Chandra X-ray Center

CIAO 3.0 Release Notes

Return to: Version History

- Tools
- Parameter Files
- Sherpa
- User Infrastructure
- Datamodel
- Libraries
- S–Lang Modules
- Documentation

Tools

Multiple Tools

- Tools now store record of all input–parameter values in history section of FITS headers
- Changes to when tools copy history info from input to output header: Tools that create a new kind of output (eg detect tools) will no longer carry history from the input. However, tools like destreak will carry full history.

acis_detect_afterglow

- When run on a file it has previously processed, now resets status bits allocated to it to 0 before reassigning. Eliminates problem where running again with more restrictive conditions failed to reduce number of afterglow events.

acis_fef_lookup

- New version of script converted to slsh making use of the caldb and paramio S–Lang wrappers

acis_process_events

- Added an exhaustive list of conditions for cc–mode time–of–arrival computations.
- Fix for full header copy to get correct CONTENT and HDU* keywords
- Fixed problems w/ CTI_CORR keyword in GRADED mode (keyword not getting written).
- Now prints a warning when the CHIPY_TARG goes out of range
- Now prints a warning when stop=none|chip and calculating cc–mode TOA
- Now uses CHIPY_TARG (instead of trow+1) for gain computations in CC mode.

- If obsfile=<filename>, tool now warns if file is not found (value of "none" or "NONE" specifies no obs.par file)
- Now applies CTI correction by default (apply_cti=yes)
- Updates for handling the various CTI conditions in acis_process_events. Problem was that in some cases, a_p_e did not select the CTI–corrected gain file.
- Fixes to allow CTI corrections to be undone and redone

acisreadcorr

- Fixed crash when aspect file is missing/bad
- Fixed mistake in parameter file indicating that random=−1 disabled randomization

aconvolve

- Added normkernel parameter. Can normalize the kernel to unit 'area' (default), 'max' pixel value renormalize to 1, or 'none' (kernel left unnormalized).
- Changed the way that tables are read. Users should now supply two columns, eg

```
aconvolve infile="my_ltc.fits[cols time,rate]"
```

  The x–axis (time in this example) needs to be evenly binned; an error will be generated if one tries to use unevenly separated grids. The output will have two columns; aconvolve and acrosscorr is just a copy of the input data, apowerspectrum converts to frequency (or equivalent) domain.
- Fixed table output when 'pad=yes'. (Affected i/o used by csmooth, dmcontour, dmimg2jpg, and arestore also)
- Fix for unint string (EDGES when using FFT)
- Fixed bug freeing same memory twice in FFT mode; only at very end would users see a problem (output file would be fine).

acrosscorr

- Changed the way that tables are read. Users should now supply two columns, eg

```
acrosscorr infile1="my_ltc.fits[cols time,rate]"
```

  The x–axis (time in this example) needs to be evenly binned; an error will be generated if one tries to use unevenly separated grids. The output will have two columns; aconvolve and acrosscorr is just a copy of the input data, apowerspectrum converts to frequency (or equivalent) domain.
- Fixed table output when 'pad=yes'. (Affected i/o used by csmooth, dmcontour, dmimg2jpg, and arestore also)

add_grating_orders

- Fix for dmgroup 3.0 version.
- Fix to a bug in the "domsg()" routine that sets the output message level according to the input
- Removed script–version message
- Included grouping option ADAPTIVE_SNR

add_grating_spectra

- Fix for problem deleting temp files
- Fix for dmgroup 3.0 version.
- Fix to a bug in the "domsg()" routine that sets the output message level according to the input
- Removed script–version message
- Included grouping option ADAPTIVE_SNR
- Fix to use 'head −1' to work around problem of getting multiple lines for certain keyword values

apowerspectrum

- Changed the way that tables are read. Users should now supply two columns, eg

  ```
  apowerspectrum infilereal="my_ltc.fits[cols time,rate]"
  ```

  The x–axis (time in this example) needs to be evenly binned; an error will be generated if one tries to use unevenly separated grids. The output will have two columns; aconvolve and acrosscorr is just a copy of the input data, apowerspectrum converts to frequency (or equivalent) domain.
- Fixed table output when 'pad=yes'. (Affected i/o used by csmooth, dmcontour, dmimg2jpg, and arestore also)

arestore

- New tool: Performs Richardson–Lucy deconvolution of an input image

asp_apply_sim

- Removed from CIAO; use asphist instead

asp_calc_offsets

- Removed from CIAO; use asphist instead

asphist

- Changed units for columns X/Y_OFFSET in output file from "pix" to "pixel".
- Fix for problems when first file has 0 rows
- dtffile can now be set to NONE for HRC and the DTCOR value in the event file will be used.

celldetect

- Added units "pixel" to R column.
- Added "ELLSIGMA" keyword indicating the user parameter used to scale the "R" column
- Removed dependence on ascfitlib by removing model evaluation of old LSF params. Old psfsize files named "psfsize.fits" and "psfsize_19991012.fits" will no longer work with celldetect.

csmooth

- Fix to allow background image to be input
- Quieted some of the misleading message about stopping iteration/etc
- Removed warning "edge treatment option will be ignored" since it is always true (since the default convolution method is FFT)

destreak

- Changed default .par value to work only on ACIS−8
- Fixed problem with not respecting datamodel filter on infile

dmappend

- Quieted some extra verbose output (not controlled by verbose)

dmarfadd

- Fixed problem with the DETNAM becoming mangled when adding non−grating ARFs (possibly causing crash)

dmcontour

- Fixed crash when specifying levels as a::b
- Fix for images with NaN values (floating point images only); tool will no longer crash

dmcoords

- Fix for problems when infile does not contain WCS and opt=sky|cel (eg infile=pcad*asol1.fits)
- Fix to set correct exit status on error

dmdiff

- New tool: Finds differences between two FITS files (header and data); see 'ahelp dmdiff' for more info

dmextract

- Fixed problem with trying to extract PHA spectra with a background stack.
- Added ability to filter wmap independently of data
- Fixed memory corruption problem on Linux.
- Fix for PHA2 files not reporting net_counts
- Added ability to weight the counts when extracting data from a table, eg

```
dmextract "foo_evt.fits[bin time=::10;dtf]"
```

(The ';dtf' part is new). Instead of counts +=1, it stores and reports counts += dtf (in this case).
- Fix for problem when region is entirely off the FOV. Will give a warning & set counts, area, and sbr to 0.
- 'dmextract' now provides support for lightcurves created from the TIME column of an event file. In versions prior to CIAO 3.0, if one binned a 10 ksec observation over 10 sec intervals, for example, the

resulting COUNT_RATE column would contain value of counts per 10 sec bin divided by 10 ksec, i.e., the full duration of the observation. By choosing 'opt=ltc1' or 'opt=ltc2', COUNT_RATE will contain the proper rate for the duration of the bin, and furthermore, the rate will include deadtime and good time interval corrections. This functionality essentially replaces one of the binning options found in the 'lightcurve' tool (i.e., binning by bin length in seconds), and users are recommended to use 'dmextract' with the ltc option in lieu of 'lightcurve'. Note that the lightcurve functionality only works for extractions based on the TIME column directly and explicitly, and does not work on other columns derived from the time column (e.g., TIME_RO, PHASE).

