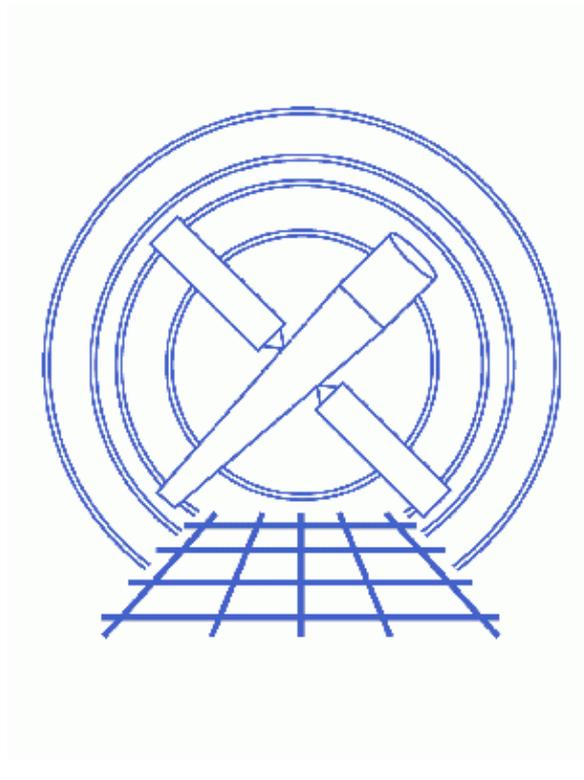


Step-by-Step Guide to Creating ACIS Spectra for Pointlike Sources



CIAO 3.4 Science Threads

Table of Contents

- **Get Started**
 - ◆ CALDB 3.3.0.1 patch
 - ◆ Using Consistent Calibration: mkrmf vs mkacisrmf
 - ◆ The ACIS dead area correction
- **Downloading acis_fef_lookup**
- **Extract Spectrum of Object**
 1. Determine Position of Source (ds9)
 2. Extract Source and Background Spectra (dmextract)
 3. Locate Centroids (dmstat)
- **Calculate the RMFs**
 - A. Using mkacisrmf (mkacisrmf)
 - B. Using mkrmf (acis_fef_lookup, mkrmf)
- **Calculate the ARFs**
 1. Compute the Aspect Histogram (asphist)
 2. Compute the ARFs (mkarf)
- **Update Spectrum Files**
 1. Group the Source Spectrum (dmgroup)
 2. Group the Background Spectrum (dmgroup)
 3. Update File Headers (dmhedit)
- **Analysis Caveats**
- **Summary**
- **Parameter files:**
 - ◆ dmextract
 - ◆ dmstat
 - ◆ mkacisrmf
 - ◆ asphist
 - ◆ mkarf
 - ◆ acis_fef_lookup
 - ◆ mkrmf
 - ◆ dmgroup
 - ◆ dmgroup
- **History**
- **Images**
 - ◆ Extraction regions overlaid on the image

Step-by-Step Guide to Creating ACIS Spectra for Pointlike Sources

CIAO 3.4 Science Threads

Overview

Last Update: 31 Mar 2008 – updated for CALDB 3.4.3: use `mkacisrmf` for –110 BI chips if TGAIN calibration has been applied

Synopsis:

Using a combination of CIAO tools, we extract source and background spectra for a pointlike source. The background spectrum is grouped, if desired. The appropriate Response Matrix Files (RMFs) and Ancillary Response Files (ARFs) are also created for both source and background.

Purpose:

To generate source and background PI (PHA) spectra of a pointlike ACIS source and build the proper RMFs and ARFs.

Read this thread if:

you are working with any ACIS observation, whether imaging or grating data.

Calibration Updates:

- **CALDB v3.3.0.1** (2 Feb 2007): The CALDB 3.3.0.1 patch corrects an indexing problem that may affects users of this thread. Read the Caveat: ACIS –120C FEF for CTI-corrected ACIS data for details.
- **CALDB v2.26** (20 Feb 2004): New FEF files containing updated information for ACIS S3 were added to the CALDB; there are CTI-corrected and uncorrected versions available. The ACIS contamination files were also included in this update.
- **CALDB v2.21** (14 Feb 2003): The new CTI-corrected FEF file (`acisD2000-01-29fef_pha_ctiN0002.fits`) contains updated information for chips S0, S1, and S3–5, eliminating the calibration problem announced in December 2002. Applying the CTI correction to these chips will no longer have a negative effect on the data.

Related Links:

- Using psextract to Extract ACIS Spectra and Response Files for Pointlike Sources: the automated version of this thread.

Proceed to the HTML or hardcopy (PDF: A4 / letter) version of the thread.

Get Started

Sample ObsID used: 459 (HETG/ACIS–S, 3C 273)

File types needed: evt2; asol1

Please ensure that you have set up ardlb to use the bad pixel file for your observation before following this thread.

