



CIAO 3.2 Release Notes

Return to: [Version History](#)

- [How CIAO 3.2 and CALDB 3.0.0 Affect Your Analysis](#)
 - [Tools](#)
 - [Parameter Files](#)
 - [Data Model](#)
 - [Sherpa](#)
 - [Libraries](#)
 - [S-Lang Modules](#)
 - [Analysis Scripts](#)
 - [Documentation](#)
-

How CIAO 3.2 and CALDB 3.0.0 Affect Your Analysis

Some tools and calibration released in CIAO 3.2 and CALDB 3.0.0 affect analyses in progress, i.e. certain processing steps should be run again for greater accuracy. This section of the release notes describes those changes, as well as the affected tasks.

ACIS QE Degradation: Spatial Component

- The ACIS QE degradation model has been enhanced to account for spatial variations in the contamination on the ACIS optical blocking filters. The contamination is now expressed as a function of time, energy, and ACIS chip coordinate.

For imaging analysis of extended sources or point sources far off-axis, there is a significant change in instrument and exposure maps when the new calibration is applied.

The spatial contamination correction is automatically incorporated into exposure maps and ARF files (made by `mkinstmap/merge_all`, `mkarf/psextract`, `mkgarf`).

See the ACIS QE Degradation why topic for additional information.

New method for creating ACIS imaging RMFs

- The tool `mkacisrmf` is a new addition in CIAO 3.2 and represents an entirely new method for creating ACIS response matrices. The tool is calibrated for ACIS chips 0–3, 5, 6, and 7 (the ACIS–I array, plus ACIS–S1, S2, and S3). The response files must be used with –120 C data that has the time-dependent gain adjustment and CTI correction (if available) applied.

The results from `mkacisrmf` and `mkrmf` are very similar. Most users will not lose any accuracy by finishing the analysis with RMFs created by `mkrmf`. The exceptions are the following cases:

- ◆ The new RMFs are 10–15% more accurate at energies lower than 1 keV. If your analysis is sensitive at this level, create a new RMF with `mkacisrmf`.

◆ Calibration for the ACIS-S1 (`ccd_id=5`) back-illuminated chip has been greatly improved. Anyone doing imaging analysis on this chip should reprocess with `mkacisrmf`. For cross-calibration consistency, users doing either of these analyses should reprocess their dataset with the new "phase 2" gain file (the default gain in CALDB 3.0.0).

The Creating ACIS RMFs in CIAO 3.2 why topic has further details on the new tool.

ACIS Hot Pixel Tools

- New tools were introduced in CIAO 3.2 to identify and flag hot pixel and afterglow events in ACIS observations: `acis_build_badpix`, `acis_classify_hotpix`, and `acis_find_hotpix`. Also included is `acis_run_hotpix`, a script that runs the ACIS hot pixel tools in the correct order.

These tools supersede `acis_detect_afterglow` and replaced that tool in standard data processing in DS 7.4.0. `acis_run_hotpix` is better because it finds some afterglows that `acis_detect_afterglow` does not, it also finds pixels that are hot or have bad bias values, and it misidentifies far fewer legitimate X-ray events as bad.

The Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread has more information on these tools and how to use them. Note that it is necessary to reprocess the data with `acis_process_events` in order to apply the new bad pixel file.

Grating Analysis Improvements

- A number of CIAO tools have had improvements that will positively impact grating analyses: `asphist`, `mkgarf`, `tg_create_mask`, `tg_resolve_events`, and `tgextract`. The changes are detailed in the Tools section of these release notes.

CALDB changes since the CIAO 3.1/CALDB 2.27 release

- CALDB v2.27 was released with CIAO 3.1 on 7 July 2004. For users who have not upgraded their CALDB since then, here is a summary of significant calibration updates. All of these changes are included in the CALDB 3.0.0 release; see the CALDB release notes for details on these and other changes.
- **ACIS QE version 5** (CALDB v2.28): contains new QE information for for chips ACIS-5 and ACIS-7 (the back-illuminated chips). Users will be able to obtain more consistent flux measurements between the front- and back-illuminated chips with this update.

There is still a residual few percent difference between the FI and BI chips which will be accounted for in a future calibration update.

- **ACIS Time-dependent gain correction** (CALDB v2.29): new time-dependent gain calibration files for February, May, and August 2004 (epochs 18 and 19) were released. Plots in the ACIS Time-dependent Gain why topic illustrate how the new calibration files affect the magnitude of the correction.

Tools

`acis_build_badpix`

- **New Tool!**

One of the new ACIS hot pixel tools; creates an observation-specific bad-pixel file.