- The meanings of various output columns generated in spatial extraction have been simplified in CIAO 3.0.

  The definitions of the output "COUNTS" columns (e.g. COUNTS, NET_COUNTS, etc.) and "RATE" columns (COUNT_RATE, NET_RATE, etc.) when exposure maps are used have been changed. Now, the units of the "COUNTS" columns are always simply "counts", and the "RATE" columns units are always "counts/sec", regardless of whether an exposure map is used. The exposure time used in rate calculations is taken from the value of the EXPOSURE keyword in the header.

  When exposure maps are used, additional "FLUX" columns (e.g., FLUX, NET_FLUX, etc.) are added to the output table. These columns always have units of "counts/cm**2/sec". MEAN_SRC_EXP and MEAN_BG_EXP columns are also added to the output in these cases. These columns represent the count−weighted averages of the exposure map values in the source and background regions, in units of cm**2. If unnormalized exposure maps have been used, these values are converted to cm**2 by dividing by the exposure time from the EXPOSURE keyword in the header. A SUR_FLUX column is also added in these cases, with units of "counts/cm**2/sec/pixel".

  Normalized (units of cm**2), unnormalized (units of cm**2−sec), and dimensionless exposure maps are now supported. Exposure map units should be the same for both source and background exposure maps. If dimensionless exposure maps are used, they are treated as unnormalized and the program issues a warning. In such cases, users should proceed with caution in interpreting the meanings of the "FLUX" columns.
- Added background surface brightness and error columns when background is not NONE
- Fixed problem where filter's specified after the "[bin " part of the infile were not getting applied
- Modified area columns to have 'pixel' units

dmfilth

- Added error exit when no data is present in region
- Fixed bug where non−square images were being indexed incorrectly.
- Fix for method=POLY and all background data = NULL. Will now warn user.
- Fixes for bad indexing (non−square images) for background regions. Improper indexing caused a very small subset of the background region to be sampled from a place on the image outside the specified background region.
- Fix for checking for regions outside the image boundary regardless of method being used.
- Fix for low 'flux' POISSION (and GLOBAL) distributions. (Source regions were not getting filled in.)
- No longer crashes if region stack is not in current directory.
- When an event file is supplied it will warn instead of allowing it to be binned
- When method=GLOBAL, the bkglist is ignored (required but ignored)

dmgroup

- Entirely new tool based on a new group lib.
- Supports new grouping methods: ADAPTIVE_SNR, BIN_WIDTH, MIN_SLOPE, MAX_SLOPE, BIN_FILE (see 'ahelp dmgroup' for info)
- Supports type II files
- 'Tabs' have been split to now also include 'stops'; the former are marked as bad, the later as good.
- The syntax of tabs and stops has changed from center:width to the usual min:max
- Removed parameters 'column' and 'bincolumn'; added 'xcolumn' and 'ycolumn', which function differently than the old parameters; see 'ahelp dmgroup' for details and the Updating dmgroup Syntax for CIAO 3.0 thread.

dmgroupreg

- New tool to parse ds9 format regions (should be in physical coordinates) into CIAO syntax. Specifically, uses ds9 'tags' to correctly group shapes together and separates into source and background components. (See 'ahelp dmgroupreg' for more info.)

dmgti/mtl_build_gti

- New dmgti script. Should be more robust with various spaces input on the command line.
- Added a new operator, ~=, which will return true if the string on the right hand side is a substring of the string on the left hand side. It will generate a data dependency error if the two values are not strings.
- Can now resolve CALDB for its lookup table parameter.
- Fix problem with checking BIT values.
- Added CALDB lookup for limits lookup file
- Can set/check floating point values against NaN, eg foo=#nan or ((goo==#nan)?(bar=1):(bar=0)). Integer NULL values are not dealt with.
- The GTI file now has a 'time' column in the empty "FILTER" extension. This should allow users to add additional time filters (will get automatically intersected w/ all GTIs), eg

```
dmcopy "dmgti_output_gti.fits[time=a:b]" foo.gti
```

  'foo.gti' should contain the intersection of a:b with each of the GTIs.
- Fixed problem with unarbitrary text between quotes in strings. (Eg "FOO.bar" would fail).
- Single quotes can now be used as string identifiers.
- Fix for problems with alloc'ing and copying non−existent wcs's.

dmhedit

- Fix for problem where tool needed one existing keyword to work properly
- Fix to allow string keyword values to be set to "". Technically in FITS they are set to " ".
- Fix so that history is not added if keyword edit failed
- Fixed problem of HisTORY and COmmeNt case sensitivity
- Fixed exit value when output file is un−writeable
- Fix for empty string after '/' comment separator

dmhistory

- New tool: Now that all tools put full history into header in std format, its possible to extract that history. Will recreate the full command line from history records for specified command (or all commands) including stack resolution. (See 'ahelp dmhistory' for more info.) Note that tools that are actually shell scripts (add_grating_orders, add_grating_spectra, psextract, tgdetect, wavdetect) do *not* produce standard histories in their output files. Hence, dmhistory may be of limited use with the output of these tools.

dmimg2jpg

- Fixed problem with extraneous grid line when non−square images are supplied.
- Fix for extra 'grid' pixel in low−left corner when grid displayed

dmimgcalc

- Integrated MTL libraries. Arbitrary calculations are now possible, using same expressions/syntax as dmtcalc. (See 'ahelp dmimgcalc' for more info.)
- Made the lookupTab optional; that is if it's set to NONE or "" then the header merge is not done −− only the values in the first header are used. This doesn't apply to the WCSs checks; only to the header merge.
- Fixed exit value to return(0) upon successfully completion in "add|sub|mul|div|tst" mode.
- Fixes problem w/ copying WCS from rmfimg−style output file
- Can now output float/double/short/etc data types

dmimghist

- Fix to use binsize=1 when bin=::. (Previously crashed.)
- Fixed exit status

dmimgpick

- New tool: Maps image values to rows in a table. (See 'ahelp dmimgpick' for more info.)

dmimgthresh

- Fix to allow upper cut−off using cut=<min>:<max> syntax. Can mix−n−match value vs value% and ":" min/max are optional, eg cut=50 –> lower only, cut=50: same, cut=:50 –> max cutoff of 50, cut=20:90% min @20 (exactly) max at 90%. Same value is used for min/max threshold (if you want a different value, run tool twice)
- If value=INDEF, the output file will contain NaN's. NOTE: It will be NaN only if the data are floating point. If it is an integer image, the value will be whatever NaN is cast to.

dmjoin

- Fix for problem when there are no more rows in the join files
- Fixed problem where if one of the values was after the end of the range covered by the interpolation file, the extrapolation was done from the wrong end of the file.
- Fixed problem of interpolation being done one row off from the correct point.

