




---

 AHELP for CIAO 3.4

# specextract

Context: [tools](#)

*Jump to:* [Description](#) [Examples](#) [Parameters](#) [CHANGES IN CIAO 3.4](#) [Bugs](#) [See Also](#)

---

## Synopsis

Extract source and background ACIS spectra for point-like and extended sources and build associated WMAPs, ARFs and RMFs.

## Syntax

```
specextract infile outroot [bkgfile] [ptype] [grouptype] [binspec]
[bkg_grouptype] [bkg_binspec] [energy] [channel] [energy_wmap]
[binwmap] [clobber] [verbose] [mode]
```

## Description

'specextract' is a script which lets the user create source and background PHA or PI spectra and their associated WMAPs, ARFs and RMFs files for point and extended sources.

specextract is designed to replace the psextract and acisspec scripts currently distributed in the CIAO Scripts package. The major improvement over its predecessors is the ability to determine when mkacismf should be used in place of mkrmf (see "Creating RMFs: mkrmf vs. mkacismf" section, below, for details). The new script, which is written in S-Lang ("ahelp slang"), is also able to accept a stack of input files in order to generate many spectra from one command. Note that specextract does *not* offer the option to coadd spectra as acisspec does.

The script combines the dmextract, mkwarf, mkrmf OR mkacismf, dmgroup and dmhedit tools; see the individual help files for information on each of these, e.g. "ahelp dmextract".

## Creating RMFs: mkrmf vs. mkacismf

The tool mkacismf was introduced in CIAO 3.2 and represents an entirely new method for creating ACIS response matrices; details on why mkacismf is more accurate than mkrmf are available in the the mkacismf help file. As of CALDB 3.1.0, released on 23 June 2005, mkacismf can be used to create RMFs for all observations taken at the -120 C focal plane temperature, which have had the appropriate calibration applied. The [Creating ACIS RMFs with mkacismf why topic](#) contains details about the tool and its calibration.

specextract reads the source and background file header keywords to determine when the observation was taken and what calibration has been applied. If all the requirements are met, the script uses mkacismf to generate the

source and background RMF(s) for the data. If the observation was not done at –120 or has not been reprocessed with the newest calibration available, specextract creates the RMF(s) with mkrmf.

## Output Files

The number of files created depends on if a background event file was provided and if source and/or background grouping was specified. In the most general case (source and background both provided and grouping given for both), the output files will be:

- ungrouped and grouped source spectra (.pi, grp.pi)
- ungrouped and grouped background spectra (.pi, grp.pi)
- source and background weight files (.fef). These files are created by mkwarf for use with mkrmf; they are not needed for any subsequent spectral analysis.
- weighted source and background RMF and ARF files (.wrmf, .warf)

## Grouping Source and Background Spectra

The user can choose to group the output source spectrum; all the grouptype options available with dmgroup are allowed. However, the dmgroup parameters "xcolumn" and "ycolumn" are hard-coded to "channel" and "counts" respectively, as appropriate for standard PHA files. If grouping is chosen, both the grouped and ungrouped source and background spectra files are written out; the grouped are designated by the "\_grp.pi" ending in the filename. The RESPFILE and ANCRFILE keywords in the header of the grouped source and background spectrum will be updated, as will the BACKFILE header keyword in the source spectra only.

## Fitting the Spectra

As mentioned, the BACKFILE, ANCRFILE, and RESPFILE header keywords in the source and background spectra are updated appropriately by specextract. This means that the spectra can then be read into Sherpa and all the supporting files will automatically be read as well; the background (if available) will be defined, as will the source and background response files. Please see the [Sherpa threads](#) for more information on using Sherpa to fit spectral data.

## Example 1

```
specextract "acis_evt2.fits[sky=region(3c273.reg)]" outroot=3c273
bkgfile="acis_evt2.fits[sky=region(3c273_bg.reg)]" grouptype=NONE
binspec=NONE
```

Extract source and background spectra from the same event file using the region files "3c273.reg" and "3c273\_bg.reg", respectively. Neither of the spectra will be binned. Default values are used for creating the WMAP and RMF files.

## Example 2

```
specextract "acis_evt2.fits[sky=region(1447_src.reg)]"
outroot=acis_1447 bkgfile="acis_evt2.fits[sky=region(1447_bkg.reg)]"
```

```
grouptype=NUM_CTS binspec=10 bkg_grouptype=NUM_CTS bkg_binspec=10
```

Similar to the previous example, except the source and background spectra are each grouped such that there are 10 counts in a bin.

