URL: http://cxc.harvard.edu/ciao3.4/releasenotes/ciao_2.0_release.html
Last modified: 11 August 2006

CIAO 2.0 Release Notes

Return to: Version History

Contents:

Tools

```
* acis_detect_afterglow * (*new*)

        Summary  identify and flag anaomlies where residual charges on
        the ccd detectors appear as events in consecutive exposures in acis
        event files, aka "Flaring Pixels". It sets specific bits in the
        status column to indicate whether or not an event is considered to
        most probably be valid or a flaring pixel. The criteria used in
        determining this status are user configurable and include− chip
        position (required), fltgrade degradation (optional), and pulse
        height amplitude [PHA] degradation (optional).

        Known Bugs  − Input event list must already have a 32bit status
        word. Re−run acis_process_events on file if needed.

* acis_process_events * −− Produce/update ACIS coordinates, PI, GRADE

        Interface & Algorithm Changes  − added support of sky_1d
        computations for continuous clocking data − bad pixel check moved
        from acis_format_events to acis_process_events − added additional
        header keywords− BPIXFILE, GRADESYS, NOMCHIPY, FSW_VER S −
        calibration database mods − status column changed from 16 bit to 32
        bit to support new bits such as flaring pixels − no longer read
        instrume parameter file to set chipsys,tdetsys,fpsys − wcs for det
        coords changed from TAN to TAN− P − aspect correction code modified
        to use any of the following files− asol, aoff, or soff. − ra_targ,
        dec_targ, and roll_nom now read from obsfile − Changed default event
        defintions to be consistent with reprocessed data including:
        floating point cooridate are now stored in single percision and the
        status column expanded to 32bits. − Removed obsolete qp* parameters

        Bug Fixes  − pi bins now start at 1 instead of zero − subspaces
        (gti extensions) maintained when rerunning acis_process_events on a
        level 1 event filt − write out tlmin/max for grade and node_id columns
```

* aconvolve * −− Convolve an N−dimensional image with a kernel

* acrosscorr * (*beta*)

* ahelp *

* apowerspectrum * (*beta*)

* asp_apply_sim * −− Add the SIM offsets to the optical axis offset

Bug Fixes − fix the problem with the sign of ROLL_OFFSETS being opposite

* asp_calc_offsets * −− Apply offsets to L1 aspect solution

Interface & Algorithm Changes − Full parameter history added to header

Known Bugs − Given a stack of files with a blank line at the end of the stack will result in an output file that does not close properly and cannot be used. Users can check their stack's with the"stk_build" tool and remove any trailing blank lines.

* asphist * −− Bin the aspect into a 3D histogram

Interface & Algorithm Changes − create an empty file when no gti is found − rearrange the parameters and read all of them in at once

Bug Fixes − fix the problem with bins off by 1 res. element − fix SEGV problem if GTI extname is not specified in 'gtifile'

* axbary * (*new*)

Summary: applies barycenter corrections to all times using the orbit ephemeris file orbitFile, ra, dec, and refFrame "FK5" (DE200) or "ICRS" (default; DE405).

* celldetect * −− Uses a "sliding cell" to search for sources

Interface & Algorithm Changes − Now accepts an exposure map (parameter "expstk"). − New parameter "ellsigma" controls size of ascii ellipse regions. − New parameter "regfile" specifies name of ascii regions file. −New source output column "expo_ratio" gives ratio of source's detect region exposure to background region exposure. − New parameter "expratio" suppresses ascii regions of sources whose expo_ratio is below specified value. − Slightly different binning for recursive blocking − source positions will be slightly different. − Source list has columns suitable for DM spatial filtering (COMPONENT, SRC_POS, ROTANG, etc). − Source SNR test is now SNR > = threshold instead of SNR > threshold. − Source position columns are now double precision. − "stall" parameter has been deleted. − Default encircled energy is now .9 (which is actually .8)

Bug Fixes − A bug which would cause celldetect to fail on HRC−S was fixed. − Bug in net_counts_error and bkg_counts_error has been fixed. − Bug in cell sizes for HRC−S fixed. − Fixed bug in source list FITS Header.

* csmooth * −− Adaptively smooth a 2−D image

* dmappend * −− Append multiple blocks/extension

Bug Fixes − add messages to remind users appending different extensions − negative signifigances can be reported when the maximum smoothing scale is reached.

* dmarfadd * −− Add multiple ARF files together weighting by exposure

− Interface & Algorithm Changes The tool now allows an arbitrary number of ARF to be added. The former restriction of 8 files is removed. − Minor changes to the documentation to clarify what happens in particular situations. − allow a non grating file to go through without requiring the TG_(M,PART,SCRID) keywords.

* dmcontour * (*new*)

Known Bugs − The region file created by dmcontour only works as a filter for simple regions. If the region has "holes" then the filter created will not work. Users can display the contours with ds9 to ensure that they have a "simple" region.

* dmcoords * (*new*)

* dmcopy * −− Filter and bin tables and images

* dmextract * −− Make a histogram table file

* dmgroup * (*new*)

Summary: Tool to group a specified column in a table, with various options: number of bins, number of counts, SNR selection, or an adaptive approach.

Known Bugs − Will have problems when the number of bins exceeds the number of rows in the file. This will typically not be a problem for users working in 'channel' space. − the grp_stat_err column is incorrect when the following grouping options are used: NUM_CTS, NUM_BINS, and SNR.

* dmgti * −− create a Good Time Interval (mtl_build_gti)

Interface & Algorithm Changes − added support for ulong/ushort. − added status column to identify which limits were exceeded. − added second extension with limits used.

Bug Fixes − detected/fixed bug in handling of last row.

Known Bugs − Spaces on the command line may not get interpreted correctly. Try allowing the tool to prompt. − Error messages will be reported for 'mtl_build_gti'.

* dmhedit * −− edit file headers

Infrastrucucutre & Algorithm Changes − Added parameters 'unit' and 'comment' for use on the command line (in addition to the previous functionality in which the unit and comment were included within the value) − Removed 'kernel' parameter − Made more robust regarding whitespace within a filelist . − Type and bounds checking when updating a keyword.

Bug Fixes − Bug fix −file access checking enhanced. − Bug fix − keywords containing characters and digits could not be added.

− any *new* numerical value keyword will get stored as a floating point number. Existing keywords will retain their datatype. − "/"'s in the "value" parameter will be treated as separators for the comment, so things like full path names cannot be written.

* dmimg2jpg * (*new*)

Summary: Generates a JPEG and an optional encapsulated level 1 postscript file from a FITS file. The tool can read in CIAO region files and place overlays in either an overlapping mode or a merged region mode. Grid and coordinate markings can be added as well as source labeling with numbers. The font marking is primitive, but allows for quick looks at the image on the command line. Given three inputs, dmimg2jpg can also create a three color image.

Known Bugs − Printing the postscript images is known to take a long time.

* dmimgcalc * −− perform arithmetic on images

Known Bugs − Needs to have exact numerical match to preserve WCS. There is a logical match of WCS from the output of dmcopy and mkexpmap, but the exact values don't match, so the WCS is lost.

* dmimghist * (*new*)

Known Bugs − puts all pixels above max and below min into max and min bins respectively.

* dmkeypar * (*new*)

Used to retrieve header keyword information. When the user supplies a filename and a keyword, dmkeypar will write the value, datatype, unit, and comment to its parameter file. COMMENT and HISTORY are invalid keywords.

* dmlist * −− list contents or structure of a file

* dmmakepar * −− Write header keywords to a parameter file

Interface & Algorithm Changes − The verbose parameter has changed to an integer value between 0 and 5.

* dmmakereg * (*new*)

* dmmerge * −− Merge two or more compatible tables into one

* dmpaste * (*new*)

Known Bugs − will not report an error if column with same name already exists, and will not paste the new column.

* dmreadpar * −− Add parameters to a file header

Interface & Algorithm Changes − The verbose parameter has changed to an integer value between 0 and 5

* dmrega2fits * −− Convert a grating ASCII region into a FITS format

* dmregrid * −− rebin a stack of 2 dimensional images

Interface & Algorithm Changes − Speed enhancements for both the 'exact' and 'approximate' algorithms.

