evselect

April 16, 2023

Abstract

evselect: event selection and creation of products

1 Instruments/Modes

All event-list generating modes:

Instrument	Mode
EPIC MOS	IMAGING, TIMING
EPIC PN	IMAGING, TIMING, BURST
RGS	SPECTROSCOPY, HIGH TIMING
OM	FAST

2 Use

pipeline processing	yes
interactive analysis	yes

In the interactive environment optionally used in combination with the dedicated GUI front-end **xmmselect**.

3 Description

Among the suite of individual SAS tasks **evselect** is of central significance. It serves two complementary purposes

1. Filter event list data controlled by user-specified selection criteria
2. Extract images, spectra, and time series from the filtered event list obtained in the first step. The filtered list is selectable for output as well.

The availability of both features within a single task allows the user to generate useful products from an event list (e.g., create a rates file in the energy band 0.5 – 3 key from the X-ray source lying within a circle of 10 arcsec radius in the center of the FOV) in a convenient manner.

3.1 Filtering stage (selector)

The filtering process is carried out on an event-by-event basis controlled by user-specified selection criteria which take the form of a single boolean expression. The expression can contain symbolic names of "intrinsic" event attributes, and numeric constants combined with logical and arithmetic operators and functions. Events for which the expression does not evaluate to true will be marked as invalid or discarded from the data set and shall not be considered in the product extraction.

evselect supports selections based on intrinsic event attributes. These attributes are present in the input event list table as corresponding columns, the names of which are to be used in the selection expression. This group further falls into the following sub-categories:

1. spatial selections in raw pixel (`RAWX/RAWY`), physical detector (`DETX/DETY`) or sky pixel (`X/Y`) space, or indeed in any spatial coordinate system (e.g. `TELX/TELY`). Special pre-defined region shapes exists such as CIRCLE, ELLIPSE, ANNULUS, BOX, POLYGON. Spatial filtering with mask images is possible as well.
2. Celestial coordinates (`RA/DEC`) are available for the CIRCLE, BOX and ANNULUS shapes,

   ```
   e.g. (RA,DEC) in CIRCLE(253.231,-2.052,0.0166)
   or (RA,DEC) in BOX(121.234,23.456,0.125,0.125,0)
   ```

   where all values are in units of degrees. These values are pre-processed into sky pixels.
3. A special construct `IMAGE` may be used to extract data from a spatial region which is defined by non-zero pixels in a supplied FITS image. This image may have axes in DETX/DETY coordinates or in celestial coordinates. Any celestial image with standard WCS header keywords and integer or boolean values is acceptable. Note that the coordinate system should be epoch 2000 (FK5), earlier 1950 epoch (FK4) images will give a significant offset.

   ```
   e.g. expression='IMAGE(myHSTimage.fits)'
   ```

   This expression is pre-processed into a BOX, encompassing the whole image logically ANDed with a MASK, e.g.

   ```
   '(X,Y) in BOX(25000,25000,15000,15000,0) && MASK(myHSTimage.fits)'
   ```

4. energy selections in the form of one or more interval specifications in either PHA or PI space.
5. time selections in the form of
   (a) one or more interval specifications
   (b) one or more Good Time Interval (GTI) files created with **tabgtigen** or any other SAS tasks which creates GTI files as output
6. any other event attributes which are present (e.g. `PATTERNID`, or `CCDID`, or `WAVELENGTH` in case of RGS)

3.1.1 Selection Expression

The selection expression is input as a string using the parameter `expression`, which is set to the value "true" by default. One can also filter an event list based on the data subspace information stored in another data block. The name of this block is specified with the `dssblock`. Which type of filter is used (expression or data subspace) is controlled with the `filtertype` parameter, with is set to "expression" by default.

For a description of the syntax rules the boolean selection expression must adhere to, as well as example expressions, please consult the documentation of the SAS package **selectlib**.

3.2 Extraction Stage (extractor)

3.2.1 Products creation

The selection stage yields a modified event list in which the events which do not satisfy the selection expression are removed or appropriately flagged. From this list the extraction stage creates images, spectra, and rates files. These products are aimed to adhere to OGIP standards as closely as possible. Please note, however, that it is likewise possible to create data sets which do not map onto any standard OGIP file type, e.g. PHA-versus-wavelength image for RGS. Evselect also provides for extracting a histogram from a column of an event list. This product is for general use and does not conform to any OGIP standard.

