



AHELP for CIAO 3.4

## dmimg2jpg

Context: [tools](#)

*Jump to:* [Description](#) [Examples](#) [Parameters](#) [Bugs](#) [See Also](#)

## Synopsis

Make a color JPEG image (or EPS) from three image files

## Syntax

```
dmimg2jpg infile greenfile bluefile outfile [regionfile] [psfile]
[lutfile] [colorstretch] [colorshift] [invert] [scalefunction]
[scaleparam] [minred] [mingreen] [minblue] [maxred] [maxgreen]
[maxblue] [regioncolor] [regionopt] [gridcolor] [showaimpoint]
[gridsize] [showlabel] [showgrid] [fontsize] [verbose] [clobber]
```

## Description

'dmimg2jpg' takes a datamodel IMAGE file and converts it to a JPEG image file (suitable for web pages). 'dmimg2jpg' can create a 3-color image from 3 input datamodel image files. User options include scaling the image(s), color table choices, CXC region overlays and WCS grid markings.

### Input Options:

- (1) Single datamodel image: The tool will apply the specified lookup table (lut) and scale the data as given by the parameters. Any optional overlays are applied to the image and it is stored as a JPEG. The lut.par parameter file provides a useful selection of color tables.
- (2) Three input datamodel images (to create 'true color' images). Input files correspond to red, green and blue colors; images are usually cut along different energy bands, with the lowest energies assigned to the "infile" (the red image) and the highest energies assigned to the "bluefile". The tool will scale the data as given by the parameters and apply a RGB colortable created from the data.

### Scaling Options:

The tool's automatic scaling makes a good first attempt, but is unlikely to show optimally the particular features that the user needs to highlight. 'dmimg2jpg' provides flexible scaling from pixel ranges to colors. The default is a log scaling of the image from the data minimum value to the data maximum value. Linear and power scaling are also available. For example, power scaling with a parameter of 0.5 (scalefunction=power scaleparam=0.5) is a square root scaling that brings out faint detail. The min and max parameters for each color can be used to emphasize or suppress particular ranges such as weak features and background events. If the color lookup tables are used, the color table can be stretched and shifted (parameters colorstretch and colorshift). The color table can also be inverted to switch the darkest color for the lightest color, which is

useful for printing.

## Overlay Options:

These assume that the input files contain WCS information. A CXC region file can be supplied to overlay sources on the image and the user can choose the method of overlay. The user can specify if they want the reference point of the image (the CRPIX values) marked with a cross-hair. A simple grid with tick marks and the coordinates can also be placed on the image.

## EPS File Output:

The tool allows an additional encapsulated postscript (EPS) file to be created, which is especially useful when preparing a publication submission. For 3-color images, the color table will be compressed from a 24-bit (JPEG) to 8-bit (postscript). The JPEG file is still produced when the EPS version is selected.

## NOTES

- There is a parameter file "colors" that has many standard color names and associated color triples. You can redirect the region/contour/grid colors to this file, eg `regioncolor="colors.white."` There are currently 655 available.
- There are several predefined color lookup tables available (taken from SAOtn). They can be accessed via redirects from the "lut.par" file, eg `lut=")lut.aips."` Available lookup tables are listed in the parameter file `lut.par`. There are currently 19 color lookup tables available.
- The input files must be datamodel image files. Event lists with no binning will cause `dmimg2jpg` to report that it could not open the file even if the file exists and is a valid FITS binary table.

For additional examples, see the [Create A True Color Image thread](#).

## Example 1

```
dmimg2jpg red_image.fits greenfile="" bluefile="" regionfile=""
outfile="my_image.jpg" invert+ lut=")lut.grey"
```

This will create a JPEG file of the image "red\_image.fits" using an inverted greyscale. The default scaling options are used.

## Example 2

```
dmimg2jpg "events.fits[energy=0:1500][bin sky=8]"
"events.fits[energy=1500:2000][bin sky=8]"
"events.fits[energy=2000:8000][bin sky=8]" regionfile=""
outfile="image_2.jpg"
```

A true-color image is created by using Data Model filters to define the three energy bands and bin the data.

## Example 3

```
dmimg2jpg image_one.fits greenfile="" bluefile=""
outfile="greyscale.jpg" regionfile="region(sources.fits)"
regioncolor=")colors.red"
```

This will create a grey scale JPEG file with the source regions specified in "sources.fits" overlaid in red.