Bug Fixes − updating the previous 'exact' algorithm also fixed a bug which occasionally caused a segmentation fault for certain combinations of parameters . − if the binning specification is given by a stack in which the second (or later) item results in a greater number of output pixels than the first binning specification, a segmentation fault resulted. The tool now uses the number of pixels resulting from binning the first image, and gives a warning message.

* dmsort * −− Limited capability table sort

Known Bugs − should only be used in the defined threads. Its is known to have serious problems (and serious limitations) in general analysis sessions.

* dmstat * −− Computes common statistics for table and images

* dmtcalc * (*new*)

Summary: Very powerful table calculator. Works with all data types, array and vector columns, as well as a number of operations and functions. It creates a new file from a stack of input files, creating and populating any new columns that the user defines in the series of arbitrarily complex arithmetic expressions input via the expression parameter. Supported operations include arithmetic, exponentiation, boolean, and smoothing. Supported functions include those in the standard math library and conditional expressions.

Known Bugs − Currently will create a TLMIN/MAX == 0 if the input columns do not already have mins and maxes attached to them.

* dmtype2split * (*new*)

Summary: Create a type 1−like output file for each row specified of a type 2−like input file. Each of the specified rows in the input file is copied to a separate output file. If the input file is a pha file, keywords required by the OGIP format are also written; otherwise, generic keywords are written. For each scalar column in the input, the coordinate keyword information is written, and the datatype, value, and unit are written as a keyword. For each input column that is an array, a column is created from the datatype, value, and unit; the row of input data is read and written to the output column.

* dmwritefef * −− create a fits embedded function file

* get_src_region * (*new*)

Summary: determines a background threshold value and then output regions that have counts higher than the background threshold value.

Known Bugs − The tool can only use the x, y physical coordinates of the source regions. So if the input is an event file, it must have the x and y columns, and the wcs infomation in the header.

* hrc_process_events * −− Correct HRC event positions, times, PHA, etc

Interface & Algorithm Changes − added support for float data type for det and sky columns − updated TLMIN/TLMAX for sky and det columns − add the tap ring correction − added event "flatness" test − added event saturation test − added event "hyperbolic" test −

display the badpixfile warning before processing events. − Keep the amplitudes as 'double' for all corrections − Saturation coeffs are a function of amp_sf − set pi=pha if no gain map is used. − bugfix to the gainmap off by one − remove the hidden parameters (chipsys, tdetsys, fpsys) from the code and use the pixlib calls to set these values for different detector.

* lightcurve * −− Bin source and background events by specified time

Bug Fixes − a non−time−sorted file will not break the tool as it does before. The tool will give out warning and list the row numbers of the data that are not in time order.

* mkarf * −− Generate an ARF for Chandra images

Interface & Algorithm Changes − Support for ACIS Windows (maskfile) − clobber, maskfile parameters added. − Energy grid specified via `engrid' parameter. This gives the user more flexibility in choosing an energy grid.

Bug Fixes − avoid coredump if no bad−pixel file was found

* mkexpmap * −− Generate a Chandra imaging effective exposure map

Interface & Algorithm Changes − The exposure map be normalized by the exposure time to produce a cm^2 map, or left unnormalized (cm^2−sec). − gridding specification changed to match other tools (engrid) − clobber and normalize parameters added.

Bug Fixes − sign error involving RA CDELTA corrected. − axis−group name changed from XXX to SKY.

* mkgarf * −− Generate a Chandra Grating ARF for one detector element

Interface & Algorithm Changes − New parameters: clobber, osipfile, maskfile − Energy grid specified via `engrid' parameter. This gives the user more flexibility in choosing an energy grid. − Support for ACIS Windows (maskfile) − Support for order sorting tables (osipfile)

Bug Fixes − typo affecting the grating ARF for HRC fixed

* mkinstmap * −− Generate a Chandra instrument map

Interface & Algorithm Changes − Support for ACIS Windows (maskfile) − New parameters: clobber, osipfile, maskfile − Support zeroth order grating instrument maps (grating)

Bug Fixes − avoid coredump if no bad−pixel file was found

* mkoif * − Create an observation index file (oif.fits) used by the /Firstlook/ GUI.

Known Bugs − does not work on Linux platforms. Users can only use the observation index file that came with their data distribution.

* mkpsf * −− Extract a point spread function

* mkrmf * −− create response matrix file

* punlearn * −− Resets tool parameters to their default values

Known Bugs − reports of problems when running on parameter files with blank lines, eg ardlib.par. If needed, run paccess to determine the path to the parameter file and then just remove the file.

* reproject_events * (*new*)

* stk_build * (*new*)

Used to build a stack from a string, a file, or a combination of the two. Output is to stdout, stderr, or a file. Useful for scripts.

* stk_count * (*new*)

Used to count the number of elements in a stack. Output is to the stk_count parameter file. The stack can be defined by a string, a file, or a combination of the two.

* stk_read_num * (*new*)

Used to read the N'th element of a stack. Output is to the stk_read_num parameter file. The stack can be defined by a string, a file, or a combination of the two.

* tg_create_mask * −− Create a region file to define spectrum sky boundaries

Interface & Algorithm Changes − The value of the ROLL_PNT header keyword instead of the value of the ROLL_NOM header keyword will be used to calculate the angle of the grating mask. If the ROLL_PNT header keyword is absent, the tool will give out a warning message, and the value of the ROLL_NOM header keyword will be used. − The value of the DEFOCUS header keyowrd instead of the value of the SIM_X header keyword will be used to calculate the radius of the mask of zeroth−order source. If the DEFOCUS header keyword is absent, the tool will give out a warning message, and the value of the SIM_X header keyword will be used. − The LESF (Drake Flat) parameters (s1_drk1_x, s1_drk1_y, s1_width_drk1, s1_length_drk1, s1_angle_drk1....) are removed, since they are not used by the tool. − The default value for the input_psf_tab parameter is now set to CALDB.

Bug Fixes − Fixed a bug that occured when the input data have the LETG and CC MODE combination. − Fixed a bug that occured when an observation has a small roll angle.

* tg_resolve_events * −− Assign events to spectral orders

Interface & Algorithm Changes − Can accept a new type of response matrix (rm) file. The new rm file's energy data are spatially dependent (i.e depends on chip coordinates). Also, the new rm file's energy limits are given absolutely, instead of as differences from the peak. The old type of rm file can still be used. − The wcs information for detector coordinates are written out using a TAN−P transform instead of a TAN wcs transform. − Out put a new column, Sky_1D, if the data are ACIS continuous clocking data. − Added support to propagate TLMIN/TLMAX from input evt file. − Centralized usage of constants to use DS lib value. − The acaofffile and alignmentfile parameters have been changed from hidden type to auto type parameters. − The default value for the rmfile parameter is now set to CALDB. This means that the tool will look at the header of the input data and then use the appropriate response matrix file to process the data. − The qp_internals, qp_pagesize, and qp_bucketlen parameters have been removed. − The stdlev1 parameter has been replaced by two new parameters (stdlev1_ACIS and stdlev1_HRC). stdlev1_ACIS holds the ACIS event format definition string, while stdlev1_HRC holds the HRC event format definition string. There are significant changes in the default event format definition string. For example, the data type of tg_r, tg_d, det, sky x, sky y, tg_m, tg_lam, tg_mlam have been changed from double to float. − The default value of the eventdef parameter has been changed from ")stdlev1" to ")stdlev1_ACIS".

Bug Fixes − Bug fix for ACIS Graded data (CORN_PHA column not getting propagated).

Known Bugs − The output of tg_resolve_events doesn't have all the ONTIMEn & related header keywords.

* tgdetect * −− Locate 0−order image (tgmatchsrc and tgidselectsrc)

Interface & Algorithm Changes − Updated with new celldetect parameters. − tgmatchsrc max_separation parameter changed from 1 to 3 arcsec.

Known Bugs − No filter can be applied to the input or reference soruce files.

