



# eoptloadmask

April 16, 2023

## Abstract

This task reads external catalogues and makes a mask image, in which the areas where the optical-loading is likely are masked.

## 1 Instruments/Modes

Instrument	Mode
EPIC	Imaging

## 2 Use

pipeline processing	yes
interactive analysis	yes

## 3 Description

### 3.1 Optical loading

X-ray CCDs are by definition sensitive in X-rays and not so in the other energy bands, such as, optical band. In particular, EPIC in XMM-Newton is equipped with optical-blocking filters, which should massively reduce the amount of low-energy (ultraviolet, optical and infrared) photons that reach the X-ray CCDs first of all.

Nevertheless the sensitivity of X-ray CCDs is non-zero even for infrared photons. When there are extremely bright sources, typically very bright (optical) stars, photons from them can excite a significant amount of electrons in the X-ray CCDs, and can be (falsely) recognised as an (X-ray) event. This phenomenon is known as optical loading.

The optical loading increases the local background level at best, which would do no good in determining the parameters of the detected sources in the area, or can even lead to the false detection of X-ray sources by itself. In EPIC, the optical loading is particularly prominent in pn (than MOSs).



There are a number of optical, infrared and even ultraviolet source catalogues available. Using those, users can make a sophisticated guess which area is likely to be affected with the optical loading. This task is designed to output the mask file to mask those (suspected) affected area.

In practice, this task reads one or more external catalogues (of optical/infrared/ultraviolet sources), applies the user-specified filter, such as filtering out faint sources in a catalogue, then outputs a mask file.

Then, users may wish to use the output mask file with **dpssflag**, to flag the sources, which are likely to be affected with the optical loading.

## 3.2 Behaviour

This task reads one or more external catalogues in the FITS table format, where both the FITS filename and Table name can be specified by users. If the latter is omitted, it is assumed the first extension of the FITS file is the source list table.

The source-list tables can contain an arbitrary number of columns. The two of them must be Right Ascension and Declination in J2000 in unit of degree with Real type. In default the column names for those are **RA** and **DEC**, however users can explicitly specify them by **colnamestyle=user** with **colra** and **coldec**, respectively. Note that these two parameters accept a list, the order of which must be the same as that of the source list. If only one element is specified for either or both of those two parameters, that value is used for all the input source-lists.

The source-list tables can contain an arbitrary number of rows. Users can filter out some sources in them by stating the filtering conditions with **expressions**, where the columns in the table are used as a parameter. Note that the columns used in **expressions** must exist in the FITS file. **expressions** accepts a list, the order of which is the same as that of the source list; in other words users can apply different filtering conditions for different input source-lists.

After filtering out uninteresting, perhaps weak, sources, this task outputs a mask file, in which the positions of the sources are masked out as a filled circle. The radius of the circle can be a fixed value **radius** (when **radiusstyle=fixed** (default); in unit of arcsec), or alternatively the input source-list files can offer a column that lists the radius in arcsec for each source. In the latter case, **radiusstyle=column** and the list of the column names of it for each input source-list **colradius** (Default: **RADIUS**) can be specified.

In default, those areas regarded to be affected with optical-loading have the value of zero, and the rest, one. This behaviour can be inverted with **isinverted=no**. The default behaviour is convenient to use of the output mask file with **dpssflag**.

## 3.3 Examples

A simple example.

```
eoptloadmask srclisttab=usno.fits:SRCLIST maskset=outmask_usno.ds \  
  templateset=P0126700201M2S002IMAGE_8000.FIT \  
  radius=20 expressions='(BMAG<10)||(RMAG<10)'
```

A slightly more complicated example with two input catalogues.



```
eoptloadmask srclisttab='usno.fits:SRCLIST iras.fits' maskset=outmask_all.ds \  
  templateset=P0126700201M2S002IMAGE_8000.FIT \  
  radiusstyle=column colradius='Radius Radius_Arcsec' \  
  colnamestyle=user colra='RA_Cat RA_J2000' coldec='DEC_Cat DEC_J2000' \  
  expressions='(BMAG<10)||(RMAG<10) FR_30GHz>1e5' isinverted=no
```

## 4 Parameters

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

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

<b>srclisttab</b>	yes	dataset list		
-------------------	-----	-----------------	--	--

The list of input (optical) source-list FITS file tables (in <dataset name>:<table name> format).

<b>maskset</b>	no	dataset	eoptloadmask.ds	
----------------	----	---------	-----------------	--

Filename of the output mask image dataset.

<b>templateset</b>	yes	dataset		
--------------------	-----	---------	--	--

This file should contain an image in the primary extension, which is used to define the pixel dimensions and World Coordinates of the output mask image.

<b>temporaryset</b>	no	dataset	eoptloadmask_tmpimage.ds	
---------------------	----	---------	--------------------------	--

Name of a temporary, internally-accessed dataset to hold the filtered source list. It is only of importance for pcms (or other parallel) calls of **eoptloadmask** (or when the current working directory is not writable for the user).

