

*AHELP for CIAO 3.4*

## mkwarf

Context: [tools](#)

[Jump to: Description Examples Parameters CHANGES IN CIAO 3.4 CHANGES IN CIAO 3.2 CHANGES IN CIAO 3.0 DESCRIPTION OF THE ALGORITHM Notes Bugs See Also](#)

## Synopsis

Generate a weighted ARF for Chandra ACIS data

## Syntax

```
mkwarf infile outfile weightfile spectrumfile egridspec [threshold]
[feffile] [mskfile] [asolfile] [mirror] [detsubsysmod] [pbkfile]
[dafile] [ardlibpar] [geompar] [clobber] [verbose]
```

## Description

`mkwarf' creates a weighted ARF file based on an input weight map (WMAP). It also produces a set of weights that are used by `mkrmf' to create the associated weighted RMF. The use of a weighted response may allow one RMF and ARF to be used to analyze a spectrum extracted from a large area that covers multiple FEF regions (and hence a range of spectral responses). Similarly it can be used to create a single response for analysing a "composite" spectrum created from multiple point sources scattered across the field of view.

## Assumptions and Caveats

Please note that the algorithm used by mkwarf and mkrmf for calculating the weighted responses assumes that there is no spatial variation in the source spectrum across the extraction region(s) used. In addition, it should only be used to create "PI" responses – ie for the analysis of spectra binned over the PI column and not the PHA column.

## The Weight Map

The WMAP can be any 2–D image that has been binned in detector coordinates (ie the DET column in a Chandra event list) and contains the necessary header–keywords (including the instrument configuration, observation date, and SIM position). Users can use `dmextract' to create such a map when creating a spectrum (i.e. opt=phal, bin on the PI column, and use the wmap option), or can use a normal image file. The WMAP is not restricted to integer pixel values, so smoothed or simulated images can also be used, as long as they are in detector coordinates. The tool will only work correctly if the image is binned in detector coordinates (and not, for instance, the chip coordinate system).

We recommend that the WMAP be created using a binning factor of 8 (see "The algorithm" section below for details) – so set the wmap parameter of dmextract to "det=8" or use the Data Model binning syntax "[bin det=8]" with dmcoppy.

## How to weight the WMAP

Ideally, the response function for a given region should be weighted by the fraction of the incident flux from the source that falls within that region. However, this is generally what we are looking to obtain from the observation! We instead know the distribution of counts (i.e. the incident flux after it has passed through the telescope mirror and been absorbed by the detector) across the regions. Below we discuss two possible approaches to the problem; the choice of which – if either – is better depends on the characteristics of the data and the science objectives of the analysis.

### a. spectrumfile=none

The simplest approach is to use the raw counts to weight the response (in which the spectrumfile parameter should be set to none). This means that the weights will include the effect of the mirror response and detector QE (including bad pixels) – ie the effective area term – which may bias the spectral fits. One way to minimise this is to pick a restricted energy range over which to make the weight map, one over which the effective area (or, more correctly, the product of the source spectrum and the effective area) does not vary strongly.

### b. using the spectrumfile parameter

Another approach is to use the spectrum of the source to correct the detected counts for the effective area of the telescope/instrument combination. Of course, we don't know the source spectrum to use to correct the model. One approach is therefore to use an iterative scheme, where the spectral model used to weight the WMAP is updated after each fit. The spectrumfile parameter is used to supply the source model to use in weighting the WMAP. The file should be in ASCII, with a format that matches the output of Sherpa's "write source" command (see the description of the spectrumfile parameter below for more details). Note that the energy range of the spectrum should be adjusted to match that used to create the WMAP.

Examples of creating weighted ARFs can be found in the [CIAO science threads](#).

## Example 1

```
mkwarf "spectrum1.fits[wmap]" spectrum1.warf spectrum1.wgt none
0.5:9.5:0.02
```

This will create a weighted ARF, called spectrum1.warf, over the energy grid of 0.5 to 9.5 keV in 20 eV bins. The weighting uses the pixel values in the WMAP extension of the input PHA file (spectrum1.fits), with no spectral weighting.

The weights file created by mkwarf would then be used in mkrmf by setting:

```
unix% pset mkrmf weights=spectrum1.wgt
unix% pset mkrmf infile=CALDB
unix% pset mkrmf axis1="energy=0:1"
```

where the infile parameter is set to CALDB since we do not need to use acis\_fef\_lookup in this case, and the

value for the energy grid is irrelevant (as long as something is supplied) since it's overridden by the energy grid in the weights file (here spectrum1.wgt).

