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

AHELP for CIAO 3.4

fullgarf

Context: tools

Jump to: Description Examples Parameters CHANGES IN FULLGARF 4.0.0 and 4.0.1 CHANGES IN CIAO 3.4 CHANGES IN CIAO 3.3 NOTES Bugs See Also

Synopsis

Create a grating arf for a particular order and grating for a given observation.

Syntax

```
fullgarf phafile pharow evtfile asol engrid dtffile badpix rootname
maskfile [pbkfile] [dafile] [osipfile] [clobber] [verbose]
```

Description

fullgarf is a script that creates a grating arf for a particular order and grating for a given observation. While the mkgarf tool will create a grating ARF for an individual chip given an aspect histogram, this script will create ARFS for each chip, creating aspect histograms as necessary. The script will then combine the individual ARFS into the full array's ARF via the dmarfadd tool (see mkgarf for further details). The order and grating can be specified either by giving the corresponding row in a Type II PHA file, or by specifying a Type I PHA file which contains a spectrum for the desired order/grating combination.

This tool carries out all of the steps involved in creating a grating arf and runs asphist, mkgarf and dmarfadd. See help on those tools for additional information. In successive invocations, the rootname parameter is used to check for the existence of asphist files in order to avoid re−creating them for the same chip (since they depend only on chip, and not grating or order).

HRC−S/LETG Data

For HRC−S/LETG data, fullgarf creates +/− 1 gARFs correctly. However, the functionality does not exist to create higher order responses. Users who wish to model more than the first order of the observation should follow the Compute LETG/HRC−S Grating ARFs thread to creater gARFs for higher orders.

HRC−I/LETG Data

This script does not operate on HRC−I/LETG data. Users doing analysis with this configuration should follow the Compute LETG/HRC−I Grating ARFs thread.

Example 1

```
fullgarf phafile=acisf00007_005N001_pha2.fits pharow=1
evtfile=acisf00007N001_evt2.fits.gz asol=pcadf085492801N001_asol1.fits
engrid="grid(acis_heg_1.rmf[cols ENERG_LO,ENERG_HI])"
```

For the first 2 examples, consider an HETG+ACIS−S observation with a pha file named "acisf00007_005N001_pha2.fits". Examining the file via dmlist (dmlist "acisf00007_005N001_pha2.fits[SPECTRUM][cols tg_m,tg_part]" opt=data) gives:

```

--------------------------------------------------------------------------------
Data for Table Block SPECTRUM
--------------------------------------------------------------------------------

ROW    TG_M TG_PART

     1   -3    1
     2   -2    1
     3   -1    1
     4    1    1
     5    2    1
     6    3    1
     7   -3    2
     8   -2    2
     9   -1    2
    10    1    2
    11    2    2
    12    3    2
```

In other words, e.g., row one corresponds to an HEG (TG_PART=1) minus−third order spectrum while row ten corresponds to an MEG (TG_PART=2) first−order spectrum.

In this case, fullgarf will create an ACIS−S, HEG ARF for the minus third order.

Example 2

```
fullgarf phafile=acisf00007_005N001_pha2.fits pharow=11
evtfile=acisf00007N001_evt2.fits.gz asol=pcadf085492801N001_asol1.fits
engrid="grid(acis_meg_1.rmf[cols ENERG_LO,ENERG_HI])"
```

This will create an ACIS−S, MEG ARF for the second order.

Example 3

```
fullgarf hrcf01715_002N001_pha2.fits 1 hrcf01715_002N001_evt2.fits.gz
hrcf01715_002N001_asoff.fits engrid="grid(hrc_leg_1.rmf[cols
ENERG_LO,ENERG_HI])" dtffile=hrcf01715_000N001_dtf1.fits
rootname=mrk421_
```

This will create an HRCS−S, LEG ARF for the minus first order.

Parameters

name	type	ftype	def	units	reqd	stacks

phafile	file				yes	no
pharow	integer				yes	
evtfile	file				yes	
dtffile	file				yes	
asol	file				yes	
engrid	string			keV	yes	
badpix	file				no	
rootname	string				yes	
maskfile	string	input			yes	
pbkfile	string	input	NONE			
dafile	string	input	NONE			
osipfile	string	input	CALDB		no	
clobber	boolean					

Detailed Parameter Descriptions

Parameter=phafile `(file required stacks=no)`

The name of the PHA file containing the order, grating and source position information. This file may be either a Type I or a Type II PHA file. (See tgextract for information on Type I/Type II PHA files.) In the case of a Type I file, this information is contained in header keywords. For a Type II file this information is contained in the data table of the SPECTRUM extension. In this case the data are specified by row number via the pharow parameter.

Note that the user does not need to specify which type of PHA file the above file is. The script will determine this via looking at the TFORMx header keywords.

Parameter=pharow `(integer required)`

Type II PHA files contain multiple spectra in the SPECTRUM binary−table extension. (Type I files contain only one spectrum.) Each row of the table contains one spectrum for a given order, grating and source. The pharow parameter specifies the appropriate row for the desired order and grating combination. (The order and grating data are contained in the TG_M and TG_PART columns). The dmlist tool can be used to determine row corresponds to which grating grating and order.

Note that if a Type I pha is specified for the phafile parameter, the pharow parameter is ignored.

Parameter=evtfile `(file required)`

This parameter is passed on to the asphist tool.

The event file provides observational configuration information via FITS keywords. It also provides good−time interval (GTI) data.

Parameter=dtffile `(file required)`

This parameter is passed on to the asphist tool.

This parameter gives the name of the file containing the dead−time correction factor. In the case of the HRC, these data are contained as a table in a FITS file. For the ACIS, a single value given by the DTCOR header keyword is used. This value is stored in the event file, so that usually (for the case of ACIS only!) one specifies the same value for this parameter as for the evtfile parameter.