- Added four new interpolation methods: closest, furthest, minimum, and maximum. Closest picks the value whose joincol value is closest to the joined row; furthest picks the one furthest away. Minimum and maximum will pick the min/max, respectively, of the closest two values.
- Fixed problem w/ interpolated values being slightly off due to casting problems.
- Correctly set exit status
- Fix for linear interpolation to use fabs() to set the weights correctly when data (not join column) are not increasing.
- Fixed segv is infile="

dmkeypar

- Modified so that it can also grab a value from a column in addition to a keyword –– thus replicating the 'ftabkey' functionality. Users can use "[#row xx]" filtering to select a given row.
- Removed minimum defaults (=0) for real and integer values in .par file.
- Added a new 'echo' parameter that prints the retrieved value to the screen. The default is no echo.

dmmakepar

- Added units of any columns to parameter description/prompt
- Fixed problem when comment or value contains a ", replaces with ' and issues as warning.
- Fixed inconsistent bool/string usage of header_filter param
- Fix for opening files w/ virtual file specifications applied
- Added logic to print max machine precision for floating point numbers; prevents numbers getting truncated

dmmakereg

- New parameter (wcsfile) to allow the user to input a reference image / eventlist. This allows a user to convert a region in World coordinates (eg WCS) to physical pixels which are necessary in CIAO analysis tasks.
- Fix when output file is unwriteable and in 'append' mode
- Fixes when appending to a file w/o write permission

dmmerge

- Input files are checked for consistency before merge begins. Previous was happening for each file as it was merged which could lead to (large) partially complete files lying around. In addition to checking the datatype of columns (already does this), it now also checks the array size. This should fix the problem with trying to merge a FAINT and VFAINT dataset.
- Allow for lookupTab="" in which case no header merging is done (first header is used)
- Hdr lib updates should have fixed the problem where CONTENT/ HDUNAME/EXTNAM was getting set to "" when outBlock was "".

dmpaste

- Now copies the range of pasted columns
- Fixed BIT/BOOL/BYTE datatypes not being written

dmreadpar

- Fixes when try to edit the header of a file w/o write permission

dmregrid

- Fix for faulty ASSERT being thrown when rotation point exactly matches rotated pixel "corner".
- Changes to deal with rotation angle of not–quite 0 causing problems in data when npoints=0 (exact algorithm).
- Fixed problem w/ not propagating units from (first) file.
- Fixed problem that multiple offsets,angles, and rot_centers needed to be specified in external (@) files

dmsort

- Added extra check against failure when kernel isn't installed.
- Previously, when one would use a ccd_id filter on an event list, the events would be filtered out, but the corresponding GTIs would remain. Now, if copyall=no then dmsort will correctly filter out the unwanted GTIs with a ccd_id filter. However, if copyall=yes, then dmsort assumes the user really wants to copy all blocks, including all GTIs.

dmstat

- Added additional precision
- Added row location of min/max
- Will only check values inside the region/DSS (if applied); no longer need to explicitly set NULL values; eg
  OLD:

  ```
  dmstat "foo_img.fits[sky=circle(100,100,5)][opt NULL=-999]"
  ```

  NEW:

  ```
  dmstat "foo_img.fits[sky=circle(100,100,5)]"
  ```

  The region is retrieved from the DSS; particularly useful when filtering an event list; then bin into an image later, eg

  ```
  dmcopy "foo_evts.fits[sky=rotbox(10,10,3,5,45)]" goo_evts.fits
  dmcopy "goo_evts.fits[bin sky=1]" goo_img.fits
  dmstat "goo_img.fits"
  ```

  the last step knows that "goo_img.fits" has the rotbox region applied and only computes stats in that region.
- For centroid=yes, ND images (N>2) are not supported; an error message is now generated in addition to a non–zero status. For centroid=no, ND images can be used.
- Fix problem when length of axes is 0
- Fix so that is nicely errors out on 1D image

dmtcalc

- Fixed the logic that caused the problem with modifying existing columns to break.
- Fix problem with checking BIT values.
- Fix for intersecting vs. merging subspace of files that could cause extra bogus subspaces (GTIs)
- Can set/check floating point values against NaN, eg foo=#nan or ((goo==#nan)?(bar=1):(bar=0)). Integer NULL values are not dealt with.
- Fixed problem with 10*phas, eg scalar*array didn't work but array*scalar did.
- Fixed problem w/ scalar*array vs array*scalar expressions.
- Fixed problem with unarbitrary text between quotes in strings. (Eg "FOO.bar" would fail).
- Single quotes can now be used as string identifiers.
- Problem parsing temp names that included component of vector column, eg .xfoo=x was internally getting set to .sky[0]foo=sky[0]
- Fixed problem w/ overwriting component of existing vector column, chipx=chipx+1.
- Fixed problem when tool would 'Abort' on some vector column expressions.
- Reworded warning when expression contains an undefined variable

dmtype2split

- Parameter 'rows' has been deprecated (will issue a warning). Use datamodel filter instead.
- Does better check of #requested rows and #outfiles
- If outfile contains a '#' character and the number of input files is not equal to the number of output files, it will replace the '#' with a number 1 to N where N is the number of files in the input stack.
- Better error message when input file has 0 rows

get_src_region

- Removed from CIAO

hrc_process_events

- Bug fix for getting *_NOM keywords from either obsfile and/or event file. Dealt w/ other keywords as well, eg SIM*.

lightcurve

- Deprecated: will not appear in future CIAO releases. Use dmextract instead.

mkarf

- Now automatically accounts for time dependence of the ACIS QE due to the buildup of contamination on the optical blocking filter. See 'ahelp ardlib' for details.

mkgarf

- Now automatically accounts for time dependence of the ACIS QE due to the buildup of contamination on the optical blocking filter. See 'ahelp ardlib' for details.
- Fixed bug in lookup of OSIP table, which sometimes caused a crash.

mkgrmf

- Now includes specific record of LSFPARMS file that is used to generate grating RMFs in HISTORY section of header.

mkinstmap

- Now automatically accounts for time dependence of the ACIS QE due to the buildup of contamination on the optical blocking filter. See 'ahelp ardlib' for details.

mkpsf

- PSFFILE keyword is added to output images identifying which hypercube the data came from.
- Fixed bug that prevented unique outpsffile filename from retaining path information
- Fixed bug that caused a failure to resolve CALDB when using ACIS (user had to override).

mkrmf

- Fixed bug that required axis1 to be energy and axis2 to be either pi or pha, and not the other way around. The axis values can now be reversed without any ill effect. (Previously, setting axis1 to pi or pha and axis2 to energy would give incorrect fit results in Sherpa.)
- Cleanup of .par file to be more consistent with other tools including making logfile hidden and def to stdout.

mkwarf

- Added ardlib 'mirror' parameter (eg so one can make mirror "ideal")
- Added min/max to verbose parameter
- Added new mskfile parameter to support mask files.
- Added new parameter "detsubsysmod" which allows ARDLIB modifiers to be added appended to internal "detsysname". For example, in mkwarf one can specify detsubsys="ACIS−3;IDEAL"; since mkwarf internally creates detector names need a way to append "IDEAL" to get ARDLIB modifiers.
- Changed FEFFILE from a history record to a keyword (consistent w/ other CALDB−enabled tools); added to the ARF output as well as the weight−file output.
- Put BASIC header into weights file instead of just PRIMARY (problem w/ DETNAM keywords)

pdump

- Can now specify path to file

pget

- Added −abort flag to exit w/ error status instead of quering user on bad parameter value

pline

- Can now specify path to file

pset

- Modified to open files with "L" mode to force learned behavior.