* tgextract * −− Bin grating wavelengths column

Interface & Algorithm Changes − Added some new background region binning parameters, so background up and background down regions for binning the spectra no longer have to be symmetric to each other. − Removed the dependency of HEG grid on MEG grid. So HEG grid is no longer scaled according to the ratio of MEG and HEG preiod. New parameters are added for the user to specify the HEG, MEG and LEG grid. − Added function to log all the tgextract user input parameters in the output file. − Added the inregion_file parameter. So user can specify the source region, background up region and background down region in the input region file. Thus the tool can bin non−retangular regions. For example, a bow−tie shape region can now be defined for extracting HRC−S/LETG spectra. − The value of the output EXPOSURE keyword is now the average value of the input EXPOSURn keywords. − Added the TG_PART column to the output region extension. − Added the BACKSCUP and BACKSCDN keywords. If the output spectra have different BACKSCAL value, the BACKSCAL keyword will be promoted to a BACKSCAL column in the output SPECTRUM extension. Ditto for BACKSCUP and BACKSCDN. − Changed the POISSERR keyword from string type to boolean type. Also set its value to FALSE. − User cannot enter HESF as the value of the tg_part_list parameter, since this instrument is not supported. − The bin_units parameter can only be set to angstrom, eV, or keV. Degrees, pixel and mm are not supported. − The tg_d_units can only be set to degrees. Arcsec, mm and pixel are not supported. − The min_bkg_tg_d and max_bkg_tg_d parameters have been replaced by the min_upbkg_tg_d, max_upbkg_tg_d, min_downbkg_tg_d, and max_downbkg_tg_d parameters. − The min_bin, max_bin, bin_size and num_bins parameters have been replaced by the min_bin_leg, max_bin_leg, bin_size_leg, num_bins_leg, min_bin_meg, max_bin_meg, bin_size_meg, num_bins_meg, min_bin_heg, max_bin_heg, bin_size_heg, and num_bins_heg parameters.

Bug Fixes − Fixed bugs in ACIS + LETG extractions. − Fixed bugs in background binning. − Fixed bugs in the output region extension. − Fixed the HEG binning bug. − Fixed the tg_d limits bug. So now if user's input min_tg_d is larger than max_tg_d, the tool will use the default min_tg_d and max_tg_d.

Known Bugs − Command line filter cannot be used with the input file.

* vtpdetect * −− Voronoi Tessellation and Percolation (VTP) source detection

Interface & Algorithm Changes − Added WCS outputs − New parameter "ellsigma" controls size of ascii region outputs. − New parameter "regfile" specifies name of ascii region output file. − Debug file outputs are no longer generated. − Source list has columns suitable for DM spatial filtering (COMPONENT, SRC_POS, ROTANG, etc).

Known Bugs − need to keep using "[cols x,y]" syntax for event lists

* wavdetect * −− Mexican−Hat Wavelet source detection (wrecon & wtransform)

Interface & Algorithm Changes −The parameter "bkgerrinput" is now a boolean rather than a file name. If true, the error image is expected in the [2] extension of the input background image. − The "mask" parameter has been removed. − New parameter "ellsigma" controls size of ascii region outputs. − New parameter "regfile" specifies name of ascii region output file. − Source list has columns suitable for DM spatial filtering (COMPONENT, SRC_POS, ROTANG, etc). − Source position columns are now double precision. − Now accepts a Chandra exposure map. − A precomputed background map will now be accepted (parameter "bkginput"). − "stall" parameter has been deleted. − A new background extrapolator has been added, which reduces the number of false detections at large wavelet scales near detector edges. − Units have been eliminated from RATE outputs.

Bug Fixes − Incorrect source ellipse inclinations fixed. Slight algorithmic changes in computation of default normalized background. − A bug with non−symmetric exposure maps has been fixed. − Bugs have been fixed which affected background error computations when 1 iteration was specified. − A bug in PSF lookup for gratings data was fixed. − Bug is source property computation fixed. Bug occured when PSF size for source was greater than the largest wavelet scale size specified.

GUIs

========

Overview

========

In addition to numerous bugfixes and generally greater stability, the primary UI contributions to CIAO2.0 are:

o The integration of an interpeted language (S−Lang) into CIAO, which greatly increases the analysis capabilities of sherpa/chips/etc by providing an interactive, IDL−like interface at the commandline.

o This work also formed the basis for the introduction of spectroscopy analysis into CIAO (current tool name: GRASP, modeled in part on the ISIS tool developed at the MIT/CSR branch of the CXC)

o Considerably improved interaction between our analysis applications, by encapsulating the notion of an "analysis session" into a class library that is linked into each of our applications.

- Switched from directly using the SM executable for plotting to using the callable SM library instead.

- Considerably improved configuration of portable end−user distributions.

- GUIs now utilize parameter files to store persistent preferences.

- Several prism improvements, including 2 new XPA commands for GRASP, solidified keyword editing (using dmtlib now instead of dmlist), support for 2D slices (along z=1) of 3D images, providing structural info for ND images (N>3)

- Several FiltWin improvements, including new menu entry and commandline switch for loading data files, undo/redo fixes, and better ciao.par usage.

- Improved interaction of GUIs through Analysis Menu

- Private colormaps in GUIs

- Faster chips performance and improved configuration of runtime files

- RMF and image I/O in varmm; better chips parsing of SLang scripts, BUT until the sherp parser can be generalized more the scripts passed to "evalfile()" may contain ONLY SLang and chips commands.

- Many minor tweaks to remove compiler warnings and improve port stability.

Greater detail on these and other improvements is given below.

Infrastructure Details

Varmm/Grasp

The Varmm [(Var)iables, (M)ath, and (M)acros] library is a moderately thin wrapper around S−Lang, and with it provides:

- Access to a powerful IDL−like interpreted language, with a wide range of arithmetic and programming capabilities.
- The ability for chips to execute scripts containing a mix of chips and SLang commands, and for sherpa to mix sherpa, chips, and SLang.
- A C++ object interface to interpreted variables created interactively during analysis, which permits applications like chips and sherpa to easily utilize those variables for plotting/fitting.
- A rewrite of most of the sherpa I/O routines to enhance performance and modularity (using DM where appropriate). These routines can be used to create either interpreted SLang variables or straight C++ objects, and currently support:

    - ASCII
    - FITS/IRAF bintable
    - FITS/IRAF images
    - ARF/RMF/PHA

  The full range of DM filtering is supported, as well the ability to skip arbitrary numbers of table rows when reading any file format.
- A replacement for GNU readline, which is less functional but free of

restrictions which formerly prohibited callable SM from being linked directly into CIAO analysis applications.

o RMF and image I/O
o Faster and more robust SLang script reading in chips (but as noted above, sherpa still needs more work)
o Better portability to Dec/Alpha and Linux
o See below for related prism work (new XPA commands)
o Bug fix to PHA I/O (aneta bug report, 6/16)
o Changed behavior of varmmGet so that variables that may be resized and still be referenced in chips by their original name (dburke, 6/16)
o writeascii() I/O function added
o nominal/imperfect arrow key support added to VI mode
o tab completion added, and command history mechanism enhanced
o added ability to retrieve varmms (for plotting/fitting/etc) by namespace ID (eg: chips> curve x myNSpace−>x y myNSpace−>y)
o Editor/Key binding preference for varmm readline taken from editor parameter in ciao.par

These new capabilities are relatively vast in scope, and thus will need some time to document completely. Until then, users can look in ~mnoble/varmm.emails for more detail and explicit examples.