CALDB 3.3.0.1 patch

The CALDB 3.3.0.1 patch, released on 02 February 2007, corrects an indexing problem that may affects users of this thread. Read the Caveat: ACIS –120C FEF for CTI–corrected ACIS data for details.

Make sure that this patch has been installed in your CALDB before continuing:

```
unix% dmlist "$CALDB/docs/chandra/caldb_version/caldb_version.fits[cols caldb_ver,ciao_ver]" data
...
 52 3.2.4      CIAO3.3.0.1
 53 3.3.0      CIAO3.4
 54 3.3.0.1    CIAO3.4
```

This file is automatically updated each time the CALDB is upgraded on your system, so the final row always indicates the current version.

Using Consistent Calibration: mkrmf vs mkacisrmf

The tool mkacisrmf is used to create RMFs for:

- all –120 ACIS data taken in (V)FAINT mode that has the time–dependent gain adjustment and CTI correction applied
- –120 ACIS GRADED mode data on the back–illuminated chips (ACIS–S1 and S3) only
- –110 ACIS data taken on the back–illuminated chips (ACIS–S1 and S3) only

*All new analyses with these types of data should be done with **mkacisrmf** instead of **mkrmf**.* This thread gives the syntax for both tools in the Calculate the RMFs section; it is up to the user to chose the appropriate method for the analysis.

It is important that the calibration applied to the event file is consistent with the RMF tool chosen, as explained in the "Using Consistent Calibration" section of the why topic. *If necessary, reprocess your data with the correct gain file before beginning this thread.*

The ACIS dead area correction

There is a fractional area loss per unit time due to cosmic ray flux incident on the ACIS detector. Calibration to account for this ACIS "dead area" was included in CALDB 3.3.0 on 15 December 2006. The correction is non–zero for the 8 front–illuminated ACIS chips; the effect is not detectable for the BI chips, so the nominal calibration value is 0.0. The resulting chipy–dependent reduction in the EA will be approximately 2.2% at the readout, and 4.0% at the top of the chip. Refer to the ACIS Dead Area Correction why topic for technical details.

In CIAO 3.4, the application of the dead area correction is *turned off* by default. However, users may opt to include it in the analysis by setting the `pbkfile` and `dafile` parameters in the **mkarf** step. Refer to the

[mkarf help file](#) for details on these parameters.

Downloading acis_fef_lookup

The most recent version of `acis_fef_lookup` is 1.20 (13 February 2007):

```
unix% grep Id `which acis_fef_lookup`  
% $Id: acis_fef_lookup,v 1.20 2007/02/13 19:36:28 egalle Exp $
```

Please check that you are using the most recent version before continuing. If you do not have the script installed or need to update to a newer version, please refer to the [Scripts page](#).

Extract Spectrum of Object

1. Determine Position of Source (ds9)

We need to define two regions, one for the source and another for the background. To do this, first display the image:

```
unix% ds9 acisf00459N002_evt2.fits &
```

In this example, we define the jet as the source with a rectangle (see [FAQ on how to rotate shapes in ds9](#)) and four 10-pixel radius circles for the background (from source-free parts of the image around the source). All the regions are shown in [Figure 1](#) . The background region can also be selected from a different event file, if desired.

To save the regions, follow these steps:

1. Region -> File Format-> Ciao
2. Region -> File Coordinate System -> Physical
3. Region -> Save Regions... -> Save As "3c273.reg" (source) and "3c273_bg.reg" (background). To select multiple regions for saving, hold down the <SHIFT> key and click on each one.

Note that both the "CIAO" and "SAOtn" region formats are supported by this script; we choose to use "CIAO" here.

The resulting region files will look something like this:

```
unix% more 3c273.reg  
# Region file format: CIAO version 1.0  
rotbox(4148.125,4043.625,7.58978,22.338761,44.516094)  
  
unix% more 3c273_bg.reg  
# Region file format: CIAO version 1.0  
circle(4119,4014.75,10)  
circle(4077,4025.75,10)  
circle(4186.5,4023.25,10)  
circle(4196.25,4064.5,10)
```

2. Extract Source and Background Spectra (dmextract)

In this example, we extract the spectra in pulse invariant (PI) space. This creates a histogram of number of counts vs. PI channel:

```
unix% punlearn dmextract
unix% pset dmextract infile="acisf00459N002_evt2.fits[sky=region(3c273.reg)][bin pi]"
unix% pset dmextract outfile=3c273.pi
unix% pset dmextract wmap="det=8"
unix% dmextract
Input event file (acisf00459N002_evt2.fits[sky=region(3c273.reg)][bin pi]):
Enter output file name (3c273.pi):
```

The wmap option in the call to dmextract creates a weight map (see [the documentation for dmextract](#)) which mkacisrmf will use to create a counts-weighted RMF over the extraction region. Creating a WMAP is extraneous if you plan on using mkrmf, but has no negative impact.