psextract

- New version that is compatible w/ dmgroup in CIAO 3.0
- Eliminated a number of high–verbose mode messages on "checking AOFF files"
- No longer supports aspect offsets files; user must provide aspect solution files
- Parameters 'aoff' and 'bgaoff' renamed 'asol' and 'bgasol'.
- Added 'clobber' parameter

psf_project_ray

- Added randseed parameter to allow ray weight filtering. randseed=−1 => no filtering; 0 = use sys time, <else>=randseed.
- Added ardlibpar parameter and code to include detector QE into rays that are allowed to pass through. Internally bpmask=0 and rays off edges are weighted by QE at chip edges so rays are still projected to semi–infinite planes w/o any discontinuities.

reproject_events

- Fixed problem with when match parameter had comma separated values; 2nd value was not being used.
- Quieted "Read foo–many events" message when verbose=0.
- Fix for crash when no time column and match=none
- Fixed problem w/ tstart/tstop values in warning when event data is outside range of aspect

rmfimg

- Made error message/checking better when file is not an RMF file (or extension is not specified when MATRIX is not first interesting block).
- Does full header copy of input rmf file to output image (including checksum/datasum which get automatically updated)

skyfov

- New tool: Makes a field–of–view region in sky coordinates. (For more info, see 'ahelp skyfov'.)

sso_freeze

- Fix the bug where code might unintentionally write a NULL value for [RA/DEC]_NOM instead of the correct value.
- Initialized RADSOL,AU,GMSOL keyword strings if missing from sso ephemeris file.
- Fixed problem where data subspace was not merged for stacks of inputs

stk_read_num

- Fixed error code when stack build fails

tg_create_mask

- Default HEG and MEG region width scaling factors were adjusted to be consistent with 'tgextract'. Prior to CIAO 3, the event cross−dispersion range provided by the region was slightly less than the background binning region, so the BACKSCAL values were in error by about 20%. The revised values also better accommodate HEG and MEG overlap at short wavelengths, and provide good extractions somewhat short−ward in wavelength of Fe K. (Previously, MEG counts spectra shortward of 1.86 angstroms were truncated due the overlap, i.e. there were fewer counts than there should have been.)
- Fix for problem with LETG angle: There is a calibration residual error in the grating and detector relative rotation for the LETG/ACIS−S combination; the code that corrected this error contained a bug, which has now been fixed. Note that this residual error is not a problem unless you use a narrow source region to bin your spectrum. For standard tgextract limits on tg_d, the spectrum is fully enclosed.
- Removed ACIS+LETG warning

tg_resolve_events

- Fix for a small−angle approximation error that caused wavelength residuals to increase with wavelength. This has most impact on LETGS data, since it has larger diffraction angles and longer wavelengths than HETGS. LETGS systematic residuals of up to 0.04A were reduced to less than 0.01A at 170A. (Other random wavelength errors are still present in LETG/HRC−S, and presumed due to detector non−linearities. These are still being analyzed.)
- Fix for sometimes showing wrong grating in pixlib history.
- Removed ACIS+LETG warning
- Made warning about number of unresolved events only appear if verbose > 1

tgdetect

- Changed default value of 'fixedcell' parameter to 15 (from 0 which implied to use recursive blocking).
- Fix for location of temp files
- Fix for HRC−I/LETG when zo_pos not specified

tgextract

- Default background outer limits have been adjusted to correspond directly to the maximum range permitted by the 'tg_create_mask' default mask size. This makes the BACKSCAL consistent with the data. (Before, the BACKSCAL for HETGS background regions was about 20% too large, and background estimates for the source region would have been too small by that amount.)

vtpdetect

- Quieted some of the verbose output (not controled by verbose parameter)
- Fixed problem requiring user supply "[x,y]" for event file
- Fixed problem with inputting an exposuremap when an image is used (instead of event list)
- Added a 2nd extension that outlines the exact src region that vtpdetect found rather than just the simple ellipse. It will contain polygons (possibly w/ excluded polygons) for each source. Can display in ds9 by doing "out_src.fits[src_region]" instead of "out_src.fits[srclist]"
- Added units "pixel" to R column.
- The R and ROTANG column are now consistent w/ the ds9/ciao interpretation. Previously, R[0] was the minor/major axis (always) and ROTANG was the PA; which was inconsistent w/ other detects and caused

many problems. Now R[0] is the angle along the "x"–axis and ROTANG is the angle from +y axis to r[0]
- Work on output values when input is an image. Values such as SRC_AREA and NET_COUNTS are now consistent between an image and event file (although they can be considerably different due to actual src region identified in each)
- Added "ELLSIGMA" keyword indicating the user parameter used to scale the "R" column

wavdetect

- New position–refinement algorithm, especially useful for off–axis source where PSF is highly asymetrical. Fixes bus error when very large scales are used.
- Made scales no longer hidden
- Fixed problem where setting the background time when background data were not entered was altering the wrecon results.
- Fixed problem with array bounds checking
- Fixed problem in which the input exptime was getting overwritten.
- Fixed bug in applying NSCALE during each loop
- Fixed the verbage of some misleading screen outputs (for bkgtime and exptime), since the times themselves are not multiplicative factors (the multiplicative factor is exptime/bkgtime)
- Added units "pixel" to R column.
- Allow x, y, and fluxscale to be comma separated
- Fix to allow multiple copies to be running simulatneously [NB: parameter dirs need to be different]
- Added "ELLSIGMA" keyword indicating the user parameter used to scale the "R" column
- Fixed problem with EXPOSURE values in output file
- Fix for NaN's in src output (problem in refined src algorithm); instead of NaN report unrefined position. 0's are still present and occur when src is Nx1 or 1xN
- Removed dependence on ascfitlib by removing model evaluation of old LSF params. Old psfsize files named "psfsize.fits" and "psfsize_19991012.fits" will no longer work with wavdetect.

wrecon

- New position–refinement algorithm, especially useful for off–axis source where PSF is highly asymetrical.
- Fixed problem where setting the background time when background data were not entered was altering results.
- Fixed problem with array bounds checking
- A problem in which the input exptime was getting overwritten was fixed.
- Fixed a bug in which some putative sources with *0* counts were getting through, causing NaNs in the output file. Such "sources" are no longer included in the output source list.
- Added units "pixel" to R column.
- Allow x, y, and fluxscale to be comma separated
- Added "ELLSIGMA" keyword indicating the user parameter used to scale the "R" column
- Fixed problem with EXPOSURE values in output file
- Fix for NaN's in src output (problem in refined src algorithm); instead of NaN report unrefined position. 0's are still present and occur when src is Nx1 or 1xN

wtransform

- Fixed bus error when very large scales are used.
- Fixed problem where setting the background time when background data were not entered was altering the wrecon results.

