



psfgen

January 12, 2017

Abstract

Generates PSF from ccf

1 Instruments/Modes

Instrument	Mode
EMOS1	All Mode
EMOS2	All Mode
EPN	All Mode

2 Use

pipeline processing	yes
interactive analysis	yes

3 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 degree (phi).



4 Usage and Examples

psfgen can be used as coordinate converter or PSF generator. In order to execute **psfgen** correctly, environment variables **SAS_CCF** and optionally **SAS_ODF** should be set appropriately. However, the user can specify a particular CIF file in parameter **ccf** at any time.

The following sub-sections show examples of usage.

4.1 Example1: coordinate converter

If output is not set, **psfgen** only prints out the following parameters values in order:

- instrument
- theta
- phi
- rotate
- coortype
- xsize
- ysize

In such cases, **psfgen** acts as a coordinate converter that converts DET/RAW/XY to TEL. The coordinate is specified by **region** or **image**. For example,

```
psfgen region='(DETX, DETY) IN box(-12962,-579,1400,1600)' instrument=M2
```

will give the following output:

```
instrument=M2 theta=648.114 phi=2.51136 rotate=270 coortype=DET xsize=128 ysize=146
```

which convert (-12962, -579) in DET to (648.114, 2.51136) in TEL.

Please be noted: because **region** only gives coordinate type and values, **ccdnr** should be given for RAW input and **rotate** should be given for XY input.

If there is a **xmmselect/evselect** image (in fits format) that contains all the necessary information, the application can be simplified as this:

```
psfgen image=selected_image.fits
```

Beware of that the input parameters (e.g. **rotate**) will overwrite the information in the image. Thus parameters can be used to supply additional information that is not in the image.

4.2 Example2: create one PSF

When a output file name is given in parameter **output**, **psfgen** will create a PSF and save it to the named file (overwrite). User should specify the required energy band (**energy**) and accuracy level (**level**). The geometry parameters should be supplied as demonstrated in Example1 4.1. For example,



```
psfgen region='(DETX, DETY) IN box(-12962,-579,1400,1600)' instrument=M2 output=m2psf.fits
level=ELLBETA energy=4500
```

will create the PSF for 4500eV energy level on EMOS2 centred at TEL(648.114, 2.51136) with accuracy level ELLBETA and save it to 'm2psf.fits'.

All the setting parameters have default values (see Parameters section 6). So

```
psfgen output=psf.fits
```

works and 'psf.fits' is a PSF for 3000eV energy level on EMOS1 centred at TEL(0, 0) with accuracy level MEDIUM.

If the user is interested in the PSF of a particular source in MEDIUM level with energy level 6000eV. The easiest way is to select and save the source image to a fits file (e.g. 'source.fits') using **xmmselect/evselect**. And then use the following command:

```
psfgen image=source.fits output=source_psf.fits energy=6000
```

The file 'source_psf.fits' is the required PSF with the same size as 'source.fits' with the PSF centre in the middle. Therefore, when selecting the source, the centre of the source should be in the middle of the source image as well.

4.3 Example3: create multiple PSFs

psfgen supplies a way to create multiple PSFs with different energy levels 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, like this:

```
psfgen image=source.fits output=source_psf.fits energy='600 2000 10000'
```

The file 'source_psf.fits' contains 4 PSFs, i.e. PSF with 600eV, PSF with 2000eV, PSF with 10000eV and their linear combination (see 10). The PSFs can be viewed using command **ds9 source_psf.fits[0] source_psf.fits[1] source_psf.fits[2] source_psf.fits[3]**.

psfgen also allows the user to give the weightings for the linear combination via parameter **weight**. For example,

```
psfgen image=source.fits output=source_psf.fits energy='600 2000 10000' weight='0.3 0.3 0.4'
```

will calculate the pixels in the combination PSF using

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

The default weighting is 1 for every energy band.

5 Description



6 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.

useodf	no	boolean	no	yes no
---------------	----	---------	----	--------

A boolean switch determining whether the procedure uses ODF.

region	no	string	(DETX, DETY) IN circle(0, 0, 2000)	valid region specifica- tion
---------------	----	--------	---------------------------------------	---------------------------------

region specification that indicates the coordinats type and region type and size. *region* can overwrite information supplied in *image*.