And for the background spectrum:

```
unix% pset dmextract infile="acisf00459N002_evt2.fits[sky=region(3c273_bg.reg)][bin pi]"
unix% pset dmextract outfile=3c273_bg.pi
unix% dmextract
Input event file (acisf00459N002_evt2.fits[sky=region(3c273_bg.reg)][bin pi]):
Enter output file name (3c273_bg.pi):
```

You can check the parameter file that was used with plist dmextract.

3. Locate Centroids (dmstat)

Since the calibration varies across the chips, we need to locate the centroid (in chip coordinates) of the source and background regions. This information is needed to create the ARFs, as well as to select which FEF (FITS Embedded Function) to use in calculating the RMFs with mkrmf. For the source:

```
unix% dmstat "acisf00459N002_evt2.fits[sky=region(3c273.reg)][cols chipx,chipy,ccd_id,x,y]"
chip(chipx, chipy)[pixel]
  min:      ( 258 350 )           @:      224
  max:      ( 309 394 )           @:      258
  mean:   ( 280.28940217 373.73097826 )
  sigma:    ( 12.242530849 11.837941816 )
  sum:      ( 206293 275066 )
  good:     ( 736 736 )
  null:     ( 0 0 )

ccd_id
  min:      7                   @:      1
  max:      7                   @:      1
  mean:     7
  sigma:    0
  sum:     5152
  good:    736
  null:    0

sky(x, y)[pixel]
  min:      ( 4138.965332 4034.1166992 )           @:      64
  max:      ( 4157.6948242 4052.0361328 )           @:      484
  mean:   ( 4146.0548958 4045.9470477 )
  sigma:    ( 3.5179725102 3.8411838536 )
  sum:      ( 3051496.4033 2977817.0271 )
  good:     ( 736 736 )
  null:     ( 0 0 )
```

The centroid of the source distribution is at chipx = 280.29 and chipy = 373.73. Note also that the source position is x = 4146.05 and y = 4045.95 and is on CCD 7 (ACIS-S3).

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

And for the background:

```
unix% dmstat "acisf00459N002_evt2.fits[sky=region(3c273_bg.reg)][cols chipx,chipy,ccd_id,x,y]"
chip(chipx, chipy)[pixel]
  min:      ( 251 296 )           @:      186
  max:      ( 340 465 )           @:      103
  mean:   ( 290.40277778 379.9537037 )
  sigma:    ( 19.837996456 51.468236206 )
  sum:      ( 62727 82070 )
  good:     ( 216 216 )
  null:     ( 0 0 )

ccd_id
  min:      7           @:      1
  max:      7           @:      1
  mean:     7
  sigma:    0
  sum:     1512
  good:    216
  null:    0

sky(x, y)[pixel]
  min:      ( 4067.6901855 4005.7412109 )           @:      186
  max:      ( 4204.6074219 4073.7333984 )           @:      209
  mean:   ( 4144.8603323 4032.6548281 )
  sigma:    ( 48.584061553 20.747670164 )
  sum:      ( 895289.83179 871053.44287 )
  good:     ( 216 216 )
  null:     ( 0 0 )
```

The centroid of the background distribution is at chipx = 290.40 and chipy = 379.95. Again, note that the mean position is at x = 4144.86 and y = 4032.65 and is also on ACIS–S3. You can check the parameter file that was used with [plist dmstat](#).

Calculate the RMFs

The observation used in this thread (ObsID 459) was taken at the –110 C focal plane temperature and the source is on ACIS–7, a back–illuminated chip. Therefore, it is [possible to use mkacisrmf to create the RMF file](#), assuming the dataset was reprocessed with CALDB 3.4.3 or higher.

The syntax for both `mkacisrmf` and `mkrmf` are given in this section. Users should choose the appropriate tool for the data and calibration.

A. Using `mkacisrmf` (`mkacisrmf`)

For the source:

```
unix% punlearn mkacisrmf
unix% pset mkacisrmf infile=CALDB
unix% pset mkacisrmf outfile=3c273_mkacisrmf.rmf
unix% pset mkacisrmf energy=0.1:11.0:0.01
unix% pset mkacisrmf channel=1:1024:1
unix% pset mkacisrmf chantype=PI
unix% pset mkacisrmf wmap=3c273.pi
unix% pset mkacisrmf gain=CALDB
unix% mkacisrmf
scatter/rsp matrix file (CALDB):
RMF output file (3c273_mkacisrmf.rmf):
WMAP file (3c273.pi):
```

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

```
energy grid in keV (lo:hi:bin) (0.1:11.0:0.01):
channel grids in pixel (min:max:bin) (1:1024:1):
channel type (PI|PHA) (PI):
gain file (CALDB):
```

This command creates a weighted RMF appropriate for the source region that was used to create the spectrum. The infile parameter is set to "CALDB" and `mkacisrmf` uses the information in the header of the input WMAP file to determine the appropriate calibration file to use.