Parameter=asol `(file required)`

This parameter is passed on to the asphist tool.

This parameter give the aspect solution or sim−corrected aspect offset file(s). It is required as input to the asphist tool. (For additional information on this, try ahelp asphist.)

Parameter=engrid `(string required units=keV)`

This parameter is passed on to the mkgarf tool.

This parameter gives the specification for the energy grid. The string may specify either a file (FITS or ASCII) which contains the energy grid or an explicit energy grid. CIAO users should, in general, use a gRMF file (created with mkgrmf) for the energy grid specification.

For example, to specify the grid contained in the MATRIX block of an RMF file name "grating_rmf.fits", specify: "engrid=grid(grating_rmf.fits[MATRIX][cols ENERG_LO,ENERG_HI])" You should NOT specify a block that contains a wavelength grid! For example, often a file will contain columns named BIN_LO and BIN_HI which may contain wavelengths. Such grids may not be specified, since the values will be interpreted as energies.

To explicitly specify a grid which runs from 0.3 keV to 10.0 keV in 0.01 keV increment steps, specify: "engrid=0.3:10.0:0.01"

To use an ASCII file which contains two columns, the energ_lo and energ_hi, specify: "engrid=grid(myfile.tbl)"

Note that the preferred grid is linear in wavelength, since this reflects the natural dispersion of photons onto the detector. For back−compatibility, we write the grid in energy units in ascending order. Grating rmfs (see mkgrmf) have energy grids in descending linear wavelength. (Also see mk_tggrid, a script which converts the wavelength grid of a pha file into an FITS−format energy grid file.)

Parameter=badpix `(file not required)`

This parameter is passed on to the ardlib.par file.

This parameter will replace the value given in ardlib.par. It is not implemented for observations involving the HRC.

Parameter=rootname `(string required)`

The rootname for all of the output files. The script will prepend this parameter to all output filenames. The script will first check for the existence of an output file (which includes this rootname). If the file already exists, the script will not create a new version.

Parameter=maskfile `(string required filetype=input)`

The mask file (msk1.fits) for the observation; used by mkgarf. The mask file is needed in particular when a window or subarray was used. A value of "NONE" indicates that no mask file will be applied.

Parameter=pbkfile `(string filetype=input default=NONE)`

The parameter block file, which defines ACIS pixel clocking parameters, is used in conjunction with the "dafile" parameter to apply the ACIS dead area correction. The correction is off by default in CIAO 3.4.

This is a standard product of pipeline processing and is available for every observation. Parameter block files have names of the form, "acisf146860615N001_pbk0.fits". The long string of digits refers to the time of observation (seconds since reference date).

See "ahelp mkgarf" for more information on this parameter.

Parameter=dafile `(string filetype=input default=NONE)`

ACIS "dead area" coefficients file, which may have the values "NONE" (no dead area computation), CALDB (for automatic lookup), or an explicit file reference to an ACIS "dead area" coefficients FITS table. The correction is off by default in CIAO 3.4.

The parameter block file is required to scale the coefficients in the dafile. If dafile=NONE, then pbkfile=NONE is assumed.

See "ahelp mkgarf" for more information on this parameter.

Parameter=osipfile `(string not required filetype=input default=CALDB)`

The Order Sorting and Integrated Probability file; used by mkgarf. The default value, "CALDB", indicates that the highest version for the most recent date before the observation date will be retrieved from the calibration database.

See "ahelp mkgarf" for more information on this parameter.

Parameter=clobber `(boolean)`

If set to yes, existing output files will be removed. Note that this value is passed on to each tool in the script. That is, the value chosen for this parameter will set the value for the clobber parameter of all tools called by the script.

CHANGES IN FULLGARF 4.0.0 and 4.0.1

fullgarf v4.0.0: New Parameters

Four new parameters have been added to the fullgarf script, all of which are used by the mkgarf tool.

- pbkfile and dafile: used to apply the ACIS dead area correction in the mkgarf step. The correction is off by default in CIAO 3.4.
- osipfile: the OSIP file to use in order−sorting. The default value, "CALDB", was previously hardwired in the script.
- maskfile: the mask file for the observation.

Read the parameter descriptions for details.

fullgarf v4.0.1: grep command

There was a minor fix to a grep command in v4.0.1.

CHANGES IN CIAO 3.4

Two new hidden parameters have been added to mkgarf − a tool called by fullgarf − to allow the ACIS dead area correction to be taken into account: "pbkfile" and "dafile". By default, the correction is "off", i.e. both parameters are set to "NONE". See the parameter descriptions in "ahelp mkgarf" for more information.

To apply this correction when running fullgarf, set the mkgarf parameters BEFORE executing the script:

```
unix% pset mkgarf pbkfile=acisf063875928N002_pbk0.fits dafile=CALDB
unix% fullgarf
```

The fullgarf script itself was not changed for CIAO 3.4.

CHANGES IN CIAO 3.3

The syntax of the asphist tool was updated for this release: the GTI filter is attached to the event file instead of the aspect solution file(s). For a detailed explanation of the syntax change, refer to the asphist help file ("ahelp asphist").

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
dm
    dmbinning
tools
    acis_bkgrnd_lookup, acis_fef_lookup, acis_set_ardlib, acisspec, add_grating_orders, add_grating_spectra, asphist, dither_region, dmarfadd, dmcopy, dmextract, dmfilth, dmgroup, dmimghist, dmregrid, get_sky_limits, mkacisrmf, mkarf, mkexpmap, mkgarf, mkgrmf, mkinstmap, mkpsf, mkrmf, mkwarf, psextract, psf_project_ray, rmfimg, specextract, tgextract

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/fullgarf.html
Last modified: February 2007