (CXC) is operated by NASA for the Smithsonian Astrophysical Observatory, 60 Garden Street, Cambridge, MA 02138 USA. Email: casper@chandra.harvard.edu or helpdesk@chandra.harvard.edu. Copyright © 2009-2010. All rights reserved.

Chandra X-ray Observatory

CXC Home | Proposer | Archive | Data Analysis

Instruments & Calibration | NASA Archives and Centers

See modified: 1 December 2009

Search the ChIPS website or contact the CXC Helpdesk

ChIPS

The Chandra Imaging Package

WHAT'S NEW | WATCH LIST

Home | About | Installation | User's Guide | FAQ | Support

ChIPS is a software package for Chandra X-ray data analysis. It is used to extract data from the Chandra X-ray Observatory and to create publication-quality figures. A range of examples are included in the [ChIPS Gallery](#), and there are a number of [presentations](#) related to point regions.

The ChIPS is intended to help Chandra users transition to new systems as smoothly as possible. If you have existing ChIPS scripts or user files, select them to see the [ChIPS Migration](#) and we will provide the ChIPS-ChIPS system to you. Refer to the [Chandra to the New System](#) for more information.

See modified: 14 December 2009

Search the CALDB website

Welcome to the Chandra Calibration Database Website

The Chandra X-ray Center introduced version 4.0.0 of the Chandra Calibration Database (CALDB) on 12 FEB 2009.

CALDB 4.0 is a brand new, newly reorganized, newly reorganized collection database for Chandra, in particular for CIAO 4.0. The directory structure, table configuration (Index Files), and the software interfaces are all changed/updated.

The CALDB 4.0.0 upgrade is a major (structural) upgrade to CALDB 3.0.0 and CALDB 4.0.0 are mutually required to implement several of the upgrades in 4.0.0 internally. CALDB 4.0.0 includes the following items:

- NEW CALDB 4.0.0 requires users, beginning with all CIAO 4.0.0 observations, with a robust package, like CIAO 4.0.0.
- NEW CALDB 4.0.0 requires users to use CIAO 4.0.0 for CIAO 4.0.0. The new CALDB 4.0.0 requires CIAO 4.0.0 or later to work properly in CIAO 4.0.0.
- NEW CALDB 4.0.0 requires users to use CIAO 4.0.0 for CIAO 4.0.0. The new CALDB 4.0.0 requires CIAO 4.0.0 or later to work properly in CIAO 4.0.0.

See details in the [CALDB 4.0.0 Release Notes](#).

CALDB 4.0.0 is for use only with CIAO 4.0.0 and may be accessed and downloaded from the [CALDB 4.0.0 website](#) and [download page](#). Previous versions of CALDB will be retained in CALDB 3.0.0 for at least throughout calendar year 2009. See the [CALDB 3.0.0 website](#) for more information.

The CALDB is the primary used indexing structure that stores and provides access to all calibration files that are required for standard processing and analysis.

These files are used in standard data processing, as well as for analysis with software packages such as CIAO. When you apply a gain map, create an exposure map, or build response files (RMF and ARF), among many other things, you are using the Chandra Calibration Database.

The CALDB serves many functions:

- Storing and archiving calibration files.
- Maintaining a working structure and header structure for all calibration files.
- Indexing calibration files, based on FITS header keywords, for software access.
- Providing updates of calibration data independent of software updates, while maintaining configuration control.
- Providing a consistent version of calibration data to the detector for maintaining monitoring.
- Providing calibration products (see below) available for processing and/or analysis.

These webpage update information about each of these topics. They are also designed to help you for your system manager's understanding how the CALDB is accessed and how it works in the Chandra software system.

Please read the [ChART](#) website with any questions or comments on the CALDB and this website.



CIAO help!

AHELP

- CIAO comes with its own help system called **ahelp**.
- Every component of the CIAO system has its own help text: tools (e.g. `dmcopy`, `wavdetect`), applications and their components (e.g. `sherpa`, `fit`, `model`), scripting language and its functions (e.g. `python`, `get_fit`, `get_data`) and numerous concepts (e.g. `regions`, `coords`, `dmsyntax`, etc.).
- The entire collection of `ahelp` files currently has more than 1000 pages!

The **ahelp** commands (see `ahelp ahelp` for details) access the CIAO on-line documentation. For example:

% **about <word>** (e.g. `about contour`) gives a list of subjects related to the query

% **ahelp <name>** (e.g. `ahelp tgextract`) provides by default the ASCII version of the help file.

% **sherpa/chips> ahelp("name") or ,ahelp name** (e.g. `ahelp("fit")` or `,ahelp fit`) provides the ASCII version of the `ahelp` file within the `sherpa` or `chips`



DATA ANALYSIS GUIDES : *start from here!*

- A roadmap through the threads.
- Arranged by instrument (ACIS, HRC) or type of analysis (e.g. extended sources)

DATA ANALYSIS THREADS (*General, Sherpa, ChIPS, Chart*)

- More than 100 CIAO processing recipes designed to teach users by leading step-by-step through a procedure.
- Added or updated as needed (look for the “**new**” or “**updated**” icons!).
- Several threads and groups of threads have been or will be made more automated in the form of scripts.
- PDF version available on-line, and threads are printer friendly.
- Quick “overview” at the beginning (synopsis, purpose, when to use etc.)



IMPORTANT CIAO WEB PAGES

Read especially the

What'sNew, "Watch out", Bugs, Caveats, Dictionary, FAQ pages, Platform support, Why topics, How does CIAOX.X and CALDBY.Y affects my analysis, Manuals and Memos, !

HELPDESK

When everything else fails....

<http://cxc.harvard.edu/helpdesk/>

and specify CIAO version, platform and OS, data, purpose etc...

Answers are very fast!!!! Because...

HelpDesk Staff



Liz



Nick



Nina

12 Years of CIAO Experience: for you on
HelpDesk and ... at this workshop!



WHAT IS CIAO?

- Collection of programs (*tools, applications, scripts, (S-Lang and) Python modules*).
- Generally run from the shell; some part can be run from GUIs.
- Source code is available to users.
- Available on several platforms; currently (as of CIAO 4.2) supported:
 - Solaris 10
 - Linux 32 bit and 64 bit
 - Mac OS X 10.5 Intel 32 bit and 64 bit and 10.5 Intel
 - Mac OS X 10.4 PPC 32 bit (**for the last time!**)

Has been successfully installed by users on other “unsupported” (= no regression testing done!) platforms: for example Mac OS X 10.6. Check the *Platform Support* page

The CIAO Environment

We recommend the use of an alias called “`ciao`” to start up the system: it sets up number of environment variables and path assignments.

