URL: http://cxc.harvard.edu/ciao3.4/caldb_slang.html
Last modified: December 2006

AHELP for CIAO 3.4

caldb

Context: modules

Jump to: Description Examples CALDB MODULE EXAMPLES CHANGES IN CIAO 3.2 Bugs See Also

Synopsis

The S−Lang interface to the CXC CALDB library

Description

The caldb module is the interface between the S−Lang interpreter (see "ahelp slang") and the CXC CALDB library (see "ahelp calibration caldb"). This document provides an overview of the features of the caldb module, and tips for using it efficiently in a S−Lang program. Detailed descriptions of each function are provided by individual ahelp pages.

The caldb module is not available by default; to use it in a S−Lang program, it must be loaded using the S−Lang require() function:

```
require("caldb");
```

Functions provided by the module

The following functions are provided by the module; use "ahelp <function>" to get a detailed description of a function:

Function name with arguments
String_Type calFindFile(String_Type, String_Type, String_Type, String_Type, String_Type, String_Type)
String_Type calFindFile(Caldb_Type)
Caldb_Type calCreateInfo([String_Type])
calSetTelescope(Caldb_Type, String_Type)
calSetInstrument(Caldb_Type, String_Type)
calSetDetector(Caldb_Type, String_Type)
calSetFilter(Caldb_Type, String_Type)
calSetExpression(Caldb_Type, String_Type)
calSetDate(Caldb_Type, String_Type)
calSetTime(Caldb_Type, String_Type)
calSetData(Caldb_Type, String_Type)
calPrintInfo(Caldb_Type)
String_Type calGetTelescope(Caldb_Type)
String_Type calGetInstrument(Caldb_Type)
String_Type calGetDetector(Caldb_Type)

String_Type calGetFilter(Caldb_Type)
String_Type calGetDate(Caldb_Type)
String_Type calGetTime(Caldb_Type)
String_Type calGetQuery(Caldb_Type)
String_Type calGetData(Caldb_Type)
Int_Type calGetError()

Querying the CALDB

The simplest case is when you want to know which version of a calibration product is valid for a particular dataset; an example would be finding which gain file should be used with a given level−2 ACIS event file. In this situation, calCreateInfo() should be called, giving it the name of the event file. This will return a variable of type Caldb_Type, which contains a CALDB structure filled with information taken from the file. The fields of this structure can not be accessed directly: they can be displayed to the screen using calPrintInfo() and read or write using the appropriate calGetXXX() and calSetXXX() functions.

Once the CALDB structure has been created, the calSetData() function should be used to set the name of the calibration product you require; this value is referred to as the "codename" in the CALDB documentation. An expression providing further constraints may also be set, if needed, using the calSetExpression() function. The calFindFile() function should then be called with the CALDB structure to find the corresponding file.

If you do not have a dataset with which to seed the CALDB structure, or wish to have full control over the query, then you can either create an empty CALDB structure and set the fields individually, or use the seven−argument form of calFindFile().

Example 1

```
chips> require("caldb")
chips> cal = calCreateInfo( "evt2.fits" )
chips> calSetData( cal, "DET_GAIN" )
chips> calSetExpression( cal, "cti_corr.eq.yes" )
chips> calPrintInfo(cal)
Telescope: Chandra
Instrument: ACIS
Detector: ACIS−01236
Filter: −
Start−Date: 2000−02−27T03:25:15
Start−Time: 03:25:15
Expression: cti_corr.eq.yes
Data: DET_GAIN
chips> gainfile = calFindFile( cal )
chips> print(gainfile)
${CALDB}/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0001.fits[AX
AF_DETGAIN]
```

Here we set up a CALDB structure using the observation details found in the file "evt2.fits". We then set up the fields to ask for the gain file appropriate for this file; since it is an ACIS−I observation we need to add a boundary condition to say whether the data has been corrected for CTI. The calFindFile() is then used to query the CALDB, where the return value is the name of the file.

Example 2

```
chips> osipfile = calFindFile("Chandra","ACIS","−","−","now","−","OSIP")
chips> print(osipfile)
```

```
${CALDB}/data/chandra/acis/cpf/osip/acisD2000−08−12osipN0006.fits[AXAF_O
SIP]
chips> gainfile =
calFindFile("Chandra","ACIS","−","−","now","−","DET_GAIN")
chips> print(gainfile)
${CALDB}/data/chandra/acis/bcf/gain/acisD2000−08−12gainN0003.fits[AXAF_D
ETGAIN]
```

In this example we used the seven−argument form of calFindFile() to find out the OSIP and DET_GAIN files that would be used for an ACIS observation taken now.

Example 3

```
chips> cal = calCreateInfo()
chips> calSetTelescope( cal, "Chandra" )
chips> calSetInstrument( cal, "ACIS" )
chips> calSetData( cal, "OSIP" )
chips> calPrintInfo(cal)
Telescope: Chandra
Instrument: ACIS
Detector: −
Filter: −
Start−Date: now
Start−Time: 00:00:00
Expression: −
Data: OSIP
chips> osipfile = calFindFile( cal )
chips> print(osipfile)
${CALDB}/data/chandra/acis/cpf/osip/acisD2000−08−12osipN0006.fits[AXAF_O
SIP]
chips> calSetData( cal, "DET_GAIN" )
chips> gainfile = calFindFile( cal )
chips> print(gainfile)
${CALDB}/data/chandra/acis/bcf/gain/acisD2000−08−12gainN0003.fits[AXAF_D
ETGAIN]
```

Here we repeat the previous example but use a CALDB structure to query the CALDB. This approach may be more useful if you intend to make multiple queries of the CALDB.

CALDB MODULE EXAMPLES

The caldb module functions are used in CIAO in the acis_fef_lookup tool (see "ahelp acis_fef_lookup"), which is now written in S−lang.

Using the module efficiently requires a knowledge of the CALDB internals and keywords. For more information, see the CALDB website, which describes the structure of the Chandra CALDB and lists keywords for all of the files in the CALDB.

CHANGES IN CIAO 3.2

The module can now be loaded by using the

```
require("caldb");
```

statement, although the previous method (loading with the import command) still works.

Prior to CIAO 3.2 the caldb and pixlib modules had to be started in a particular order (caldb then pixlib) when used together, otherwise a warning message was generated. This restriction has been removed.

Bugs

See the bugs page for the caldb library on the CIAO website for an up−to−date listing of known bugs.

See Also

caldb

calcreateinfo, calfindfile, calgetdata, calgetdate, calgetdetector, calgeterror, calgetfilter, calgetinstrument, calgetquery, calgettelescope, calgettime, calprintinfo, calsetdata, calsetdate, calsetdetector, calsetexpression, calsetfilter, calsetinstrument, calsettelescope, calsettime

calibration

ardlib, caldb

modules

pixlib

tools

quizcaldb

URL: http://cxc.harvard.edu/ciao3.4/caldb_slang.html
Last modified: December 2006