## Example 2

```
mkwarf "spectrum2.fits[wmap]" spectrum2.warf spectrum2.wgt none
0.5:9.5:0.02 threshold=0.01
```

As with example 1, but this time only those FEF regions which contain at least 1 percent of the total counts are included in the calculation of the weighted ARF. This option can be used to speed up mkrmf by removing the need to calculate RMFs for regions which do not significantly contribute to the response. The choice of threshold depends on both the distribution of counts across the WMAP and the science objectives of your analysis.

## Example 3

```
mkwarf "sources.pi[WMAP]" sources.warf sources.wgt
egridspec=0.01:11:0.01 pbkfile=acisf063875928N002_pbk0.fits dafile=CALDB
```

Create a weighted ARF over the energy grid of 0.01 to 11 keV in 0.01 eV bins. The pbkfile and dafile parameters are set to apply the dead area correction.

## Example 4

```
mkwarf csmooth_out.fits spectrum3.warf spectrum3.wgt spec.txt
"grid(spectrum1.rmf[MATRIX][cols ENERG_LO,ENERG_HI])"
```

This will use the (supposedly) smoothed image in 'csmooth\_out.fits' as the WMAP with the spectral weights in the ASCII file 'spec.txt' to create the ARF and RMF weights on an energy grid that matches that of the RMF stored in spectrum1.rmf.

## Parameters

name	type	ftype	def	min	max	reqd
<u>infile</u>	file	input				yes
<u>outfile</u>	file	output				yes
<u>weightfile</u>	file	output				yes
<u>spectrumfile</u>	file	input				yes
<u>egridspec</u>	string					yes
<u>threshold</u>	float		0	0	1	
<u>feffile</u>	file	ARD	CALDB			
<u>mskfile</u>	file					
<u>asolfile</u>	file					
<u>mirror</u>	string		HRMA			
<u>detsubsysmod</u>	string					

<u>pbkfile</u>	string	input				
<u>dafile</u>	string	input				
<u>ardlibpar</u>	file		ardlib			
<u>geompar</u>	file		geom			
<u>clobber</u>	boolean		no			
<u>verbose</u>	integer		0	0	5	

## Detailed Parameter Descriptions

### Parameter=infile (file required filetype=input)

*Input WMAP file name. Be sure to include the DM filter "[WMAP]" if you are using a WMAP attached to a PHA file.*

The input WMAP parameter name. The WMAP should be binned in DET coordinates and include the SIM positions in its header. If the data has been CTI corrected then the data must also contain the CTI\_CORR keyword with a value of TRUE (CIAO tools will retain this information in the output file if the original event file contained the keyword, although it may be lost if non-CIAO tools have created/manipulated the image).

### Parameter=outfile (file required filetype=output)

*The output file name for the weighted ARF file.*

The weighted ARF created by mkwarf is OGIP compliant.

### Parameter=weightfile (file required filetype=output)

*The output file name for the weights file.*

This file is needed by mkrmf to create the weighted RMF (the weights parameter). It contains a list of FEF regions used to create the weighted ARF, along with the weight used for each region.

### Parameter=spectrumfile (file required filetype=input)

*Input spectral weights file*

If the raw counts in the input weight map are to be used to weight the spectral response then set this parameter to "none" (or left blank). Otherwise this parameter should be set to the name of an ASCII file containing columns for energy (in keV) and a weight value (such as the spectral model in photon/cm<sup>2</sup>/s/keV). This file can be produced by Sherpa's "write source" command as shown below.

The energy range of the spectrum should be adjusted to match that used to create the WMAP. Note that the weights column is normalized before being used (unlike the spectrumfile parameter for mkinstmap).

**How to create a weights file in Sherpa**

In the following fragment of a Sherpa session we create an ASCII file (wgt.dat) that can be used as a weight file for a WMAP created over the range of 0.3 to 8.0 keV. For this example we use a very simple spectral model (a single power law with a photon index of 1.8 and no Galactic absorption); the value for the normalisation of the model is unimportant (as long as the values are not so small that they appear as 0.0 in the output file).

The lines beginning with a '#' are comments to help indicate the steps taken. Please see the [Sherpa threads](#) and ahelp files ("ahelp sherpa") for more information on how to use Sherpa.

```
# set the source model (a single powerlaw)
sherpa> source = xspowerlaw[po]
po.PhoIndx parameter value [1] 1.8
po.norm parameter value [1]
# set the energy grid to evaluate the model over
sherpa> dataspace (0.3:8:0.01) histogram
# write out the source model to the file wgt.dat
# [the units are listed as Photons/bin because we have defined
# no instrument model (eg ARF/RMF) for the source]
sherpa> write source wgt.dat
Write X-Axis: Bin Y-Axis: Flux (Photons/bin)
# review the file
sherpa> $head wgt.dat
  0.305      8.479
  0.315      8.001
  0.325      7.563
  0.335      7.161
  0.345      6.792
  0.355      6.452
  0.365      6.137
  0.375      5.845
  0.385      5.575
  0.395      5.323
sherpa>
```

**Parameter=egridspec (string required)***Output energy grid specification*

This parameter defines the energy grid of the output ARF. It accepts the same syntax as the engrid parameter of mkarf.

