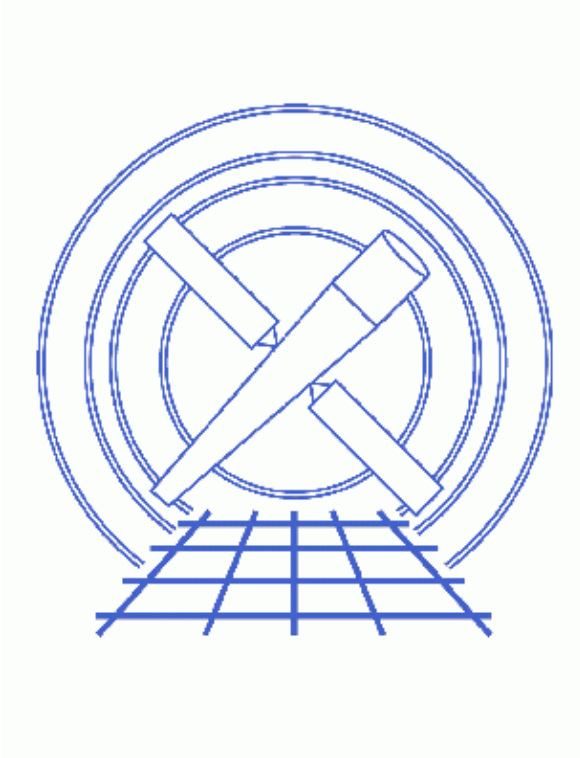
Creating ACIS RMFs with mkacisrmf



CIAO 3.4 Science Threads

Table of Contents

- ***Get Started***
  - ♦ ACIS GRADED Mode Data
  - ♦ ACIS −110 C Data
- ***Using mkacisrmf***
  - A. Creating an RMF at a specific location
  - B. Creating an RMF to match an extracted spectrum
  - C. Using mkacisrmf with the specextract script
  - D. Using mkacisrmf with the psextract script
- ***Update the Spectrum File Header***
- ***Caveat: Setting the channel type***
- ***Caveat: Matching the ARF and RMF energy grids***
  - ♦ Match an existing ARF
  - ♦ Create the RMF first
- ***History***

Creating ACIS RMFs with mkacisrmf

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, see −110 C Data section

Synopsis:

The tool `mkacisrmf`, available since CIAO 3.2, represents an entirely new method for creating ACIS imaging response matrices. This tool contains all the functionality of the previous tool `mkrmf`. Unlike its predecessor, however, `mkacisrmf` separates the RMF calculation process into two components: an "ideal" component which describes the CCD spectral response prior to the effects of CTI, and a spatially varying component which incorporates the changes in the response produced by CTI.

Technical details on `mkacisrmf` are available from the Creating ACIS RMFs why topic.

Purpose:

To create an RMF for an ACIS imaging observation (or zeroth−order grating) with the newest calibration available. The Using mkacisrmf with the specextract script section shows how to run `mkacisrmf` on the output of `specextract`, when necessary.

Read this thread if:

- you are working with −120 C ACIS imaging data or the zero−order of a grating observation taken in (V)FAINT mode. `mkacisrmf` is fully calibrated for all ten ACIS chips in these modes.
- you are working with −120 C ACIS imaging data or the zero−order of a grating observation taken in GRADED mode on one of the back−illuminated chips (ACIS−S1 and S3). The ACIS GRADED Mode Data section has more information.
- you are working with −110 C ACIS imaging data or the zero−order of a grating observation taken on one of the back−illuminated chips (ACIS−S1 and S3). The −110 C Data section has more information.

Get Started shows how to check the mode of your observation.

Calibration Updates:

- ***CALDB v3.4.0*** ***(16 May 2007):*** The P2_RESP lookup was updated so that `mkacisrmf` can select the correct calibration file based on the information in the WMAP file header for `GRADED` mode data. The ACIS GRADED Mode Data section has more information
- ***CALDB v3.3.0*** ***(18 Dec 2006):*** The version 6 "phase 2" response file (`acisD2000-01-29p2_respN0006.fits`) was added to CALDB 3.3.0. Only calibration for the BI chips (S1, S3) has changed in this file; calibration for the FI chips is identical to the v5 file. The How CIAO 3.4 and CALDB 3.3.0 Affect Your Analysis section of the CIAO release notes explains how the file will affect your analysis.