For the background, repeat the command with the background spectrum:

```
unix% pset mkacisrmf outfile=3c273_bg_mkacisrmf.rmf
unix% pset mkacisrmf wmap=3c273_bg.pi
unix% mkacisrmf
scatter/rsp matrix file (CALDB):
RMF output file (3c273_bg_mkacisrmf.rmf):
WMAP file (3c273_bg.pi):
energy grid in keV (lo:hi:bin) (0.1:11.0:0.01):
channel grids in pixel (min:max:bin) (1:1024:1):
channel type (PI|PHA) (PI):
gain file (CALDB):
```

You can check the parameter file that was used with plist mkacisrmf.

If you use `mkacisrmf` to create the RMFs, you can now continue to the Calculate the ARFs step.

B. Using `mkrmf` (`acis_fef_lookup`, `mkrmf`)

First acis_fef_lookup is needed to determine the correct FEFs. For the source:

```
unix% punlearn acis_fef_lookup
unix% acis_fef_lookup acisf00459N002_evt2.fits 7 280.29 373.73
/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phan0002.fits[FUNCTION][ccd_id=7,chipx=2
```

and for the background:

```
unix% acis_fef_lookup acisf00459N002_evt2.fits 7 290.40 379.95
/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phan0002.fits[FUNCTION][ccd_id=7,chipx=2
```

You can check the parameter file that was used with plist acis_fef_lookup.

Now that we have the FEFs, we can compute RMFs with `mkrmf`. It is important that the axes are defined correctly. The energy range (keV) for `axis1` should cover the detector response range, which is ~0.2–10 keV for ACIS–S. The default for extraction in PI space is `axis2=1:1024:1`; in PHA space it would be `1:4096:2`.

For the source:

```
unix% punlearn mkrmf
unix% pset mkrmf infile="/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phan0002.fits[F
unix% pset mkrmf outfile=3c273.rmf
unix% pset mkrmf axis1="energy=0.1:11.0:0.01"
unix% pset mkrmf axis2="pi=1:1024:1"
unix% mkrmf
name of FEF input file (/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phan0002.fits
[FUNCTION][ccd_id=7,chipx=257:288,chipy=353:384]):
name of RMF output file (3c273.rmf):
axis-1(name=lo:hi:btype) (energy=0.1:11.0:0.01):
axis-2(name=lo:hi:btype) (pi=1:1024:1):
```

and for the background:

```

unix% pset mkrmf infile="/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phaN0002
unix% pset mkrmf outfile=3c273_bg.rmf
unix% pset mkrmf axis1="energy=0.1:11.0:0.01"
unix% pset mkrmf axis2="pi=1:1024:1"
unix% mkrmf
name of FEF input file (/soft/ciao/CALDB/data/chandra/acis/cpf/fefs/acisD1999-09-16fef_phaN0002.
[FUNCTION][ccd_id=7,chipx=289:320,chipy=353:384]):
name of RMF output file (3c273_bg.rmf):
axis-1(name=lo:hi:btype) (energy=0.1:11.0:0.01):
axis-2(name=lo:hi:btype) (pi=1:1024:1):

```

You can check the parameter file that was used with [plist mkrmf](#).

Calculate the ARFs

1. Compute the Aspect Histogram (asphist)

We now need to create the aspect histogram, which is a binned representation of aspect motion during the observation:

```

unix% punlearn asphist
unix% pset asphist infile="@pcad_asol1.lis"
unix% pset asphist outfile=3c273.asphist
unix% pset asphist evtfile="acisf00459N002_evt2.fits[ccd_id=7]"
unix% asphist
Aspect Solution List Files (@pcad_asol1.lis):
Aspect Histogram Output File (3c273.asphist):
Event List Files (acisf00459N002_evt2.fits[ccd_id=7]):
Live Time Correction List Files for HRC ():

```

In some cases there will be more than one asol1.fits file for an observation. *All* the files must be input to the [infile](#) parameter, either as a list or as a stack (see [ahelp_stack](#) for more information). For example, here we used:

```

unix% more pcad_asol1.lis
pcadf063874624N002_asol1.fits
pcadf063875522N002_asol1.fits
pcadf063902942N002_asol1.fits

```

You can check the parameter file that was used with [plist asphist](#).