For more information on using this tool, see the Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread.

acis_classify_hotpix

- **New Tool!**

One of the new ACIS hot pixel tools; determines whether a suspicious pixel is hot or affected by an afterglow.

For more information on using this tool, see the Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread.

acis_find_hotpix

- **New Tool!**

One of the new ACIS hot pixel tools; identifies pixels that have an unusually large (or small) number of events.

For more information on using this tool, see the Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread.

acis_process_events

- unneeded `cc_chipy` parameter removed
- Bug fix: if the string "qp" is in the output filename the tool segmentation faults (fix in dslib)
- updates to allow afterglow bits to be set from hotpixel-amended bad pixel file.

acis_run_hotpix

- **New Tool!**

A script which runs the three hot pixel tools (`acis_build_badpix`, `acis_find_hotpix`, and `acis_classify_hotpix`) in the correct order.

For more information on using this tool, see the Identify ACIS Hot Pixels and Cosmic Ray Afterglows thread.

acisreadcorr

- Bug fix: default value of `bgroup` parameter is changed to 10

aconvolve

- enabled support for 1D images
- FFT speed improvement
- Bug fix: errors instead of crashing when `kernel_spec` parameter is left blank
- Bug fix: added error checking to `kernel_spec` for extra/missing parentheses or commas

arestore

- changed order of parameters make consistent with other tools

asphist

- Bug fix: the aspect histogram can now be made for nominal aimpoints which do not fall on a detector element at all during the observation. This was done intentionally for grating observation of Sco X-1 to prevent detector damage by an intense zero order image, while allowing two of the 4 HETG spectra to fall partially on the ACIS-S array.
- Bug fix: nominal aim falling into a chip gap caused `asphist` to fail
- The `asphist` computation time has been reduced by about a factor of two by removing an unnecessary second pass through the data.

The speed improvement updates required replacing the `max_bin` parameter with two new parameters: `delta_xy` and `delta_roll`.

- Bug fix: corrected typo in error message ("DTF" in place of "LTF")

axbary

- increased keyword precision in the output file

celldetect

- output source list has improved metadata (TLMIN/MAX)
- Bug fix: added "double" checking: will still output duplicate srcs, but the `double` flag/column will be set for one of them to make removing the duplicates easier

csmooth

- small speed improvement in FFT
- Bug fix: bad redirect in the `sclmax` parameter (`)sclmax` changed to `)sclmin`)

destreak

- The `mask` parameter which used to be for a region to ignore during the processing has been generalized so more than just spatial filters can be applied. The upshot being that you can now `destreak` a L1 event file with `mask=" [status=0,grade=0,2,3,4,6] "` and it will use only those values when processing the data.
- implemented a maximal-row-loss algorithm; new parameter: `max_rowloss_fraction` controls the max number of rows per frame to be streaking.

dmcoords

- `dmcoords` does not automatically take into account drifts in the Chandra geometry when calculating sky coordinates. These drifts were negligible near the beginning of the mission, but have grown to the order of 10 arcseconds and so cannot be neglected. The [Using dmcoords to Convert between CHIP and SKY Coordinates](#) thread shows how to account for this change.

A new parameter (`asolfile`) was added to use the aspect solution files to account for the geometry drifts in the calculations.

- Bug fix: the `"set nominal"` command now respects the `celfmt` parameter to use either HMS or DEG format

- Bug fix: made internal help consistent with interactive mode setting for SIM/DISPLACE

dmcopy

- Bug fix: improvements for handling image cubes (fix in the Data Model library)
- Bug fix: set pixels outside of region filter to NULL for >2D images (fix in the Data Model library)

dmdiff

- The tool now checks:

- ◆ images (1D and 2D)
- ◆ WCS
- ◆ DSS
- ◆ Units
- ◆ Comments
- ◆ Datatypes
- ◆ Missing keys
- ◆ Ranges

Many new parameters have been added to support the extended comparisons. See the `dmdiff` `ahelp` file for a complete list.