- ***CALDB v3.2.1*** ***(15 Dec 2005):*** The gain for the back−illuminated (BI) chips − ACIS−S1 and S3 − for −120 GRADED mode observations has been upgraded. It is now possible to use mkacisrmf when creating responses for these chips only in GRADED mode. See How CALDB 3.2.1 Affects Your Analysis for details.
- ***CALDB v3.2.0*** ***(21 Nov 2005):*** The version 5 "phase 2" response file (acisD2000−01−29p2_respN0005.fits) was added to CALDB 3.2.0, released on 21 November 2005. Information on the changes in CALDB 3.2.0 is in the How CIAO 3.3 and CALDB 3.2.0 Affect Your Analysis section of the CIAO release notes.
- ***CALDB v3.1.0*** ***(23 Jun 2005):*** The version 4 "phase 2" response file (acisD2000−01−29p2_respN0004.fits) was added to the CALDB. In conjunction with the ACIS CTI and TGAIN calibration improvements in that release, ***the mkacisrmf tool can now be used to create RMFs for all −120 observations***, regardless of detector configuration.
- ***CALDB v3.0.3*** ***(9 May 2005):*** A repaired calibration file, acisD2000−01−29p2_respN0003.fits, was released to correct sorting errors in the version 2 file.
- ***CALDB v3.0.0*** ***(15 Dec 2004):*** The calibration files necessary to run mkacisrmf were released, including the new gain file (acisD2000−01−29gain_ctiN0003.fits).

Related Links:

- mkacisrmf bugs page: a list of all known issues with this tool.
- Why topic: Creating ACIS RMFs
- Updates to the RMF model in the ACIS FI CCDs (PDF, 5 pages)
- the mkacisrmf help file

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

Get Started

Although mkacisrmf does not require an observation−specific data file as input, the responses it creates are intended for use with −120 C data that has the time−dependent gain adjustment and CTI correction (if available) applied.

The responses that mkacisrmf creates are intended for use with data that has the time−dependent gain adjustment and CTI correction applied.

Make sure that your data are taken are the proper focal plane temperature (any chip at −120 C, BI chips at −110 C) and that an acceptable gain has been applied:

```
unix% dmkeypar evt2.fits fp_temp echo+
153.60722351

unix% dmkeypar evt2.fits gainfile echo+
/soft/ciao/CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0006.fits
```

Any CTI−corrected gain file since version 4 (acisD2000−01−29gain_ctiN0004.fits) is good enough for use with this tool; ***read the Creating ACIS RMFs why topic for information on using consistent calibration.***

Check the mode of the observation, which is recorded in the DATAMODE header keyword:

```
unix% dmkeypar evt2.fits DATAMODE echo+
```

```
FAINT
```

`mkacisrmf` can be used on any observation taken in `(V)FAINT` mode, including continuous−clocking `CC(33)_FAINT` mode. The tool can also be used on GRADED mode data on the BI chips.

ACIS GRADED Mode Data

There is limited use for `mkacisrmf` when working with data taken in `GRADED` mode. The data must have the time−dependent gain adjustment applied; there is no CTI correction for GRADED mode.

```
unix% dmkeypar acis_evt1.fits DATAMODE echo+
GRADED
```

New gain files were released in CIAO 3.2.1 (15 December 2005) which make it possible to use `mkacisrmf` when creating responses for observations taken in GRADED mode on the ***back−illuminated chips (ACIS−S1 and S3) only***.

Responses for the ***front−illuminated chips*** in GRADED mode data should still be created by running `mkrmf`.

Running mkacisrmf

In CALDB 3.4.0 (16 May 2007), the P2_RESP lookup was updated so that `mkacisrmf` can select the correct calibration file based on the information in the WMAP file header. This change means that users can run `mkacisrmf` for the extracted spectrum case as well as the specific location case, depending on which best suits the data analysis.

Running specextract

Since calibration is only available for the two chips in GRADED mode, the `specextract` script will use `mkrmf` to create the RMF response files. If your source is on ACIS−S1 or S3, follow the instructions in the Using mkacisrmf with the specextract script section to create a new response file with `mkacisrmf`.