## Example 4

```
dmimg2jpg red.fits green.fits blue.fits rgb.jpg
regionfile="region(src.reg)" regioncolor=")colors.red" scaleparam=.5
scalefunction=pow
```

This will create a 3-color image that is square-root scaled. The source regions specified in the region file "src.reg" will be overlaid in red.

## Parameters

name	type	ftype	def	min	max	units	reqd	autoname
<u>infile</u>	file	input					yes	
<u>greenfile</u>	file	input	none				yes	
<u>bluefile</u>	file	input	none				yes	
<u>outfile</u>	file	output					yes	yes
<u>regionfile</u>	file	input					no	
<u>showaimpoint</u>	boolean		yes				no	
<u>lutfile</u>	file	input	)lut.grey				no	
<u>colorstretch</u>	real		1				no	
<u>colorshift</u>	integer		0				no	
<u>invert</u>	boolean		no				no	
<u>scalefunction</u>	string		log				no	
<u>scaleparam</u>	real		3				no	
<u>minred</u>	real		INDEF				no	
<u>mingreen</u>	real		INDEF				no	
<u>minblue</u>	real		INDEF				no	
<u>maxred</u>	real		INDEF				no	
<u>maxgreen</u>	real		INDEF				no	
<u>maxblue</u>	real		INDEF				no	
<u>regioncolor</u>	string		)colors.green				no	
<u>regionopt</u>	string		individual				no	
<u>showlabel</u>	boolean		no				no	
<u>showgrid</u>	boolean		yes				no	
<u>gridcolor</u>	string		)colors.white				no	
<u>gridsize</u>	real		120			arcsec	no	
<u>fontsize</u>	integer		2	1	3		no	
<u>psfile</u>	file	output					no	
<u>verbose</u>	integer		0	0	5		no	
<u>clobber</u>	boolean		no				no	

## Detailed Parameter Descriptions

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

*Input file or red color file*

The primary input image file. The full Data Model filtering and binning syntax is supported. This cannot be a stack of images.

### **Parameter=greenfile (file required filetype=input default=none)**

*Green color file*

The image file name for the green channel color image. Can be set to "none" to indicate that a color lookup table should be used. This cannot be a stack of images.

### **Parameter=bluefile (file required filetype=input default=none)**

*Blue color file*

The image file name for the blue channel color image. Can be set to "none" to indicate that a color lookup table should be used. This cannot be a stack of images.

### **Parameter=outfile (file required filetype=output autoname=yes)**

*Name of the output JPEG file.*

This is the name of the output JPEG file. This file must always be created even when the user wants a postscript file. A single period "." is allowed. This will create an output file in the same directory as the input file, with the same name except for the new extension being "jpg."

### **Parameter=regionfile (file not required filetype=input)**

*Region file or stack*

The file name for the region file. This tool follows the region syntax. Use "ahelp dmregions" for more information. Can be set to "none." If the regionopt parameter is set to combine, overlapping shapes will become a single shape.

### **Parameter=showaimpoint (boolean not required default=yes)**

*To place a cross-hair at the aim point or not.*

The showaimpoint parameter allows for the placement of a cross-hair at the location of the WCS recorded aim point. The default is to not place the aim point.

### **Parameter=lutfile (file not required filetype=input default=)lut.grey)**

*Color table (lookup table)*

Color lookup table file. This lists the color values for each scaled pixel value in red green, blue order scaled from 0 to 1. This is ignored if the user supplies three images for true color scaling.

**Parameter=colorstretch (real not required default=1)**

*Amount to stretch the color lookup table.*

Colorstretch allows a stretching of the color lookup table. This is similar to a gamma correction. This currently only applies to images that are using a colortable.

**Parameter=colorshift (integer not required default=0)**

*Amount to shift the color scale*

Amount to shift the color scale. This currently only applies to images that are using a colortable.

**Parameter=invert (boolean not required default=no)**

*Invert the color table*

Invert will allow the color table to be inverted. This currently only applies to images that are using a colortable. True color images cannot be inverted.

**Parameter=scalefunction (string not required default=log)**

*Function to scale the color table by. [options=log/linear/power]*

This parameter defines the scaling function used to view the data. Typically "log" scaling is used. Other options are linear and power. The log base and power exponent are controlled by the scaleparameter.

**Parameter=scaleparam (real not required default=3)**

*The parameter used for scaling in the function defined by scalefunction.*

When a non-linear scale function is defined, the scaleparam is used as the exponent or base for the scale function.

**Parameter=minred (real not required default=INDEF)**

*The minimum color for scaling the red image.*

The minimum value to consider when scaling the infile data. INDEF can be used to let the program determine the min and max from the data. In the case of a single input, this parameter controls the minimum.

