



AHELP for CIAO 3.4

caldb

Context: [calibration](#)*Jump to: [Description](#) [Examples](#) [See Also](#)*

Synopsis

CALibration DataBase (CALDB)

Description

The Chandra CALDB is the directory and indexing structure that stores and provides access to all calibration files that are required for standard processing and analysis, provided that they have been put into FITS format. The Chandra X-ray Observatory calibration database is modeled after the HEASARC style CALDB. It is the directory and file structure which stores the calibration data used in standard data (a.k.a. pipeline) processing, as well as for analysis with systems such as CIAO.

When you receive Chandra data products from the pipelines, you are getting data which has been generated using Level 1, 1.5, and 2 calibration files, which reside in the currently-active version of the Chandra CALDB at the Operations Control Center (OCC). When you analyze data, build response matrices or ARFs, use SHERPA with the PSF library, you are using your local version of the Chandra CALDB to access the calibration files (PSF's, FEF's, quantum efficiencies, grating efficiencies, HRMA axial effective areas and vignetting functions). The CALDB provides you with the most recently approved calibration data for distribution to the public, without your having to specify it yourself (unless you so desire).

The Chandra CALDB is currently in a state of development, both because of software enhancements and because of calibration updates. The active version of the CALDB used at the OCC will have profound affects on your standard data products (SDP). Similarly, the particular version of the CALDB you have installed on your local system will have profound affects on your analysis results.

Therefore you will be keenly interested in:

- maintaining the most current version of the CALDB for your analysis,
- understanding what version was active at the time your data were processed through the SDP pipelines,
- knowing when updates you need to improve your analysis will be available, and when they are released.

This information can be found in the [CALDB](#) and [CIAO](#) web sites.

INTERFACE TO TOOLS

Many CIAO tools can access the CALDB directly by getting the necessary meta from an input file already supplied. Things like the DATE, DETECTOR, and other instrumental configuration parameters are extracted and can be used to query the CALDB directly. The simplest way is to set the parameter value to "CALDB", eg

```
gainfile=CALDB
```

The parameters of tools that can access the CALDB directly are noted in their respective help files. Users can also override the parameters that the tool would have picked up from the header. This may be useful to work around some indexing problems, for example finding 'HRC-SI' calibrations which are indexed as 'HRC-S'. The syntax for overriding the various parameters that are used to make the query is

```
CALDB(param1=value1;param2=value2;...)
```

Adjustable parameters in CALDB lookup

parameter	Description
TELESCOPE	Change the telescope (TELESCOP) name
INSTRUMENT	Change the instrument (INSTRUME) name
DETECTOR	Change the detector (DETNAM) name
FILTER	Change the filter value (FILTER)
START_TIME	Start time of observation (DATE-OBS), HH:MM:SS format
STOP_TIME	Stop time of observation (DATE-END), HH:MM:SS format
START_DATE	Start DATE of observation (DATE-OBS), YYYY-MM-DD format
STOP_DATE	Stop DATE of observation (DATE-END), YYYY-MM-DD format
QUERY	Query expression to further restrict search

Example

Prior to CIAO 2.3 – when the ability to dynamically re-bin PHA to PI FEF files was introduced – the default value of the feffile parameter was set to

```
CALDB ( QUERY=CHANTYPE.EQ.PI )
```

to ensure that the search was restricted to files suitable for analysis of PI data (from CIAO 2.3 the default value is now just CALDB since this particular restriction is no longer necessary).

Example 1

```
acis_process_events evt2.fits gainfile=CALDB ....
```

The correct gain file for the observation will be selected from the CALDB using the header information in "evt2.fits".

Example 2

```
hrc_process_events my_hrc-si.fits tapingfile='CALDB(DETECTOR=HRC-S)'  
...
```

This will load all the necessary meta data from 'my_hrc-si.fits' but will override the DETECTOR, aka DETNAM, to look for HRC-S data since the CALDB doesn't know about the 'HRC-SI' DETECTOR.

See Also

calibration

[ardlib](#)

chandra

[coords](#), [guide](#), [isis](#), [level](#), [pileup](#), [times](#)

chips

[chips](#)

concept

[autoname](#), [parameter](#), [stack](#), [subspace](#)

dm

[dm](#), [dm binning](#), [dmcols](#), [dmfiltering](#), [dmimages](#), [dmimfiltering](#), [dmintro](#), [dmopt](#), [dmregions](#), [dmsyntax](#)

gui

[gui](#)

modules

[caldb](#), [paramio](#), [pixlib](#), [stackio](#)

slang

[overview](#), [slang](#), [tips](#)

tools

[quizcaldb](#)



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

URL:
<http://cxc.harvard.edu/ciao3.4/caldb.html>
Last modified: December 2006