- added better checking for Inf and NaN in image data
- Bug fix: ignoring subcomponent tolerance ranges

dmextract

- Bug fix: in `ltc1` or `ltc2` mode, the value of `TIMEDEL` is set to the time resolution of the lightcurve. Similarly `TIMEPIXR` is set to 0.5 since the value in the `TIME` column is the mid-bin.
- Bug fix: read the `TLMIN`, `TLMAX`, and `TDBIN` keywords when a binning specification is not given
- added new way of handling binning on `EXPNO`
- fixed problem with uncertainties when user supplies variance image
- name of the vector column from the input image is kept (was hard-coded to be `POS`)

dmfilth

- added `NULL`, `NaN`, and `DSS` image checking
- change to `randseed` parameter limits to allow value of 0

dmgroup

- Bug fix: allow `BIN` to work on ascending or descending data (e.g. to bin wavelengths that decrease)
- Bug fix: `BIN_WIDTH` always working on row/channel instead of on `xcolumn` data
- Bug fix: problem with `bin=: <value>`
- Bug fix: problem with `BIN_FILE` when non-integer `xcolumn` is used
- Bug fix: last element now marked as `quality=2` when `SNR` grouping (fix to the group library)
- Bug fix: typo in the error message for `groupype=NUM_BINS`

dmgroupreg

- Bug fix: tool name is included in error messages (uses `errlib`)

dmgti

- Bug fix in smoothing: last $n/2$ rows were off-by-one where n is the smoothing factor/size
- allow `.[0-9]+` to be parsed as a number without requiring lead 0
- Bug fix: casting when converting numeric values from one format to another with `(int)`
- Bug fix: `<array>**<scalar>` giving array all equal to first element/pixel
- Bug fix: not using correct array index in expressions (e.g. `sum=(phas[2]+phas[3])`). Note that setting individual elements in an array still does not work, e.g. `(phas[2]=32)`

dmhedit

- Bug fix: empty lines in the file list cause problems

dmhistory

- **New functionality!**

A new `action` parameter has been added to determine what `dmhistory` should do with the information it obtains. The default is `get` as before (i.e. send it to the `outfile`); new options are `pset` (set the parameter file for the requested tool) and `put` (the parameter file for the given tool is written to the data file).

- Bug fix: tool keeps trailing whitespace when necessary

dmimg2jpg

- Bug fix: modified region labels (`showlabel` parameter). corrected problems where lines were too long and source numbers would appear in the wrong place

dmimgcalc

- allow `.[0-9]+` to be parsed as a number without requiring lead 0
- Bug fix: casting when converting numeric values from one format to another with `(int)`
- Bug fix: `<array>**<scalar>` giving array all equal to first element/pixel
- Bug fix: tool will error is `imgout` expression does not contain an `"imgout="` variable
- Bug fix: if the input file is missing, `dmimgcalc` dies with a segmentation fault
- Bug fix: table in second extension opened instead of image
- Bug fix: typo in error message ("Could [not] correctly parse expression")

dmimghist

- output includes `bin_low` and `bin_hi` columns to output denotes the lower end and (non-inclusive) upper limit, respectively, of pixel values in a bin
- added NULL, NaN, and DSS image checking

dmimgthresh

- added NULL, NaN, and DSS image checking

dmlist

- Bug fix: listing bits with array option not working (fix in the Data Model library)

dmmakereg

- Bug fix: there is better error reporting when the tool cannot parse a region
- Bug fix: ASCII output when region contains Points

dmregrid

- **New functionality!**

It is possible to specify the bin grid in either image or physical coordinates. A new parameter (`coords_sys`) controls whether bin is interpreted as logical (the default, old behavior) or physical.

Images covering different physical pixel ranges and at different resolutions (binning) can now be combined without applying the `x/yoffset` parameters.

- allow for non-integer bin and min/max x/y
- Bug fix: clobber would not clobber outfile that had extension supplied

dmsort

- Bug fix: corrected behavior when filtering columns and `copyall=yes`

dmstat

- Note that `dmstat` only works on images if they are 2-dimensional.
- improved handling of min/max location values

dmtcalc

- Bug fix in smoothing: last $n/2$ rows were off-by-one where n is the smoothing factor/size
- allow `.[0-9]+` to be parsed as a number without requiring lead 0
- Bug fix: casting when converting numeric values from one format to another with `(int)`
- Bug fix: `<array>**<scalar>` giving array all equal to first element/pixel
- Bug fix: not using correct array index in expressions (e.g. `sum=(phas[2]+phas[3])`). Note that setting individual elements in an array still does not work, e.g. `(phas[2]=32)`
- Bug fix: the tool SEGVs when setting bits in a file (Linux only)
- Bug fix: will return non-zero exit status if unable to evaluate expression

get_src_region

- restored to CIAO (had been removed at the CIAO 3.0.0 release). `get_src_region` determines a background threshold value from an image or event file and outputs regions with counts higher than that threshold.
- now able to process non-integer (floating point) images
- new parameter (`binning`) to control image histogram binning specification
- added NULL, NaN, and DSS image checking

mkacisrmf

- **New Tool!**

The tool `mkacisrmf` is a new addition in CIAO 3.2 and represents an entirely new method for creating ACIS response matrices. Unlike its predecessor `mkrmf`, this new tool separates the RMF calculation process into two components: an "ideal" component which describes the CCD spectral

CIAO 3.2 Release Notes – CIAO 3.4

response prior to the effects of CTI, and a spatially varying component which incorporates the changes in the response produced by CTI.