2. Compute the ARFs (mkarf)

The [grating](#) parameter in [mkarf](#) is set to reflect whether or not grating data is being used. For non-grating observations, this parameter is left at the default (NONE). You can double-check the grating information in the file header:

```

unix% dmkeypar acisf00459N002_evt2.fits GRATING echo+
HETG

```

In this case, since we are using an ACIS/HETG observation, [mkarf](#) will extract the 0th order spectrum and ARF. The [sourcepixelx](#) and [sourcepixely](#) were found in the [Locate Centroids](#) step. The energy grid ([engrid](#)) must be the same as that used for the axis in [mkrmf](#).

```

unix% punlearn mkarf
unix% pset mkarf grating=HETG
unix% pset mkarf detsubsys=ACIS-S3
unix% pset mkarf outfile=3c273.arf
unix% pset mkarf asphistfile="3c273.asphist[ASPHIST]"

```

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

```
unix% pset mkarf obsfile="acisf00459N002_evt2.fits[EVENTS]"
unix% pset mkarf engrid="grid(3c273.rmf[cols ENERG_LO,ENERG_HI])"
unix% pset mkarf sourcepixelx=4146.05
unix% pset mkarf sourcepixely=4045.95
```

To obtain an accurate ARF at the very edge of a CCD, subarray or window, include the mask file (`msk1.fits`):

```
unix% pset mkarf maskfile="acisf00459_000N002_msk1.fits"
```

If you wish to include the ACIS dead area correction (not applied in this thread), set the `pbkfile` and `dafile` parameters:

```
unix% pset mkarf pbkfile=acisf063875928N002_pbk0.fits dafile=CALDB
```

Run `mkarf` for the source:

```
unix% mkarf
Aspect Histogram File (3c273.asphist[ASPHIST]):
Output File Name (3c273.arf):
Source X Pixel (4146.05):
Source Y Pixel (4045.95):
Energy grid spec (grid(3c273.rmf[cols ENERG_LO,ENERG_HI])):
Name of fits file with obs info (evt file -- include extension) (acisf00459N002_evt2.fits[EVENTS]):
Verbosity (0:5) (0):
Detector Name (ACIS-S3):
Grating for zeroth order ARF (NONE|LETG|HETG) (HETG):
NONE, or name of ACIS window mask file (NONE):
```

and for the background (note that we only need to pset the parameters that have *changed* since the last run):

```
unix% pset mkarf outfile=3c273_bg.arf
unix% pset mkarf engrid="grid(3c273_bg.rmf[cols ENERG_LO,ENERG_HI])"
unix% pset mkarf sourcepixelx=4144.86
unix% pset mkarf sourcepixely=4032.65
unix% mkarf
Aspect Histogram File (3c273.asphist[ASPHIST]):
Output File Name (3c273_bg.arf):
Source X Pixel (4144.86):
Source Y Pixel (4032.65):
Energy grid spec (grid(3c273_bg.rmf[cols ENERG_LO,ENERG_HI])):
Name of fits file with obs info (evt file -- include extension) (acisf00459N002_evt2.fits[EVENTS]):
Verbosity (0:5) (0):
Detector Name (ACIS-S3):
Grating for zeroth order ARF (NONE|LETG|HETG) (HETG):
NONE, or name of ACIS window mask file (NONE):
```

You can check the parameter file that was used with plist mkarf.

Update Spectrum Files

1. Group the Source Spectrum (dmgroup)

The source spectrum is grouped to have a minimum number of 15 counts per new channel:

```
unix% punlearn dmgroup
unix% pset dmgroup infile=3c273.pi
unix% pset dmgroup outfile=3c273_grp.pi
unix% pset dmgroup grouptype=NUM_CTS
unix% pset dmgroup grouptypeval=15
unix% pset dmgroup xcolumn=channel
```

```

unix% pset dmgroup ycolumn=counts
unix% dmgroup
Input dataset name (3c273.pi):
Output dataset name (3c273_grp.pi):
Grouping type (NONE|BIN|SNR|NUM_BINS|NUM_CTS|ADAPTIVE|ADAPTIVE_SNR|BIN_WIDTH|MIN_SLOPE|MAX_SLOPE)
Grouping type value (15):
Binning specification (:):
Name of x-axis (channel):
Name of y-axis (counts):
    
```

You can check the parameter file that was used with [plist dmgroup](#).

2. Group the Background Spectrum (dmgroup)

The background spectrum is binned by a factor of 20:

```

unix% punlearn dmgroup
unix% pset dmgroup infile=3c273_bg.pi
unix% pset dmgroup outfile=3c273_bg_grp.pi
unix% pset dmgroup binspec=1:1024:20
unix% pset dmgroup grouptype=BIN
unix% pset dmgroup xcolumn=channel
unix% pset dmgroup ycolumn=counts
unix% dmgroup
Input dataset name (3c273_bg.pi):
Output dataset name (3c273_bg_grp.pi):
Grouping type (NONE|BIN|SNR|NUM_BINS|NUM_CTS|ADAPTIVE|ADAPTIVE_SNR|BIN_WIDTH|MIN_SLOPE|MAX_SLOPE)
Grouping type value (0):
Binning specification (1:1024:20):
Name of x-axis (channel):
Name of y-axis (counts):
    
```

You can check the parameter file that was used with [plist dmgroup](#).