### Example 3

```
specextract
"acis_5027_evt2.fits[sky=region(5027_srcA.reg)],acis_5027_evt2.fits[sky=
region(5027_srcB.reg)]" outroot=5027 bkgfile="" ptype=PI grouptype=NONE
binspec=NONE
```

Two event files are given as a stack, so two sets of output files are created based on the outroot value "5027": 5027\_src1 and 5027\_src2. No grouping is applied to the source spectra, and no background spectra are created. Default values are used for creating the WMAP and RMF files.

### Example 4

```
specextract @input.lis @output.lis
```

Both the input event files and output filenames are defined as a stack:

```
unix% cat input.lis
acisf00459N002_evt2.fits[sky=region(3c273.reg)]
5027_repro_evt2.fits[sky=region(5027_src.reg)]

unix% cat output.lis
459_3c273
5027_sl4
```

Default values are used for all the other parameters.

## Parameters

name	type	ftype	def	reqd	stacks
<u>infile</u>	string	input		yes	yes
<u>outroot</u>	string			yes	yes
<u>bkgfile</u>	string	input		no	yes
<u>ptype</u>	string		PI	yes	
<u>grouptype</u>	string		NUM_CTS	yes	
<u>binspec</u>	string		15	yes	
<u>bkg_grouptype</u>	string		NONE	no	
<u>bkg_binspec</u>	string			no	
<u>energy</u>	string		0.3:11.0:0.01	yes	
<u>channel</u>	string		1:1024:1	yes	
<u>energy_wmap</u>	string		300:2000	yes	

<u>binwmap</u>	string		det=8	yes	
<u>clobber</u>	string		no	yes	
<u>verbose</u>	string		0	yes	

## Detailed Parameter Descriptions

**Parameter=infile (string required filetype=input stacks=yes)**

*The source event file(s)*

The primary input to this tool is an event file (or stack of event files) with a filter defining the extraction region for the spectrum. See "ahelp dmfiltering" for information on defining region filters, and "ahelp stack" for details on using stacks of files.

**Parameter=outroot (string required stacks=yes)**

*Root for the output filenames or a stack of filenames*

If a string is given, it is used in naming all files created by specextract. In this case, the tool creates output files designated as "src1", "src2", etc. The background files are named "bkg1", "bkg2", etc., to correspond to the source

Alternatively, a stack of output filenames can be specified, one for each input file. See "ahelp stack" for details on using stacks of files.

**Parameter=bkgfile (string not required filetype=input stacks=yes)**

*The background event file(s)*

The background event file (or stack of event files) with a filter defining the extraction region for the spectrum. See "ahelp dmfiltering" for information on defining region filters, and "ahelp stack" for details on using stacks of files.

If background spectra are desired, the source and background stacks must contain the same number of elements. It is allowable to leave this parameter value blank, which results in no background spectra being created.

**Parameter=ptype (string required default=PI)**

*Spectrum type to extract*

Determines if the RMF grids are energy vs PHA or energy vs PI. Valid values are PHA and PI; these values are case-sensitive.

Read the "channel" parameter description for important information about PI and PHA binning.

**Parameter=grouptype (string required default=NUM\_CTS)**

*Source spectrum grouping type*

## Ahelp: specextract – CIAO 3.4

The allowed values the same as those in dmgroup: NONE, BIN, SNR, NUM\_BINS, NUM\_CTS, ADAPTIVE, ADAPTIVE\_SNR, BIN\_WIDTH, MIN\_SLOPE, MAX\_SLOPE, and BIN\_FILE.

- NONE: The data are grouped using maximum resolution.
- BIN: Explicitly state the binning boundaries, e.g. "10:40:5".
- SNR: Data are grouped until the square root of the number of counts in each group exceeds the given signal-to-noise value.
- NUM\_BINS: Data are grouped into a number of equal-width groups.
- NUM\_CTS: Data are grouped until the number of counts in each group exceeds the minimum number of counts specified.
- ADAPTIVE: Keeps bright features ungrouped, while grouping low SNR regions.
- ADAPTIVE\_SNR: similar to the ADAPTIVE method. Uses the SNR ratio to determine cutoffs instead of a count threshold.
- BIN\_WIDTH: Data are grouped into bins of the specified width.
- MIN\_SLOPE: Groups are formed when the absolute value of the slope of the data is less than the minimum slope threshold given in grouptypeval.
- MAX\_SLOPE: Groups are formed when the absolute value of the slope of the data is greater than the minimum slope threshold given in grouptypeval.
- BIN\_FILE: Data are grouped to match a previously grouped file.

More information on grouping is available from "ahelp dmgroup".