`ciao -v` tell what version you are using (useful when reporting problems)



General Concepts

- File format
- Parameter Files
- Filters
- Regions
- Good Times Intervals (GTI)
- Scripting language



File Format

- Chandra data is stored in FITS format. ASCII (text) files can now be handled by all tools and applications through the new software library known as the “ASCII kernel”
- When CIAO operates on data it stores processing state/information along with data (keywords, subspace).
- A single file can contain multiple “datasets” (e.g. data, GTI, weight map, regions) stored in “blocks”.
- Blocks can contain image or table data. Table columns can be vectors.
- **dmlist** (a command line tool) or **prism** (a GUI) are available to view file contents.



Prism: file viewer, editor, plot and histogram launcher

Header

Plot Dialog Window

Blocks

The screenshot shows the Prism software interface with three main components:

- Header:** A window titled 'prism (on devel6)' showing the FITS header for 'rprofile_exclmore_DTCOR.fits'. It lists keywords like TSTART, TSTOP, COMMENT, TIMEUNIT, TIMESYS, MJDREF, TIMEUNIT, TASSIGN, and TIMEZERO.
- Data:** A table window showing a histogram with columns for 'units', 'BG_ERR', 'BG_COUNTS', 'BG_RATE', 'BG_SUR_BR', 'BG_SUR_BR_ERR', 'NET_COUNTS', 'NET_ERR', 'NET_RATE', and 'ERR_RA'. The data shows a decreasing trend of counts as area increases.
- Plot Dialog Window:** A 'Plot Dialog' window with 'Plot Settings'. It shows 'Curve X' set to 'AREA', 'Y' set to 'NET_COUNTS', and 'Errs' set to 'NET_ERR'. It also allows for selecting line styles, symbol styles, and error styles.

Plot



Parameter Files (ahelp parameter)

- Processing parameters for CIAO tools can be set on the command line or, as with IRAF and FTOOLS, using parameter files.
- Parameters files are
 - ✓ stored in $\$HOME/cxcds_param4/$
 - ✓ are called $\langle tool \rangle.par$ (e.g. `dmcopy.par`)
 - ✓ are ASCII files.

CAVEAT: Always delete or rename the *cxcds_param* when upgrading operating system or CIAO version

- A “Parameter Editor GUI” (`peg`) and a number of routines (e.g. `plist`, `pset`, `punlearn`) are provided to read and write to these files. A S-Lang interface to



Parameter Files (ahelp parameter)

- Processing parameters for CIAO tools can be set on the command line or, as with IRAF and FTOOLS, using parameter files.

```
antonell@bonsai:/home/antonell — ssh — 75x13
ciao-65: plist dmlist

Parameters for /home/antonell/cxcds_param4/dmlist.par

    infile = /tmp/test.fits    Input dataset/block specification
      opt = cols,subspace     Option
(outfile = )                 Output file (optional)
  (rows = )                   Range of table rows to print (min:max)
  (cells = )                  Range of array indices to print (min:max)
(verbose = 0)                 Debug Level(0-5)
  (mode = ql)

ciao-66: █
```




Parameter Files (ahelp parameter)

- Processing parameters for CIAO tools can be set on the command line or, as with IRAF and FTOOLS, using parameter files.
- Parameters files are
 - ✓ stored in $\$HOME/cxcds_param4/$
 - ✓ are called $\langle tool \rangle.par$ (e.g. `dmcopy.par`)
 - ✓ are ASCII files.

