



psfgen

January 27, 2025

Abstract

Generates a PSF image

1 Instruments/Modes

Instrument	Mode
EMOS1	Imaging
EMOS2	Imaging
EPN	Imaging

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

psfgen is designed to return the point spread function (PSF) of the EPIC cameras for a given position in the field of view. It allows the position to be specified in any of the common XMM-Newton coordinate systems optionally using an input image to translate between sky (XY) or celestial (RA,DEC) coordinates and telescope coordinates.

Within the XMM-Newton calibration system the PSF has been described by several different models, referred to as 'Accuracy Levels'. The current best model, called ELLBETA, is a 2-dimensional King function which includes instrumental distortions and the 'spokes' (see ref 1.). An alternative model, called MEDIUM, has been produced based on ray-tracing performed with the science simulator (SCISIM), while a simpler analytical model, EXTENDED, consists of a 1-dimensional King function. A further two models, LOW and HIGH, are included for completeness but have serious shortcomings and are not recommended.

The task calculates the PSF based on the input parameters and writes the result into an output FITS file. Due to limitations of the PSF implementation within the SAS, the resolution of the output image pixels are fixed to be 1 arcsecond by 1 arcsecond for the EXTENDED and ELLBETA modes and 1.1



x 1.1 arcseconds for MEDIUM mode. Integer rebinning of these images may be performed later by the *ftool fimgbin*.

3.1 Definitions

For the sake of brevity, the following terms will be used in this document:

- 'DET' coordinates refers to the CAMCOORD2 system, but expressed in units of *0.05 arcsec* instead of mm.
- 'RAW' coordinates refers to the PIXCOORD1 system.
- 'XY' coordinates refers to the projected sky coordinates of the source relative to the nominal pointing position, in units of 0.05 arcsec.
- 'TEL' coordinates refers to telescope coordinates (theta, phi), in units of arcsec (theta) and radians (phi).

4 Usage and Examples

psfgen is used to create an image of the point spread function (PSF) of one of the XMM-Newton EPIC telescopes.

4.1 Example 1: an on-axis PSF

```
psfgen instrument=EMOS2 energy=1000 coordtype=TEL x=0 y=0 level=MEDIUM output=psf.fits
```

This command produces an image called *psf.fits* containing the MEDIUM-level PSF of the MOS2 camera, at the optical axis (off-axis angle=0, azimuthal-angle=0) for photons of 1000 eV. The image is binned into 1.1 x 1.1 arcsecond square pixels (for MEDIUM mode) and has a default size of 199 x 199 pixels.

4.2 Example 2: an off-axis PSF image with 400 x 400 pixels

```
psfgen instrument=EPN energy=3000 coordtype=TEL x=600 y=1.0466 level=ELLBETA xsize=400  
ysize=400 output=psf.fits
```

This command produces an image containing the PSF of the EPIC-pn camera, at a position 600 arcseconds off axis and with an azimuthal angle of 1.0466 radians (60 degrees), for photons of 3000 eV. The image is binned into 1.0 x 1.0 arcsecond square pixels (ELLBETA mode) and has a size of 400 x 400 pixels.

4.3 Example 3: a PSF centred on a particular detector coordinate position

```
psfgen instrument=EMOS1 energy=3000 coordtype=det x=600 y=4200 xsize=400  
ysize=400 output=psf.fits
```

Produces a PSF centred on the detector coordinate (DETX=600, DETY=4200) for the MOS1 camera.



4.4 Example 4: Rotated PSF image

```
psfgen instrument=EPN energy=3000 coordtype=TEL x=600 y=1.0466 level=ELLBETA xsize=400 ysize=400
output=psf.fits rotate=162.0
```

As example 2 but with the PSF rotated by 162 degrees. Note that the rotation angle needed to allow comparison with a sky coordinate image would typically be PA degrees, where PA is the position angle of the observation, given by the PA_PNT keyword.