The product creation process can be controlled via various command line parameters such as binning factors, cut-off boundaries, etc. Only one product file of each kind can be constructed per **evselect** run. If more than one file is required, e.g. time series files in different energy bands, **evselect** has to be run repeatedly. The parameters `blockstocopy` and `attributestocopy` can be used to copy selected blocks and attributes from the input file to the output product files. See the following subsections for a more detailed description of the production of the individual data product types.

For binned products, binning of a given column is in general controlled by a minimum value, a binning factor, and a maximum value. For integer-valued columns, these 3 parameters are all required to be integer-valued. In this case, any event with a value equal to the minimum value will be put into the first bin, and any event with a value equal to the maximum value will be put in the last bin. Thus for example, if the minimum value is 4, and the binning factor 3, then events with values of 3, 4 and 5 will be put into the first bin.

Information about the filtering criteria used to produce the product files will be stored in data blocks and attributes of the files using a data subspace model as described in [1]. See also documentation on the **dsslib** package.

3.2.2 Event List Output

In addition to the above product files, the result of the filtering stage can be propagated into the output as well. This can be done in four different ways depending on the setting of the two task parameters `flagcolumn` and `filteredset`. The first parameter specifies the name of a column in the table which can be used to flag an event. In this non-destructive mode, non-selected events are merely flagged by setting a corresponding bit in the specified column, which must be integer-valued. The bit posistion to be manipulated must be set using the parameter `flagbit`. If the `flagcolumn` parameter is not set,

non-selected events are physically removed, and thus will not be present in the final filtered event list. If non-destructive filtering is selected and the named flag-column does not exist, it will be created.

Thus setting of the `filteredset` parameter determines whether the input event list is modified according to the above two schemes or a new event list file is created accordingly. The following table provides a quick overview over the four possible cases:

	`flag column not set`	`flag column set`
`output file name not set`	the input file is modified; non-selected events will be removed	the input file is modified by adding an integer flag column indicating each event's selection status
`output file name set`	input file is not modified; selected events will be copied to new output file	input file is not modified; all events (selected and non-selected) get copied to new event list file with an integer flag column

If the `filteredset` parameter is not set, then the parameter `keepfilteroutput`, which is set to false by default, will prevent one from modifying the input table. If one wishes to modify the input file (either destructively or by flagging) one needs to set `keepfilteroutput` to true. Note that when **evselect** produces an output table file, it does so by first creating an exact copy of the input table file, and then preforming all filtering operations on the copy. Thus any extensions in the input file will also appear in the output file.

3.2.3 Spectrum Extraction

Spectrum extraction is turned on by setting the parameter `spectrumset`, which specifies the name of the output spectrum file. A spectrum, as defined here, is an OGIP compliant FITS file, made by histogramming the energy column of an input table. The energy column must be integer-valued, and any specified binning factor or energy ranges must be integer-valued as well. The column to histogram is specified with the parameter `energycolumn`. More general histogramming of real-valued columns can be performed using the histogram extraction utility in **evselect**. If a real-valued column is fed to the spectrum extractor, then a warning is issued, and all values in the column are truncated to integers before being accumulated into a spectrum. Note that the parameter `energycolumn` is also used in light curve extraction (see below).

evselect performs true binning of spectral files, as opposed to OGIP grouping. The binning factor is set via the parameter `spectralbinsize`, which is set to 10 by default. The energy range for the spectrum is determined in one of three ways. First, if the parameters `specchannelmin` and `specchannelmax` are set, they are used for the spectral range. If these values are not set, then the input data set is queried for the `TLMIN` and `TLMAX` attributes of the energy column, and if available these are used for the energy range. Finally, if no other information is available, then the actual range of the energy column data itself is used as the minimum and maximum values for extraction of the spectrum. Note that if `specchannelmin` or `specchannelmax` exceed the allowed data range set by `TLMIN` and `TLMAX`, then a warning message is issued, and the range is adjusted to be within the allowed limits. If the parameter `ignorelegallimits` is set to true, then the keywords `TLMIN` and `TLMAX` are ignored, that is they are treated as if their values are not set. All range and binning parameters are in the units of the column from which the data is being extracted.