ACIS −110 C Data

There is limited use for `mkacisrmf` when working with data taken at the −110 C focal plane temperature. The data must have the time−dependent gain adjustment applied; the TGAIN calibration for −110 C was first released in CALDB 3.4.3. There is no CTI correction for this temperature.

```
unix% dmkeypar acis_evt1.fits FP_TEMP echo+
163.952042
```

It is appropriate to use `mkacisrmf` for observations taken at −110 C with the source on a ***back−illuminated chip (ACIS−S1 or S3) only***.

For these chips, run `mkacisrmf` as shown in the extracted spectrum case or the specific location case, depending on which best suits the data analysis. However, in either case, the `infile` parameter must be set to the ***version N0005 P2_RESP file***:

```
unix% pset mkacisrmf \
      infile=$CALDB/data/chandra/acis/cpf/p2_resp/acisD2000−01−29p2_respN0005.fits
```

Responses for the ***front−illuminated chips*** at −110 C should still be created by running `mkrmf`.

Running specextract

Since calibration is only available for the two chips, `specextract` script will use `mkrmf` to create the RMF response files. If your source is on ACIS−S1 or S3, follow the instructions in the Using mkacisrmf with the

specextract script section to create a new RMF file with `mkacisrmf`.

Using mkacisrmf

A. Creating an RMF at a specific location

For this case, it is assumed the user wishes to create an RMF file for a specific location on one of the ACIS CCDs, independent of a given dataset or extraction region. In such a situation, the user must explicitly supply the value of the `infile` and `gain` parameters, since the tool does not have access to any header information to determine which calibration files should be used.

First the `dmkeypar` tool is used to make sure that the same gain file used in the data processing is provided to `mkacisrmf`:

```
unix% dmkeypar evt2.fits gainfile echo+
/soft/ciao/CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0006.fits
```

This event file has been reprocessed with the version 6 gain file; refer back to the Get Started section for an explanation of why this is important. If the gain file was applied in standard data processing at the CXC, the path will need to be altered for the `mkacisrmf` command to point to your local CALDB installation.

Choose the `mkacisrmf` calibration file (called a "P2_RESP" file). The P2_RESP and gain files are released in pairs of matching versions, so simply choose the file which has the same "N000x" version as the gain:

```
unix% ls −1 $CALDB/data/chandra/acis/cpf/p2_resp/
acisD2000−01−29p2_respN0002.fits
acisD2000−01−29p2_respN0003.fits
acisD2000−01−29p2_respN0004.fits
acisD2000−01−29p2_respN0005.fits
acisD2000−01−29p2_respN0006.fits
```

For this gain, acisD2000−01−29gain_cti***N0006***.fits, we should use the acisD2000−01−29p2_resp***N0006***.fits file. This is the value for the `infile` parameter.

Now we can run `mkacisrmf`. Note that when specifying the input response and gain files, the path must be included.

```
unix% mkacisrmf \
      infile=$CALDB/data/chandra/acis/cpf/p2_resp/acisD2000−01−29p2_respN0006.fits \
      outfile=location_rmf.fits \
      wmap=none \
      energy=0.3:10.0:0.005 \
      channel=1:1024:1 \
      chantype=PI \
      ccd_id=7 \
      chipx=645 \
      chipy=10 \
      gain=$CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0006.fits
```

This command creates an RMF at a position of (645, 10) in CHIP coordinates on CCD 7.

The corresponding `mkrmf` syntax is:

```
unix% mkrmf \
      infile="$CALDB/data/chandra/acis/cpf/fefs/acisD1999−09−16fef_phaN0002.fits[FUNCTION][ccd_id=7,chi
      outfile=3c273.rmf axis1="energy=0.1:11.0:0.01" axis2="pi=1:1024:1"
```

B. Creating an RMF to match an extracted spectrum

This method is the recommended way of constructing RMFs with `mkacisrmf`.

In this scenario, the user has extracted a spectrum from a given region using `dmextract` and wishes to compute a matching RMF. For example:

```
unix% dmextract \
      "data_evt2.fits[(x,y)=circle(4232,3289,30)][bin pi=1:1024:1]" \
      test_src_pi.fits  wmap="det=8"

unix% mkacisrmf infile=CALDB \
      outfile=test_wrmf.fits \
      energy=0.3:9.5:0.005 \
      channel=1:1024:1 \
      chantype=PI \
      wmap=test_src_pi.fits \
      gain=CALDB
```

The energy range is chosen as `0.3:9.5:0.005` in order to avoid an error from `mkwarf` later in the analysis; see the `mkwarf` bugs page for details.

