URL: http://cxc.harvard.edu/ciao3.4/acis_fef_lookup.html
Last modified: February 2007

AHELP for CIAO 3.4

acis_fef_lookup

Context: tools

Synopsis

Find the FITS Embedded Function file for use by mkrmf

Syntax

```
acis_fef_lookup  infile chipid chipx chipy [outfile] [verbose]
```

Description

This tool returns the location of the FITS Embedded Function (FEF) file needed by mkrmf to make either a PI or a PHA RMF for a point source observed by ACIS. It is not needed if you are creating a weighted response (ie for an extended source) using mkwarf and mkrmf, since the CALDB can be automatically queried in this case, although acis_fef_lookup can be used if desired.

Output

When run, the tool prints the location of the FEF file (including any necessary spatial filter) to the screen and to its outfile parameter. This makes it easy to use from the shell, or script, since pget can be used to access the file name:

```
  unix% set feffile = `pget acis_fef_lookup outfile`
```

or the parameter redirection syntax (see "ahelp parameter") can be used:

```
  unix% pset mkrmf infile=")acis_fef_lookup.outfile"
```

Using acis_fef_lookup

The script can be used in two modes:

- Point source
- Extended source

FEF for a point source

You must supply the location of the source in CHIP coordinates − ie the value of the ccd_id, chipx, and chipy columns from the event file. The returned file name includes a DM spatial filter which specifies the part of the FEF file to use.

Since the dither pattern of Chandra means that even a point source moves over a non−negligible number of pixels, and likely over multiple FEF tiles, the decision of what location to use is not a simple one, and depends on your science goals and the location of the source (eg is it near a chip gap or does it fall on more than one node).

FEF for an extended source

If you wish to use mkrmf with a weight map, created by mkwarf, then you need the location of the FEF file without the spatial filter used in the point source case. This is achieved by setting chipid to "none" (the values of the chipx and chipy parameters are ignored in this case). However, this step is not needed since mkwarf and mkrmf can accept a CALDB query (e.g. feffile=CALDB) for the FEF file when using weights files.

The CIAO thread pages discuss in greater detail the steps necessary to calculate the ARF and RMF for a source.

Example 1

```
unix% acis_fef_lookup evt2.fits 5 512 512
.../acisD2000−01−29fef_pha_ctiN0004.fits[FUNCTION][ccd_id=5,chipx=449:51
2,chipy=449:512]
(the full path has been excluded for brevity)
```

This gives the file name of the FEF file applicable for the observation and position on the detector.

In this example the input events file is for an obervation at a focal plane temperature of −120 degrees C and which has had the CTI correction algorithm (see acis_process_events) applied to it.

One can then run mkrmf by either explicitly setting its infile parameter to the output of this tool, for instance by using pget (using tcsh shell syntax and again ignoring the full path) −

```
  unix% set feffile = `pget acis_fef_lookup outfile`
  unix% echo $feffile
  .../acisD2000−01−29fef_pha_ctiN0004.fits[FUNCTION][ccd_id=5,chipx=449:512,chipy=449:512]
```

− or by using the redirection capabilities of the parameter interface −

```
  unix% mkrmf infile=")acis_fef_lookup.outfile" ...other parameters..
```

See "ahelp parameter" for more on the parameter interface.

Example 2

```
unix% acis_fef_lookup evt2.fits none
.../acisD2000−01−29fef_pha_ctiN0004.fits[FUNCTION]
(the full path has been excluded for brevity)
```

Find the name of the FEF file but without any spatial filter. This is useful when creating weighted responses for extended sources.

Example 3

```
unix% acis_fef_lookup spectrum_cti.pi none
.../acisD2000−01−29fef_pha_ctiN0004.fits[FUNCTION]
(the full path has been excluded for brevity)
```

Here we repeat the previous example but using a spectrum (PHA format; the column that was binned on − ie PHA or PI − is unimportant) which was extracted using dmextract from an events file that had been CTI corrected.

Parameters

name	type	ftype	def	min	max	reqd
infile	string	input				yes
chipid	string		none			yes
chipx	integer		1	1	1024	yes
chipy	string		1	1	1024	yes
outfile	string	output				no
verbose	integer		0	0	5	no

Detailed Parameter Descriptions

Parameter=infile (string required filetype=input)

Source file (event or spectrum)

The infile parameter can accept any file that contains enough information in its header for the quizcaldb tool to work; event files (L1 and L2) and PHA−format files produced by dmextract are both suitable.