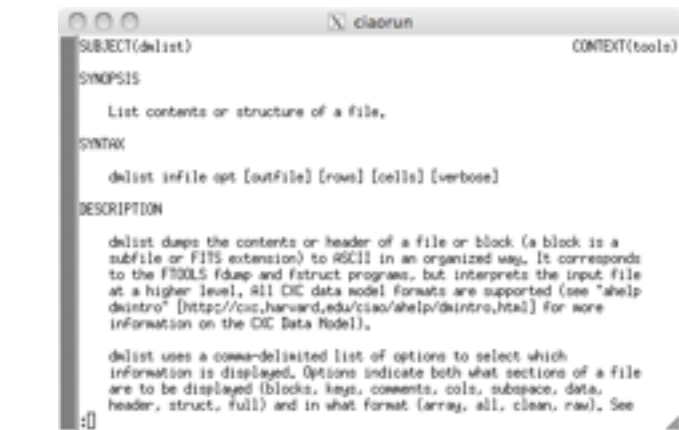
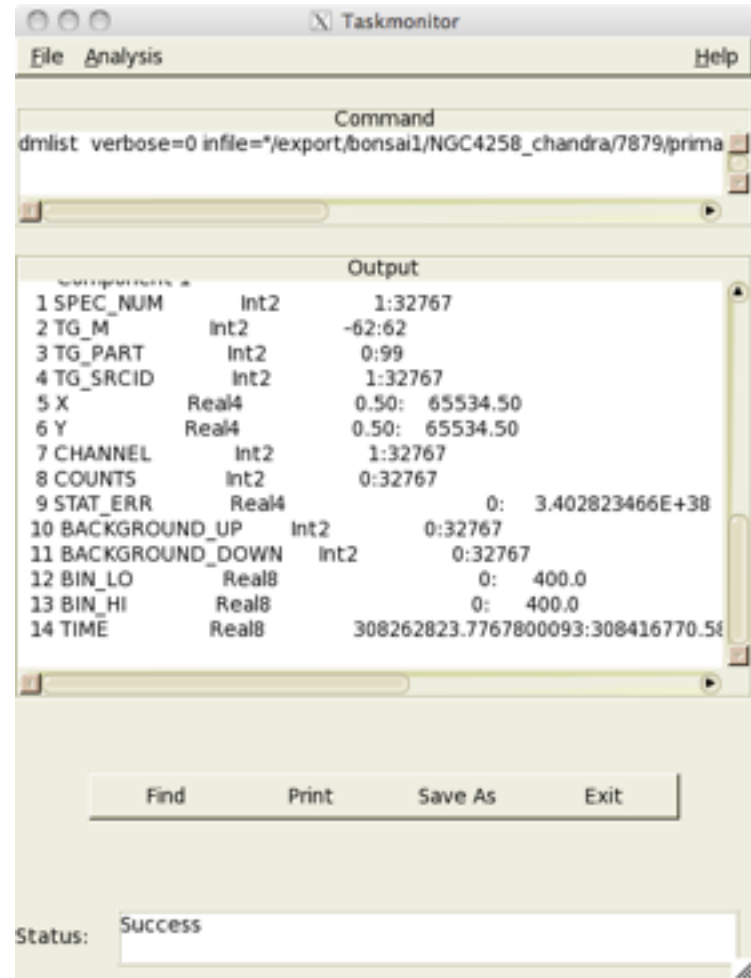
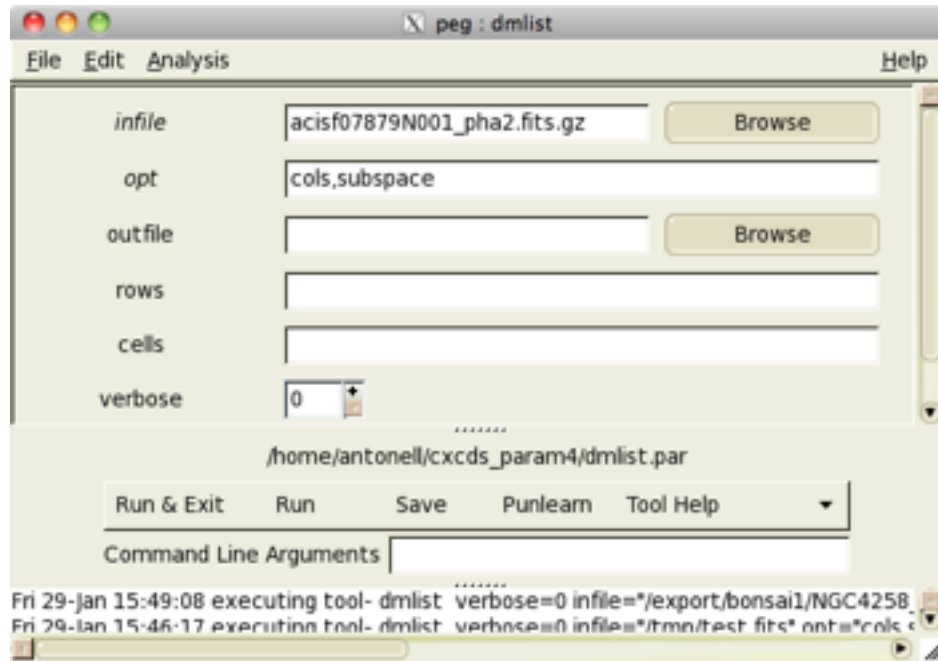
CAVEAT: Always delete or rename the *cxcds_param* when upgrading operating system or CIAO version

- A “Parameter Editor GUI” (`peg`) and a number of routines (e.g. `plist`, `pset`, `punlearn`) are provided to read and write to these files. A S-Lang interface to



Parameter Editor GUI (PEG)

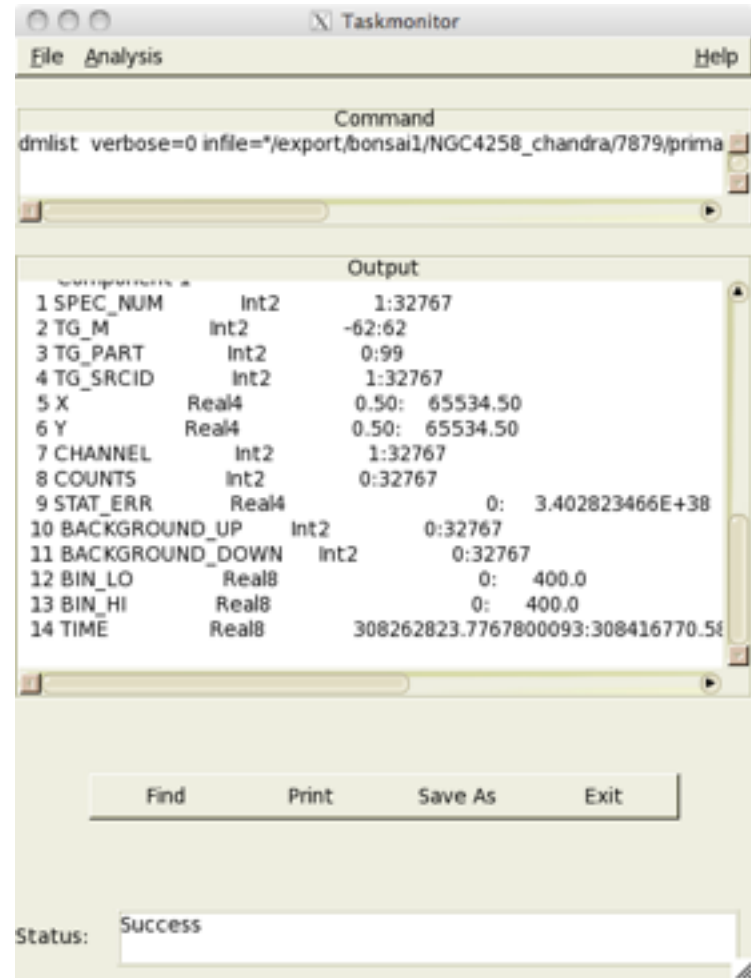
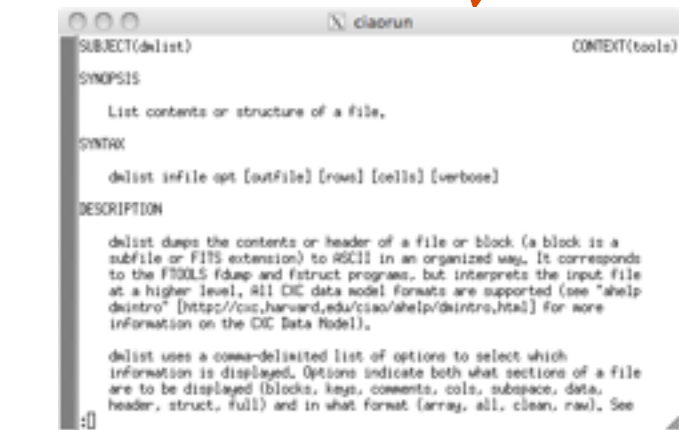
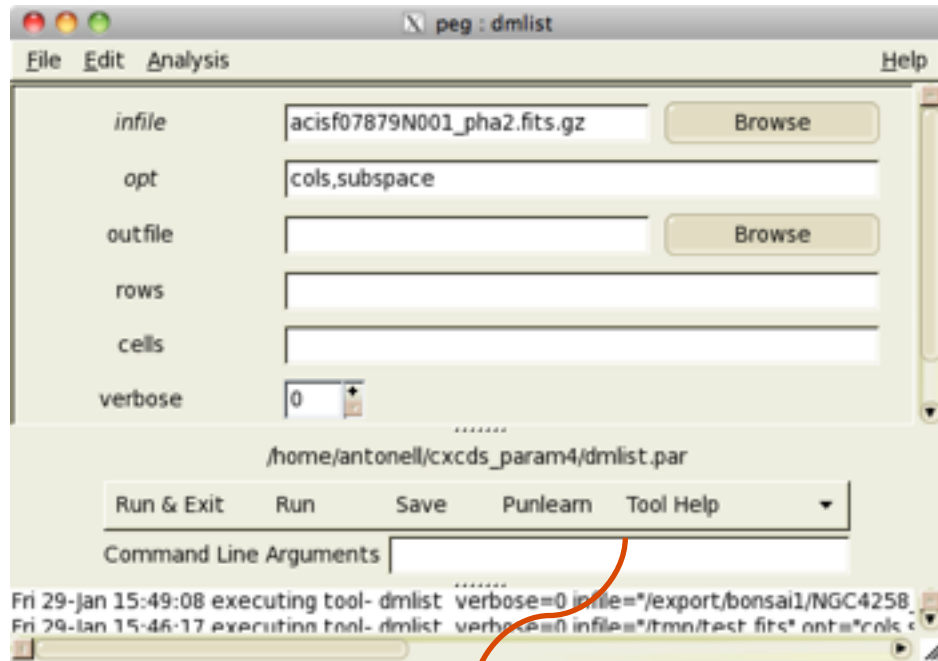
unix% peg dmlist