The algorithm contained in `mkacisrmf` is based upon the algorithm developed by Alexey Vikhlinin and implemented in his `calcrmf2` tool. As compared to the original tool (`mkrmf`) it includes a number of enhancements, including an improved interpolation scheme to calculate the response at intermediate energies between the available calibration points. All functionality of `mkrmf` is available including the ability to produce weighted response matrices for arbitrary spatial regions.

For more information on this tool, see the [Creating ACIS RMFs in CIAO 3.2](#) why topic.

mkrmf

- Bug fix: tool crashed if input FEF filename contained an error

mkwarf

- Bug fix: added support for 0th order gratings
- Bug fix: entire response region is bad or outside of the mask

psf_project_ray

- new parameter, `detsubsysmod`, (similar to the `mkwarf detsubsysmod` parameter) used to set ARDLIB BPMASK qualifier; was previously hard-coded as `BPMASK=0`
- the tool will now work with multi-OBI event files (`RA_PNT` vs `RA_NOM` keywords when available and different)
- Bug fix: passes correct detector name to ARDLIB for HRC data

reproject_events

- Bug fix: if a GTI does not exist for the first event in the file, the remainder of the file is skipped
- Bug fix: `rand_seed` not getting initialized correctly, which could cause problems with the background files

skyfov

- Bug fix: for problems seen with near-zero roll angles. More robust convex hull algorithm uses 8 points at each corner at min/max ra/dec/roll to determine polygon, and the polygon is not limited to 4 corners.
- Bug fix: file is empty or ObsID does not match

stk_build

- upper limit of verbose range in parameter file is 5

stk_count

- new parameter: `echo` determines if the retrieved value is printed to the screen in addition to being stored in the parameter file
- upper limit of verbose range in parameter file is 5

stk_read_num

- new parameter: `echo` determines if the retrieved value is printed to the screen in addition to being stored in the parameter file
- upper limit of verbose range in parameter file is 5

tg_create_mask

- Mask geometric parameters have been changed to allow floating-point regions. This is to support better extractions in crowded fields when fractional pixels are important for the zero order radius or diffraction arm widths.
- Bug fix: handling of `width` and `radius` parameter; values in the output R column are expected to be different by up to half a pixel. Changed datatype of those parameters from integer to read/float.

tg_resolve_events

- The spatial filter mask now supports polygonal regions instead of just boxes. This has no impact on pipeline processing, which will still use default regions from `tg_create_mask`. The enhancement is to support development of better background region definition and HEG-MEG extraction regions which can be customized near zero order to avoid overlap to shorter wavelengths.
- The coordinate transformation errors have been fixed for cases when the zero order instantaneous (or mean) position is off the detector array. Since grating event coordinates have their origin at the zero order, this deficiency prevented proper assignment of wavelengths and orders. (There is only one observation to date for which this was intentionally the case: Sco X-1, offpointed in order to avoid the risk of detector damage.)

tgdetect

- Bug fix: naming of temporary files works correctly
- new parameters for temporary and diagnostic files: `temproot`, `keeptemp`, and `keepexit`
- added "double" checking: will still output duplicate srcs, but the `double` flag/column will be set for one of them to make removing the duplicates easier (change made in `celldetect`)

tgextract

- Source spectral binning region max and min in the cross-dispersion direction were adjusted (to a half-width of 0.002 degrees) to compensate for an anomalous residual rotation of the LETG spectrum when imaged onto the ACIS-S array. The expansion of the region will prevent excessive clipping of the spectrum.

The cause of the residual rotation is under investigation. When resolved, we will revert to the previous default half-width of 0.00133 degrees.

The LETG/ACIS-S rotation compensation technical note (http://space.mit.edu/CXC/docs/technote_2004.06.11_la_rotation.ps.gz) describes the problem in more detail.

vtpdetect

- Bug fix: fixed problem with WCS not getting attached to column
- Bug fix: changed ASCII region output to be compatible with `dmcopyp` & `ds9`
- added `TLMIN/TLMAX` keywords to output POS column
- Bug fix: added warning when recursive routine goes into an infinite loop with event list input

wavdetect

- some speed improvement in FFT convolutions
- Bug fix: will exit with non-zero status if the tool fails

wrecon

- added TLMIN/TLMAX keywords to output POS column
-

Parameter Files

Since some of the parameter files have new entries or default values, it is recommended that users delete all old parameter files before running CIAO 3.2. The FAQ on cleaning up parameter files has more information.