3. Update File Headers (dmhedit)

Finally, update the appropriate header keywords in the both the ungrouped and grouped source PHA files:

```

unix% punlearn dmhedit
unix% dmhedit infile=3c273.pi filelist="" operation=add key=BACKFILE value=3c273_bg.pi
unix% dmhedit infile=3c273.pi filelist="" operation=add key=RESPFILE value=3c273.rmf
unix% dmhedit infile=3c273.pi filelist="" operation=add key=ANCRFILE value=3c273.arf

unix% dmhedit infile=3c273_grp.pi filelist="" operation=add key=BACKFILE value=3c273_bg.pi
unix% dmhedit infile=3c273_grp.pi filelist="" operation=add key=RESPFILE value=3c273.rmf
unix% dmhedit infile=3c273_grp.pi filelist="" operation=add key=ANCRFILE value=3c273.arf
    
```

in the ungrouped background PHA file:

```

unix% dmhedit infile=3c273_bg.pi filelist="" operation=add key=RESPFILE value=3c273_bg.rmf
unix% dmhedit infile=3c273_bg.pi filelist="" operation=add key=ANCRFILE value=3c273_bg.arf
    
```

and in the linearly-grouped background PHA file:

```

unix% dmhedit infile=3c273_bg_grp.pi filelist="" operation=add key=RESPFILE value=3c273_bg.rmf
unix% dmhedit infile=3c273_bg_grp.pi filelist="" operation=add key=ANCRFILE value=3c273_bg.arf
    
```

Analysis Caveats

Users should be cautious about analyzing the data for sources near the edges of the ACIS CCDs.

1. For X-rays passing through the mirrors, the very bottom of each CCD is obscured by the frame store. As a result, some of the events in rows with CHIPY ≤ 8 are not detected. (The set of rows affected varies from CCD to CCD.) Since the CIAO tools do not compensate for this effect, the ARFs and exposure maps for sources in these regions may be inaccurate.
2. For sources within about thirty-two pixels of any edge of a CCD, the source may be dithered off the CCD during part of an observation. The aspect histogram, which is used to create ARFs and exposure maps, is designed to compensate for this effect.
3. A contaminant has accumulated on the optical-blocking filters of the ACIS detectors, as described in the [ACIS OE Degradation why topic](#). Since there is a gradient in the temperature across the filters (the edges are colder), there is a gradient in the amount of material on the filters. (The contaminant is thicker at the edges.) Within about 100 pixels of the outer edges of the ACIS-I and ACIS-S arrays, the gradient is relatively steep. Therefore, the effective low-energy (~ 1 keV) detection efficiency may vary within the dither pattern in this region. The ARF and instrument map tools are designed to read a calibration file which describes this spatial dependence.

Summary

If you would like to fit the background-subtracted source spectrum using a common RMF and ARF for source and background, simply read the source spectrum FITS file into *Sherpa*, subtract the background, and fit it. See the [Introduction to Fitting PHA Spectra](#) thread for details.

To fit source and background spectra simultaneously with proper and distinct RMFs and ARFs, load the source and background as different datasets. This procedure is discussed in the [Independent Background Responses](#) thread.

Parameters for /home/username/cxcds_param/dmextract.par

```
#-----
#
# DMEXTRACT -- extract columns or counts from an event list
#
#-----
    infile = acisf00459N002_evt2.fits[sky=region(3c273_bg.reg)][bin pi] Input event file
    outfile = 3c273_bg.pi          Enter output file name
    (bkg = )                       Background region file or fixed background (counts/pixel/s) subtracti
    (error = gaussian)             Method for error determination(poisson|gaussian|<variance file>)
    (bkgerror = gaussian)         Method for background error determination(poisson|gaussian|<variance
    (bkgnorm = 1.0)               Background normalization
    (exp = )                      Exposure map image file
    (bkgexp = )                   Background exposure map image file
    (sys_err = 0)                 Fixed systematic error value for SYS_ERR keyword
    (opt = phal)                  Output file type: phal
    (defaults = ${ASCDS_CALIB}/cxo.mdb -> /soft/ciao/datacxo.mdb) Instrument defaults file
    (wmap = )                     WMAP filter/binning (e.g. det=8 or default)
    (clobber = no)                OK to overwrite existing output file(s)?
```

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

(verbose = 0) Verbosity level
(mode = ql)