Parameter Editor GUI (PEG)

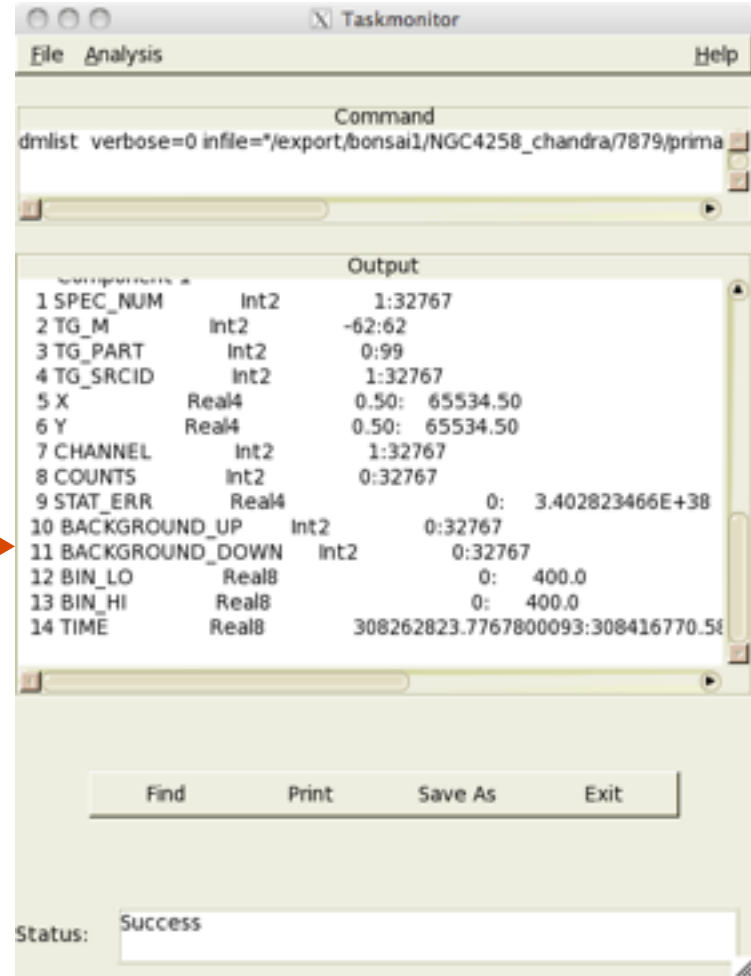
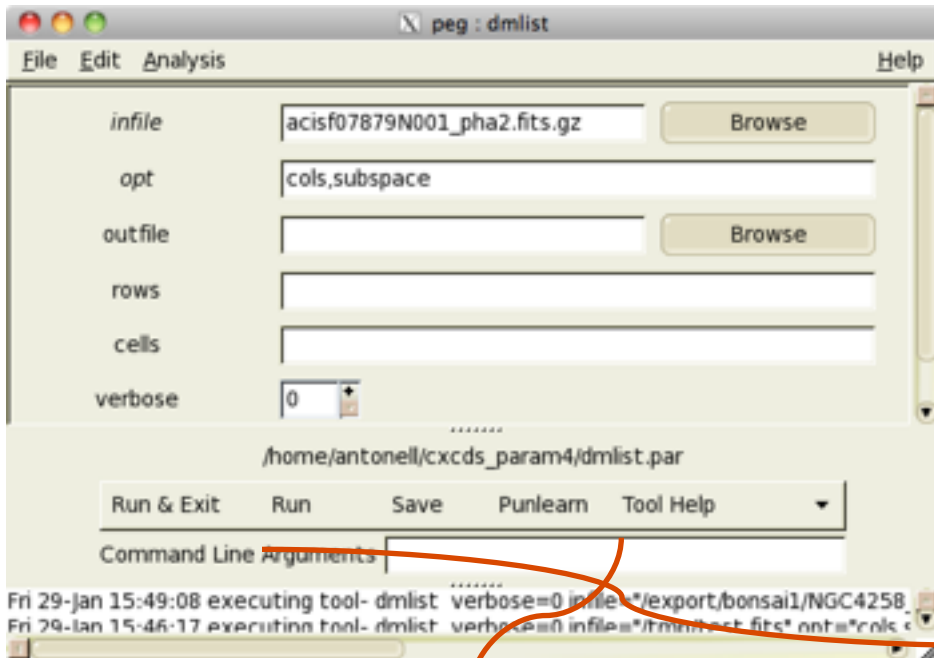
unix% peg dmlist