As the energy range and binning factor of the spectrum is needed by down-stream software (for example

to produce an RMF file), this information is written to keywords in the spectrum file, using a WCS style specification. The keyword `SPECPIX` gives the value of the reference channel in the spectrum. The keyword `SPECVAL` then is set to the value of the energy column which corresponds to the center of the spectrum's reference channel. The keyword `SPECDELT` is set to the binning factor.

For XMM only data, one can produce a spectrum of rates rather than counts, using the exposure information available in the XMM event files. In this case a "weight" column must also be specified, and is used to specify the weight given to each event in the event list. This is activated by setting the name of the weighting column using the parameter `zcolumn`. An error on the weight can also be specified using the `zerrorcolumn` parameter. If this parameter is not set, then the error on the weight is set to the value of the weight itself. Each weight is divided by the exposure time for the particular CCD from which the event originated, and these values are summed on a bin by bin basis to create the spectrum of rates. See the section on exposure information for more details on how the CCD exposures are calculated.

evselect does not calculate the `BACKSCAL` keyword, but simply sets its value to unity. The task **backscale** calculates a proper value for this keyword.

The downstream matrix creation software, **arfgen** and **rmfgen**, expect to receive a spectrum extracted with a limited set of event PATTERNS and over a fixed set of PI channels. If a non-standard spectrum is attempted to be produced the task throws a warning. The warning can be avoided by setting `nonStandardSpec`=true.

3.2.4 Image Extraction

Image extraction is enabled by setting the name of the output file using the parameter `imageset`. The name of the columns to use for the x and y axes of the image are set using the parameters `xcolumn` and `ycolumn`. The range of the columns used for image extraction is determined using the same algorithm as for the energy column in spectral extraction, and is controlled and determined independently for the x and y columns. The parameters used for the x column are `ximagemin` and `ximagemax`, with similarly named parameters for the y column.

There are two options available for binning of images. By default, the binning factor for the image is set such that the image will be of a certain size. The size of the image is controlled by the parameters `ximagesize` and `yimagesize`, and is set to 600×600 by default. Note that non-integral binning of integer-valued columns is not allowed. Thus when extracting an image from integer-valued columns using `ximagesize` and `yimagesize`, the image binning factor will be rounded up to the nearest integer number, so that the actual extracted image may be smaller than what was requested. It is also possible under these circumstances that the x and y bin sizes could be set to different values, even though a square image was requested. In this case a square image can be produced by setting the parameter `squarepixels` to true. Then the x and y bin sizes will both be set to whatever is the larger of the two values.

Alternatively, one can control the binning factor directly by setting the parameters `ximagebinsize` and `yimagebinsize`.

By default, an image is accumulated simply as counts of the number of rows (events) falling within each pixel. However, if the parameter `zcolumn` is set, then the image is accumulated by summing up the values of the specified column for rows that fall within each pixel.

By default, evselect tries to determine the best data type for the image created, under the assumption that the smallest data type possible should be used. If `zcolumn` is not set, an integer-valued image will be created. If `zcolumn` is set, a real-valued image will be created. The default type of image can be over-ridden by setting the parameter `imagedatatype`.

Any World Coordinate System (WCS) information available for the columns used in image extraction will be propagated to the output image. If no WCS information is available for a column, then the `CTYPE` keyword for this axis of the image is set to the name of the column, and the other WCS keywords are set to reflect the range and binning of this column, as is done for `SPEC*` keywords in spectral extraction. If the image extraction columns do have WCS information, then one can request **evselect** to shift the image so that it is centered on a particular point in the sky. This is activated by setting the coordinate center using the parameters `raimagecenter` and `decimagecenter`. Coordinates must be given in decimal degrees.

As well as the standard WCS keywords described above, **evselect** also writes keywords to the image describing the "physical" coordinate system, ie the native coordinate system of the columns used for extracting the image. In order to maintain backward compatibility, there are currently two sets of redundant keywords written for this purpose. The first set uses the `LTMn_m` and `LTVn` keywords to specify the transformation, where n and m can take on the values of 1 or 2. The transformation from the physical (X_{ph}, Y_{ph}) to the image (X_{im}, Y_{im}) coordinates is

$$\begin{pmatrix} LTM_{11} & LTM_{12} & LTV_1 \\ LTM_{12} & LTM_{22} & LTV_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{ph} \\ Y_{ph} \\ 1 \end{pmatrix} = \begin{pmatrix} X_{im} \\ Y_{im} \\ 1 \end{pmatrix}.$$

As there is no rotation in the transformation, $LTM_{12} = LTM_{21} = 0$.