4.5 Example 5: Coordinates from an input image

A PSF for a particular source in a sky-coordinate image can be produced by supplying the name of the image and the position in sky (XY) or celestial (RA,DEC) or image pixel coordinates. e.g.

```
psfgen image=myskyimage.ds energy=3000 level=ELLBETA coordtype=POS x=27540 y=15620 output=psf.fits
```

or

```
psfgen image=myskyimage.ds energy=3000 level=ELLBETA coordtype=EQPOS x=221.7612 y=68.8493
xsize=400 ysize=400 output=psf.fits
```

```
psfgen image=myskyimage.ds energy=3000 level=EXTENDED coordtype=IMPIX x=239 y=186 output=psf.fits
```

where the first command produces a 199 x 199 pixel (default size) PSF image centred on the sky coordinate (XY) position (27540,15620), the second produces a 400x400 pixel image centred on the celestial position, RA: 221.7612, DEC: 68.8493 and the third produces an image centred on the position of the input image pixel (239,186). The task uses the world coordinate system information encoded in the header of the input image to identify the location of the PSF centre and uses the position angle of the observation (keyword=PA_PNT) to rotate the output image appropriately. Naturally, these values have to be correctly set in the input image, which they should be if they have been created by the SAS tasks **evselect** or **xmmselect**.

The PSF is made for the camera contained in the INSTRUME keyword in the header of the file mysky-image.ds. This may be overridden by setting the instrument parameter directly. e.g.

```
psfgen image=myskyimage.ds energy=3000 level=EXTENDED coordtype=IMPIX x=239 y=186 instrument=EPN
```

4.6 Example 6: create multiple PSFs

psfgen supplies a way to create multiple PSFs with different photon energies in a single call. This is useful when the user is interested in for example comparing different energy band PSFs at a particular position. To do this, supply **energy** with a list instead of a single value, e.g.:

```
psfgen image=source.fits coordtype=eqpos x=256.345 y=-34.209 output=source_psf.fits energy='600
2000 10000'
```

The file 'source_psf.fits' will contain 4 images, a PSF with energy 600eV, one at 2000eV, one at 10000eV



and their linear combination. The combined PSF is stored in the first extension and the individual energy PSFs in the subsequent extensions. The PSFs can be viewed using the command `ds9 'source_psf.fits[0]' 'source_psf.fits[1]' 'source_psf.fits[2]' 'source_psf.fits[3]'`.

4.7 Example 7: create a PSF weighted over several energies

To produce a weighted PSF use the command in example 4.6 with the `weight` parameter. For example,

```
psfgen image=source.fits output=source_psf.fits energy='600 2000 10000' weight='0.3 0.3 0.4' coordtype=pos x=17500 y=25000
```

will calculate the pixels in the combination PSF using

$$p_{comb}^{i,j} = 0.3 * p_{600}^{i,j} + 0.3 * p_{2000}^{i,j} + 0.4 * p_{10000}^{i,j} \tag{1}$$

The default weighting is 1 for each energy band and weights are normalised to one within the program.

5 Parameters

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

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

Event List Parameters

image	no	string	NULL	valid file name
--------------	----	--------	------	-----------------

A name of a fits file that was generated from `xmmselect/evselect`. No image is used if NULL.

level	no	string	ELLBETA	valid accuracy level value
--------------	----	--------	---------	----------------------------

Accuracy level: LOW, MEDIUM, EXTENDED, HIGH, ELLBETA

binning	no	int	0	valid binning value
----------------	----	-----	---	---------------------

On-chip-binning factor. Only works for certain camera mode

coordtype	yes	string	TEL	valid coordinate value
------------------	-----	--------	-----	------------------------

Coordinate Type: TEL (theta,phi), DET (detector coords), RAW (chip coords), POS (X,Y - sky coordinates, EQPOS (Ra, Dec), IMPIX (image pixel). The last three (POS,EQPOS,IMPIX) can only be used if an image is supplied.

x	no	real	0	
----------	----	------	---	--

The X coordinate of the centre of the PSF. If `coordtype = TEL`, then `x` is the off-axis angle (theta) which is expected in units of arcseconds. If `coordtype = EQPOS` then `x` is the right ascension in units of degrees. For other coordinate systems it should be entered in units of pixels.

y	no	real	0	
----------	----	------	---	--

The Y coordinate of the centre of the PSF. If `coordtype = TEL`, then `y` is the azimuthal angle (phi) which is expected in units of radians. If `coordtype = EQPOS` then `y` is the declination in units of degrees. For other coordinate systems it should be entered in units of pixels.