Parameter Editor GUI (PEG)

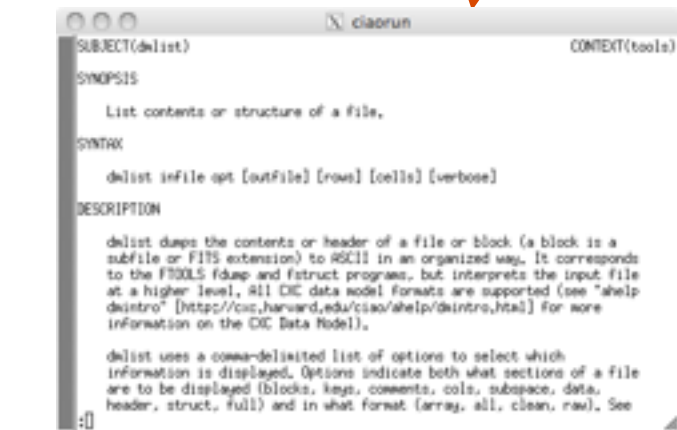
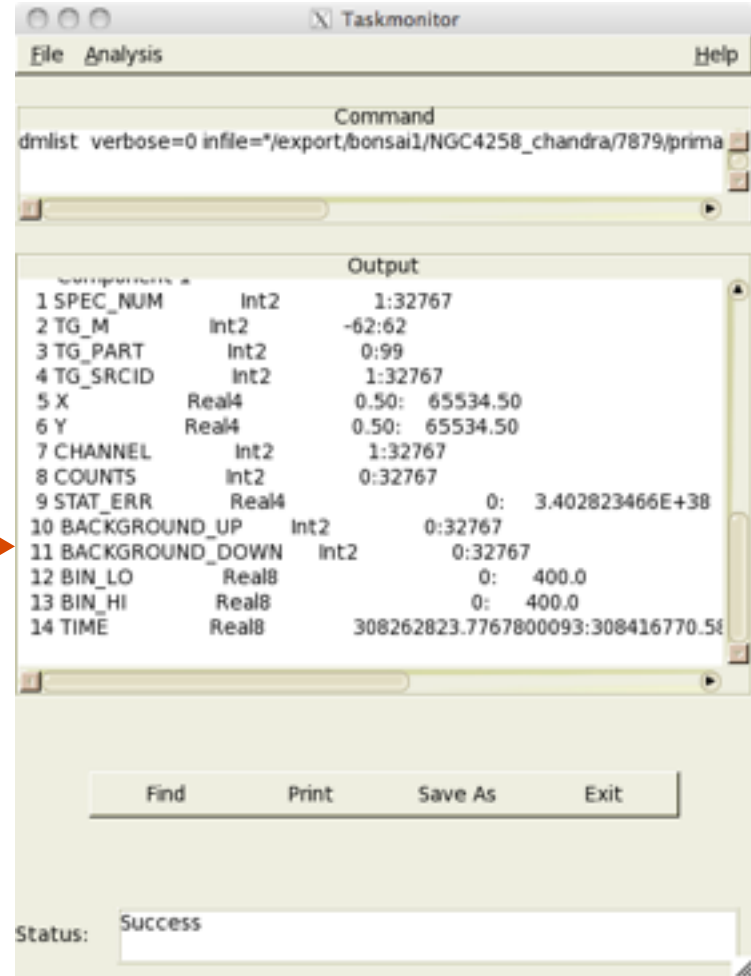
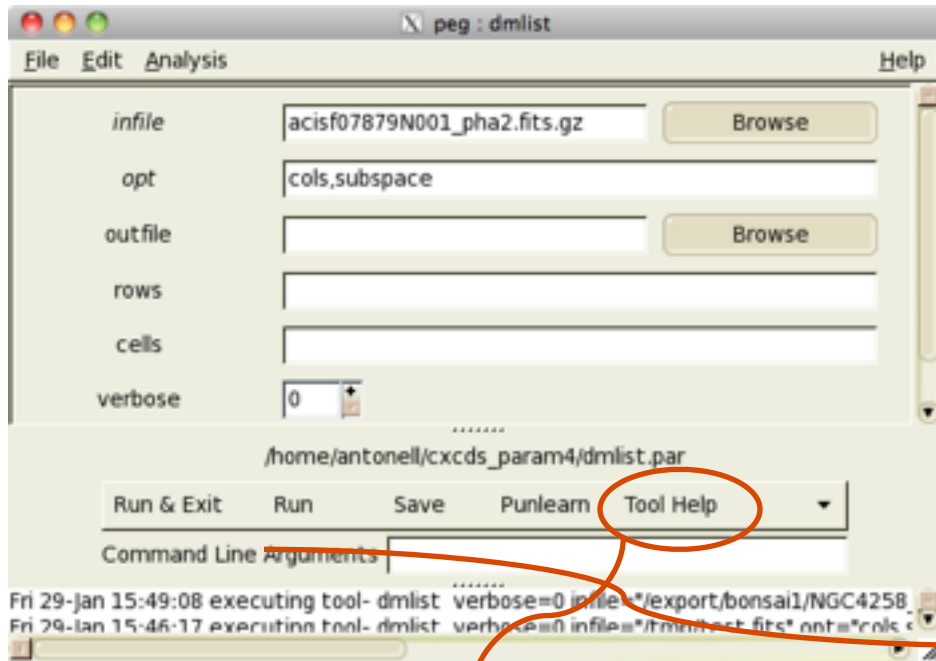
unix% peg dmlist





Parameter Editor GUI (PEG)