Session

The interaction of our analysis GUIs/apps has been enhanced by encapsulating the notion of an "analysis session" into a class library that is linked into each of our applications. This class object manages a shared memory segment and provides/supports:

o Multiple instances of chips running concurrently
o Imager exits ONLY when the application which invoked it is terminated or when a 'session' is exited (thus non−CIAO invocations of ds9 or SAOtng are NOT terminated when the user exits a CIAO applications)
o Ability to save the state of a session from filtwin or prism
o Ability to restore a session provided the user can access all of the data files
o Better communication between the GUIs and toolagent (eg, tools invoked from prism will default to running with the file currently being viewed within the prism window)
o The ability for multiple users to login to a single machine under a single login (eg, dutysci) without interfering with one another, provided each uses a distinct DISPLAY.
o Introduced a ciao.par parameter file, which allows the user to set various session−wide options like: the number of filter states to save or the maximum number of applications to run in a session, etc.
o Updated session_save option to utilize ciao.par clobber parameter if output file already exists
o Updated session_save menu to generate a warning dialog if unable to save the session info
o Some internal enhancements for shared memory segments / session IDs.
o Session management utilities:

```
    cxc_shmutility  renamed to ciaoshmem; −f option added to display full
                    ciao session configuraton.
    session_save    renamed to   ciaosave
    session_start   renamed to   ciaostart
```

o See parameter files section below.

GUI library

o CIAO guis can now utilize a private colormap. The $HOME/.CXCdefaults forcePrivateColormap entry may be set to True, False, or dynamic (the default), which respectively allow the the GUIs (eg, prism) to always use a private colormap, never use a private colormap, or switch to a private colormap at runtime if necessary.   This should address most of the "GUIs are invisible/unusable" problems that have historically occurred when the X colormap has been hogged by other applications (like Netscape).

NOTE: This has been tested in mwm, twm, fvwm, and CDE, but the "right thing to do" is ultimately a subjective question best decided by individual preference. In essence we have simply traded one ugliness (whitewashed GUIs from empty colormaps) for another (psychedelic flashing when moving the mouse from one app to another), although majority opinion deems the latter less unsavory.

o Fixed the Analysis menu pulldown's 'launch filtwin' option so that the application gets launched and an appropriate status message is reported

o Updated the Analysis Pulldown menu's launch commands so that the imager, browser (prism), and filtwin are launched with the data that is loaded in the invoking application

o File selection dialog behavior for initial directories and seach patterns is now dependent on the settings of parameters in ciao.par

o Information/Error dialogs updated to display application name.

o Old parameter files automatically detected and replaced with latest system version (previous is copied to a .bak file).

o Exiting from the window manager (for example using the window's close menu option or {alt}{f4}) now performs the exit cleanup that occurs when an application is exited from the app's file pulldown menu exit option. Previously, the application terminated but the application's cleanup procedures were not invoked.

Ahelp
−−−−−

CIAO2.0 ahelp commandline option synopsis:

```
−s [subject][..context] :
−m [subject][..context] :
−l [subject][..context] : short,medium,long versions of help output

−w [subject][..context] : use web browser to display information

−k keyword              : like appropros (searches for subjects related
                          to specified keyword)

−s −f file              :
−m −f file              :
−l −f file              : processes file and display as formatted ascii
                          text in short, medium, and long formats
−f file                 : equivalent to −l −f

−r                      : used to maintain the help database; creates
                          index files necessary for search/retrieval
```

* Format of xml files changed − ENTRY entity has only four attributes: key, context,refkeywords and seealsogroups.
* New entities have been added to dtd:

```
  ADESC       − like DESC but appended after PARAM block ,
  HREF        − used to keep information for hyperlinks in xml docs
                translated to HTML
```

```
    VERBATIM    − used to suppress formating of text within this block,
    XMLONLY     − text used together with PASSTHRU to specify alternative
                  text for PASSTHRU to be displayed only in ascii mode.
  * PARAMINFO entity removed as it was duplicating functionality of PARAM
  * ADESC and PARA can have titles via title attribute
  * ahelp keeps a database of subjects and references, which fosters
    efficient dynamic help search/retrieval
  * maintenance of xml files has been reduced via ahelp database reindexing
    facilities (−r option)
  * ahelp automatically creates intermediate web pages while reindexing
    database if mulitple ADDRESS blocks are encountered
  * extensive checking for compliance with the current dtd
  * Some changes to the public C++ ahelp interface (see the code)



=====================
Data Analysis Details
=====================

  Filtwin
  −−−−−−−

  o Integrated session library:
    − ability to receive updates from multiple instances of chips/prism
    − dynamically updates list of currently running chips/prism processes
    − ability to save/restore state of the entire analysis session
  o Now displays a variable number (scrollable) of table columns, rather
    than simply the static Temporal/Spatial/Housekeeping filter fields.
  o When displaying a FITS table: column data type, tlmin, tlmax, and
    units displayed through tooltips
  o Better interaction with ToolAgent, feeding it the current file and
    last filter which was applied.
  o Ability to restore sessions via a filtwin commandline option
  o Only one instance of filtwin may be invoked per analysis session
  o Persistent user preferences are saved across invocations to filtwin.par.
  o Uses ciao.par filtwin_init parameter to determine from where initial
    data will be loaded {browser, imager, plotter, none}. Filtwin will try
    to use the lowest available instance of the specified application.
  o Updated to allow the user to specify an initial data file or a session
    resume file as a commandline option. Starts a restore if the file is
    a session file, otherwise the filename is put in the file text field
  o Update filtwin instantiation from chips so that undo/redo work
    correctly
  o Update filtwin to prevent the filter field scroll window from flashing
    all of the columns of a file when an undo/redo is done on a file for
    plotting (only the axes columns should be displayed)
  o Update so that an 'apply' command to a table file attempts to image the
    data instead of automatically assuming that the data should be plotted.
  o Renamed 'Plotter' in the options toggle to 'Plot Zoom' to better indicate
    that it performs a chips pickpoints operation.
  o Feedback provided when a user performs a save data command.
  o If three axes are provided in the 'Cols' field, hitting [APPLY] will
    plot col1 as X  vs  col2 as Y  with col3  as the errs.

  Chips
  −−−−−

  o Switched from using IPC−based communication with background SM process
    to using callable SM library.  This makes plots from multiple apps
    independent of one another, should provide a substantial memory savings
    for large files, generally increase performance, and simplifies the
  software, configuration, and distribution.
o Integrated Varmm interface to S−Lang interpeted language into Chips.
  This allows users to plot directly from interpreted variables created
  during interactive analysis, write powerful functions/scripts, and
  provides a broad amount of other arithmetic and programmatic capability.
o no longer need to write an .sm.cxcds to $HOME
o Engine and Enginemode commands disabled.
o API now supports curve/contour/surface creation from float*, double*,
  short*, and long* arrays (previously only float* was supported)
o Improved performance of contour plotting.
o Several bugfixes:
  − ds9 would display blank screen if an absolute filepath not provided
  − at exit time threads no longer generate "killed" message on Linux
  − minor fixes to axes width, width, ln width, and errs commands
  − small parsing fixes
o Info command has been updated to display more information
o SAO/DS9 2.0 is bundled with CIAO2.0, and SAOtng has been removed
  from the package.  Users may still utilize SAOtng (1.9.1 or later) if
  they have it installed on their systems and edit $HOME/.CXCdefaults to
  swap the default imager entry from ds9 to saotng.

Firstlook
−−−−−−−−−

o Now exits when it receives a close request from the window manager.
o Generated lightcurves now display err bars.
o The name and path of generated lightcurve/spectrum files is echoed.

Online Help
−−−−−−−−−−−

o Online help output is automatically sized to fit the screen width.

Prism
−−−−−

o Now utilizes dmlist for header keyword display
o DataModel−visible header keywords may now be edited.
o Easier and smoother navigation:
    − added Navigate menu and more useful keyboard accelerators
    − no longer pops up annoying "working ..." window in average case,
      but only when large # of rows is being scrolled
o Attempting to open files that cannot be read by DM no longer leaves
  prism in an undefined, semi−unusable state.
o Removed references to status, error, and quickhelp "external message
  files," to simplify the codebase and make it easier to debug/maintain
o Bulk column/row deselection
o Persistent user preferences are saved across invocations to prism.par.
o Added "moveto {block#}" XPA command to enable switching the current
  block being displayed via remote control.
o Added "selected rows|cols" XPA command to retrieve (via remote control)
  the rows/cols currently selection−highlighted
o Support 3D image reading by taking 2D slice along z=1
o Correctly reports structure of 3D, 4D, etc images, instead of issuing error
  (images > 3D, however, will still not be read).
o Also, image info is presented more naturally/correctly as dim1 x dim2 x ...
  instead of treating all images as 2D tables of "dim1 rows, dim2 cols"
o Directly uses dmtlib to display keys (instead of dmlist executable) to
  to ensure header display is always in−sync with file contents.
o Clear out the data matrix when a block is either empty or cannot be read.
  Previous behavior left the old data intact
o Small improvement to image I/O performance.
```

- Several bugfixes:
  - first cell of an expanded vector column no longer 'selected'.
  - eliminated XmScrollBar warning messages when changing to files with fewer columns than the currently displayed file.
  - fixed segv seen when toggling between two extensions in FITS file

ToolAgent

- Toolagent behavior now configurable via ciao.par 'toolagent' parameter. If set to DISPLAY, a GUI taskmonitor is launced to display the stderr and stdout of the launched tool. If set to 'BACKGROUND', only the tool exit status is displayed, in an small popup window.