- Fixed bug in applying NSCALE during each loop
- Fixed the verbage of some misleading screen outputs (for bkgtime and exptime), since the times themselves are not multiplicative factors (the multiplicative factor is exptime/bkgtime)
- Allow x, y, and fluxscale to be comma separated

Parameter Files

In order to accomodate the new Parameter Editor GUI (Peg), filename parameters in *all* parameter files have been changed from type 's' to type 'f'. Since all files have been modified, users should delete all their old parameter files before running CIAO 3.0.

acis_process_events

- Now applies CTI correction by default (apply_cti=yes)

aconvolve

- Added normkernel parameter. Can normalize the kernel to unit 'area' (default), 'max' pixel value renormalize to 1, or 'none' (kernel left unnormalized).

dmgroup

- Removed parameters 'column' and 'bincolumn'. Added 'xcolumn' and 'ycolumn', which function differently than the old parameters. See 'ahelp dmgroup' for details and the Updating dmgroup Syntax for CIAO 3.0 thread.
- Added parameters 'stopspec', 'stopcolumn', 'errcolumn'. See 'ahelp dmgroup' for details.

dmimg2jpg

- Changed default scaleparam from 1 to 3 so that by default the log & pow methods will not produce the same results as linear

dmkeypar

- Removed minimum defaults (=0) for real and integer values in .par file.
- Added a new 'echo' parameter that prints the retrieved value to the screen. The default is no echo.

dmmakereg

- New parameter (wcsfile) to allow the user to input a reference image / eventlist. This allows a user to convert a region in World coordinates (eg WCS) to physical pixels which are necessary in CIAO analysis tasks.

mkrmf

- Cleanup of .par file to be more consistent with other tools including making logfile hidden and def to stdout.

mkwarf

- Added ardlib 'mirror' parameter (eg so one can make mirror "ideal")
- Added min/max to verbose parameter
- Added new mskfile parameter to support mask files.
- Added new parameter "detsubsysmod" which allows ARDLIB modifiers to be added appended to internal "detsysname". For example, in mkwarf one can specify detsubsys="ACIS−3;IDEAL"; since mkwarf internally creates detector names need a way to append "IDEAL" to get ARDLIB modifiers.

psextract

- Parameters 'aoff' and 'bgaoff' renamed 'asol' and 'bgasol'.
- Added 'clobber' parameter

psf_project_ray

- Added randseed parameter to allow ray weight filtering. randseed=−1 => no filtering; 0 = use sys time, <else>=randseed.
- Added ardlibpar parameter to include detector QE into rays that are allowed to pass through.

tg_create_mask

- HETGS default hetg_width_factor changed to 35

tgdetect

- Changed default value of 'fixedcell' parameter to 15 (from 0 which implied to use recursive blocking).

wavdetect

- Made scales no longer hidden

Sherpa

Most Noteworthy Changes

- Sherpa plot configuration is now done via the configuration variable sherpa. See AHELP SHERPA for an overview.
- The Sherpa/S−Lang module has been enhanced considerably. See AHELP SHERPA−MODULE for an overview.
- The Sherpa/S−Lang module is importable into other S−Lang−aware applications using `import("sherpa")`.
- Configuration of the parameter estimation commands (e.g., projection) has moved from internal Sherpa functions (accessed by SHOW) to the Sherpa configuration variables (e.g., sherpa.proj, sherpa.regproj). See AHELP SHERPA for an overview.
- Levenberg−Marquardt (LM) is now Sherpa's default optimizer.
- The instrument stack has been generalized such that it can contain more than one instrument model.

New (as Opposed to Enhanced) Functionality

- Two new optimization methods: MONTE–LM, and SIGMA–REJECTION (based on IRAF SFIT)
- A new statistic: CSTAT (equivalent to CASH as implemented in XSPEC).
- BFLUX: [command] returns the photon flux for background spectra.
- [B]EFLUX: [commands] returns the energy flux.
- [B]DCOUNTS: [commands] returns a summation of observed counts.
- [B]MCOUNTS: [commands] returns a summation of model counts.
- COORD: [command] sets 2D coordinate system (logical,physical,world) for fits.
- [UN]GROUP: [commands] applies grouping schemes read in via READ GROUPS.
- KERNEL: [command] defines an expression to be used as a kernel in 1D source model convolution.
- [B]NOISE: [commands] defines an expression to be used for modeling detector noise in source and background datasets.
- [B]SYSERRORS: [commands] analogous to ERRORS; defines systematic errors.
- BEQWIDTH: [command] equivalent to EQWIDTH, for background datasets.

Changes to I/O

- Data input from a FITS table (FITSBIN format) are no longer automatically grouped if a GROUPING column is present.
- Background data are no longer automatically regrouped to match the source data grouping, except to perform background subtraction.
- A configuration variable, sherpa.multiback, has been added; the default setting of 0 dictates that only one background dataset is allowed per source dataset. If 1, the user must identify each background dataset with a string, e.g., back 1 foo background.dat or back 3 A background.dat.
- The columns BACKGROUND UP and BACKGROUND DOWN in Chandra grating PHA files are now automatically combined into one dataset, unless the configuration variable sherpa.multiback is set to 1.
- The command BACK => DATA has been disabled.
- Quality flags may be read in using READ QUALITY.
- DATASPACE commands are saved in SAVE ALL.

Changes to Fitting

- Levenberg–Marquardt optimization works with source and background datasets of different sizes.
- Sherpa supports the fitting of data with BACKSCAL arrays (XMM data); subtraction and simultaneous fitting of data and background are allowed. See AHELP BACKGROUND for a caveat regarding background normalizations.
- SOURCE model stacks, et al., may now include user–defined model stacks, e.g., sherpa> foo = gauss; sherpa> source = foo. (User–defined stacks may also be embedded within other user–defined stacks.)

Changes to Sherpa Models

- BBODY has a new parameter, space, which allows the user to choose energy or wavelength blackbody formulae.
- Point–source modeling done with PSFFROMFILE is now done with the new models PTSRC1D and PTSRC[2D].
- There are several new instrument models: FARF[1D], FARF2D (FEXPMAP), FRMF, FPSF1D, TPSF1D, RSP2D.
- The EEARF parameter has been removed from RSP. Use the FARF model instead. See AHELP INSTRUMENT for examples.

Changes to Fitting/Modeling Utilities

- There are many new enhancements to the parameter estimation methods (e.g., REGION−PROJECTION supports logarithmic gridding).
- Sherpa will now refit data if a lower statistic is found by the UNCERTAINTY method. (This affects *all* parameter estimation methods, since UNCERTAINTY is used as at least a scale−setter for all.)
- A new parameter, FAST, has been added to PROJECTION−style methods; if 1, Levenberg−Marquardt is always used as the optimizer.
- Users are now alerted when PROJECTION is finished for a given parameter.
- The FLUX command now returns the photon flux only.
- All flux/counts−computing commands now work with 2−D data.
- All flux/counts−computing commands now work with user−defined model stacks.
- One may now filter data in 2D in logical, physical, or world coordinates.
- 2−D region filters must be quote−delimited (e.g., notice filter "circle(100,100,10)")
- FAKEIT can produce 2−D image data.
- GUESS (parameter value guessing) may be turned ON/OFF.

Changes to Visualization

- There are several new options for LPLOT, including UDATA and UFIT, which show unconvolved data and fits, SYSERRORS, and several new background visualization options. See AHELP LPLOT.
- IMAGE displays the product data*filter, over the full unfiltered range.
- Imaged data are now automatically resized to fit the ds9 frame.
- CPLOT and SPLOT displays the product data*filter, over the smallest rectangular region containing all noticed data.
- FLUX is now a synonym for SOURCE, e.g. lp flux.
- MCOUNTS is now a synonym for MODEL, e.g. lp mcounts.