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

CCD number for interested area. Only useful for RAWX/RAWY coordinates. If *ccdnr* > 0 specified, it will overwrite the information from *image*. If CCD number can not be detected by *image*, default to 1.

node	no	string	PRIMARY	valid node value
-------------	----	--------	---------	------------------

CCD readout node: PRIMARY, REDUNDANT

cammode	no	string	PrimeFullWindow	valid camera mode value
----------------	----	--------	-----------------	----------------------------

Camera mode. See Xmm.h for details

filter	no	string	Open	
---------------	----	--------	------	--

Filter option. See Xmm.h for details

ccdtemp	no	real	120.0	
----------------	----	------	-------	--

CCD temperature (K)

camtemp	no	real	120.0	
----------------	----	------	-------	--

Camera temperature (K)

temptrack	no	boolean	no	yes no
------------------	----	---------	----	--------

Dis/enable retrieving temperature from HK data

starttime	no	time	1999-12-10T14:32:00.000	valid time specifica- tion
------------------	----	------	-------------------------	-------------------------------

Observation start time. If *image* is given, the start time is specified in the image file unless *duration* > 0.

duration	no	real	0	
-----------------	----	------	---	--

Observation duration in seconds. If *duration* is 0 and *image* is given, both *starttime* and *duration* are specified by the information in the image file. If *duration* is 0 but *image* is not given, default to 3000.

level	no	string	MEDIUM	valid arccuracy level value
--------------	----	--------	--------	--------------------------------

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

parameter	no	int	0	0
------------------	----	-----	---	---



Auxiliary parameter, always 0. Used in the future

random	no	boolean	yes	yes no
---------------	----	---------	-----	--------

En/disable randomize.

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

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

ccf	no	boolean	NULL	
------------	----	---------	------	--

CCF file path and name. NULL if not specify CCF file.

ccflog	no	boolean	no	yes no
---------------	----	---------	----	--------

Print CCF Access Log information to standard error.

coortype	no	string	XY	valid coordinate value
-----------------	----	--------	----	------------------------

Coordinate Type: XY, DET, 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	
---------------	----	-----------	-----	--

Energy weighting to construct summation PSF

theta	no	real	-1	[-1 : 1800]
--------------	----	------	----	-------------

Off-axis angle in arcsec. Note that there is a minimum-maximum range. *theta* < 0 means it is decided by *region* or *image*. If it can not be decided, default to 0.

phi	no	real	-1	[-1° : 360°]
------------	----	------	----	--------------

Azimuth angle. (theta/phi) are given in the TELCOORD reference system indicating location of the PSF center. *phi* < 0 means it is decided by *region* or *image*. If it can not be decided, default to 0.

rotate	no	real	-1	(-1° : 360°)
---------------	----	------	----	--------------

Rotation angle of the PSF (PA). When *rotate* = -1, the PA value is decided by *image*. If *image* is not given or $-1 < rotate < 0$, rotation angle is not used. A warning will be displayed.

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

PSF width in pixels, 0 means decided by the *image* or *region*. If can not be decided, default to 199.

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

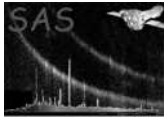
PSF height in pixels, 0 means decided by the *image* or *region*. If can not be decided, default to 199.

output	no	string	NULL	valid file name
---------------	----	--------	------	-----------------

File name of the PSF image. NULL means no output, only display the theta, phi, rotation, xsize and ysize.

7 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.



InvalidInstState (*error*)

The instrument state is not correct.

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.)

NoCifSpecified (*error*)

The file specified in parameter `ccf` is not a valid CIF.

InvalidPA (*warning*)

Position angle is not supplied.

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

CCFLog (*warning*)

Involved CCF details are printed out.

corrective action: Continue

UserOwnCCF (*warning*)

Will use a particular CIF file instead of the one specified in `SAS_CCF`.

corrective action: Continue

InvalidPosWCSInfo (*warning*)

One or more of the global attributes `REFXCTYP`, `REFXCRPX`, `REFXCRVL`, `REFXCDLT`, `REFYCTYP`, `REFYCRPX`, `REFYCRVL`, `REFYCDLT` are not presented when converting in XY.

corrective action: Provide a dummy WCS setting, for the POS coordinate system and continue.

InvalidRegionInfo (*warning*)

Region information is incorrect or not supplied.

corrective action: Set to default '`(DETX, DETY) IN circle(0, 0, 2000)`' and continue.

UnknownModeString (*warning*)

If the observing mode is not recognised then the software assumes that the common Prime-FullWindow mode was in use.

corrective action: PrimeFullWindow

InvalidCCDNR (*warning*)

The CCD number is not supplied when converting in RAW.

corrective action: Set to default and continue

8 Input Files

1. event image file generated from `xmmselect/evselect` to produce PSF (optional)

9 Output Files

1. PSF image(s) (optional)



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()-iccd()-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()`

References