Portable System Details

Installation

- This release will eliminate the admittedly clumsy convention of using CXCDS_ROOT_{system} as the base CIAO install directory, and instead offers the user the ability to name this directory whatever they please. This will also simplify the installation by reducing the number of times the user must change their current working directory, as all files can now be untarred from the same level in the tree.
- More intelligent searching of X and Motif/LessTif libraries, which also helps reduce the size of our Solaris OTS.
- Supports multiple types of distributions (CIAO analysis / NRA3), while making it easy to add more.

Building from source:

We now utilize more of the power of Autoconf/Configure scripts. The user can now customize the configuration by using any of the following:

```
--x-includes=DIR        X include files are in DIR
--x-libraries=DIR       X library files are in DIR
```

Note: The location of the X libraries is determined automatically however if you want to use X from a non-standard location or if configure has difficulty finding the X libraries or headers you might want to use these switches.

```
--with-comp  Compile or just configure. Required for compilation.
--with-top=/top Explicitly specify top of distribution
```

This switch is used if you are installing ciao on a read-only file system such that the physical location is different from where the users will see it.

```
--with-irafdir=DIR Indicate where the IRAF directory is. The default
  is /iraf

--with-fits      Include FITS kernel
--with-iraf      Include IRAF kernel  (Solaris only)
--with-iraflibs  Use IRAF libraries provided in package  (Solaris only)

--with-nothreads Disable threads
```

This switch should only be used on older systems that do not fully support threads.

−−with−debug If the users are compiling the software, this switch will setup the make scripts to compile with the −g switch by default.

−−with−opt=−O3 If the users are compiling the software, this switch will setup the make scripts to compile with the select optimization switch. −O3 is the default level if no argument is provided.

−−with−tcltk={base tcldir} If the users are compiling the software. This sets the directory where the make scripts will lib and include directories that contain the tcl/tk libraries and includes. If this switch is omitted the default is /usr (this should be correct for most linux distributions).

−−with−tclver=8.0 If the users are compiling the software, this switch select the version of tcl/tk to use. Currently 8.0 is the only version that has been tested. The default is 8.0.

−−with−lesstif={base dir for lesstif / Mofit} If the users are compiling the software, this switch sets the base directory to look for the Motif lib and include directories. The default location is /usr/X11R6/LessTif

Environment/Configuration

- SM resource files are GNU−configured during CIAO installation
- Removed XPA_SHORT_TIMEOUT and added XPA_METHOD=local to the $HOME/.cxcds.(c)sh configuration, the latter of which helps avoid problems on machines (like laptops) not attached to networks.

Runtime Configuration/Customization

'ciao' alias (ciao.csh and ciao.sh)

- New switches

  −v Version switch: Prints the versions of all CXC software installed in $ASCDS_INSTALL tree.

  −i Ciao info switch: This switch activates a script that dumps you system configuration information the screen and the file $HOME/ciao_info.report This file can then be used to help debug users problems.

  −o Force setup switch: This switch allows the users to over−ride their current ciao setup and setup a new version of ciao. This switch will also remove the old setup for the users path and LD_LIBRARY_PATH to avoid conflicts and path bloat.

Parameter Files

```
  o new parameters in ciao.par to customize file selection dialogs:
      pattern − specifies the default filter pattern to use the first time
        a file selection dialog box is invoked from a GUI.
      dir_option − specifies whether the default directory for a file dialog
        should be based on the last directory used by the application or the
        last directory referenced by any application in the session. This
        value may be set to either APPLICATION or SESSION.
      workdir − if dir_option is set to SESSION, this value specifies the
        default directory to list when a file selection dialog is invoked.
  o new parameter in ciao.par to customize readline key binding preference
      editor − allows user to select emacs or vi type keybindings for readline
  o $HOME/.CXCdefaults ehanced to provide better GUI colormap support
    (background/foreground colors) under CDE.  Colormap problems are mostly
    eliminated under other window mangers.


UI CAVEATS
==========

  * Retrieval of region specifications from ds9 in filtwin is currently
    performed in physical coordinates. Unfortunately, there is an inconstency
    between filtwin and dmcopy where dmcopy expects logical coordinates when
    working on iamge files. As a result, when attempting to apply a region
    filter from filtwin to an image in ds9, the user currently needs to
    convert the region descriptor into logical coordinates.

  * The region filtering interface in ds9 was changed in ds9 v1.9.4 and
    was changed back insubsequent versions of the imager. As a result,
    filtwin's region handling code works with various versions of ds9
    except for v.1.9.4.
```

Sherpa

```
* New Commands *

    * ANALYSIS: allows the user to specify space for analysis: energy,
      wavelength, channel
    * BACKERRORS: allows the user to specify errors for the background
      dataset read in with "BACK." (Can be used with READ to read in
      errors from a file.)
    * COVARIANCE: calculation of the covariance matrix and error estimates.
    * DEMINIT: a command to be used with the GUIDE package only.
      Prepares an intermediate file with flux and line positions from
      each model, and temperature and emissivity grids culled from the
      APED line list file.
    * FEFFILE: specifies a FITS Embedded Function file for Sherpa to
      read in.
    * FEFPLOT: allows the user to plot the line−spread function for a
      given photon energy in keV, based on information read in with
      FEFFILE.
    * INTERVAL−PROJECTION: creates a 1D plot showing the fit statistic
      as a function of the value of the specified parameter, with all
      other thawed parameter values allowed to float.
    * INTERVAL−UNCERTAINTY: creates a 1D plot showing the fit statistic
      as a function of the value of the specified parameter, with all
      other thawed parameter values held to their best−fit values.
    * LPLOT: formerly the Sherpa PLOT command. Uses all the same options
      formerly understood by the PLOT command; may be abbreviated LP.
    * PRECISION: allows the user to change the number of significant
      digits in some values output to the screen with SHOW.
```

  * PROJECTION: calculation of error estimates.
  * REGION−PROJECTION: creates a 2D plot showing fit statistic contours as a function of the values of the two specified parameters, with all other thawed parameter values allowed to float.
  * REGION−UNCERTAINTY: creates a 2D plot showing fit statistic contours as a function of the values of the two specified parameters, with all other thawed parameter values held to their best−fit values.
  * SETBACK: allows the user to set the time, backscal, or areascal for a background dataset read in with the BACK (or READ BACK) command.
  * SETDATA: allows the user to set the time, backscal, or areascal for a dataset read in with the DATA (or READ DATA) command.
  * TRUNCATE: resets negative model values to zero. Automatically set if the statistic is CASH or BAYES.
  * UPDATE: stores relevant information about a fit in an S−Lang variable called "fitResults".
  * WCS: allows the user to use WCS coordinates, if available, when fitting two−dimensional models.