These commands extract a spectrum from within the indicated region and create a weighted RMF appropriate for that region. Here, 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. The `infile` parameter is set to "`CALDB`" and `mkacisrmf` uses the information in the header of the input wmap file to determine the P2_RESP calibration file to use.

Similarly, the gain parameter should be set to "`CALDB`" in order to ensure that the gain file which matches the P2_RESP file is used. By using the CALDB value for the `infile` parameter, `mkacisrmf` will automatically select the appropriate file.

If you prefer, it is always acceptable to give a full path to the gain file in the CALDB, e.g.

```
unix% pset mkacisrmf \
      gain=$CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0006.fits
```

The corresponding `mkrmf` syntax is:

```
unix% mkrmf infile=CALDB outfile=sources.wrmf \
      axis1="energy=0:1" axis2="pi=1:1024:1" weights=sources.wgt
```

C. Using mkacisrmf with the specextract script

`specextract`, a new script for creating ACIS spectra, was released in CIAO 3.3 (21 November 2005). The major feature is the script's ability to determine when `mkacisrmf` should be used in place of `mkrmf`.

`specextract` reads the source and background file header keywords to determine when the observation was taken and what calibration has been applied. The script then does a CALDB lookup to compare the calibration applied to the event file and the most recent file available in the CALDB. This means that the data needs to have been processed with the very newest calibration, not just a "good enough" version; that is, even if the new calibration will have a minimal effect on your data, `specextract` requires that it be applied for the "use `mkacisrmf`" switch to be triggered.

Users who don't wish to reprocess can still use `specextract` and run `mkacisrmf` afterwards to create a new RMF file. This approach is also appropriate for users who have run `specextract` on GRADED mode data or −110 C data and want to create an RMF for a source on ACIS−S1 or S3.

```
unix% dmkeypar acis_evt2.fits GAINFILE echo+
acisD2000−01−29gain_ctiN0005.fits
```

Since this file has an older CTI−corrected gain file applied, `specextract` used `mkrmf` to create the RMFs. The version 4 calibration is sufficient for `mkacisrmf`, though, so we run the tool independently of the script to create a new response file.

Choose the `mkacisrmf` calibration file (called a "P2_RESP" file). The P2_RESP and gain files are released in pairs of matching versions, so simply choose the file which has the same "N000x" version as the gain:

```
unix% ls −1 $CALDB/data/chandra/acis/cpf/p2_resp/
acisD2000−01−29p2_respN0002.fits
acisD2000−01−29p2_respN0003.fits
acisD2000−01−29p2_respN0004.fits
acisD2000−01−29p2_respN0005.fits
acisD2000−01−29p2_respN0006.fits
```

For this gain, acisD2000−01−29gain_cti***N0005***.fits, we should use the acisD2000−01−29p2_resp***N0005***.fits file. This is the value for the `infile` parameter.

Please read the "Matching the ARF and RMF energy grids" caveat before continuing. That information applies to users who intend to use XSpec instead of *Sherpa* to model and fit the data.

Now we can run the tool:

```
unix% mkacisrmf \
      infile="$CALDB/data/chandra/acis/cpf/p2_resp/acisD2000−01−29p2_respN0005.fits" \
      outfile=acis_new_rmf.fits \
      energy=0.3:10.0:0.005 \
      channel=1:1024:1 \
      chantype=PI \
      wmap=acis_src1.pi \
      gain=$CALDB/data/chandra/acis/bcf/gain/acisD2000−01−29gain_ctiN0005.fits
```

The `wmap` parameter is set to the source spectrum file created by `specextract`. The script always creates a `WMAP` block, so this approach will work with any `specextract` output. The new RMF file is name `acis_new_rmf.fits`.

Be sure to update the spectrum file header to contain the name of the new RMF file.

D. Using mkacisrmf with the psextract script

The `psextract` script uses the `mkrmf` tool to create the RMF. Users who have the correct calibration applied to the data may run `mkacisrmf` independently to create a new RMF.

```
unix% dmkeypar acis_point_evt2.fits GAINFILE echo+
acisD2000−01−29gain_ctiN0005.fits
```

The file has an appropriate CTI−corrected gain applied to it, so we can use `mkacisrmf`. For the point source case, `mkacisrmf` is run in "Creating an RMF at a specific location" mode.