The necessary keywords that infile must contain are: TELESCOP, INSTRUME, DATE−OBS, and DATE−END. The following keys are also used if found: DETNAM, FILTER, CTI_APP, and CTI_CORR.

Parameter=chipid (string required default=none)

ACIS chip number

This parameter is used to define which ACIS chip you want the FEF for. The allowed parameters are 0−9 (for the 10 chips ACIS−0 to ACIS−9) or the string "none". This last option makes the tool ignore the values of the chipx and chipy parameters and just report the FEF file (with block name).

Parameter=chipx (integer required default=1 min=1 max=1024)

ACIS chip x coordinate

The spectral response of the ACIS chips depend on location within the chip (ie in which FEF tile the source falls). The chipid, chipx and chipy parameters specify this position, where chipx/y are in chip coordinates (ie 1 to 1024). If chipid=none then this value is neither prompted for nor used.

Parameter=chipy (string required default=1 min=1 max=1024)

ACIS chip y coordinate

The spectral response of the ACIS chips depend on location within the chip (ie in which FEF tile the source falls). The chipid, chipx and chipy parameters specify this position, where chipx/y are in chip coordinates (ie 1 to 1024). If chipid=none then this value is neither prompted for nor used.

Parameter=outfile (string not required filetype=output)

FEF file to use

This parameter is filled by the tool when it has run, and contains the name of the FEF file to use, including any spatial filter (if chipid was not set to none).

The value can be accessed using pget/plist/pdump; for instance:

```
unix% set feffile = `pget acis_fef_lookup outfile`
unix% echo $feffile
.../acisD2000−01−29fef_pha_ctiN0004.fits[FUNCTION][ccd_id=5,chipx=449:512,chipy=449:512]
```

Parameter=verbose (integer not required default=0 min=0 max=5)

Verbose level

If set to a non−zero value then the tool will output extra information (other than the name of the FEF file) when run. This information is only likely to be useful when debugging the tool.

CHANGES IN ACIS_FEF_LOOKUP 1.20

When looking up certain chipid values with CTI−corrected data, the script prints a warning that users should use mkacisrmf to create the RMF response, not acis_fef_lookup and mkrmf. The warnings are printed at verbosity > 0. The Creating ACIS RMFs with mkacisrmf thread has information on using that tool.

CHANGES IN CIAO 3.4

The tool has been updated to support the CTI_APP keyword and will no longer exit if the CTI_CORR keyword is stored as an integer, rather than a boolean, in the file header.

If the chipid is set to "none" then the chipx and chipy parameters will not be asked for.

CHANGES IN CIAO 3.0

The tool will now exit with an error if the CTI_CORR keyword exists in the header of the input file but it is not a boolean.

HOW IS THE FEF CHOSEN?

There are separate FEF files for each focal plane temperature since the spectral response varies with chip temperature. Prior to the CIAO 2.3 release (November 2002), there were separate FEF files for PI and PHA RMFs; with the introduction of the "dynamic rebinning" (aka "PI on the fly") mode of mkrmf there are now only PHA FEF files. As the CTI correction algorithm recovers much of the resolution lost due due to the radiation damage there are also separate FEF files for use with data that has and has not been CTI corrected.

Each ACIS chip is split up into a number of tiles and the response is assumed to be constant within each tile. The size of these tiles depends on the type of chip (whether it is front or back illuminated) and the focal−plane tempertature. Currently, as of the CIAO 3.0 release in summer of 2003, the minimum size for a tile is 32 by 32 pixels and the largest size is 256 by 32 pixels (note that no tile crosses a node boundary).

NOTES

This script is not an official part of the CIAO release but is made available as "contributed" software via the CIAO scripts page. Please see the installation instructions page for help on installing the package.

Bugs

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

See Also

calibration

ardlib

tools

acis_bkgrnd_lookup, acis_set_ardlib, acisspec, add_grating_orders, add_grating_spectra, asphist, dither_region, dmarfadd, dmfilth, dmregrid, fullgarf, mkacisrmf, mkarf, mkexpmap, mkgarf, mkgrmf, mkinstmap, mkpsf, mkrmf, mkwarf, psextract, psf_project_ray, rmfimg, specextract

---

URL: http://cxc.harvard.edu/ciao3.4/acis_fef_lookup.html
Last modified: February 2007