* GUIDE * −− Grating User Interacitve Data Extension

  * A new package for analyzing grating data, called GUIDE (Grating User Interactive Data Extension) has been created. This package can be accessed from Sherpa and consists of a small number of Sherpa commands and a large number of S−Lang scripts. Please see the GUIDE Manual for more information.
  * Access to S−Lang has been added to Sherpa. S−Lang is an interpreted language that may be embedded into an application to make it extensible. Thus it is now possible for the user to write their own S−Lang scripts, which incorporate both S−Lang and Sherpa commands, and run them inside Sherpa.
  * Data from S−Lang variables (which were read in via the VARMMIO read functions) can be assigned to Sherpa data sets.
  * A new format for storing information relevant to a fit, called the Model Descriptor List (MDL), has been created. The MDL stores all relevant information about a fit (e.g., names of data files, source and instrument models, and model parameter values). The MDL can be written to a FITS file, or stored in an S−Lang variable called "fitResults". This variable is automatically updated after every fit, and can also be updated manually via the UPDATE command. The MDL is used by GUIDE, but may also be used for non−GUIDE analysis.

* Changes to I/O *

  * The user can now read in PHA datasets from PHA Type II files.
  * The user can now read in/analyze double (real*8) data.
  * Support for a Sherpa preference file, .sherparc, has been added.
  * SAVE I/O has been altered; vestigial parameters DELTA and EPSILON are no longer output; errors are specified by err_min, err_max, and err_met, where err_met = c (covariance), p (projection), or u (uncertainty).
  * The command record now places the file search.dat.[process ID] into the ASCDS_WORK_PATH directory, normally /tmp.
  * Users can now update multiple parameter values using GETX, GETY, instead of just one.
  * Users may save a Model Descriptor List (MDL) to a FITS file with the WRITE command.
  * Users may read in from an MDL file with the READ command. This has the effect of reading in all specified data files, defining all source and instrument models, and setting all parameter values.

* Changes to General Fitting *

* The means by which one specifies datasets in commands has been generalized. Commands that are affected: ANALYSIS, BACKGROUND, COVARIANCE, FIT, GOODNESS, INSTRUMENT, IGNORE, INTERVAL−PROJECTION, INTERVAL−UNCERTAINTY, NOTICE, PROJECTION, REGION−PROJECTION, REGION−UNCERTAINTY, SOURCE, SUBTRACT, TRUNCATE, UNCERTAINTY, UNSUBTRACT, WEIGHT, WCS
* The user can now jointly fit data specified with DATA and BACK (before, all BACK data could be used for was background subtraction). Two backgrounds are allowed per dataset (specifically, BACK UP and BACK DOWN; the names reflect the tie−in with grating analysis). Also, the background can be fit alone.
* The user can now specify background errors (BACKERRORS), either on the command line or from a file. (Equivalent to ERRORS.)
* The user can specify the time, backscal, or areascal of an input dataset (SETDATA) or background dataset (SETBACK).
* The user can change the analysis space (e.g. from energy−space to wavelength−space), or determine the current analysis space, using the command ANALYSIS. Note that ANALYSIS is reset automatically when PHA files are read in.
* Sherpa can now fit arbitrarily filtered histogram ASCII input.
* Fluxes can now be determined for individual model components.
* Counts fluxes can now be determined for individual model components.
* The user may filter data using NOTICE/IGNORE ENERGY/WAVE.
* Parameter value guesses for background data fits are now based on the background data, not the source data.

* Changes to Responses *

* The user can now specify an "encircled energy ARF" (EEARF) as the third parameter of the model RSP. Sherpa also looks for the EEARF in the RMF fits file; if there, it need not be specified separately.
* The user−specified ARF file no longer needs to have the same binning as the user−specified RMF file. (However, the EEARF binning must currently match the ARF binning.)
* The user can read in a FEF file using the command FEFFILE and display line−spread functions for a given photon energy using the command FEFPLOT. Note that the user cannot yet *use* the displayed line−spread function as a response.
* Sherpa can now apply PSFs to filtered 2D data.

* Changes to Optimization Methods *

* The LEVENBERG−MARQUARDT method has been fixed, allowing it to find the actual local minimum in parameter space in fewer steps.
* The implementation of the SIMPLEX method is improved, such that it is less likely now that it will stop short of the local minimum.
* GRID & GRID−POWELL have been changed so that the user may specify the number of grid points on each axis separately.
* The method parameter SHOW interface has been improved; notably, parameters which do not actually used by the method are not listed (e.g. the delta parameter for Sherpa 1.0).

* Changes to Error Analysis Methods *

* The uncertainty algorithm has been updated to allow the user to find uncertainties for subsets of the thawed parameters; it also can find uncertainty bounds beyond the min and max parameter values stipulated by the user. Also, one can apply uncertainty to a subset of the entered data (e.g. to one specific dataset when N

datasets have been entered).

  * Two new error analysis algorithms have been added: projection and covariance. The former determines error estimates numerically by varying each chosen thawed parameter in turn, and allowing the other parameter values to drift to new best−fit values. The latter estimates errors using the covariance matrix, the inverse of the matrix of parameter space second−derivatives at the best−fit point.
  * Four new error analysis/visualization routines have been added: interval−projection, interval−uncertainty, region−projection, and region−uncertainty. The former two allow the user to display a 1D plot showing the statistic value as a function of the value of the specified parameter (using either the projection or uncertainty algorithms). The latter two allow the user to display 2D contour plots showing the statistic value as a function of the values of the two specified parameters (using either the projection or uncertainty algorithms).

* New Sherpa Models *

  * BPL/BPL1D: broken power law
  * CONST/CONST1D: a positive constant, one dimension
  * CONST2D: a positive constant, two dimensions
  * DELTA/DELTA1D: a delta function, one dimension
  * DELTA2D: a delta function, two dimensions
  * EDGE: absorption edge as a function of energy or wavelength, one dimension
  * LINEBROAD: a line−broadening model, one dimension
  * NGAUSS/NGAUSS1D: a normalized Gaussian model, one dimension

* Altered Sherpa Models *

  * BB/BBODY: now includes amplitude parameter (ampl)
  * LORENTZ/LORENTZ1D: functional form changed, now normalized
  * PSF/PSFFROMTCD: parameter size changed to parameters xsize/ysize (i.e., Sherpa supports asymmetric kernels); nsigma changed from two to a user−specifiable parameter
  * RSP: new parameter eearf (encircled−energy ARF file)
  * SHEXP: the former Sherpa EXP model; functional form altered
  * SHEXP10: the former Sherpa EXP10 model; functional form altered
  * SHLOG/SHLOGE: the former Sherpa LOG/LOGE model; functional form altered
  * SHLOG10: the former Sherpa LOG10 model; functional form altered
  * SQRT: functional form altered
  * APEC: APEC models xsapec and xsvapec have been added. These are models to calculate emission from a plasma with a DEM. Data from either the APEC plasma emission file or a Raymond−Smith file may be used (the APEC file is used by default). The model xsvapec allows certain elemental abundances to be varied.

* General Model Changes *

  * The evaluation of all two−dimensional models now includes (numerical/analytical) integration (if desired).
  * All two−dimensional models may now be evaluated in "WCS coordinate space," i.e. as a function of RA and Dec.
  * A new command, TRUNCATE, has been added. This resets model values that are less than zero to zero, allowing the statistics BAYES and CASH to be used without difficulty. Note that when those statistics are specified, TRUNCATE is automatically turned on.
  * Sherpa's algorithms for guessing parameter values were overhauled: (1) better guesses are made for some parameters (such as fwhm of gauss); (2) time and arf information is taken into account for

guesses in the analysis of PHA data; (3) if the source model expression and the dataset number are specified in the same command (e.g. source 4 = pow[p]), then that dataset is used for guessing, and not the first dataset.

* Changes to Statistics *

  * For model variance chi^2 (CHI MVAR), error propagation in background subtraction has been fixed, and post−fit plots show properly sized errorbars (instead of using a default errorbar size of one).
  * For Primini chi^2 (CHI PRIMINI), error propagation in background subtraction has been fixed, and post−fit plots and goodness computations use the correct error values.
  * Plots of fits done using the statistics BAYES and CASH now show errorbars estimated using the Chi Gehrels variance (rather than showing no errorbars at all).
  * Error propagation has been generalized for background subtraction: (1) errors are properly propagated if the errors for either the data or the background or both are read in rather than computed; (2) errors are properly propagated if two background files are specified (UP and DOWN for Chandra grating analysis).

