



# srcdisplay

April 16, 2023

## Abstract

Displays an image overlayed with the positions of detected sources.

## 1 Instruments/Modes

Instrument	Mode
MOS, PN, OM	All imaging modes

## 2 Use

pipeline processing	no
interactive analysis	yes

## 3 Description

This task displays a given image and plots the positions of sources listed in a separate sourcelist file.

The input image is specified via the parameter **imageset** and the sourcelist file (generated by eg **eboxdetect**, **emldetect**, **srcmatch** or **ewavelet**) is specified through the **boxlistset** parameter.

Circles are used to depict the source positions. The radius of these circles can be set using the **sourceradius** parameter. Alternatively the statistical error on the position error contained in the source list can be used by specifying **useposerr=yes**. In this case an additional systematic error may be added in quadrature using the **syserr** parameter.

An optional ID label can also be displayed alongside the circle, corresponding to the row number of that source in the input source list. This can be enabled through the **uselabel** parameter. This helps the user to refer back to source properties documented in the source list.

These circles are in fact Ds9-type regions, which can be written out to a file for future use (for example, when running a later DS9 session) by setting **withregionfile** to true, and specifying the desired file name via the **regionfile** parameter.



Note that the current version computes the regions in terms of equatorial coordinates, as opposed to raw image coordinates in previous versions. This means that the task can overlay source positions taken from a given source list onto *any* image, eg a PN sourcelist can be overlayed on a MOS1 image, as long as the image contains WCS information that maps the pixel grid onto equatorial coordinates.

### 3.1 Examples

To overlay the sourcelist `srclist.ds` generated previously by **eboxdetect** on an image `image.ds` the following command can be used:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds
```

This would overlay circles, each of radius 5 pixels, around each source detected by **eboxdetect**. To overlay circles of radius 0.01 degrees, use the following:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds sourceradius=0.01
```

To perform the same as above, but also write out a DS9 region file named `regions.txt`, detailing the source regions corresponding to the displayed circles, the following command could be used:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds sourceradius=0.01 withregionfile=true  
regionfile=regions.txt
```

To use a source radius determined from the RADEC\_ERR, position error, column in the sourcelist use:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds useposerr=yes
```

To use a source radius determined from a position error column with a user supplied name:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds useposerr=yes poserrcol="POSERR"
```

NB: The errors are assumed to be in units of arcseconds and the column must be of type REAL\*32.

To add a systematic error of 2 arcseconds to the position error, use:

```
srcdisplay imageset=image.ds boxlistset=srclist.ds useposerr=yes syserr=2.0
```

## 4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

<b>boxlistset</b>	yes	data-set	boxlist.ds	none
-------------------	-----	----------	------------	------

Name of dataset containing source list, output from **eboxdetect**.

<b>withimageset</b>	no	boolean	true	true false
---------------------	----	---------	------	------------

If set to **true** a new image data set is given in parameter **imageset**.



<b>imageset</b>	no	data-set	image.ds	none
-----------------	----	----------	----------	------

The image on which the positions of detected sources are displayed. For **withimageset=false** no new image data set shall be displayed but the sources in **boxlistset** will be marked on the image in the currently active Ds9 session.

<b>sourceradius</b>	no	float	0.01	0.0001 $\leq$ <b>sourceradius</b> $\leq$ 2
---------------------	----	-------	------	--

The radius of the displayed circles for each source, in degrees.

<b>useposerr</b>	no	boolean	false	none
------------------	----	---------	-------	------

If true, then the circle radius is taken as the error on the position in the sourcelist.

<b>poserrcol</b>	no	string	"RADEC_ERR"	none
------------------	----	--------	-------------	------

The name of a column in the source list which gives the position error in arcseconds. The column must be of type REAL\*32.

<b>syserr</b>	no	float	0.0	
---------------	----	-------	-----	--

A systematic error to be added in quadrature to the positional error when **useposerr=true** is set.

<b>includesources</b>	no	boolean	true	none
-----------------------	----	---------	------	------

If true, then the DS9 region descriptor generated for each source is such that it INCLUDES the area bounded by the circle; otherwise it includes everything BUT this area.

<b>uselabel</b>	no	boolean	false	none
-----------------	----	---------	-------	------

If true, a number is displayed alongside the source position, corresponding to the row number of that source in the input source list.

<b>withregionfile</b>	no	boolean	true	none
-----------------------	----	---------	------	------

If true, then a file of name **regionfile** is written out, containing a list of DS9-type regions for each source, corresponding to the displayed circles.

<b>regionfile</b>	no	file-name	regionfile.txt	none
-------------------	----	-----------	----------------	------

The name of the region file.



<b>overlay</b>	no	boolean	false	true false
----------------	----	---------	-------	------------

If set to **false** any previously existing source region markers on the active image will be deleted before the new ones are displayed. **overlay=true** suppresses this so any existing source markers will be left unmodified. This is useful for e.g., comparing the results of two independent source detection runs on the same image.

<b>srccolor</b>	no	string	white	black	white	red
				green	blue	cyan
				magenta	yellow	

The color of the source markers - useful in conjunction with the **overlay** parameter to control the display of two or more source lists on the same image.

## 5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

### **error(UnableToCreateFile** (*error*)

Cannot open a file of corresponding to the parameter **regionfile**.

### **invalidWCS** (*warning*)

the image does not contain a WCS entry describing a projection from image coordinates to equatorial coordinates.

*corrective action:* continue

### **NoBoxSizePresent** (*warning*)

the source list does not contain a column **BOX.SIZE**. This has no affect on the plotting but may affect other tools using the **srcdisplay** library.

*corrective action:* continue

### **NoPosnErrorsPresent** (*warning*)

the source list does not contain a column **RADEC\_ERR**. This has no affect on the plotting but may affect other tools using the **srcdisplay** library.

*corrective action:* continue

## 6 Input Files

1. A **eboxdetect** source list output. This is a dataset in which the first table must contain the following columns:

- RA, of type real-64.
- DEC, of type real-64.



and optionally:

- `RADEC_ERR`, of type `real-32`.
  - `BOX_SIZE`, of type `real-32`.
2. An image dataset. This must contain a block corresponding to an  $n \times m$  array. It must also contain WCS information that describes the mapping from image coordinates to equatorial coordinates.

## 7 Output Files

1. A DS9-compatible region file. This is a text file containing region descriptions for each source. This is only written out if `withregionfile` is true.

## 8 Algorithm

1. Read in source list and generate source regions for each source, using their positions in pixels. Write these regions to a temporary file if `withregionfile` is false, otherwise to a permanent file with a name corresponding to the parameter `regionfile`.
2. Invoke an `imgdisplay` session, using the image specified in `imageset`, and the region file name as arguments.

## 9 Comments

- The ID labeling is currently based on the ROW number. A source list could have its own identification scheme, eg a `SRC_NUM` or `SRCNUM` column that contains an ID number for each source, which may cause slight confusion. Support is planned for that in the near future.

## References