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

AHELP for CIAO 3.4

calFindFile

Context: caldb

Jump to: Description Examples CHANGES IN CIAO 3.1 Bugs See Also

Synopsis

Query the Calibration Database for a file

Syntax

```
String_Type calFindFile( Caldb_Type cal )
String_Type calFindFile( String_Type telescope, String_Type instrument,
String_Type detector, String_Type filter, String_Type date_time,
String_Type expression, String_Type product )

Error Return Value: NULL − also see calGetError()
```

Description

Given a query, this routine returns the full path to the matching file in the Calibration Database (CALDB), if one exists. If the query is not complete − so that the CALDB can not work out what file, such as the gain or OSIP, to look for − then an error message is printed to stderr and NULL is returned. If the query returns multiple matches − as is sometimes possible with the way data is indexed in version 2 of the CALDB − then a warning message is printed to stderr and the first file found is returned. The calGetError() can be used to find out information about the errors.

The calFindFile() routine can be called with either 1 or 7 arguments. The one−argument form requires a CALDB structure created by the calCreateInfo() routine. This version is convenient when you want to find out the file for a particular observation (since you can use an event file with calCreateInfo() to set up most of the fields), or you want to query the CALDB for several files. The 7−argument form of calFindFile() requires that you give all the fields that the CALDB needs for the query (using "−" for unimportant fields), as described below.

Values for the 7−argument form of calFindFile().

Argument	Value
telescope	See calSetTelescope().
instrument	See calSetInstrument().
detector	See calSetDetector().
filter	See calSetFilter().

date_time	This should either be "now", to indicate the calibration file currently relevant, or a string in YYYY−MM−DDTHH:MM:SS format.
expression	See calSetExpression().
product	See calSetData().

Example 1

```
chips> require("caldb")
chips> cal = calCreateInfo( "evt2.fits" )
chips> calSetData( cal, "DET_GAIN" )
chips> file = calFindFile( cal )
```

Here we have used the caldb module to find the gain file that should be used for the observation stored in the file evt2.fits (which is assumed to be an event file).

Example 2

```
chips> file = calFindFile("chandra","acis","−","−","now","−","DET_GAIN")
```

Here we have used the 7−argument form of the function to find out the gain file for an observation taken now. For CALDB 2.27 this returns the file:

```
$CALDB/data/chandra/acis/bcf/gain/acisD2000−08−12gainN0003.fits[AXAF_DETGAIN]
```

The path_basename() function from the S−Lang Run−Time Library can be used to extract the name of the file from the full path returned by calFindFile(). For the example above,

```
path_basename( file )
```

would return

```
acisD2000−08−12gainN0003.fits[AXAF_DETGAIN]
```

The other "path" functions, as well as strtok() and strchop(), can also be useful when manipulating these paths.

Example 3

```
chips> file =
calFindFile("chandra","acis","−","−","now","cti_corr.eq.yes","DET_GAIN")
```

In this example we added an expression −

```
cti_corr.eq.yes
```

− to find the gain file that corresponds to CTI−corrected data. For CALDB 2.27 this returns:

```
$CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0001.fits[AXAF_DETGAIN]
```

Example 4

```
chips> cal = calCreateInfo
chips> calSetTelescope( cal, "chandra" )
chips> calSetInstrument( cal, "acis" )
chips> calSetExpression( cal, "cti_corr.eq.yes" )
chips> calSetData( cal, "det_gain" )
chips> gainfile = calFindFile( cal )
```

This returns the same answer as the previous example but using the one−element form of calFindFile(). The advantage to this approach is that you can re−use the structure to ask for other files; for example to also find the PHA FEF for this observartion you could say

```
chips> calSetData( cal, "fef_pha" )
chips> feffile = calFindFile( cal )
```

which returns

```
$CALDB/data/chandra/acis/cpf/fefs/acisD2000−01−29fef_pha_ctiN0003.fits[FUNCTION]
```

for CALDB v2.27.

CHANGES IN CIAO 3.1

In CIAO 3.0 the documentation for the multi−argument version of calFindFile() did not mention the "filter" argument.

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

modules

caldb

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