acis_process_events

- unneeded `cc_chipy` parameter removed

acisreadcorr

- Bug fix: default value of `bgroup` parameter is changed to 10

ardlib

- To support the spatial contamination model, the `ardlib.par` `AXAF_ACIS_CONTAM_FILE` parameter has been replaced by a set of 10 parameters (one per CCD): `AXAF_ACIS0_CONTAM_FILE` to `AXAF_ACIS9_CONTAM_FILE`. The default value of these parameters is `CALDB`.

arestore

- changed order of parameters make consistent with other tools

asphist

- The speed improvement updates required replacing the `max_bin` parameter with two new parameters: `delta_xy` and `delta_roll`.

csmooth

- Bug fix: bad redirect in the `sclmax` parameter (`)sclmax` changed to `)sclmin`)

destreak

- implemented a maximal-row-loss algorithm; new parameter: `max_rowloss_fraction` controls the max number of rows per frame to be streaking.

dmcoords

- A new parameter (`asolfile`) was added to use the aspect solution files to account for the geometry drifts in the calculations.

dmdiff

- Many new parameters have been added to support the extended comparisons. See the `dmdiff` `ahelp` file for a complete list.

dmfilth

- change to `randseed` parameter limits to allow value of 0

dmhistory

- A new `action` parameter has been added to determine what `dmhistory` should do with the information it obtains. The default is `get` as before (i.e. send it to the `outfile`); new options are `pset` (set the parameter file for the requested tool) and `put` (the parameter file for the given tool is written to the data file).

dmregrid

- It is possible to specify the bin grid in either image or physical coordinates. A new parameter (`coords_sys`) controls whether bin is interpreted as logical (the default, old behavior) or physical.

get_src_region

- new parameter (`binning`) to control image histogram binning specification

psf_project_ray

- new parameter, `detsubsysmod`, (similar to the `mkwarf` `detsubsysmod` parameter) used to set ARDLIB BPMASK qualifier; was previously hard-coded as `BPMASK=0`

stk_build

- upper limit of verbose range in parameter file is 5

stk_count

- new parameter: `echo` determines if the retrieved value is printed to the screen in addition to being stored in the parameter file
- upper limit of verbose range in parameter file is 5

stk_read_num

- new parameter: `echo` determines if the retrieved value is printed to the screen in addition to being stored in the parameter file
- upper limit of verbose range in parameter file is 5

tg_create_mask

- Bug fix: handling of `width` and `radius` parameter; values in the output R column are expected to be different by up to half a pixel. Changed datatype of those parameters from integer to read/float.

tgdetect

- new parameters for temporary and diagnostic files: `temproot`, `keeptemp`, and `keepexit`

Data Model

Ability to perform weighted binning

- In the DM syntax, we have always been able to make a 2D image by picking two columns from a table (e.g. an event list):

```
unix% dmcoppy "evt.fits[ccd_id=3][bin time=1000:2000:0.4,energy=300:1000:2]" time_energy.img
```

This defines a 2500 x 350 pixel image where the x-axis is time in 0.4 second pixels and the y-axis is energy in 2 eV pixels. For every row in the table which passes the filter (e.g. `ccd_id=3`), we add 1 to the relevant pixel in the image.

In the new "weighted binning", you can specify a "weighting column" from the table by adding its name to the end of the binning spec, preceded by a semicolon:

```
unix% dmcoppy "evt.fits[ccd_id=3][bin x=4000:5000:2,y=3000:4000:2;energy]" subset.img
```

This means that instead of adding 1 to a pixel for each row that passes, we add the value of the energy column in that row. (Note that the energy column has a floating-point data type, and the output image is automatically forced to that type).

This will be particularly useful with tables which are already histograms or tabulated functions: suppose you have a table with columns `WAVELENGTH` and `FLUX`; this will let you make a 1D FITS image from the table at any desired resolution, for use with spectral programs that work on images rather than tables. Or, suppose you have a table which gives a grid of parameter values `NH`, `ALPHA` and goodness-of-fit values `CHI2`:

```
unix% dmcoppy "grid.table[bin nh=19.0:22.0:0.1,alpha=-2.0:2.0:0.05;chi2]" grid.image
```

Improved Keyword Support

- Support for FITS WCS Rotation keywords (`CROTA2`, `PC`, `CD`) and unofficial IRAF keywords (`LTM`, `LTV`)

All X-ray sky images have north at top; since all X-ray satellites perform aspect reconstruction from event data, we have the freedom to reorient the pixels. In contrast, optical astronomical images retain their original pixelization and use rotation information in the WCS header keywords to specify the alignment between north and the pixel x-axis. We didn't handle this correctly in earlier releases of CIAO; now we correctly propagate the rotation metadata.