* Plotting *

  * Improved labeling for plots, by checking for exposure times, etc.
  * One can now create composite image filters using NOTICE/IGNORE IMAGE and the include/exclude region feature of ds9.

* Miscellaneous *

  * Added function to wrapper class, to switch model integration on and off from tools which use ascfitlib.
  * RESET command changed so that only model parameters are reset, and not the method parameters as well.

* Bug Fixes since v. 1.1 (not including items mentioned above) *

* I/O *

  * A bug which caused improper fit results after the entry of two−dimensional ASCII data entry was fixed.
  * Fixed problem where chips was being started from Sherpa when the user issued a help command, or made a typographical error.
  * Fixed bug in WRITE ERRORS so that the correct, filtered errors are actually output to the screen.
  * Fixed bug causing the uncertainty minimum to be written sometimes with "−−" in SAVE ALL.
  * Sherpa now checks PHA data for the column STAT_ERR_RATE if the column STAT_ERR is not found (bug fix for Einstein data).
  * The command RESET FILTER now immediately causes the data to be refiltered; before, if RESET FILTER was followed by, e.g., FIT, the fit was done using the old filter.
  * Typing "q" while in GETX (setting a parameter value from a plot) no longer leads to the parameter value being reset to zero.

* Plotting *

  * The user can apply IGNORE/NOTICE PLOT to a log plot to get filter boundaries.
  * One can LPLOT/CPLOT/SPLOT/IMAGE negative data without having to set ERRORS or READ ERRORS.

```
    * Fixed parser bug so that commands like LPLOT DATA will open the
      plotter automatically if chips has not yet been started.
    * Attempts to LPLOT, SPLOT, CPLOT or IMAGE an instrument when no
      instrument is loaded now cause an error message to be shown,
      rather than cause a Seg.V.
    * Fixed bugs relating to SPLOT/CPLOT SOURCE/BG.
    * Contour plots of rectangular filtered regions now work correctly
      (row-major intermediate information was changed to column-major).
    * IMAGE [model name] now works properly for arbitrarily filtered
      data; the imager no longer displays a 1 x N image.
    * IMAGE ERRORS now shows the errors rather than the input data.
    * SPLOT/CPLOT FILTER improved to give better zmin/zmax or contour
      levels.

* Fitting *

    * Bugs relating to timescales and data grouping in FAKEIT were fixed.
    * A bug which prevented Sherpa from fitting a (small) subset of
      filtered PHA datasets has been fixed for CIAO 1.1-1.1.1.
    * Fixed bug in the computation of residuals for 2D data.
    * An error which caused the number of degrees of freedom to be
      sometimes incorrectly tabulated when background-subtracted data
      was analyzed is fixed.
    * PSF now keeps FFT of the kernel in memory--speeding the convolution.
    * Defining a SOURCE for an empty dataset (made empty by IGNORE ALL)
      no longer results in badly set (NaN) model parameter values.
    * Parameter links can now be reversed after the link is broken (e.g.
      m1 => m2+5 ; ... ; m1 = 20 ; ... ; m2 => m1-5 is now allowed).
    * Using GUESS to reset model parameter values after a link has been
      set no longer causes the link to be broken.

* Methods *

    * A type-conversion error which caused garbage numbers to be written
      to the search.dat file was fixed.
    * Fixed bug in opt.C that caused the user's choice for random number
      seed in some optimization algorithms to be overwritten with "14391."

* Models *

    * A bug which caused Sherpa to Seg.V. when nested models were
      defined was fixed.
    * Increased the maximum number of model components per source and
      background models to be 512 (was 32).
    * Bugs in model dered fixed.
    * Bug causing bbodyfreq to output zero amplitude fixed.

* Known Bugs and Limitations *

* Relating to plotting/imaging *

    * The command CLOSE PLOT now longer deletes the plot window from the
      screen. (This is not a bug, but rather related to the fact that
      ChIPS is no longer run within Sherpa, but as a separate process.)
    * The user may not use SPLOT or CPLOT to visualize an arbitrarily
      filtered (i.e. non-rectangularly filtered) two-dimensional dataset.
    * ChIPS curves are plotted through regions that have been filtered out.
    * A Sherpa user may not overplot several datasets, although
      overplotting is allowed when using ChIPS itself.
    * Multiple plots in one ChIPS window that show different datasets
      that have been filtered in the same manner may be shifted with
      respect to one another.
```

    * One cannot create images which show only those data which lie within a NOTICE filter region.
    * The ChIPS UNDO command cannot be used properly from Sherpa; labels and axes are restored, but not the curves themselves. This is due to the fact that in ChIPS, storing the state of a plot works when the data were read from file, but not when read from S−Lang variables. Sherpa sends ChIPS data via variables, instead of dumping to temporary files, and hence UNDO (as well as STORE and RESTORE) do not work well for plots generated via sherap LPLOT commands. This fix for this will be for ChIPS to store plots even when data came from S_Lang variables.

* Relating to filtering *

    * A user may not read in a filter from a file, and then add to that filter by specifying more filters from the command line
    * In the analysis of PHA data, the user may not combine IGNORE BAD with other filter commands; the desired filter will not be created.
    * Read−in files related to a single dataset must have the same precision. For instance, improper results will be obtained if the user reads in single−precision data and a double−precision filter for those data.
    * Filter bounds that are given in double precision are truncated to single precision, regardless of whether the data are expressed in single or double precision. This can lead to data being improperly left in or out of a filter region.
    * Extra characters added to the end of otherwise valid NOTICE/IGNORE commands will (i) cause S−Lang error messages to be shown, although Sherpa *will* properly interpret the filter, and (ii) be shown when the SHOW command is issued
    * Some attempts to filter FITS table (FITSBIN) data using the column name will fail because the Sherpa parser mistakes the column name for a parser token. Similar errors will be seen if the column name is channel. An exhaustive list of bad column names has not been compiled.

* Relating to GUIDE *

    * When the MDL is updated, the flux of each model component is determined. Currently, the error on the flux is determined only if the model component is a normalized Gaussian, a Lorentzian, or a delta function. (For these models, the error on the flux is equivalent to the error on the amplitude, which is simple to estimate using COVARIANCE.)
    * Currently, the error on the flux can only be determined for the models listed above in (a) if the amplitude parameter is thawed, i.e., not linked to another model's amplitude parameter.

* Miscellaneous *

    * BUG: for the Linux version of Sherpa (compiled with gcc, v. egcs−2.91.66), an attempt to read in non−existent files will cause a core dump.
    * After the command JOURNAL [filename] is given, all commands issued to Sherpa are recorded, including mis−typed/wrong commands that cause error messages to appear. This can lead to problems if the output file is used as a script for a subsequent Sherpa session.
    * Issuing the DATA or READ DATA commands causes Sherpa to first delete all information that may already exist for the numbered dataset.
    * If one uses STATISTIC CHI MVAR or CHI PRIMINI, and issues the command SHOW ERRORS after the fit, the errors shown will not be

```
      correct. Do SHOW FILTER ERRORS instead.
```