<b>radiusstyle</b>	no	string	fixed	fixed—column
--------------------	----	--------	-------	--------------

Specify how to give the radius or radii of the masked area for each (optical) source.

<b>radius</b>	yes	real	20	$0 \leq \text{radius}$
---------------	-----	------	----	------------------------

Radius in arcsecond for masking the area around bright sources. Mandatory if **radiusstyle**=fixed.

<b>colradius</b>	no	string	RADIUS	
------------------	----	--------	--------	--

Column name in the input FITS tables for the radius in arcsecond to mask. This is only read if **radiusstyle**=column.

<b>colnamestyle</b>	no	string	default	default—user
---------------------	----	--------	---------	--------------

If 'default', 'RA' and 'DEC' are assumed to be the column name for the coordinates used in the input FITS tables. If 'user', **colra** and **coldec** are read. If 'file', **colnamefile** must be given.

<b>colra</b>	yes	string list		
--------------	-----	----------------	--	--

Column name list for the 'RA' (Right Ascension) in the input FITS tables. The unit has to be in degree. The equinox and epoch of the coordinates must match those in the given source list (**srclisttab**).

<b>coldec</b>	yes	string list		
---------------	-----	----------------	--	--

Column name list for the 'DEC' (Declination) in the input FITS tables. The unit has to be in degree. The equinox and epoch of the coordinates must match those in the given source list (**srclisttab**).



<b>withexpressions</b>	no	boolean	no	
------------------------	----	---------	----	--

Whether specify **expression** or not.

<b>expressions</b>	no	string list		
--------------------	----	----------------	--	--

A selection expression used to filter the source list. The selection expression should adhere to the **selectlib** grammar and should contain as arguments only keywords and column names actually present in the source list table. The list (as opposed to a single string) is allowed for this parameter, each of element of which corresponds to each input source-list; for that reason, spaces must be used only to separate the expressions among selection-expressions for the different source lists.

<b>withexpressions</b>	no	boolean	no	
------------------------	----	---------	----	--

Whether specify **expression** or not.

<b>isinverted</b>	no	boolean	yes	yes—no
-------------------	----	---------	-----	--------

If true, the pixels affected with optical-loading in the output mask file are zero, if not, one. For the later use of the output mask file with **dpssflag**, **isinverted=yes** is more convenient (default).

## 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.

### **noOutputMade** (*error*)

When no source passes the filtering condition, Warning **noSourcesSelected** is issued, then if the output mask file already exists, this task raises this error to abort.

### **noInputCatalogue** (*error*)

### **badRadiusStyle** (*error*)

### **badColnameStyle** (*error*)

### **noSourcesSelected** (*warning*)

No source is selected after the specified filtering is applied. In other words, all the sources are filtered out.

*corrective action:* No source is selected hence it stops.

### **allSourcesTooFar** (*warning*)

All the sources in an input source-list are more than 90 degrees away from the field of view, and are hence ignored.

*corrective action:* All the sources are out of field of view.



## 6 Input Files

1. (Mandatory) a FITS dataset with a table of source list, which contains the celestial coordinate columns in J2000 in unit of degrees.
2. (Mandatory) an image dataset, which provides the reference coordinate frame for the output mask file.
3. (Optional) a template filename used internally during the processing.

## 7 Output Files

1. (Optional) a mask image file. See the command-line parameter `maskset` and the main text for detail.

## 8 Algorithm

Check the number of rows in each input source-list

```
Do i=1, numOptSrcList ! (for each source-list)
  clone()
  doFilter(srcTable)
  call readColDataToFixed(srcTable, 'RA', srcPos(i,:) %raDeg)
  call readColDataToFixed(srcTable, 'DEC', srcPos(i,:) %decDeg)
  if (trim(RadiusStyle) == 'column') then
    call readColDataToFixed(srcTable, 'RADIUS', srcPos(i,:) %radiusArcsec)
  end if
End Do
```

```
call getArraySize(TemplateImageSetName, xiSize, yiSize)
allocate(maskImage(xiSize, yiSize))
call readWcs(TemplateImageSetName, wcs)
```

```
maskImage = .false.
```

```
Do i=1, numOptSrcList ! (for each source-list)
  Do j=1, numOptSrcRowsFiltered(i) ! Loop over the sources in each file
    call raDecToPixels(srcPos(i,j) %raDeg, %decDeg, wcs, xPixel, yPixel)
    where (position < radius) maskImage = .true.
  End Do
End Do
```

```
if (IsInverted) call invertMask(maskImage)
maskImageInt = 0
where(maskImage) maskImageInt = 1
```

```
call saveImage(maskImageInt, TemplateImageSetName, MaskSetName &
  , TemporaryImageSetName, wcs=wcs)
```



## 9 Comments

## References