Miscellaneous Changes

- The ERASE command has been extended to allow the deletion of user−defined model stacks, source datasets, and background datasets.
- Data output to PHA format files are automatically output as counts, regardless of the setting of ploty.
- TRUNCATE is no longer a synonym for TRUNCATE ON; the former now displays the trunction setting.
- The RECORD command has been changed: RECORD shows the record setting, while RECORD ON/OFF toggles RECORD on and off.

Commands Removed for CIAO 3.0

- EXPMAP: assigned via new instrument model FEXPMAP.
- PRECISION: assigned via new sherpa.output configuration variable.
- SET: assigned via sherpa plot configuration variables (see AHELP SET).
- WCS: coordinate system now set via new command COORD.
- WEIGHT: eliminated; use READ WEIGHTS (or module function set_weights) to input statistical weights.

User Infrastructure

Paramter Editor Gui (PEG)

- *NEW* New to CIAO 3.0 is peg, a Graphical User Interface which allows users to view and modify their parameter files. Most of the tools packaged within the Chandra X Ray Center's (CXC) data analysis package, CIAO, utilize parameter files to control their behavior. The tool provides built in range checking, the ability to restore default parameter values, and to retain settings over multiple executions of a tool. PEG is not geared towards any specific application. It is a general purpose parameter editor which can be used for any parameter file that is compatible with the IRAF/CXCDS parameter interface.

ascii2fits

- *NEW* The ascii2fits script provides a quick way to convert ASCII files to FITS format. The ASCII file must contain columns of numeric data only or be in RDB format.

ahelp

- ahelp can now use regular expressions to search for matches:

        ahelp /^get_/

  returns a list of all subjects that begin with "get_", and

        ahelp tools /par/

  lists those subjects in the "Tools" concept that contain the text "par".
- ahelp can now return specific parts of a document:

        ahelp -b QEXAMPLELIST dmextract

  returns the "Examples" section of dmextract and

        ahelp -b PARAMLIST dmextract

  its list of parameters.
- The "−c" flag is no longer required to specify the context when the subject of the search is also given.
- The output format used by ahelp has been updated to improve legibility.

ChIPS

- S−Lang interface: The S−Lang curve() command has been enhanced to allow plotting of errors on the Y and X axes. A number of S−Lang functions have been added to control and query the state of ChIPS; use "ahelp /chips_/" to list these functions.
- Postscript output: The postscript created by ChIPS' PRINT command can be controlled by the COLORSYS and PAGESIZE commands or by the colorsys, pagewidth, pagelen, and unit fields of the ChIPS configuration variable (state object).
- Auto−scaled limits: The automatic scaling of limits for plots now includes a border to improve legibility. This can be controlled by new fields in the ChIPS configuration variable (state object): padfactor, fullautolimits, and lowerloglimit.

- New symbol types: Three new symbols – SOLIDDIAMOND, SOLIDUPTRI, and SOLIDDOWNTRI – have been added.
- Contour plots: Contour plots can now be switched between log and linear scales on either – or both – axes. Linear interpolation is used when converting between the two scales; the mingridsize field of the ChIPS configuration variable (state object) is used to control this interpolation. Contour plots can no longer be overlaid on other plots; in CIAO 2.3 this was possible but would result in incorrect plots if the axis limits did not agree.
- Store files: The format of the files created by the STORE command has changed: the first two lines are now commented out which means that they can be read by loaded when ChIPS is started by saying chips filename The previous format is still understood by the RESTORE command.
- New linewidth field: The linewidth field has been added to the ChIPS configuration variable (state object) to allow control over the width of lines.

Prism

- ASCII and RDB files: Simple ASCII files – those that can be processed by the ascii2fits program – and RDB format files can now be viewed by Prism.
- Plotting Errors: If you choose 3 components to plot, this will plot the three components as X, Y and error, in that order.
- Histograms: A histogramming function was added to PRISM, using ChIPS as the plotting engine. This allows a user to create a simple histogram – with evenly spaced bins – very quickly. More advanced binning schemes can be read in from a file. The data and routines are available from S–Lang to allow user customisation.
- Exporting data to ChIPS: The selected columns can now be sent to ChIPS, where they can be accessed via the S–Lang variable prism_data.
- Cell editing: The restriction on cell editing to using values that fit within the displayed cell has been removed.
- Image display: The default display of image data has been changed to more– closely match that of DS9. This can be changed by the imgdisplay parameter.

Taskmonitor

- The exit status of a tool is displayed in the "Status" field.
- The default behavior for the "Print" and "Save As" options is to use both the command string and the output text (previously it was just the output text).
- Added the "find" command to allow searching of the output from the task.
- The name of the window has been changed from "TaskWindow" to "TaskMonitor". The "Dismiss" button has been re–named "Exit" and the "Save" button re–named to "Save As".

Obsvis

- Dates: Extended ability to display roll and visibilities graph using textual Calendar date, not only numerical Julian date
- Field of View: Extend ability to display gifs of roll/vis, as well as field

Firstlook

- Sorting of OIFs now works correctly and it is now possible to display more than 10 OIFs at once.
- If given a directory name when started, firstlook will scan that directory for OIFs.

The Analysis Menu

- The format used to configure the contents of the Analysis Menu has been changed to match that used by DS9. As well as allowing the same file to be used to configure both CIAO tools and DS9, the format allows a much greater level of control. The old format is still understood (but deprecated).
- All CIAO GUIs – except for firstlook – understand this format.

Session

- Simplified interprocess communication through XPA by using simpler access point names. For example, when you launch PRISM, "prism" becomes the access point name. (in CIAO 2.3, the access point name is prism:username.host.display.instance #).
- The "–xpa" command–line option can be used to change the name of the XPA access point of those CIAO GUIs which can be controlled and queried by XPA.

Iconic option

- CIAO GUIs can now be started with the "–iconic" command–line option which will start the GUI in "iconified" (or hidden) form. The actual behavior depends on the window manager being used (including ignoring the request).

Datamodel

The CXC Datamodel library, which underlies CIAO's data I/O and filtering capabilities, has been completely rewritten for greater efficiency and maintainability. The new structure will make it easier for us to add new capabilities, but in CIAO3.0 the behaviour should be essentially identical.

User Caveats

- In some circumstances you may see different GTI blocks after filtering FITS files; the DM data subspace should be the same, though, so CIAO should give the same answers. (For instance, some DM operations used to gratuitously create a GTI from the TSTART/TSTOP keyword values when there wasn't one in the input file; and sometimes DM would retain a GTI block that applied to a chip whose data had been filtered out).
- There is an unfortunate bug with handling some XMM spectral PHA files with attached regions in detector coordinates. A workaround is to dmcopy the file without using 'opt=all'; this will lose the offending region extension.
- Some keywords may be written to output files in a different order.