Users should note, however, that region files using the ellipse and rotbox shapes still define the rotation angle in those shapes as being with respect to pixel x, not celestial east, and so are not portable between different datasets.

We have also added support to recognize IRAF's nonstandard `LTM` and `LTV` coordinate keywords to define physical coordinate systems.

Virtual Column Filtering/Binning for Tables

- It is now possible to filter on a coordinate as well as a column. For example:

```
unix% dmcoppy "evt.fits[phi=0.00:0.02]" onaxis.fits
```

selects only photons within 0.02 deg of the axis. Using "dmlist evt.fits cols" you can see that phi is not one of the columns in the event list, it's just a coordinate, which is defined in the header by a coordinate transformation on the detx and dety columns. The filtering applies the transform "on-the-fly." (In CIAO 3.1 it was possible to do "dmlist "evt.fits[cols phi]" data", but you couldn't filter on a coordinate.)

Warning 1: you can't omitting the lower bound, e.g. [phi=:0.02], because the lower/upper bound range of coordinates aren't currently stored in the file header.

Warning 2: filtering on 2D regions (e.g. of RA and Dec) is problematic for reasons to do with the spherical nature of these coordinates. Our regions (circle, polygon, etc.) are always defined as shapes on the tangent plane; you should continue to use

```
unix% dmcoppy "evt.fits[sky=circle(14:13:11.2,-00:02:23.5')]" ...
```

rather than [eqpos=circle...] or [(ra,dec)=circle...]. 1D filters in degrees, such as "ra=270.0:275.0", are fine.

Other Enhancements

- Allow users to type-cast images using [option type=i4] syntax:

```
unix% dmcoppy "integer_image.fits[opt type=r4]" real_image.fits
```

- Some improvements for handling image cubes, e.g.

```
unix% dmcoppy "cube.fits[#2=12][bin #1,#3]"
```

makes a 2D image from plane 12 of the second axis of a 3D image.

Bug Fixes

- **The Data Model bug page lists all currently known bugs, as well as those fixed in CIAO 3.2.**
- dmlist will display logical values as T/F even when opt=raw (instead of 1/0)
- correctly handle files with only NULL and GTI blocks
- print bit columns properly in array mode (fix affects dmlist)
- print all elements of a string array when opt=array (fix affects dmlist)
- allow numbered image axes (e.g. #1,#2) grouping using bin or filter syntax
- proper support for variable-length arrays: if you dmcoppy an RMF file, it will retain the correct FITS variable-length format
- fixed bug in column selection on second component of a coordinate vector, e.g. evt.fits[cols y]
- set pixels outside of region filter to NULL for >2D images
- keep the LONP/LAT coordinate system on DETX/DETY (THETA,PHI) when copying
- allow image enlargement using bin filter; restrict range/region filters to image field
- catch bad filter syntax, such as "a:b:c" when not binning
- error out when given a dataset with no blocks. This was reported as a tool crashing when given a directory instead of a filename, which it interpreted as a dataset with no blocks.
- propagate subspace region blocks like GTI blocks: if an event file includes keywords saying that another block (HDU) in the file is a region describing the area covered by the event file, that block will get copied along with the event block

Sherpa

Convolution with PSF Models

- Improvements to the file-based PSF models (`fpsf1d`, `fpsf2d`) and the point-source models (`ptsrc1d`, `ptsrc2d`) have been made.
- The `ptsrc1d` and `ptsrc2d` have been updated to accommodate both FITS and ASCII files.
- A bug in file-based PSF convolution (`fpsf` models) has been corrected.

Fitting Methods

- The Levenberg–Marquardt fitting method has improved handling of problem datasets.

Error Estimation

- Improvements have been made to the `projection` method to handle the case of overflow and underflow errors:
 - ◆ the `projection` method now catches overflows and underflows, and calculates the errors for the remaining parameters rather than immediately exiting, and
 - ◆ the allowable range on many model parameters has been restricted to reduce the chances of overflow occurring.
- The error-estimation commands can now be run without first performing a fit. This allows users to load in a previous fitting session (saved with the "SAVE" command) and continue their analysis. This only holds for the Sherpa commands (e.g. `PROJECTION`, `INTERVAL-UNCERTAINTY`) and not the S-Lang versions (e.g. `run_proj()`, `run_intunc()`).