Parameters for /home/username/cxcds_param/dmstat.par

```
infile = acisf00459N002_evt2.fits[sky=region(3c273.reg)][cols chipx,chipy,ccd_id,x,y] Inp
out_columns = chipx,chipy,ccd_id,x,y Output Column Label
out_min = 258,350,7,4138.965332,4034.1166992 Output Minimum Value
out_min_loc = 224,726,1,64,394 Output Minimum Location Value
out_max = 309,394,7,4157.6948242,4052.0361328 Output Maximum Value
out_max_loc = 258,142,1,484,26 Output Maximum Location Value
out_mean = 280.28940217,373.73097826,7,4146.0548958,4045.9470477 Output Mean Value
out_median =                      Output Median Value
out_sigma = 12.242530849,11.837941816,0,3.5179725102,3.8411838536 Output Sigma Value
out_sum = 206293,275066,5152,3051496.4033,2977817.0271 Output Sum of Values
out_good = 736,736,736,736,736 Output Number Good Values
out_null = 0,0,0,0,0              Output Number Null Values
out_cnvrqd =                      Converged?
out_cntrd_log =                   Output Centroid Log Value
out_cntrd_phys =                  Output Centriod Phys Value
out_sigma_cntrd =                 Output Sigma Centriod Value
  (centroid = yes)                Calculate centroid if image?
  (median = no)                   Calculate median value?
  (sigma = yes)                   Calculate the population standard deviation?
  (clip = no)                     Calculate stats using sigma clipping?
  (nsigma = 3)                    Number of sigma to clip
  (maxiter = 20)                  Maximum number of iterations
  (mode = ql)
```

Parameters for /home/username/cxcds_param/mkacisrmf.par

```
infile = CALDB                    scatter/rsp matrix file
outfile = 3c273_bg_mkacisrmf.rmf RMF output file
wmap = 3c273_bg.pi                WMAP file
energy = 0.1:11.0:0.01           energy grid in keV (lo:hi:bin)
channel = 1:1024:1                channel grids in pixel (min:max:bin)
chantype = PI                     channel type
ccd_id =                          filter CCD-ID
chipx =                           filter chipx in pixel
chipy =                           filter chipy in pixel
gain = CALDB                      gain file
(logfile = )                      log file
(contlvl = 100)                   # contour level
(geompar = geom)                  pixlib geometry parameter file
(thresh = 1e-6)                  low threshold of energy cut-off probability
(clobber = no)                    overwrite existing output file (yes|no)?
(verbose = 0)                     verbosity level (0 = no display)
(mode = ql)
```

Parameters for /home/username/cxcds_param/asphist.par

```
#-----
#
#        Parameter file for the ASPECT HISTOGRAM Tool
#
```

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

```
#-----
infile = @pcad_asol1.lis Aspect Solution List Files
outfile = 3c273.asphist Aspect Histogram Output File
evtfile = acisf00459N002_evt2.fits[ccd_id=7] Event List Files
dtffile = Live Time Correction List Files for HRC
(geompar = geom) Parameter file for Pixlib Geometry files
(res_xy = 0.5) Aspect Resolution x and y in arcsec
(res_roll = 600.) Aspect Resolution roll in arcsec
(max_bin = 10000.) Maximal number of bins
(clobber = no) Clobber output
(verbose = 0) Verbose
(mode = ql)
```

Parameters for /home/username/cxcds_param/mkarf.par

```
asphistfile = 3c273.asphist[ASPHIST] Aspect Histogram File
outfile = 3c273_bg.arf Output File Name
sourcepixelx = 4144.86 Source X Pixel
sourcepixely = 4032.65 Source Y Pixel
engrid = grid(3c273_bg.rmf[cols ENERG_LO,ENERG_HI]) Energy grid spec
obsfile = acisf00459N002_evt2.fits[EVENTS] Name of fits file with obs info (evt file -- include o
detsubsys = ACIS-S3 Detector Name
grating = HETG Grating for zeroth order ARF
maskfile = NONE NONE, or name of ACIS window mask file
verbose = 0 Verbosity
#
(pbkfile = NONE) NONE, or the name of the parameter block file
(dafile = NONE) NONE, CALDB, or name of ACIS dead-area calibration file
#
(mirror = HRMA) Mirror Name
#
(ardlibparfile = ardlib.par) name of ardlib parameter file
(geompar = geom) Parameter file for Pixlib Geometry files
(clobber = no) Overwrite existing files?
#
(mode = ql) Enter mode for parameter file.
```

Parameters for /home/username/cxcds_param/acis_fef_lookup.par

```
infile = acisf00459N002_evt2.fits Source file (event or spectrum)
chipid = 7 ACIS chip number
chipx = 290 ACIS chip x coordinate
chipy = 379 ACIS chip y coordinate
(outfile = /soft/ciao/CALDB/data/chandra/acis/cpf/feFs/
acisD1999-09-16fef_phaN0002.fits[FUNCTION][ccd_id=7,chipx=289:320,chipy=353:384]) FEF file to use
(verbose = 0) Verbose level
(mode = ql)
```