**Parameter=threshold (float default=0 min=0 max=1)***Fractional threshold cut for FEF regions*

Any FEF region that has a weight (counts or spectrum weighted counts) below the fractional threshold of the total will be ignored from the calculation of the ARF and will not go into the weights file. This can be used to exclude regions that only minimally contribute to the over-all spectrum/flux and thus speed up the generation of the RMFs by mkrmf.

**Note 1**

The final fractional contribution (in the 'FRACTION' column in the weight file output) can be less than the threshold parameter since the thresholding is done and then the fractions of the remaining regions are re-normalized.

**Note 2**

The total weight/flux can be reduced by more than the threshold percent if, for example, there are many low counts regions and a single bright region. The low count regions individually could be less than the threshold of the total, but the sum of the flux/counts from these regions could be more (considerably more) than the threshold.

**Parameter=feffile (file filetype=ARD default=CALDB)**

*The FEF file containing the calibration areas.*

In most case this parameter should be left at its default value of CALDB. The name of the FEF file selected by the tool is output if the verbose value is set to 2 or greater and is also included in the header of the output files as the FEFFILE keyword.

The verbose=2 output when running mkwarf looks like:

```
Mapping response regions to FEF regions
FEF File: /soft/ciao/CALDB/data/chandra/acis/cpf/feffs/acisD2000-01-29fef_pha_ctiN0001.fits
Mapping response regions to FEF regions. Done
```

and the FEFFILE value can be found from the header using the dmkeypar tool:

```
unix% dmkeypar spectrum.wgt feffile echo+
/soft/ciao/CALDB/data/chandra/acis/cpf/feffs/acisD2000-01-29fef_pha_ctiN0002.fits
```

**Parameter=mskfile (file default=)**

*The detector mask file*

**Parameter=asolfile (file)**

If the aspect solution file(s) for the observation are provided, the average dy, dz, and dtheta values are computed and are used to adjust the SIM alignment. This reduces problems seen when the WMAP is at the edge of the chip, causing the mapping from detector to chip coordinates to fail.

**Parameter=mirror (string default=HRMA)**

*The ARDLIB mirror parameter; can use additional qualifiers to change mirror response.*

**Parameter=detsubsysmod (string default=)**

*A modifier that is added to the detector name sent into ARDLIB.*

Unlike some other ARDLIB enabled tools; mkwarf runs on multiple chips and as such does not have a detsubsys parameter. This parameter allows one to modify the internal detsubsys value to allow the response product to be modified. Things such as detector QE/U can be overridden (see "ahelp ardlb" for more information).

**Parameter=pbkfile (string filetype=input)**

The parameter block file, which defines ACIS pixel clocking parameters, is a standard product of pipeline processing and is available for every observation. This file contains information which defines how long any pixel is exposed before being read-out, which is related to the probability that any pixel will be disabled ("deadened") by cosmic rays. See the description of the "dofile" parameter for more information on ACIS dead area.

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) and "N001" is a revision number.

Since the pbkfile contents are observation-dependent, there is no default other than "NONE". In this case, no dead area correction is applied, and dafile is assumed to have the value "NONE". If pbkfile is set to a valid file, then the dafile parameter must also refer to a calibration file.

The pbkfile parameter is ignored for HRC data.

**Parameter=dafile (string filetype=input)**

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 ACIS dead area refers to a slight decrease in detector efficiency due to the background cosmic ray flux which temporarily renders some pixels useless. The calibration product is a coefficient (per CCD) which gives the fractional area lost (or "deadened") per second. Since the area lost increases with time, the magnitude of the effect depends upon the ACIS clocking parameters (number of rows, window location, frame-time) which determine how long a pixel was exposed to the background cosmic ray flux during the primary exposure and during electronic readout from the frame-store area. For full-frame timed-exposure, the dead area is about 4% at maximum CHIPY and about 2% at the readout. It is smaller for subarrays.

The ACIS clocking parameters required to scale the coefficients in the dafile are contained in the observation-specific parameter block file, which can be set by the associated parameter of this tool, "pbkfile". If dafile=NONE, then pbkfile=NONE is assumed.

The dafile parameter is ignored for HRC data.

**Parameter=ardlibpar (file default=ardlib)**

*The name of the ARDLIB parameter file.*

**Parameter=geompar (file default=geom)**

*The name of the Pixlib Geometry parameter file.*

**Parameter=clobber (boolean default=no)**

*Overwrite output files if they already exist?*

**Parameter=verbose (integer default=0 min=0 max=5)**

*Controls amount of debugging output is sent to screen.*

## CHANGES IN CIAO 3.4

### Parameter Changes

Two new parameters have been added to mkwarf 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

individual parameter descriptions for more information.