S-Lang Integration

- In a number of cases, cleaned up extraneous output from S-Lang functions. These include functions such as `get_fit()`, `run_proj()`, and `sherpa_eval()`.
- Sherpa can now be used with the S-Lang `require()` function; this change is in conjunction with the S-Lang module changes for `provide` and `require`.

Miscellaneous Changes

- In the `jdpileup` model, some default parameter values and limits have been changed:
 - ◆ value of `alpha` changed from 0.95 to 0.5
 - ◆ `f` is now thawed
 - ◆ the upper limit for the `n` parameter is 100 (was 2048)
 - ◆ the lower limit for the `g0`, `ftime`, and `fracexp` parameters is $1e-120$ (was 0)
 - ◆ value of `fracexp` changed from 1 to 0.987
- The CIAO 3.2 version of Sherpa links in the XSPEC v11.3.1 model library, and includes interfaces to the new models in this library. The XSpec model function help file lists the models which are not available in CIAO 3.2.
- The CIAO 3.2 version of Sherpa makes use of version 0.4.0 of the S-Lang XPA module, in its CIAO 3 compatibility mode. This means that the `ds9` package can not be used with Sherpa.
- Sherpa no longer overrides the ChIPS `redraw` setting. This means that ChIPS commands will be executed immediately, so users no longer need to issue the `REDRAW` command.

Bug Fixes

- *The Sherpa bug pages list all currently known bugs, as well as those fixed in CIAO 3.2.*
 - `get_fit` always returned goodness-of-fit information for all datasets
 - `sherpa_eval()` does not return `-1` on a failed `use` command
 - due to a round-off/casting problem, creating an ARF in double precision causes the source model to contain 0's when it shouldn't
 - if a parameter value is at the maximum boundary, running `projection` on it may result in a core dump (Linux only)
-

Libraries

ardlib

- To support the spatial contamination model, the `ardlib.par` `AXAF_ACIS_CONTAM_FILE` parameter has been replaced by a set of 10 parameters (one per CCD): `AXAF_ACIS0_CONTAM_FILE` to `AXAF_ACIS9_CONTAM_FILE`. The default value of these parameters is `CALDB`.

cxcpam

- use `strncpy` to make sure string buffer doesn't overflow
- parser did not set error condition when parameter file had too many field or field was not parseable
- crash when bad parameter file lived in the system part of the `PFILES` search path, so it could not be copied to the user path

dslib

- changed length of `ds_autoname` string routines
- corrected "kernal" -> "kernel" spelling
- tweaked autonaming to avoid `_extname` when filename begins with an underscore
- event file names containing "qp" that are not qpoe files caused crash
- fixed autonaming so that empty strings do not cause array bound reads
- added range to `WCS` struct to allow detect tools to include `TLMINTLMAX` on output columns

errlib

- Updated version to reflect CIAO 3.2.

fl_acisio

- allow afterglow bits to be set from `hotpixel-ammended` `badpixel` file

grplib

- allow `BIN` to work on ascending or descending data (e.g. to bin wavelengths that decrease)
- `BIN_WIDTH` always working on row/channel instead of on `xcolumn` data
- problem with `bin=: :<value>`
- problem with `BIN_FILE` when non-integer `xcolumn` is used
- last element now marked as `quality=2` when SNR grouping (fix to the group library)
- use `groupypeval` when `binspec` is unspecified
- typo in the error message for `groupype=NUM_BINS`

hdrlib

- Updated version to reflect CIAO 3.2.

mtl_libs

- SEGVs when setting bits in a file (Linux only)
- allow `.[0-9]+` to be parsed as a number without requiring lead 0
- casting when converting numeric values from one format to another with `(int)`
- `<array>**<scalar>` giving array all equal to first element/pixel
- not using correct array index in expressions (e.g. `sum=(phas[2]+phas[3])`). Note that setting individual elements in an array still does not work, e.g. `(phas[2]=32)`
- minor speed improvements
- table in second extension opened instead of image

regionlib

- ASCII output when region contains Points
- problem writing ASCII files with rectangles and circles

rmflib

- crash if input FEF filename contained an error

stklib

- changed formatting internally for `pgrid`, `rgrid`, and `lgrid` strings
- added `igrid` to force integer type grid
- problem with `lgrid` giving 1 more than necessary
- calling `stk_delete_current` when `current==0` caused crash
- `stk_build` on an `@` file that doesn't exist will now return a NULL stack pointer
- `stk_change_current` exhibited memory problems

tcdio

- enabled support for 1D images
- added warning when image has 0 axes

tcdlib

- minor code cleanup that gives some noticeable speed improvements in FFT convolutions

S-Lang Modules

Changes for `provide` and `require`

- To keep up to date with S-lang enhancements, CIAO modules were upgraded so that they could be loaded with `"provide"` and `"require"`; this also affected the `varmm` and `chips` modules. The changes were made to be backwards-compatible such that loading a module with `"require"` is the same as loading it with `"import"`. Other changes were made to `varmm` so that `varmm`-based software automatically set proper slang load paths and loaded the `slsh` resource file (`slsh.rc`). The `slsh` resource file will not be loaded by `varmm` explicitly if the the file is not located in the

`$ASCDS_INSTALL/etc` directory.