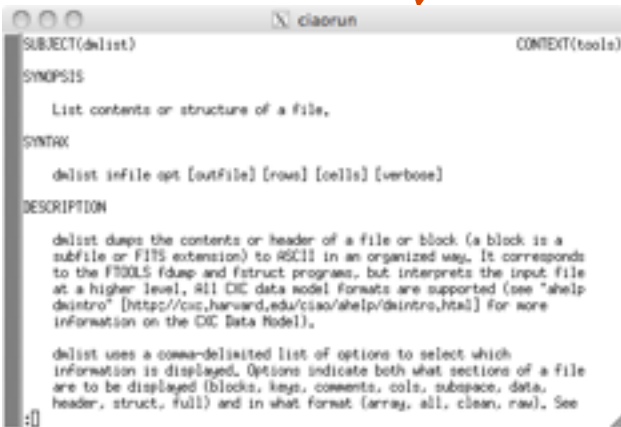
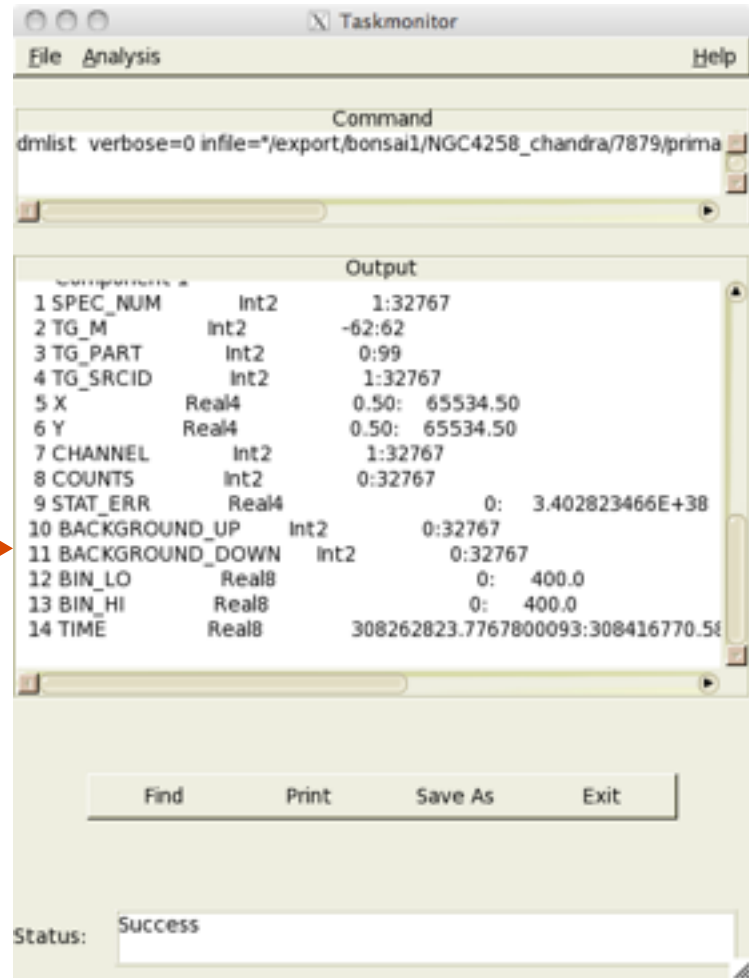
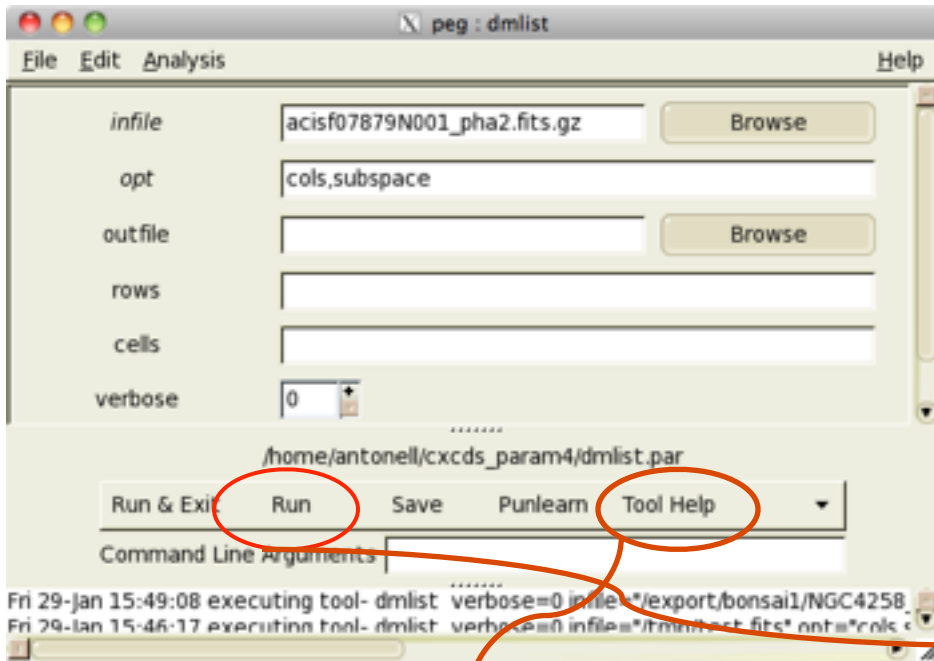
unix% peg dmlist





Parameter Editor GUI (PEG)

unix% peg dmlist





Filters, Regions, and GTIs

- filtering (removal of unwanted events) is an essential part of X-ray analysis - e.g. to remove periods of high background or poor aspect solution, exclude uninteresting sources from an image etc.
- the DataModel (DM) provides great filtering flexibility: e.g. `dmstat "evt2.fits [EVENTS][energy>300][cols -grade]"` (see [ahelp filtering](#), [ahelp dmimgfiltering](#))
- GTIs (Good Time Intervals) are used to define what times periods of the observation can be used (i.e. contain valid data). They are generally stored as a block in the event list (see [ahelp chandra times](#))
- Regions are used to define the source and background areas of an image. They are text files that can be created manually or within ds9, and are used as a filter (e.g. `"[sky=region(source.reg)]"`). (see [ahelp dmregions](#))
- Subspace records the filters applied to a file; `dmlist` can read this history using `opt=subspace` (see [ahelp subspace](#))



Scripting Languages

- The main scripting (or interpreted = no compilation is necessary) language supported in CIAO via Sherpa and ChIPS is Python.
S-Lang (introduced in CIAO2.0) is still available in CIAO4.2 but no further development is planned and will be phased out.
- Sherpa and ChIPS are importable modules for Python
- You DO NOT need to know Python to use Sherpa and Chips, but IF YOU DO, you will be able to use its capabilities in your analysis (see Doug's talk on "Scripting in CIAO")



CIAO overview

Data manipulation: copy, filter, extraction, stats, etc.