A third new parameter is "asolfile". If the aspect solution file(s) for the observation are provided in this parameter, the average dy, dz, and dtheta values are computed and are used to adjust the SIM alignment.

## CHANGES IN CIAO 3.2

Support has been added for 0th-order grating spectra. The grating efficiencies are taken into account, correctly determining the number of photons detected. The tool gets the grating information from the file header.

## CHANGES IN CIAO 3.0

### FEFFILE keyword

The name of the FEF file is now stored in the headers of both the ARF and the weights file in the FEFFILE keyword, and so can be accessed using dmkeypar. Previously it was only available in the tool history section.

### New parameters

Several new parameters – all by default hidden – have been added to mkwarf to allow more control over the ARF calculation. The "mirror" and "detsubsysmod" parameters allow you to control what parts of the mirror and detector systems ARDLIB uses to calculate the response. See "ahelp ardlib" for information on how to configure ARDLIB using these parameters ("detsubsysmod" equates to the "detsubsys" parameter of other ardlib-enabled tools, the different name is necessary because mkwarf can operate on multiple ACIS chips in one call).

As an example, if you want to turn off the ACIS contamination model – i.e. calculate the response if there was no contamination on the ACIS filter cover – then you would specify

```
detsubsysmod="contam=no"
```

The mskfile parameter, also hidden, has been added to support ACIS mask files.

## DESCRIPTION OF THE ALGORITHM

The ACIS chips are split up into 32 by 32 pixel regions – referred to as chip regions below – and the WMAP examined to see how many counts fall into each of these regions. Since the mapping uses the center of each WMAP pixel, the WMAP should be binned at a scale smaller than 32 pixels; we suggest a value of "det=8". Using a much smaller binning value than this will lead to larger files (the WMAP) and longer processing times. Note that these chip regions may be smaller than the FEF tile size for the chip (the size of a region over which the spectral response is assumed to be constant).

The conversion from detector to chip coordinate systems requires knowledge of the SIM position. The value stored in the WMAP header – the SIM\_X/Y/Z keywords – corresponds to the average value during the observation; although the change in SIM position during an observation is normally small, it is possible that a small fraction of WMAP events at the edges of the chips are not mapped onto any chip region. In these situations mkwarf ignores these counts, and prints out a warning message giving the location and number of counts. Note that the coordinate conversion is made using the center of each chip region.



If a spectrumfile is supplied, then the counts detected in each chip region are weighted by the expected number of events given the the input spectral model and the ARF for the region.

Once all the chip regions have been analysed, the fractional contribution of each region is calculated – this is referred to as the region weight below. If the threshold parameter is not set to 0, then those regions with weights below the threshold are removed from the list and the weights re-calculated (this pass happens only once).

An ARF is calculated for each chip region using the energy grid specified by the egridspec parameter and multiplied by the weight for the region. The chip regions are then accumulated up into FEF regions (i.e. the weights for all the chip regions within a single FEF region are added), and the resultant set of weights is output as the weights file. The weighted ARF is calculated as the sum of these weights.

## Notes

- The algorithm assumes that the source spectrum is spatially uniform, so that only the normalisation of the spectrum varies with position.
- The algorithm is designed to represent the weighted ARF over a large region. If the exposure varies strongly over most of the extraction region then the ARF may not fully account for this variation. This is mainly of concern for regions at the edge of an ACIS chip.
- The weights file must only be used with mkrmf to create a PI RMF; PHA RMFs are not supported.
- Response regions are currently fixed to be 32x32 pixel regions on each ACIS chip.

## Bugs

See the [bugs page for this tool](#) on the CIAO website for an up-to-date listing of known bugs.

## See Also

*calibration*

[ardlib](#)

*dm*

[dmbinning](#)

*tools*

[acis bkgnd lookup](#), [acis fef lookup](#), [acis set ardlib](#), [acis spec](#), [add grating orders](#), [add grating spectra](#), [asphist](#), [dither region](#), [dmarfadd](#), [dmcoppy](#), [dmextract](#), [dmfilth](#), [dmgroup](#), [dmimghist](#), [dmregrid](#), [fullgarf](#), [get sky limits](#), [mkacisrmf](#), [mkarf](#), [mkexpmap](#), [mkgarf](#), [mkgrmf](#), [mkinstmap](#), [mkpsf](#), [mkrmf](#), [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.  
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:  
<http://cxc.harvard.edu/ciao3.4/mkwarf.html>  
Last modified: December 2006

Ahelp: mkwarf – CIAO 3.4