Module Conflicts

- Bug fix: checks error lib before trying to initialize. Reported as "If functions are called in a certain order, a benign warning is printed to the screen" (fix to caldb module)

caldb

- checks error lib before trying to initialize.
- fixed array over-run
- changes for provide/require; back-compatible with `import`

group

- fixed array over-run
- problem where `grpBinFile()` caused crash (seen on os-x)
- changes for provide/require; back-compatible with `import`

paramio

- `paramopen` take NULL for the file name and it will get it from the argv list. This enables "@@" opening and removes the need to have taskname in both argv array and individual input.
- `paccess` now has two modes of operation. The new mode is enabled when the first argument is a `Param_File_Type` variable, in which case the function returns 1 if the second argument is the name of a variable defined in the parameter file, or 0 otherwise.
- fixed array over-run
- changes for provide/require; back-compatible with `import`

pixlib

- added optional "dtheta" argument to `pix_set_simoffset()` routine
- added a new routine, `pix_fpc_to_msc()` which is a wrapper around `fpc_to_mnc()` and `mnc_to_msc()` to get off-axis angles
- fixed array over-run
- changes for provide/require; back-compatible with `import`

region

- new `regExtent` function: calculates the bounding box of a region
- new `regRegionString` function: complements `regPrintRegion` routine
- fixed array over-run
- changes for provide/require; back-compatible with `import`

stack

- return NULL when @ file is not valid
 - `stk_read_next` and `stk_read_num` return NULL at the end of a stack
 - corrected usage message of `stk_expand_n`
 - fixed array over-run
 - changes for provide/require; back-compatible with `import`
-

Analysis Scripts

The CIAO contributed scripts package is available from the CIAO Scripts page and is considered a required part of the installation. The script installation instructions explain where the scripts should be unpacked within the CIAO directory tree.

New Additions

- Four scripts have been moved from the CIAO distribution to the scripts package:

- ◆ `psextract`
- ◆ `acis_fef_lookup`
- ◆ `add_grating_orders`
- ◆ `add_grating_spectra`

Users that follow the script installation instructions will not notice any difference in using these scripts between CIAO 3.1 and 3.2.

Changes for `provide` and `require`

- A number of scripts were updated in conjunction with the "provide" and "require" S-lang enhancement; see the S-Lang Modules section of these release notes for details. Again, this change will be transparent to users.

`get_sky_limits`

- use `require` for loading modules instead of `import`

`paramest.sl`

- updated for fixes to `run_regproj/run_regunc()` and `run_intproj/run_intunc()`; removes an error message seen in CIAO 3.1
- the script can be loaded via `require("paramest");`
- use `require` for loading modules instead of `import`

`setplot.sl`

- use `require` for loading modules instead of `import`

`sherpa_plotfns.sl`

- added a `provide("sherpa_plotfns");` line
- use `require` for loading modules instead of `import`
- Use `get_fit()` rather than `get_goodness()`

`sherpa_utils.sl`

- the script can be loaded via `require("sherpa_utils");`
- use `require` for loading modules instead of `import`

`tg_osort_img`

- new path to ISIS: `$ASCDS_INSTALL/contrib/share/`
- removed `'() = mk_osum_img...;'` since there is no return value from `mk_osum_img()`

Documentation

New Threads

- Creating ACIS RMFs with mkacisrmf
- Identify ACIS Hot Pixels and Cosmic Ray Afterglows

Updated Threads

- Extract ACIS Spectra for Pointlike Sources and Make RMFs and ARFs
- Step-by-Step Guide to Creating ACIS Spectra for Pointlike Sources
- Extracting Extended Source Spectra and Responses
- Coadding Spectra and Weighted Responses

Why Topics

- *New:* Creating ACIS RMFs in CIAO 3.2
- *Updated:* ACIS QE Degradation

CALDB Website

- The Chandra Calibration Database website has been updated for the CALDB 3.0.0 release.
-
-

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
http://cxc.harvard.edu/ciao3.4/releasenotes/ciao_3.2_release.html
Last modified: 11 August 2006