Programmer Caveats

- The public API for the Datamodel library is unchanged.
- All old bugs may or may not exist at this point; please report any problems as new bugs even if they were known problems in the previous version.
- Changes in behaviour:
    - '[opt SCALE=NO]' doesn't work the same as it used to. This is the mechanism to get access to the raw values for FITS files whose columns have TSCAL/TZERO (or BSCAL/BZERO) keywords. The original way we used SCALE=NO let you use dmCoordGetTransform to get the scaled values later, but this doesn't work any more. Code should be changed so that you don't open the block with SCALE=NO any more, but instead use

```
scaled_col = dmTableOpenColumn( block, colname );
unscaled_col = dmCoordGetParent( scaled_col );
```

and then calls to e.g. dmGetScalar will return scaled or unscaled parameters depending on which descriptor is used as the argument, while dmCoordGetTransform_d(scaled_col, ...) will return the scaling parameters.

- Other notes (mostly also applied to DM in CIAO2):
    - ♦ Types (relevant on Alpha machines): Note that the DM types have a fixed external representation, so that dmLONG always is used to handle values that can fit in a 4 byte integer (since the DM's goal is to provide file interoperability). The DM api uses a C datatype that is always big enough to hold these values – but may be bigger. So, on the Alpha, dmGetScalar_l returns a C long, which takes up 8 bytes of memory, but you should not use data values which would exceed 4 bytes of memory, since DM will write only the lowest 4 bytes to disk. The point here is that we try and win 2 different ways: you always get the same output file on disk, but you don't have to rewrite your code from machine to machine either.
    - ♦ If there are multiple open blocks pointing to the same physical block which are being written to concurrently, some data may not get flushed to the file.
    - ♦ dmSetCptName() will not work for subspace columns (ie. the START/STOP for time cols and the foo_MIN/foo_MAX for other cols).
    - ♦ TLMIN/TLMAX and DSVAL values don't get clamped by the column's type. So, it is possible to generate keywords with values beyond the legal range of the datatype.
    - ♦ Can't change the aspect ratio of image data (e.g. with different binning factors on x and y) while region filtering:

        ```
        dmcopy "img.fits[sky=circle(300,300,10)][bin x=3,y=4]"
        ```

        gives bad results.
    - ♦ When a subspace region is attached to a vector column via the DSREF/DSVAL mechanism, it will only be read correctly if the column is named "SKY" or "POS". The user should use a non–DM utility to remove the region block and/or the DSREF/DSVAL keywords pointing to the region block.
    - ♦ Can't filter to select NULL values (would like to be able to do dmcopy evt.fits[tg_m=NULL], for instance.)
    - ♦ When a subspace range is set via the DSVAL = min:max mechanism, the min value MUST be smaller than the max value.
    - ♦ QPOE:
        - ◊ kernel isn't effective with BIT columns that aren't aligned to tbe byte boundary.
        - ◊ doesn't do a very good job writing .imh files.
        - ◊ There are many more. Use at your own risk.
    - ♦ FITS:
        - ◊ When retrieving non–byte aligned BIT data, the bits outside of the BIT data are set randomly. We should manually set these bits to zero.

Libraries

Note that some of the changes listed in this section were also included in the CIAO 2.3 patch. However, in that release, the library changes applied *only* to the tools that were patched. In CIAO 3.0, the changes now apply to all of CIAO.

ardlib

- The time dependence of the ACIS QE, due to the buildup of contamination on the optical blocking filter, is now accounted for automatically. A new ARDLIB qualifier, CONTAM, for ACIS−S/I, controls this feature. The filter contamination model accounts for the time−dependence of the QE but does not account for spatial variations in the contamination. See 'ahelp ardlib' for more information.

caldb lib

- Cleaned up and tweaked error messages
- Fixed uninit memory read in FTOOLs src.
- Increase path length to find 'caldb.config' file (was only 80 characters)
- Enabled http:// access to CALDB; need to have 'caldb.config' local but other files can be remote, eg

```
% setenv CALDB http://mysite.edu/~me/CALDB
% quizcaldb none chandra acis − − det_gain now 00:00:00 − \
http://mysite.edu/~me/CALDB/data/chandra/acis/bcf/gain/\
acisD2000-08-12gainN0003.fits[AXAF_DETGAIN]
```

cxcparam

- Added ":" to prompt when only min or only max is specified, eg "foo limit (0:)" now indicates that 0 is the min with no specified max; "foo limit (:1)" shows a max with no min.
- Added %xpa indirection form to obtain parmeter values from XPA−enabled applications (e.g. Prism, ds9); see 'ahelp parameter' for details
- Fixed integrated help; ??toolname will display 'toolname's full help file
- Fix for nonexistantkey="" that caused crash on Linux
- Fixed problem with empty .par file causing dmreadpar and dmmakepar to crash

dmtlib

- Now uses same history routines are rest of ciao (for compatibility with dmhistory).

dslib

- New routine to help with merging event header with data from obspar file.
- Updates to better merge keywords for event processing tools.

errlib

- New error message format, removes PID, adds error code, less white space
- Added release version to error message
- Added new error codes
- Added single quotes "'" around string values in the general lib.
- Tweak to some of the signal handling code; specifically making sure all errors are going to stderr.

grplib

- Library rewritten; used by dmgroup and group S−Lang module

hdrlib

- New library
- Added 'CTI_CORR' keyword so that analysis tools will propagate value so CALDB lookups can work.
- Better treatment of date–obs default keywords for unexpected data–types (eg MJDREF being a 'long')
- Fix for internal cast problems when keyword datatype is not expected (eg Long vs. Double)
- Fix to propagate keyword comments for some special keywords
- Added MJD_OBS keyword
- Correctly copies keyword units from input to output
- EXPOSUR* now part of basic header

histlib

- Fix for crashing if history record is > 1024 characters

mtllibs

- Added a new operator, ~=, which will return true if the string on the right hand side is a substring of the string on the left hand side. It will generate a data dependency error if the two values are not strings.
- Can set/check floating point values against NaN, eg foo=#nan or ((goo==#nan)?(bar=1):(bar=0)). Integer NULL values are not dealt with.
- Fixed some memory leaks, and some ABR/W with string columns.
- Fixed problem with 10*phas, eg scalar*array didn't work but array*scalar did.
- Fixed problem w/ scalar*array vs array*scalar expressions.
- Fixed problem with unarbitrary text between quotes in strings. (Eg "FOO.bar" would fail).
- Single quotes can now be used as string identifiers.

pixlib

- Fix for error handling when larger number of internal errors are encountered.

psflib

- Fixed bug that prevented unique outpsffile filename from retaining path information
- Fixed bug that caused a failure to resolve CALDB when using ACIS (user had to override).
- Removed dependence on ascfitlib by removing model evaluation of old LSF params. Old psfsize files named "psfsize.fits" and "psfsize_19991012.fits" will no longer work with celldetect (and tgdetect) or wrecon (or wavdetect).

region lib

- Fixes for problems with field() areas
- Fixes for pie() filtering
- Fixed error that affected a region with multiple shapes which are close to one another (so that their bounding boxes overlap) but not within each other (their bounding boxes do not entirely overlap). The region was incorrectly trimmed to use only the area within the bounding box of the first shape.
- Fix for sector area causing a crash
- Fix for problem w/ area getting counted twice when needing to 'pixelate' region.
- Added support for DS9 2.3b1 regions which include additional comment lines.