ccdnr	no	int	0	valid CCD number
--------------	----	-----	---	------------------



CCD number. This is mandatory if coordtype=RAW.

energy	no	real list	3000.0	[100 : 150000]
---------------	----	-----------	--------	----------------

Energy value list. Note that there is a minimum-maximum range

weight	no	real list	1.0	
---------------	----	-----------	-----	--

Weighting to construct summation PSF from multiple PSFs with different energies.

rotate	no	real	0.0	($-1^\circ : 360^\circ$)
---------------	----	------	-----	----------------------------

Rotation angle of the PSF in units of degrees. If rotate is non-zero then the PSF will be rotated by this angle. The rotation is in the same direction as the position angle (PA). If rotate is not set but an image has been given via the **image** parameter then the PSF will be rotated by the position angle (PA_PNT) stored in the image file header. If rotate is non-zero then it will override the value in the input image.

xsize	no	int	199	[0 : 1024]
--------------	----	-----	-----	------------

Width of the X-axis output image in pixels.

ysize	no	int	199	[0 : 1024]
--------------	----	-----	-----	------------

Width of the Y-axis output image in pixels.

output	no	string	psfimage.fits	valid file name
---------------	----	--------	---------------	-----------------

File name of the PSF image.

6 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.

noCCDNumber (*error*)

Raw coordinates have been requested but no CCD has been entered.

noImageSupplied (*error*)

When sky or image coordinates are entered the task needs an input image to find the detector position.

invalidCoordType (*error*)

An invalid coordinate type has been requested.

InvalidEnergyValue (*error*)

The energy level is not given. (This should not happen normally because it has default value. If it happens, it indicates a unknown bug.)

InvalidAccLevel (*error*)

Unknown or incorrect accuracy level. (This should not happen normally because it has default value. If it happens, it indicates a unknown bug.)

InvalidPA (*warning*)

Position angle is not supplied.

corrective action: Rotation is not employed or set to 0. Continue



7 Input Files

1. EPIC image file generated by `xmmselect/evselect` (optional)

8 Output Files

1. One file containing PSF image(s) of pixel size 1 by 1 arcsecond (`ELLBETA`, `EXTENDED`) or 1.1 by 1.1 arcseconds (`LOW`,`MEDIUM`,`HIGH`) depending on the PSF level selected. The image is centred on the selected celestial position or on RA=0, DEC=0 if coordinates were entered in TEL,DET or RAW form.

9 PSF accuracy levels

There are several implementations of the PSF which can be selected with the `level` parameter.

- *ELLBETA* This is the most complex PSF implementation and is used in the processing pipeline for source searching and by default in the task `arfgen` to calculate the effective area. See [1] for more details.
- *EXTENDED* A model based on the King function which was the forerunner of the *ELLBETA* model.
- *MEDIUM* Based on images generated by the XMM-Newton event simulator (`SCISIM`).
- *HIGH* An early simple PSF which isn't recommended for general use.
- *LOW* An early even simpler model which is also not recommended for general use.

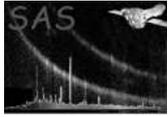
10 Comments

- `psfgen` produces the PSF image(s) at the specific position in TEL.
- `psfgen` creates a PSF image for each energy value in energy list and one summation PSF image using the weightings. Therefore, if there n energy values in the list, $n + 1$ PSF images will be created, except for $n = 1$, which only one image will be created.
- The first PSF image is always the summation PSF image. Other PSF images are named with their energy values. All PSF images are saved in one fits file.

11 Developernotes

11.1 CAL usage

1. `calServer.setInstrument()`



2. `calServer.setCcf()`
3. `calServer.CcfAccessLog()`
4. `calServer.state()-i.ccd()-i.set()`
5. `calServer.pushState()`
6. `calServer.popState()`
7. `calServer.setState()`
8. `calServer.getAtom(ModeParameters *)`
9. `calServer.getAtom(GeometryDataServer *)`
10. `calServer.getAtom(MiscDataServer *)`
11. `calServer.getAtom(Boresight *)`
12. `calServer.getAtom(PsfDataServer *)`
13. `PsfDataServer.psf()`
14. `Psf.setPosAngle()`
15. `Psf.image()`

12 References

- [1] Read, A., Rosen, S., Saxton, R. & Ramirez, J. 2011, A&A 534, 34

References