The second method uses the multiple WCS conventions to specify the physical coordinate system for the image. In this scheme a second set of WCS keywords is written, except each of the standard keywords has a letter appended to it, which we have chosen to be "L". In addition we specify the following keywords to identify this coordinate system:

```
WCSNAMEL = 'PHYSICAL'
WCSAXESL = 2
```

Keyword(s) which give the CCD integration time in seconds, are added to the PRIMARY header of EPIC images. These are:

```
FRMTIMcc (EPIC-MOS; where cc is the CCD number)
TIMEDEL (EPIC-pn)
```

3.2.5 Light Curve Extraction

Light curve extraction is controlled by setting the parameter `rateset`, which is the name of the output file containing the ligh curve. Light curves are OGIP compliant histogram files of counts as a function of time. The name of the time column is set using the parameter `timecolumn`. Unlike spectrum extraction, light curve extraction supports real-valued columns. The range of times used is determined using the same algorithm as for the energy column in spectral extraction. The parameters used to control this are `timemin` and `timemax`. The bin size is set with the parameter `timebinsize`. All range and binning parameters are in the units of the column from which the data is being extracted.

The OGIP standard allows for several variations on how a light curve can be stored in a FITS table. In the first variation, only a `COUNTS` column is required in the table. Timing information is stored in the keywords `TSTART`, `TSTOP` and `TIMEDEL`. Each row in the table represent a sequential time bin, with the first bin starting at `TSTART` and a bin size of `TIMEDEL`. This is the type of light curve which evselect produces by default.

In a second variation, the table also contains a `TIME` column, giving the time of each bin in the light curve. Production of this type of light curve is enabled in evselect by setting to true the parameter `maketimecolumn`.

In either of the variations above, it is possible to replace the `COUNTS` column with a `RATE` column. In addition, an `ERROR` column, giving the error on the rates, is also supplied. Use of the `RATE` column is activated by setting the `makeratecolumn` parameter to true. It is false by default.

The OGIP standard also specifes the keywords `CHANMIN`, `CHANMAX` and `CHANTYPE`, which describe the energy range of the events used to extract the light curve. These keywords are set in **evselect** by querying information from the input table column. As with images, rather than just accumulating counts in each bin of a light curve, one can sum up the values of a another column in the input table, by specifying the name of the column in the parameter `zcolumn`. In the case of light curves one can also specify the error on the value of the z column. A column in the input table containing this information should be specified using the parameter `zerrorcolumn`. If `zerrorcolumn` is not set, then the values in `zcolumn` are used as the error values.

Note that no exposure correction of time bins is done in evselect. Thus, for example, the possibility that a time bin might partially overlap with a GTI is not corrected for. Exposure correction of light curves is handle by the task **epiclccorr**.

3.2.6 Histogram Extraction

evselect also contains the facility to produce a general FITS histogram product, which does not have the constraints imposed on it by the OGIP compliance of light curves and spectra. This is enabled by seting the histogram output file, which is specified with the parameter `histogramset`. The column from which to extract the histogram is set with the parameter `histogramcolumn`.

The range of the histogram is set using the same algorithm as for setting the energy range in a spectrum, using the parameters `histogrammin` and `histogrammax`. The bin size is set with the parameter `histogrambinsize`. Real valued columns are supported. All range and binning parameters are in the units of the column from which the data is being extracted.

When a histogram table is created, it contains two columns, one named after the column from which it was extracted, and another named `COUNTS`. As with light curves, one can accumulate the values of a column from the input table, instead of just counts, by setting the parameter `zcolumn`. In this case, an `ERROR` column is also created in the output table, and its value is based on the parameter `zerrorcolumn`, as with light curves.

3.3 Data Subspace Support

evselect supports the reading and writing of data subspace information into a data set, which allows one to track the filtering history of a file. A data set with data subspace information can be used as an input to evselect, in place of a filtering expression, using the parameter `dssblock`. **evselect** will also take the input filtering expression and convert it into a data subspace specification, which will then be written to all output files.

*Not all possible filtering expressions allowed by **selectlib** can be converted to a data subspace specification. See the **dsslib** documentation for a detailed description of the capabilities of the data subspace parser.*

The writing of data subspace information is enabled by default, and controlled by the parameter `writedss`.