Data preparation (or Chandra-specific instrument tools): update calibration, correct for instrumental effects, find & extract grating data, create aspect histograms

Response tools: exposure map, PSF, RMF and ARF

Source Detection: celldetect, wavdetect, vtpdetect

Timing & Background tools: lighcurve, power spectrum, barycenter correction

Convolutions, Transforms, & Smoothing: csmooth, aconvolve, acrosscorr, apowerspectrum

Plotting: ChIPS (*)

Modeling/Fitting: Sherpa (*)

S-Lang: modules (parameter, region, group, pixlib (coord. transformation), caldb, stack, xpa, varmio); shell (“slsh”) to execute S-Lang scripts on the command line

GUIs: DS9, prism,peg

(*) powerful data manipulation and scripting capabilities are now possible in these applications through the Python and S-Lang interpreted language. ChIPS and



The “Data Model” and the Data Manipulation Tools

- The CXC analysis and processing software is built on a common versatile interface library called the CXC Data Model (or just DM).
- The DM provides users with a powerful built-in data filtering and binning capability.
- The name “Data Model” reflects the fact that the interface can be used on data files of different format (all described by a single abstract description - the same “model”) in a transparent way.
- The latest addition to the DM is the “ASCII kernel” which gives the ability to operate on ASCII (text) file the same way as on FITS files (eg for filtering, plotting etc.)



- An important characteristic of the DM is that ANY program that asks for a data file name as input accepts a “*virtual file*” string which causes the program to see a filtered version of the file in question.
- The “virtual file” syntax is also commonly used to create on disk a filtered version of the input file.
- Another important characteristic of the DM is that all columns of event lists are treated “equally”: for example binning is allowed not only in spatial coordinates but also in e.g. time, or energy coordinate, giving the ability of creating multidimensional images in space-energy, or space-time, etc.



Data Manipulation Tools

The four DM “core” tools are:

dm`list`: list contents or structure of a file

dm`copy`: filter and bin tables and images

dm`extract`: make a histogram table file (e.g. PHA file, lightcurve file) from a table column. Generate count histogram on supplied regions for a spatial table or image file.

dm`gti`: create custom Good Time Intervals (GTIs) from a constraint expression

30+ data manipulation tools are included in CIAO



DATA MODEL SYNTAX (`ahelp dmsyntax`)

- All CIAO tools use the DM library and therefore accept as input “virtual files” described using the DM syntax.
- In the DM context a “virtual file” is represented by a filename followed by a series of optional qualifiers in square brackets []:

“filename[block][filter][columns/binning][options][rename]”

where:

block - is the “section” of the file to use

filter- is the filter to be applied

columns/binning - specifies either the columns from a table to be included in an output table or the binning. When binning the data to generate an n- dimensional image, the range and binsize (min:max:bin) must be specified.

options - a sequence describing special options for the DM library

rename - specifies a name for the new block

Note that:

- the order of the qualifiers generally matters, however...
- not all qualifiers need to be present always



Simple examples of “virtual files”:

- Select the first three columns of the EVENTS block by number:

```
acisf01843N001_evt2.fits[EVENTS][time=84245787:84247000]  
[cols #1,#2,#3]
```

or by name:

```
acisf01843N001_evt2.fits[EVENTS][grade=0,2,3][cols  
time,ccd_id,node_id]
```

after filtering in time or grade

- Bin an events file to create a PI spectrum for a specified region

(input of dmextract): **acisf01843N001_evt2.fits[EVENTS]
[sky=region(mysrc.reg)][bin pi=1:1024:1]**

or an image (input of dmcop): **acisf01843N001_evt2.fits**



In the examples above:

block: [EVENTS]

filter: [time=84245787:84247000]

[grade=0,2,3]

[sky=region(mysource.reg)]

[pha<1000]

columns/binning:

[cols time,ccd_id,node_id]

[cols #1,#2,#3]

[bin pi=1:1024:1]

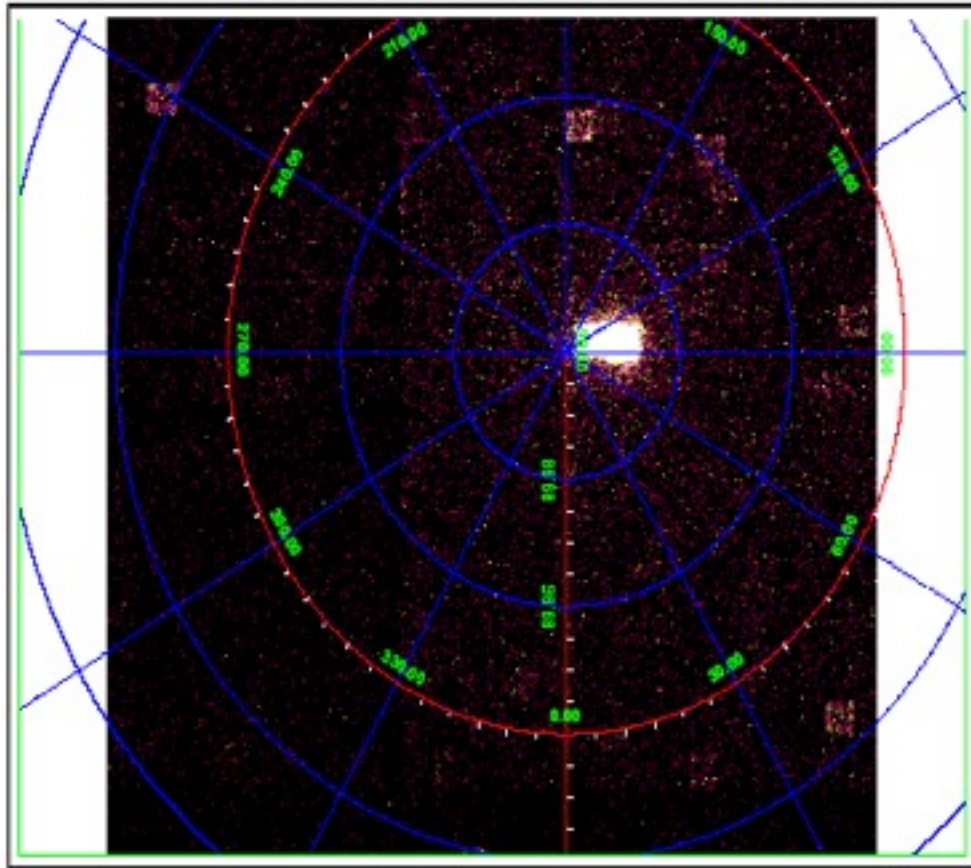
[bin x=320:480:4,y=320:480:4]