Data Model

```
*  "ahelp dm" gives high level on line help.

*  Support for new DS9 region formats.

*  dmlist now gives higher precision for double precision
   quantities.

*  dmcopy binning on a table which is filtered on the binning
   axes now shrinks the image to surround the filtered region.
     dmcopy  evt.fits[sky=circle(4300,4200,100)][bin sky]
   will make a 100x100 image. The command [bin/f sky]
   will make the full image replicating the older behaviour.


*  Multiple component ("logical OR") command line filtering:

   You can now do

   "m33.fits[(tdetx=4740:4750,5608:5650||
      ((detx,dety)=circle(5786,5010,2)&&(node_id=3||pi=58))||pi=122)
      &&time=:63950001][cols time,tdet,det,pi,ccd_id,node_id]"

    or
   "m33.fits[@clean.filter]"
   where clean.filter is a file containing

(ccd_id=0:6,8:9,chipx=256,512,768,fltgrade=8)||
(ccd_id=0:6,8:9,chipx=257,513,769,fltgrade=16)||
(ccd_id=0:6,8:9,chipx=512,fltgrade=104)||
(ccd_id=0:6,8:9,chipx=513,fltgrade=208)||
(ccd_id=7,chipx=512:513,fltgrade=2,64)||
(ccd_id=0,chipx=81)||(ccd_id=6,chipx=58,585)||(ccd_id=2,chipx=559,571)


This can make the data subspace very complicated; adding the string
[opt −ssupdate] will suppress the updating of the data subspace if you
find this is giving you trouble.



* "Exclude" syntax
   You can do
     dmcopy "evt1.fts[exclude pha=2,time=4]" bad.fits
     dmcopy "evt1.fts[exclude @flt.fits]" bad.fits
   or
     dmcopy "evt1.fts[@!flt.fits]" bad.fits
   and get the stuff that doesn't pass the flt.fits filter.
   (Be careful with the @! notation which is quite limited,
   you can't combine an
   exclude and an include so [pha=2,@!flt.fits] is an error,
   it in fact also excludes pha=2.)

*  LIVETIME/ONTIME keywords now updated by filtering

   If you do
     dmcopy "evt1.fits[@flt1.fits]" evt2.fits
   the GTIs are updated as before, but now the LIVETIME
```

and ONTIME (and LIVTIMn ONTIMEn) keywords are updated. ONTIME is the sum of the GTI intervals, and LIVETIME is ONTIME times DTCOR, (with DTCOR assumed 1 if not present). The keywords are only updated if they are present in the input file.

* There have been some improvements to the copying of GTI extensions. GTIs are part of the file's data subspace and dmcopy recreates them rather than copies them, but now it preserves a few crucial keywords that got lost before.

* Bin image to image;
  We finally implemented a simple image rebinning option. You can rebin an image by an integer factor.

```
 dmcopy   image.fits[bin 4]   blocked_image.fits
 dmcopy   image.fits[bin x=2:20:4,y=1:1000:4]   blocked_image.fits
```

* String filtering: (FITS only)

```
 dmcopy   bpix1.fits[shape=point:rectangle]
```

  If a table has a string column, you can filter on ASCII−ordered ranges. The above example passes rows whose value of SHAPE lies between "point" and "rectangle" in alphabetical order. The range is a closed interval, i.e. both "point" and "rectangle" pass the filter.

* Bit filtering (FITS only)

```
 dmcopy   bpix1.fits[status=001x0x0]
 dmcopy   bpix1.fits[status!=001x0x0]
```

  This filter matches 001000, 0011010, 0011000, etc: the 'x' means accept either 1 or 0. Leading zeroes may be omitted so that 1x0x0 and 001x0x0 are the same.

  Currently you can only do this on columns of bit datatype. Note that XMM data uses an integer column for its FLAG bitmask; we plan that a later DM release will let you use bit filters on integer columns too, but CIAO2.0 can't do this.

* Bin to size:

```
 dmcopy  evt.fits[bin sky=1:1024:#200]
```

  bin from 1 to 1024 generating a total of 200 bins, as an alternative to specifying the step size directly.

* The syntax [cols #1,#2] now works and selects columns number 1 and 2.

* Multiple syntax sections:

```
 dmcopy  bpix1.fits[shape=point][cols status][status=001x0x0]
```

  is now interpreted as identical to

```
 dmcopy  bpix1.fits[shape=point,status=001x0x0][cols status]
```

  This will be helpful for programmers wishing to append extra filters to an input DM spec. The block name, if present, must still be the first syntax section:

```
  dmcopy bpix1.fits[events][shape=point]
```

  not

```
  dmcopy bpix1.fits[shape=point][events]
```

* Control option syntax:

```
 [opt option−list]
```

```
    dmcopy evt.fits[bin time,pha][opt mem=200]

   The supported options are:
     mem=n       Max memory is n megabytes. Used to control how
                 big an image you can make when binning.
     −ssupdate   Suppress updating the data subspace.
     type=i2     Control data type of output binning image.
     type=i4
     type=r4
     type=r8

Subroutine interface enhancements
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

These are relevant to anyone programming directly with the DM interface.

*  WCS enhancements: CROTAn, EQUIn, LONPn, LATPn recognized.
   New routine dmCoordSetParam allows you to set LONPn.



*  Support for null subspaces. If you filter e.g.
    dmcopy evt.fits[time<0] flt.fits
   this gives a data subspace which is empty.
   DM didn't use to handle this; now it writes a
     DSVALn = "NULL"
   keyword and filtering recognizes this and acts accordingly
  (so filtering on a file with any such keyword will give no
  output rows).



 Bug fixes:

*   Binning xmin:xmax:step  generates bins from xmin to xmin + n * step
    where n is such that xmax <= xmin + n * step
    A bug in binning created an extra bin in the common case where the step
    size divided into the range exactly (in other words, the eqn above had
    a < instead of a <=). This is now fixed so your images may end
    up 1 pixel smaller than they used to.

*   Column selection/filtering bug: specs of the form
      evt.fits[pha=20:40][cols time,grade]
    in some circumstances failed to apply the pha filter if it was not
    in the list of selected columns. This is fixed properly now − I hope!


*   Compound areas now calculated correctly (e.g. BACKSCAL in dmextract)

*   Multiple component filtering support.
    You can now do
    dmcopy "evt1.fits[@flt.fits]" evt2.fits
    and get the correct answer! Earlier implementations were flawed
    when multiple GTIs were present.

*   Data subspace failed to handle default filters in multiple−component
    subspaces correctly.


*   The FITS kernel occasionally got confused about which HDU
    it was meant to be writing to. I think this is fixed now.
```

```
*   The QPOE kernel was inconsistent about block names, sometimes
    appending ".qp" to them; I fixed this, the block name will
    not include ".qp".

 Outstanding bugs
 ----------------

 There are still problems with inserting header
 keys; still no support for writing variable length arrays or
 filtering on row number. Support for filtering regions from
 images is still primitive, with logical and physical coordinates
 not being well supported; using world coords (RA and Dec) is safer.
```

Other Libraries

```
Pixlib
------

  - The various pix_*.par parameter files are now gone.  They have
    been replaced with a set of CALDB indexed FITS files.  The files
    used by pixlib are set by the "geom.par" file.


TCD
---

  - The transformations/convolutions/ and deconvolutions library has
    been upgraded to use the "FFTW" algorithm.  It has also been
    upgraded for futher efficiency for 2D FFTs.  Tools such as
    wavdetect, csmooth, and aconvolve should experience great
    improvements in speed.


CALDB (*new*)
-------------

  - Based on version 4.2 of FTOOLs, the CALDB lib allows some selected
    tools to access the calibration database (CALDB) directly.  Each
    tool has to know what kinds of files it is looking for (a choice
    made by the developer+scientist) and then when the string "CALDB"
    is used for the calibration file name the tool will automatically
    find the file that matches the current observation.  Tools that
    have been CALDB-enabled include:  acis_process_events,
    hrc_process_events, tg_resolve_events, tg_create_mask, tgextract,
    mkgarf, mkarf, mkinstmap, and pixlib.


Parameter Interface
-------------------
  - Will try to detect if an out-of-date parameter file is present
    in the user's home directory and will move it out of the way
    so the tool can run without any problems.
  - Uses a different command line interface library so things like
    tab completion, back-arrows when the line wraps, emacs & vi
    style key bindings, etc may not either look or function as in
    previous versions of CIAO.

DSLIB
-----
  - Various physical constants were updated with the most recently
    accepted values.
```

```
REGION
−−−−−−
  − Bug was fixed in polygon filtering where two adjacent points
    shared the same x−coordinate.
```

The Chandra X−Ray Center (CXC) is operated for NASA by the
Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

URL:
http://cxc.harvard.edu/ciao3.4/releasenotes/ciao_2.0_release.html
Last modified: 11 August 2006