Choose the `mkacisrmf` calibration file (called a "P2_RESP" file). The P2_RESP and gain files are released in pairs of matching versions, so simply choose the file which has the same "N000x" version as the gain:

```
unix% ls −1 $CALDB/data/chandra/acis/cpf/p2_resp/
acisD2000−01−29p2_respN0002.fits
acisD2000−01−29p2_respN0003.fits
acisD2000−01−29p2_respN0004.fits
acisD2000−01−29p2_respN0005.fits
acisD2000−01−29p2_respN0006.fits
```

For this gain, acisD2000−01−29gain_cti***N0005***.fits, we should use the acisD2000−01−29p2_resp***N0005***.fits file. This is the value for the infile parameter.

To get the necessary coordinate information (ccd_id, chipx, chipy), convert the source region information to chip coordinates with dmcoords:

```
unix% pset dmcoords \
      asolfile="pcadf063874624N002_asol1.fits,pcadf063875522N002_asol1.fits,pcadf063902942N002_as
unix% dmcoords acisf00459N002_evt2.fits
dmcoords>: sky 4148.125 4043.625
(RA,Dec):     12:29:05.953    +02:02:55.94
(RA,Dec):      187.27481        2.04887 deg
THETA,PHI           36.4"          22.24 deg
(Logical):        4148.12       4043.62
SKY(X,Y):         4148.12       4043.62
DETX,DETY         4164.90       4124.47
CHIP ACIS-S3       286.57        370.93
TDET              4203.57       2072.93
...
```

Now we have all the information to run the tool:

```
unix% mkacisrmf \
      infile="$CALDB/data/chandra/acis/cpf/p2_resp/acisD2000-01-29p2_respN0005.fits" \
      outfile=acis_point_rmf.fits \
      energy=0.3:10.0:0.005 \
      channel=1:1024:1 \
      chantype=PI \
      ccd_id=7 chipx=286.57 chipy=370.93 \
      gain=$CALDB/data/chandra/acis/bcf/gain/acisD2000-01-29gain_ctiN0005.fits
```

The new RMF file is name acis_new_rmf.fits.

Making a new ARF:

Please read the "Matching the ARF and RMF energy grids" caveat before continuing. That information applies to users who intend to use XSpec instead of *Sherpa* to model and fit the data.

psextract users who intend to finish the data analysis in XSpec should now remake the ARF file to ensure that the grid matches the RMF. To do so, get the mkarf command from the history in the ARF file. Change the engrid parameter to use the new RMF file, and rerun mkarf.

```
unix% dmhistory 3c273.arf tool=mkarf
mkarf asphistfile="3c273.asphist" outfile="3c273.arf" sourcepixelx="4145.9587485" sourcepixely="4
engrid="grid(3c273.rmf[MATRIX][cols ENERG_LO,ENERG_HI])" obsfile="acis_dstrk_evt2.fits" pbkfile="
dafile="NONE" mirror="HRMA" detsubsys="ACIS-S3" grating="HETG" maskfile="NONE"
ardlibparfile="ardlib.par" geompar="geom" verbose="0" clobber="no"

unix% dmhistory 3c273.arf tool=mkarf action=pset

unix% pset mkarf outfile="3c273_new.arf"
unix% pset mkarf engrid="grid(acis_point_rmf.fits[MATRIX][cols ENERG_LO,ENERG_HI])"

unix% mkarf
```

Be sure to update the spectrum file header to contain the name of the new RMF (RESPFILE keyword) and ARF (ANCRFILE keyword).

Update the Spectrum File Header

Update the RESPFILE keyword in the header of the source (and background) spectrum file with the name of the new RMF. For example:

```
unix% dmhedit infile=acis_src1.pi filelist="" operation=add \
      key=RESPFILE value=acis_new_rmf.fits
```

If the spectrum was created with one of the CIAO scripts (e.g. specextract or acisspec), the header keyword will contain the name of the file created with mkrmf. The wrong file could be selected during fitting if the keyword is not updated.

Caveat: Setting the channel type

All the examples in this thread use the default chantype value of "PI". If you extracted a spectrum in PHA space instead, you need to change this parameter setting, as well as the channel value:

```
unix% pset mkacisrmf chantype=pha channel=1:4096:2
```

As explained in the dmextract help file, the default binning used for PHA files is "1:4096:2".