**Parameter=mingreen (real not required default=INDEF)**

*The minimum color for scaling the green image.*

The minimum value to consider when scaling the green data. INDEF can be used to let the program determine the min and max from the data.

**Parameter=minblue (real not required default=INDEF)**

*The minimum color for scaling the blue image.*

The minimum value to consider when scaling the blue data. INDEF can be used to let the program determine the min and max from the data.

**Parameter=maxred (real not required default=INDEF)**

*The maximum color for scaling the red image.*

The maximum value to consider when scaling the red data. INDEF can be used to let the program determine the min and max from the data. For true color use, it is advised to lower the max values for greater detail. Setting the parameter verbose to 1 or higher will display the min and max values used. For a single image, this is the parameter that controls the maximum value.

**Parameter=maxgreen (real not required default=INDEF)**

*The maximum color for scaling the green image.*

The maximum value to consider when scaling the green data. INDEF can be used to let the program determine the min and max from the data. For true color use, it is advised to lower the max values for greater detail. Setting the parameter verbose to 1 or higher will display the min and max values used.

**Parameter=maxblue (real not required default=INDEF)**

*The maximum color for scaling the blue image.*

The maximum value to consider when scaling the blue data. INDEF can be used to let the program determine the min and max from the data. For true color use, it is advised to lower the max values for greater detail. Setting the parameter verbose to 1 or higher will display the min and max values used.

**Parameter=regioncolor (string not required default=)colors.green)**

*The RGB color to mark regions with.*

This parameter specifies the color to mark regions in the regionfile with. This is a triple of numbers, between 0 and 1, that represent the red, green and blue factors. The file colors.par contains a large number of colors broken into the RGB triple.

**Parameter=regionopt (string not required default=individual)**

*The method of marking the regions [options=individual/combine]*

This parameter specifies the method to mark regions. If individual, then shapes are marked separately even if they overlap. The combine option will combine shapes that overlap.

**Parameter=showlabel (boolean not required default=no)**

*Labels the items marked from the region file.*

This parameter specifies whether items marked in the region file should be labeled. If set to yes, a line is drawn from the source to the label. Items are currently labeled in the order from the file (or file filter).

**Parameter=showgrid (boolean not required default=yes)**

*Display the grid.*

This parameter specifies whether a grid should be displayed. The related parameters are gridcolor, gridsize and fontsize.

**Parameter=gridcolor (string not required default=)colors.white)**

*The RGB color to mark grid and aim point with.*

This parameter specifies the color to mark the grid and the aim point with. This is a triple of numbers, between 0 and 1, that represent the red, green and blue factors. The file colors.par contains a large number of colors broken into the RGB triple.

**Parameter=gridsize (real not required default=120 units=arcsec)**

*Size in arc seconds for grid separations.*

This determines how far apart the tick marks are placed in the image. The parameter is reported in arcseconds.

**Parameter=fontsize (integer not required default=2 min=1 max=3)**

*Size of the font to mark the RA and Dec on the grid.*

The font is ugly and very simple but functional. This controls how big the font is displayed.

**Parameter=psfile (file not required filetype=output default=)**

*The name of the output encapsulated postscript file.*

This allows the user to create an encapsulated postscript file. A JPEG file will always be produced. A blank or a "none" will not produce an encapsulated postscript file. A single period "." is allowed. This will create an output file in the same directory as the input file, with the same name except for the new extension being "ps."

**Parameter=verbose (integer not required default=0 min=0 max=5)**

*Report information to the user.*

This parameter defines how much information is passed to the user. A level of 0 reports nothing, a level of 5 reports the most. Level 1 will report the input parameters.

**Parameter=clobber (boolean not required default=no)**

*Remove files if they already exist?*

Clobber will remove the output files if they already exist before creating the new files.

## Bugs

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

## See Also

*concept*

[subspace](#)

*dm*

[dmimages](#), [dmimfiltering](#), [dmregions](#)

*gui*

[firstlook](#), [prism](#)

## Ahelp: dmimg2jpg – CIAO 3.4

*tools*

[dither\\_region](#), [dmappend](#), [dmcontour](#), [dmfilth](#), [dmimgcalc](#), [dmimghist](#), [dmimgpick](#), [dmimgthresh](#),  
[dmmakereg](#), [dmregrid](#), [get sky limits](#), [get src region](#), [tg\\_create\\_mask](#)

---

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/dmimg2jpg.html>  
Last modified: December 2006