- Fix for problem w/ getting the bounding box for included/excluded regions. The problem report was that the area of ellipse!*sector produced the wrong value; this was due to bad bounding box causing a very large pixel size to be used.
- Increase string length from 1K to 32K
- Added analytical algorithm for computation of polygon areas.

stack lib

- Allow continuation in @ file by ending line with '\'
- Added routines to change current and specific item numbers to API: stk_change_current and stk_change_num.
- Added two new features to external file expansion.
    - "@−" will now override the default behavior of (many) tools to NOT prepend the path to files items listed in the stack file. This has been a continual problem with for example region files.
    - "@+" will now override the default behavior of recursively expanding items in the stack; ie it will cause files in the list file to be opened. Thus if @+foo contains @goo and @bar, the contents of goo and bar are added to stack. Its not truely recursive, if @bar contains @moo; @moo will not be expanded; however @+bar will cause @moo to be expanded.
- Fix for problem with files containing rgrid, lgrid, or pgrid but are not stack identifiers.
- Fixed problem with stk_expand_n being limited to 100 values.
- Fix for quietly breaking lines at 1024 characters

tcdlib

- Some minor speed improvements to Lucy deconvolution
- Fix for bug in getting origin of the kernel
- Fix for RL deconvolution on alpha (division by "0").
- Fix for slide convolve when kernel is asymetric (was getting flipped before).
- General speed enhancements to slide convolve
- Fix for low statistics PSF
- Fix for Nx1 or 1xN PSF (origin was getting set incorrectly)
- Fix for problem w/ edges=renom w/ normkern=none|unit and w/ method=slide. The kernel was getting area normalized despite normkern!=area.

S−Lang Modules

CIAO contains many libraries of code that perform 'under−the−hood' tasks, such as interpreting parameter files, converting between coordinate systems, and so on. CIAO 3.0 is the first release where many of these libraries now can be called from S−lang directly. Advanced users can develop tools for their specific needs using these libraries. Some CIAO 3.0 tools are already written in S−lang using these libraries, e.g. acis_fef_lookup, and more are expected.

caldb

- Interface to CXC CALDB library. Contains routines to search the CALDB for relevant calibration files. See 'ahelp caldb' for more information, or 'ahelp quizcaldb' for an example of a tool that uses the caldb library.

chips

- The curve() command has been enhanced to allow plotting of errors on the Y and X axes.
- A number of S–Lang functions have been added to control and query the state of ChIPS. Use "ahelp /chips_/" to list these functions.

group

- Interface to CXC group library. Contains routines to group, or rebin, arrays or files of data using a variety of algorithms. Can be used to rebin datasets in Sherpa dynamically See 'ahelp group group' for more information, or 'ahelp dmgroup' for an example of a tool that uses the group library.

paramio

- Interface to CXC parameter interface. Contains routines to read, set, and restore CIAO parameter files. This library is often useful when creating S–lang scripts that use a parameter file for input/output. See 'ahelp paramio' for more information.

pixlib

- Interface to CXC coordinate transformation library. Contains routines that facilitate conversion between the various Chandra coordinate systems, such as the detector plane, sky pixels, and the various grating coordinate systems. Allows for explicit inclusion of the aspect solution. See 'ahelp pixlib' for more information.

region

- Interface to CXC region library. Contains routines to read CIAO regions and to determine whether or not a given point is in or out of the region, as well as calculating the total area in a region. Can be used in conjunction with pixlib to determine how regions map onto various Chandra coordinate systems. See 'ahelp region' for more information.

sherpa

- The Sherpa/S–Lang module has been enhanced considerably. See AHELP SHERPA–MODULE for an overview.
- The Sherpa module is now importable into other S–Lang–aware applications using `import("sherpa")'.

stackio

- Interface to CXC stack library. Contains routines to read and evaluate CIAO stacks, which frequently begin with the @ sign in CIAO tools. Useful when developing general–purpose scripts that should be able to use either a single file or a stack of files, for example when using pixlib with an aspect solution. See 'ahelp stackio' for more information.

varmm

- It is now possible to import just Varmm using import("varmm"); Previously you had to import ChIPS to access the Varmm functions.

- readrdb() has been added to allow RDB files to be read.
- The structure returned by the read family of routines – such as readfile() and readarf() – contains an extra field "_filetype" which is a numeric identifier of the file type. A number of constants (such as ASCII and PHA_I) are also available to help identify these values.
- apropos() has now been documented and fits_bitpix() has been added to the library.

xpa

- Available via import("xpa"); from a S–Lang script – provides the functionality of the command–line XPA tools (see http://hea−www.harvard.edu/RD/xpa/). With this module you can control external programs which have an XPA access point – such as DS9 and most of the CIAO GUIs – by sending them commands or retrieving information from them. See 'ahelp xpa' for more info.

Documentation

Ahelp

- Global revision of all AHELP files: improved description, increased number of examples; checked consistency with CIAO3.0; added section with changes in CIAO3.0 to many tools.
- The S–Lang Run–Time Library documentation is now available via ahelp in the "slangrtl" context. To list the available documentation, type

```
ahelp −c slangrtl
```

CIAO Threads

- New Threads
  - Introduction to Peg [replaces the "Introduction to Toolagent" thread]
  - Introduction to the Analysis Menu
  - Creating Source and Background Files
  - Updating dmgroup Syntax for CIAO 3.0
  - Correcting Responses for ACIS Contamination
  - Measure Grating Dispersion Distance
- Updated Threads
  - Introduction to Prism
  - Filtering Lightcurves
  - ACIS Background Subtraction (Blank–Sky Files)
  - Basic Lightcurves
  - Create an Image of Diffuse Emission
  - Use merge_all Script to Compute ACIS Exposure Maps and Fluxed Images
  - Compute Single Chip ACIS Exposure Map and Fluxed Image Step–by–Step
  - Extract ACIS Spectra for Pointlike Sources and Make RMFs and ARFs
  - Step–by–Step Guide to Creating ACIS Spectra for Pointlike Sources

Sherpa Threads

- New Threads
  - Introduction to Fitting PHA Spectra
  - Sherpa Configuration: Using the State Objects

- ♦ Customizing Sherpa with a Resource File
- ♦ Sherpa and Scripts
- ♦ Introduction to the Sherpa S–Lang Module
- ♦ Changing the look of Sherpa plots using setplot.sl
- ♦ Step–by–Step guide to changing the look of Sherpa plots
- ♦ Estimating Errors and Confidence Levels
- ♦ Step–by–Step Guide to Estimating Errors and Confidence Levels
- ♦ Accessing fit results using S–Lang
- Updated Threads
  - ♦ Using an Exposure Map in Fitting Image Data
  - ♦ Advanced customization of Sherpa plots
  - ♦ Calculating K–corrections using S–Lang and Sherpa
  - ♦ Using the ACISABS model in Sherpa

ChaRT Threads

- Updated Threads
  - ♦ Preparing to Run ChaRT

Why Topics

- New Why Topics
  - ♦ ACIS QE Degradation
  - ♦ Continuous Clocking Mode
  - ♦ Destreaking ACIS Data
  - ♦ Timing Analysis with Lightcurves

CIAO Web Site

- Streamlined navigation bar (on the left of all CIAO web pages) with direct link to the most used pages (Analysis Guides/Threads, Why? Topics, AHELP System)

Sherpa Web Site

- All Sherpa–related documents have moved to the new Sherpa web site (http://cxc.harvard.edu/sherpa/).