



AHELP for CIAO 3.4

## dmregions

Context: dm

*Jump to:* [Description](#) [Examples](#) [CHANGES IN CIAO 3.3](#) [Issues with CIAO regions](#) [Bugs](#) [See Also](#)

## Synopsis

CIAO region filtering syntax

## Description

Regions are two dimensional filters that can be used to include or exclude data from a given file. They may be applied to any `vector column' in the file.

Regions are made up of a number of shapes that are described below. The shapes can either be inclusive or exclusive. The different shapes can then be combined using either a boolean AND or boolean OR operation which leads to a degree of flexibility to design arbitrary regions.

### The recognized shapes.

Name	Arguments
RECTangle	(xmin,ymin,xmax,ymax)
BOX	(xcenter,ycenter,width,height)
BOX	(xcenter,ycenter,width,height,angle)
ROTBBOX	(xcenter,ycenter,width,height,angle)
CIRcle	(xcenter,ycenter,radius)
ANNULUS	(xcenter,ycenter,iradius,oradius)
ELLipse	(xcenter,ycenter,xradius,yradius,angle)
SECTor	(xcenter,ycenter,minangle,maxangle)
PIE	(xcenter,ycenter,iradius,oradius,minangle,maxangle)
POLYgon	(x1,y1,x2,y2,x3,y3,...)
POInt	(xcenter,ycenter)
REGION	(file)
FIELD	()

### Names of shapes

In the table above, the text in capital letters is the minimum allowed for each shape. The shape names are case insensitive, so

```
circle(4023.45,3769.23,120)
```

and

dmregions

```
CIR(4023.45,3769.23,120)
```

both refer to the same shape.

## Angles

Angles are given in degrees from the X axis, with positive values representing the counter-clockwise direction.

## Lengths and Radii

This applies to all regions that take a length parameter (box, rotbox, circle, annulus, ellipse, and pie). The length or radius is interpreted as a physical coordinate value (e.g. sky pixels). If a trailing d, ', or " is added (e.g. 130d), then the value is interpreted as degrees, minutes, or seconds of arc, respectively. Two single quotes will also work for arcseconds.

## Polygons

For the polygon shape, if the last point is not equal to the first point, it will add another point onto the end to close the polygon. At present, polygons must be simple; that is they must not intersect themselves.

## Combining shapes

Regions can be combined using boolean AND (intersection) or boolean OR (union) operations. The syntax for these is given in the table below, where shape1() and shape2() refer to any recognised shape except for "region()".

### Shape arithmetic

Operation	Syntax	Syntax	Syntax
OR	shape1()+shape2()	shape1() shape2()	shape1()shape2()
AND	shape1()*shape2()	shape1()&shape2()	
NOT	!shape1()	shape1()-shape2()	

Region expressions are parsed using the normal precedence rule of the mathematical "\*" and "+" operators, with all "-" operators replaced by "!". This means that

```
circle(0,0,10)+rect(10,-10,20,10)-circle(10,0,10)
```

is read as

```
circle(0,0,10)+rect(10,-10,20,10)*!circle(10,0,10)
```

so that the second circle is only excluded from the rectangle, and not the first circle. To exclude the second circle from both shapes you would have to write

```
circle(0,0,10)-circle(10,0,10)+rect(10,-10,20,10)-circle(10,0,10)
```

An alternative method of excluding shapes is to use the exclude filter, described in "ahelp dmfiltering". So, to exclude all rows within a distance of 20 from (x,y)=(4000,4000) you could use either

```
"[(x,y)=field()-circle(4000,4000,20)]"
```

or

```
"[exclude (x,y)=circle(4000,4000,20)]"
```

## Using region files

The "region(file)" directive can read in ASCII region files produced by SAOImage, SAOtngr, and PROS (SAODs9 can write out region files in any of these formats). It can also read CXC–FITS REGION files, e.g. as produced by the source–detection tools such as wavdetect or by the dmmakereg tool. The region directive can not be combined with other shapes.

### Example 1

```
dmstat "evt2.fits[sky=circle(4000,4000,20)]"
```

Here the dmstat tool is given only those rows of the file evt2.fits that pass the filter, in this case a circle centered at (4000,4000) and with a radius of 20 using the SKY values. Since "sky" is the name of a vector column in the file and translates to "(x,y)", the command could have been written as

```
dmstat "evt2.fits[(x,y)=cir(4000,4000,20)]"
```

where we have also taken advantage of the fact that the region name does not have to be fully spelled out.

Note that the default behavior for region filters is to exclude the entire field and then start adding to it.