Caveat: Matching the ARF and RMF energy grids

Sherpa allows you to use different energy grids for your ARF and RMF files, but *XSpec* does not. *XSpec* will still run if the grids do not match, but it issues a warning and sets all values in the ARF to unity (1).

There are two approaches to creating an ARF−RMF pair with the same gridding: Match an existing ARF or Create the RMF first.

Match an existing ARF

If the specextract or acisspec scripts were used, you already have an ARF file for the data. Rather than remake both the RMF and ARF, get the grid information from the history in the ARF file:

```
unix% dmhistory acis_src1.warf tool=all
# dmhistory (CIAO 3.4): WARNING: Found "pixlib" library parameters

# dmhistory (CIAO 3.4): WARNING: Found "ardlib" library parameters

mkwarf infile="acis_src1.[WMAP]" outfile="acis_src1.warf" weightfile="acis_src1.wfef" spectrumfile=""
egridspec="0.3:9.5:0.01" threshold="0" feffile="CALDB" mskfile="" mirror="HRMA"
detsubsysmod="" ardlibpar="ardlib" geompar="geom" clobber="no" verbose="2"
```

Use the egridspec value as input for the energy parameter in mkacisrmf:

```
unix% pset mkacisrmf energy="0.3:9.5:0.01"
```

This method cannot be used with the ARFs created by psextract. That script creates the RMF first and uses it to define the ARF gridding, so the engrid parameter value looks like "grid(3c273.rmf[MATRIX][cols ENERG_LO,ENERG_HI])". The Using mkacisrmf with the psextract script section has special instructions for those users.

Create the RMF first

Since `mkacisrmf` can change the requested grid to match the calibration data, create the RMF first and then use it to define the energy grid when creating the ARF. This will work for both `mkarf` and `mkwarf`:

```
unix% pset mkarf \
      engrid="grid(sources_ciao.wrmf[cols ENERG_LO,ENERG_HI])"

or

unix% pset mkwarf \
      egridspec="grid(sources_ciao.wrmf[cols ENERG_LO,ENERG_HI])"
```

History

15 Dec 2004	original version, new for CIAO 3.2
12 Jan 2005	created Matching the number of energy bins
28 Feb 2005	updated `chipx/y` values in Using mkacisrmf Instead of mkrmf: Example 1; expanded information in Matching the number of energy bins section
12 Apr 2005	clarification on how gain is specified in Using mkacisrmf Instead of mkrmf section
09 May 2005	a corrected calibration file has been released in CALDB 3.0.3. All users should download the patch to obtain the updated calibration.
23 Jun 2005	CIAO 3.2.2 patch: new calibration file in CALDB 3.1.0; the Using mkacisrmf section was rewritten to be more descriptive
01 Aug 2005	changed energy range to `energy=0.3:9.5:0.005` in Creating an RMF to match an extracted spectrum due to an issue with `mkwarf`
15 Dec 2005	updated for CIAO 3.3: new calibration files in CALDB 3.2.0; added ACIS GRADED Mode Data section
15 Feb 2006	created Using mkacisrmf with the specextract script section
14 Jun 2006	corrected link in "Calibration Updates"; clarified information on `GRADED` mode data; added information on choosing a P2_RESP file for a given gain
25 Jul 2006	created Using mkacisrmf with the psextract script section
25 Aug 2006	showed how to find ARF energy grid in Caveat: Matching the ARF and RMF energy grids section (renamed from "Matching the number of energy bins"). In the case where one of the CIAO scripts was used (e.g. `psextract`), it is not necessary to remake both the ARF and RMF in order to have the grids match.
18 Dec 2006	updated for CIAO 3.4: new calibration files in CALDB 3.3.0; CIAO version in warnings
30 Mar 2007	changed how the tool is run in the Using mkacisrmf with the psextract script section; added Caveat: Setting the channel type
15 May 2007	updated for CALDB 3.4.0: CALDB lookup works for ACIS GRADED mode, so users can run `mkacisrmf` for the extracted spectrum case as well as the specific location case
31 Mar 2008	updated for CALDB 3.4.3: use `mkacisrmf` for −110 BI chips if TGAIN calibration has been applied, see −110 C Data section