Parameters for /home/username/cxcds_param/mkrmf.par

```
infile = /soft/ciao/CALDB/data/chandra/acis/cpf/feFs/
acisD1999-09-16fef_phaN0002.fits[FUNCTION][ccd_id=7,chipx=289:320,chipy=353:384] name of FEF input file
outfile = 3c273_bg.rmf name of RMF output file
axis1 = energy=0.1:11.0:0.01 axis-1(name=lo:hi:btype)
axis2 = pi=1:1024:1 axis-2(name=lo:hi:btype)
```

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

```
(logfile = STDOUT)      name of log file
(weights = )            name of weight file
(thresh = 1e-5)        low threshold of energy cut-off probability
(outfmt = legacy)      RMF output format (legacy|cxc)
(clobber = no)         overwrite existing output file (yes|no)?
(verbose = 0)          verbosity level (0 = no display)
(axis3 = none)         axis-3(name=lo:hi:btype)
(axis4 = none)         axis-4(name=lo:hi:btype)
(axis5 = none)         axis-5(name=lo:hi:btype)
(mode = ql)
```

Parameters for /home/username/cxcds_param/dmgroup.par

```
infile = 3c273.pi       Input dataset name
outfile = 3c273_grp.pi  Output dataset name
grouptype = NUM_CTS     Grouping type
grouptypeval = 15      Grouping type value
binspec =              Binning specification
xcolumn = channel      Name of x-axis
ycolumn = counts       Name of y-axis
(tabspec = )          Tab specification
(tabcolumn = )        Name of tab column
(stopspec = )         Stop specification
(stopcolumn = )       Name of stop column
(errcolumn = )        Name of error column
(clobber = no)        Clobber existing output file?
(verbose = 0)         Verbosity level
(maxlength = 0)       Maximum size of groups (in channels)
(mode = ql)
```

Parameters for /home/username/cxcds_param/dmgroup.par

```
infile = 3c273_bg.pi   Input dataset name
outfile = 3c273_bg_grp.pi Output dataset name
grouptype = BIN        Grouping type
grouptypeval = 0      Grouping type value
binspec = 1:1024:20   Binning specification
xcolumn = channel     Name of x-axis
ycolumn = counts      Name of y-axis
(tabspec = )         Tab specification
(tabcolumn = )       Name of tab column
(stopspec = )       Stop specification
(stopcolumn = )     Name of stop column
(errcolumn = )      Name of error column
(clobber = no)     Clobber existing output file?
(verbose = 0)     Verbosity level
(maxlength = 0)   Maximum size of groups (in channels)
(mode = ql)
```

History

14 Dec 2004 updated for CIAO 3.2: created [Using Consistent Calibration](#) and [Downloading acis_fef_lookup](#) sections

23 Jun 2005 CIAO 3.2.2 patch: new calibration for mkacisrmf is available (see the [Using Consistent](#)

Make ACIS Point-like Spectra/RMFs/ARFs (Manually) – CIAO 3.4

Calibration: mkrmf vs mkacisrmf section); change to `asphist` parameter file

- 15 Dec 2005 updated for CIAO 3.3: Calculate the RMFs section has been updated to show syntax for both `mkrmf` and `mkacisrmf`; default value of `dmextract` error and `bkgerror` parameters is "gaussian"; updated syntax for `asphist` (GTI filter is associated with the event file rather than the aspect solution); parameter file changes (`dmextract`, `dmstat`)
- 01 Feb 2006 added link to [specextract thread](#)
- 05 Apr 2006 changed `specextract` thread link to `psextract` thread link
- 14 Jun 2006 corrected link in "Calibration Updates"; clarified information on GRADED mode data
- 01 Dec 2006 updated for CIAO 3.4: `acis_fef_lookup` version 1.19 (changes for CTI lookup); parameter file updates for `mkarf` (`obsfile` parameter is set to event file instead of aspect histogram file)
- 02 Feb 2007 updated for [CALDB 3.3.0.1 patch](#)
- 26 Feb 2007 `acis_fef_lookup` v1.20: Script issues a warning to use `mkacisrmf` instead of `mkrmf` when running with CTI-corrected data on certain chips.
- 06 Mar 2007 added [ACIS dead area correction section](#) and example of setting the `pbkfile` and `dafile` parameters
- 11 Apr 2007 fixed typo in `mkarf` parameter file listing: as of CIAO 3.4, `obsfile` parameter takes the event file as input instead of redirecting to the `asphistfile` value.
- 31 Mar 2008 updated for CALDB 3.4.3: use `mkacisrmf` for -110 BI chips if TGAIN calibration has been applied

URL: <http://cxc.harvard.edu/ciao/threads/pieces/>

Last modified: 31 March 2008

Image 1: Extraction regions overlaid on the image