### Example 2

```
dmimghist infile="img.fits[field()-circle(30,40,10)]"
```

The region used to filter the image "img.fits" consists of the whole image (as indicated by the "field()" shape) with a circular region removed from it. Since no coordinate system was given in the filter expression, the logical system is used, so that the above is a short form for

```
dmimghist infile="img.fits[(#1,#2)=field()*!circle(30,40,10)]"
```

To use other coordinate systems for the filter, include the name of the axes in the expression; for example

```
"img.fits[(x,y)=circle(4010,3943,25)]"
```

### Example 3

```
my_tool "foo[(pha,pi)=region(ds9.reg)]"
```

Filter the file on the energy and time columns with the region stored in the file ds9.reg. Although the pha, pi pair do not form a vector column, you are allowed to define a paired column like this on the fly provided their datatypes are the same.

## CHANGES IN CIAO 3.3

A number of bugs have been fixed in the calculation of the areas of composite regions.

## Issues with CIAO regions

There are several subtleties with CIAO regions that are worth noting. These arise particularly when using region files made by imagers like DS9.

### Celestial coordinate regions

In order for a CIAO tool to make use of a region in world coordinates, e.g.

```
circle(06:37:11,-40:20:00,5')
```

it must know about the mapping between world and physical coordinates for the dataset in question. Not all CIAO tools have this ability yet. If you have problems with celestial coordinates, try using a region in physical pixel coordinates instead.

DS9 region files saved in celestial coordinates in degrees are not currently supported in CIAO; celestial coordinates are only recognized in sexagesimal format.

### Stacks of regions versus regions with many shapes

It's important to understand the difference between two concepts: the REGION STACK and the REGION FILE.

A region file defines a single 'region' which may have multiple shapes. For example, a region file with

```
circle(4096,4096,20)
+circle(5000,2000,100)
```

is a single region consisting of two circles. Applying a region file to some data lets CIAO decide whether points are in the region, but no knowledge is returned about which of the shapes a point is in.

A region stack contains multiple regions. CIAO will figure out each of the regions separately. The most common use of region stacks is in dmextract, where binning on a region stack creates a histogram where each bin corresponds to one region.

The syntax is

```
unix% dmextract "evt.fits[bin sky=@reg.stk]" ...
```

to operate on a region stack and make a histogram of counts versus region. In contrast,

```
unix% dmextract "evt.fits[bin sky=region(reg.stk)]" ...
```

specifies a single (albeit complicated) region, which will generate a single output bin.

### Includes and excludes

The CIAO syntax requires excluded regions to follow included regions. If you draw four circles on DS9, two includes and two excludes, and then save the region, the resulting file depends on the order the regions were drawn. If the order is `-circle+circle-circle+circle`, CIAO will not understand because of the leading `-circle`.

Note that

```
+circle(4096,4096,1000)
-circle(4096,4096,250)
+circle(4096,4096,50)
```

will have the inner 50-radius circle included, and the 50-250 radius annulus excluded, but

```
+circle(4096,4096,50)
-circle(4096,4096,250)
+circle(4096,4096,1000)
```

will include everything, as it's interpreted as "(R50 minus R250) OR R1000", and the R1000 includes everything. Therefore, you must be careful about order when generating region files in DS9.

## DS9 regions

CIAO does not understand all the shapes used in DS9-format region files. This includes elliptical and box annuli, the panda shape, and format used for writing grids of annuli.

## Region areas (BACKSCAL)

For simple regions, the area can be calculated using analytical methods. For complicated regions where shapes overlap, or are close enough that it's not obvious they don't overlap, we must use a different method; CIAO divides up the area into bins (which may be smaller than a single pixel) and counts the number of bins which pass the test 'am I inside the region?'. This approach using discrete bins has a finite error, which is usually of order one percent of the area.

Region areas are most commonly used in the BACKSCAL keyword added to PHA/PI spectral files generated by dmextract. The error in the area is almost always small compared to other calibration uncertainties.

Be particularly careful when making regions comparable in size to the pixel size. When applying such a region to a table which contains fractional pixel positions, filtering does occur in exactly the area specified, although the user should be careful about interpretation – for instance, the true radial profile of a source may be blurred by aspect, PSF and instrument pixelization. When applying the region to an image file, filtering will accept all the counts in a pixel if the pixel center is within the region, and none of them if the pixel center is outside the region; therefore, the effective area sampled is not quite the same as the area calculated.

## Bugs

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

## See Also

*calibration*

[caldb](#)

*chandra*

[coords](#), [guide](#), [isis](#), [level](#), [pileup](#), [times](#)

*chips*

[chips](#)

*concept*

[autoname](#), [parameter](#), [stack](#), [subspace](#)

*dm*

[dm](#), [dmbinning](#), [dmcols](#), [dmfiltering](#), [dmimages](#), [dmimfiltering](#), [dmintro](#), [dmopt](#), [dmsyntax](#)

*gui*

[gui](#)

*modules*

[paramio](#), [pixlib](#), [stackio](#)

*slang*

[overview](#), [slang](#), [tips](#)

*tools*

[dither\\_region](#), [dmcontour](#), [dmimg2jpg](#), [dmmakereg](#), [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/dmregions.html>  
Last modified: December 2006