The data subspace also has a cleaning option, which will eliminate any components from the data subspace which do not select any events in the event list. This can result in the loss of important filtering information, and thus it is recommended that data subspace cleaning be used by experts only. The parameter for enabling data subspace cleaning is `dssclean`, and is set to false by default.

3.4 Exposure Information

evselect will optionally update exposure information for event lists and and product files (no exposure information is written to histograms). This can be done only for finalized XMM event lists, and if the event list does not have the right structure, a warning will be issued. By default exposure updating is turned on. It can be turned off by setting the boolean parameter `updateexposure` to false. Note that data subspace writing must be turned on in order for exposure correction to work correctly (`writedss` set to true).

For EPIC event lists, **evselect** uses information in the data subspace and in the EXPOSU*nn* extensions of the event lists to update the `ONTIME`, `ONTIME`*nn*, `LIVETIME`, and `LIVETI`*nn* keywords. Here *nn* refers to the relevant CCD number. For spectrum and rate files, **evselect** determines the `EXPOSURE` keyword by doing an average of the live times of the CCDs in the selection region, weighted by the number of selected events in each CCD. For images, the the `EXPOSURE` keyword is set to the maximum value of all the available `ONTIME`*nn* values.

As the EXPOSU*nn* extensions for an event list can be quite large, evselect can filter these extensions, using the same timing information as is applied to events from the corresponding CCD. This option is controlled by the parameter `filterexposure`, which is set to true by default.

Exposure correction cannot be performed for the RGS within **evselect**. Instead, **evselect** simply propagates exposure information for RGS to any spectral files produced. A warning is issued if any TIME filtering is done for RGS event lists, as this would invalidate the RGS exposure information.

3.5 Usage Examples

- Extract a light curve from the table `event.fits:EVENTS`, using the default column name `TIME`. Write the light curve to the file `rate.fits`.

  ```
  evselect table='event.fits:EVENTS' rateset=rate.fits
  ```

- Extract an image from the table `event.fits:RAW EVENTS`, using the default columns `RAWX`, `RAWY` when the value of `RAWX` is greater than 50. Write the output to the file `image.fits`. The results of the filtering is not saved.

  ```
  evselect table='event.fits:RAW EVENTS' expression='RAWX>50' imageset=image.fits
  ```

- Extract a spectrum from the first extension in the file `event.fits`, in a circle in `X`, `Y` space. Write the spectrum to the file `myspec.fits`, and write the filter results back into the input file. Accumulate the spectrum using the default energy column `PHA`:

  ```
  evselect table='event.fits' expression='circle(152.5,194.0,14.5,X,Y)'
    spectrumset=myspec.fits keepfilteroutput=true
  ```

- Filter the table `event.fits:EVENTS`, using the data subspace information in the block `dss.fits:SPECTRUM`. Rejected events are removed from the output file, which is written to `good_events.fits`.

  ```
  evselect table='event.fits:EVENTS' dssblock='dss.fits:SPECTRUM'
    filteredset='good_events.fits'
  ```

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

Event List Parameters

table	yes	table specifier		event list table specifier

A table specifier which must point to an event list table in a data set. It must be in either of the forms `setname` or `setname:tableid` where `setname` must be the name of an existing data set and `tableid` the name of a table in the specified data set. If the first form, `setname`, is used, the event data are sought in the *first* block of the named data set.

filteredset	no	string	`filtered.fits`	valid file name

The name of the file to which the filtered event list is to be written. If this parameter is set, then `keepfilteroutput` is automatically set to true.

keepfilteroutput	no	boolean	false	true|false

A boolean switch determining whether the result of the filtering process is to be kept or not. In the former case, the parameters `flagcolumn` and `filteredset` specify the form of the output from the filtering stage (see Sect. 3.2.2).

flagcolumn	no	string	"EVFLAG"	column name

The name of the integer column used to store the selection status of a table row in non-destructive (see Sect. 3.2.2 for details) filtering mode, which is activated if this parameter is set. If the column does not exist, it will be created with type *int8*, otherwise the existing column will be re-used. If a table with a flag column which has been added by a previous non-destructive **evselect** run is subsequently filtered in destructive mode (see Sect. 3.2.2 for details) the flag column gets deleted at the end of the filtering process. In case the flag column has not been created by **evselect**, the bit specified by the parameter `flagbit` (see Sect. 4) is cleared at the end.

flagbit	no	integer	-1	-1–31

In non-destructive (see Sect. 3.2.2 for details) filtering mode: the position of the bit in the integer column named via the `flagcolumn` (see Sect. 4) parameter used to store the row selection status. 0 signifies the least- and the default value -1 the most-significant bit (i.e. 31 for *int32*, 15 for *int16*, 7 for *int8*) respectively. This bit being *set* flags an event as *selected*.