DM Examples 1: Detector Image

Imaging on multiple coordinate systems: first, let's look at a region in detector coordinates, filtered on energy and time.

```
dmcopy "merge3e.fits[energy=500:2000, time=:63940080, 63940180:] [bin
```





DM Examples 2: Sky Image

Now look at the same photons but in sky coordinates

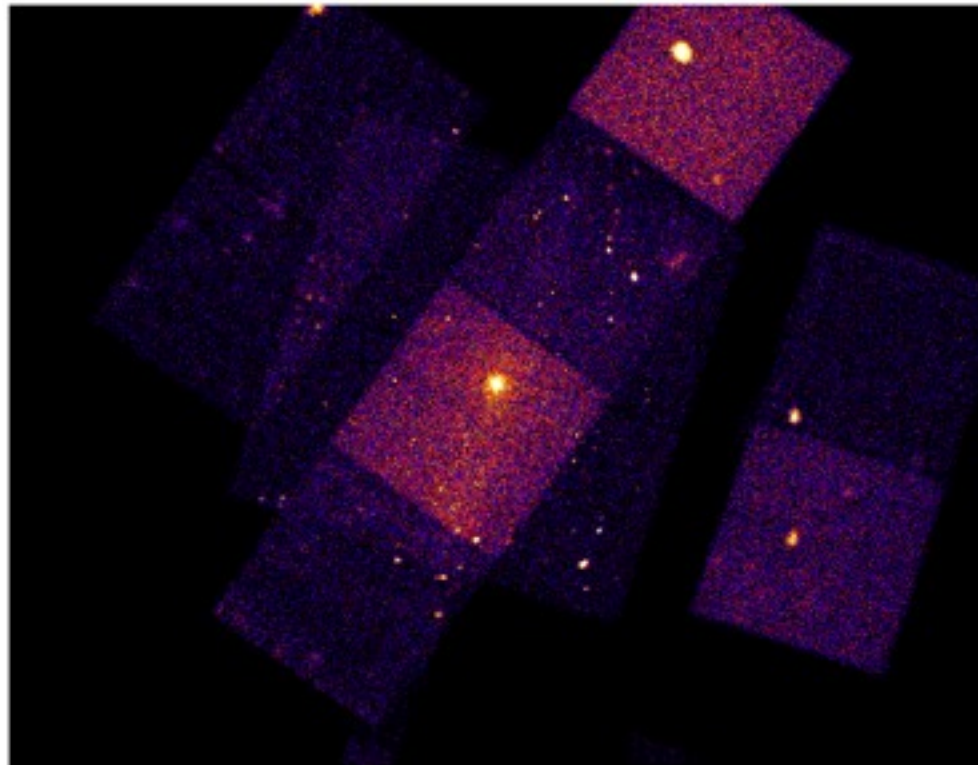
```
dmcopy "merge3e.fits[energy=500:2000, time=:  
63940080,63940180:,detx=3500:4500,dety=3500:4500][bin x=3200:4800:2,  
y=3200:4800:2]" sky.img
```





DM Examples 3: Merged sky image

The whole field was created by merging three separate observations.

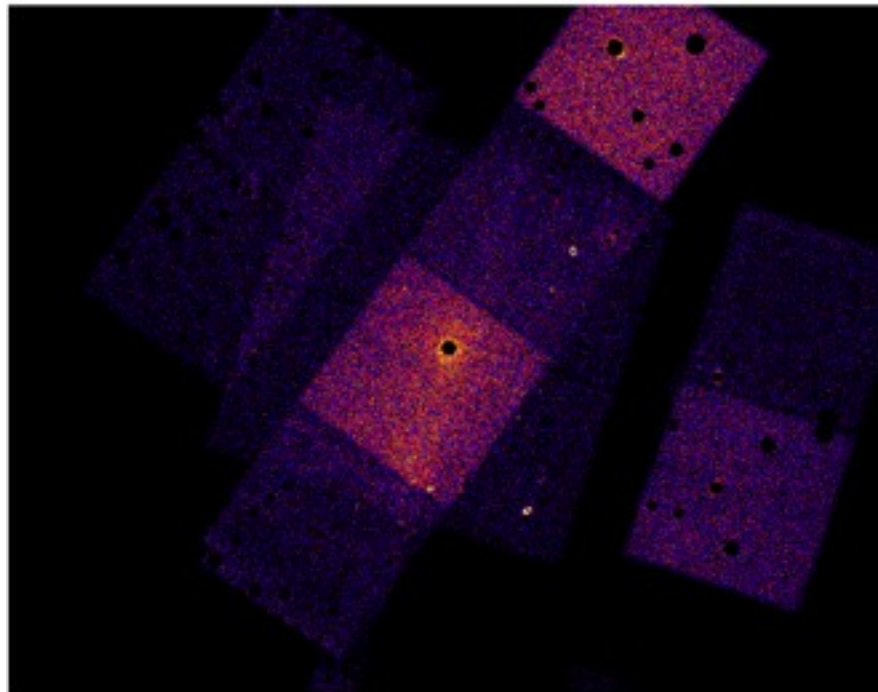




DM Examples 4: Removing sources

We can generate a background image by removing sources found by the automatic source detection program.

```
dmcopy "merge3e.fits[exclude sky=region (gg.reg)]" exclude.fits
```





DM Examples 5: Infrared spectroscopy data

ISO data: LWS LSAN file. This is a very simple file by wavelength and flux for the different detectors and scans are mixed together. We can use the DM tools to isolate a single scan and dump wavelength versus flux for it.

```
dmcopy "lasan59901083.fits[lsancnt=4][cols lsanwav, lsanflx]"
```

```
subset.fits
```

```
dmlist "lasan59901083.fits[lsanscnt=4][cols lsanwav, lsanflx]"
```