**Parameter=binspec (string required default=15)**

*Source spectrum grouping specification*

A numerical value used for the grouping method. The format of the grouping specification depends on what is chosen for the "grouptype" parameter. Here are some example values for each option:

- grouptype=NONE, binspec=NONE (no grouping)
- grouptype=BIN, binspec="10:40:5" (start:stop:stepsize)
- grouptype=SNR, binspec=30 (signal-to-noise value)
- grouptype=NUM\_BINS, binspec=10 (number of bins)
- grouptype=NUM\_CTS, binspec=30 (minimum number of counts per group)
- grouptype=ADAPTIVE, binspec=100 (counts value for low SNR regions)
- grouptype=ADAPTIVE\_SNR, binspec=50 (signal-to-noise value for low SNR regions)
- grouptype=BIN\_WIDTH, binspec=15 (width of the bins)
- grouptype=MIN\_SLOPE, binspec=50 (minimum slope of the data)
- grouptype=MAX\_SLOPE, binspec=100 (maximum slope of the data)
- grouptype=BIN\_FILE, binspec=grouped.fits (name of a file that provides the grouping to match)

More information on grouping is available from "ahelp dmgroup".

**Parameter=bkg\_grouptype (string not required default=NONE)**

*Background spectrum grouping type*

Parameter=binspec (string required default=15)

## Ahelp: specextract – CIAO 3.4

The allowed values are the same as those in dmgroup: NONE, BIN, SNR, NUM\_BINS, NUM\_CTS, ADAPTIVE, ADAPTIVE\_SNR, BIN\_WIDTH, MIN\_SLOPE, MAX\_SLOPE, and BIN\_FILE; see the "grouptype" parameter description for details. More information on grouping is available from "ahelp dmgroup".

### **Parameter=bkg\_binspec (string not required)**

#### *Background spectrum grouping specification*

A numerical value used for the background grouping method. The format of the grouping specification depends on what is chosen for the "bkg\_grouptype" parameter; see the "binspec" parameter description for details. More information on grouping is available from "ahelp dmgroup".

### **Parameter=energy (string required default=0.3:11.0:0.01)**

#### *Energy grid in keV*

The grid is specified by giving the lower bound, upper bound, and binning step, separated by the ":" character. For example, the default value "energy=0.3:11.0:0.01" bins from 0.3 to 11.0 keV in steps of 0.01 keV.

The default value of this parameter is suitable for most analyses.

### **Parameter=channel (string required default=1:1024:1)**

#### *RMF binning specification in pixels*

The grid is specified by giving the minimum, maximum channel, and binning step, separated by the ":" character. For example, "channel=1:1024:1" bins from channel 1 to 1024 in steps of 1. The default value is suitable for most analyses of PI spectra. A binning of "channel=1:4096:2" is suitable for most analyses of PHA spectra.

NB: the default PHA binning used by dmextract is "1:4096:2"; refer to the "defaults" parameter in "ahelp dmextract" for more information. If you intend to finish your analysis in XSpec, you must set "channel=1:4096:2", so that the spectrum and RMF have the same binning. Sherpa does not require that the spectrum and response be binned the same.

If you want an unbinned PHA spectrum (e.g. 1:4096:1), you will have to run dmextract independently of specextract. Setting "channel=1:4096:1" in specextract will produce an RMF and ARF appropriate for use with the unbinned PHA spectrum.

### **Parameter=energy\_wmap (string required default=300:2000)**

#### *Energy range for WMAPs*

The WMAP (weight map) file is an image of the extraction region. Specifically, the PHA FITS file has in its primary header an image containing a low resolution map of the source in detector coordinates. This allows downstream software to determine the appropriate weighting for calibrations which depend on detector position (for instance, effective areas may depend on the off-axis angle).

An energy filter may be included when creating a WMAP to better represent the event distribution in a particular energy range, e.g. the default value of 300:2000 keV.

This parameter is used in conjunction with the "binwmap" value.

**Parameter=binwmap (string required default=det=8)**

*Binning factor for WMAPs.*

The binwmap parameter allows the user to define the binning of the WMAP; see the "energy\_wmap" parameter for a description of what a WMAP is. We recommend 'binwmap="det=8"' (the default) to create a map in DETX,DETY coordinates binned by a factor 8.

This parameter is used in conjunction with the "energy\_wmap" value.

**Parameter=clobber (string required default=no)**

*Specifies if an existing output file should be overwritten.*

**Parameter=verbose (string required default=0)**

*Specifies the level of verbosity (0–5) in displaying diagnostic messages.*

## CHANGES IN CIAO 3.4

### Output File Stacks

The "outroot" parameter can now take a stack of output filenames. It is still possible to supply a root string; specextract will then name the files <root>\_src1, <root>\_src2, and so forth.

## 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](#)

*tools*

[acis bkgnd lookup](#), [acis fef lookup](#), [acis set ardlb](#), [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](#)

Ahelp: specextract – CIAO 3.4