Filtering Parameters

expression	no	string	"true"	valid filter expression according to **selectlib** syntax rules

This is the expression controlling the filtering process. It's length is in principal unlimited, please note however, that this is not true for the length of a command line in most UNIX shells. With this limit being in the range of several kByte, it should not pose a real problem in practice, though. This parameter and the `dssblock` parameter cannot be used at the same time. If both are specified, the one listed on the command line first takes precedence.

dssblock	no	block specifier		

Specifies a block containing data subspace information. This is converted into a filter expression which is used to filter the event list. Note that this parameter and the `expression` parameter cannot be used at the same time. If both are specified, the one listed on the command line first takes precedence.

General Output Product Parameters

writedss	no	boolean	true	true|false

Controls the writing of data subspace information to the output data files.

cleandss	no	boolean	false	true|false

Controls the use of data subspace cleaning, which deletes components from the data subspace which select no events from the event list. Recommended for use by experts only.

updateexposure	no	boolean	true	true|false

Update exposure information in event lists and in spectrum files. For the event lists this means that new values of the keywords LIVETIME, LIVETI*nn*, ONTIME, and ONTIME*nn* will be calculated, where *nn* is the CCD number. For spectrum files, the EXPOSURE keyword is calculated; note that this part of evselect is XMM specific

filterexposure	no	boolean	true	true|false

Filter EXPOSU*nn* extentions of an EPIC event list with the same time filters applied to events in the corresponding CCD; XMM specific.

blockstocopy	no	string list		

List of extensions in the input data set to be copied to any extracted data product file. This does not apply to output event lists.

attributestocopy	no	string list		

List of attributes in the input table to be copied to any extracted data product table or array.

energycolumn	no	string	`PHA`	name of existing column

Name of energy column, used for spectral extraction and for determining the energy range for light curve extraction.

zcolumn	no	string	`WEIGHT`	name of existing column

Column from which to populate image, spectrum or light curve values.

zerrorcolumn	no	string	`EWEIGHT`	name of existing column

Column containing the error on the `zcolumn` value, used to calculate errors for extracted products when accumulating using the `zcolumn` column.

ignorelegallimits	no	boolean	false	true|false

If true, ignore the `TLMIN` and `TLMAX` keywords when determining column extraction ranges.

Image Extraction Parameters

imageset	no	string	`image.fits`	file name

Name of the image product file.

xcolumn	no	string	`RAWX`	name of existing column

Name of column with spatial X-coordinates (for image creation and spectral area determintation).

ycolumn	no	string	`RAWY`	name of existing column

Name of column with spatial Y-coordinates (for image creation and spectral area determintation).

squarepixels	no	boolean	false	true|false

When `ximagesize` and `yimagesize` are set, forces the x and y bin sizes to be the same. The larger of the two bin sizes is used.

ximagesize	no	integer	600	> 0

If set, the size of the image (ie. number of image pixels) along the x axis; for extraction using integer valued columns, the extracted image size may be somewhat smaller than the requested image size.

ximagebinsize	no	real	1	> 0

If set, the binning factor for x axis in image creation

yimagesize	no	integer	600	> 0

If set, the size of the image (ie. number of image pixels) along the y axis; for extraction using integer valued columns, the extracted image size may be somewhat smaller than the requested image size.

yimagebinsize	no	real	1	> 0

If set, the binning factor for y axis in image creation.

ximagemin	no	real	1	

If set, the lower limit of x coordinate for image extraction.

ximagemax	no	real	640	

If set, the upper limit of x coordinate for image extraction.

yimagemin	no	real	1	

If set, the lower limit of y coordinate for image extraction.

yimagemax	no	real	640	

If set, the upper limit of y coordinate for image extraction.

imagedatatype	no	string	Real64	Int8|Int16|Int32|Real32|Real64

Data type to use for the output image. If not set, **evselect** decides for itself what data type to use.

raimagecenter	no	real	0	

If set, right ascension for the center of the output image, in decimal degrees.

decimagecenter	no	real	0	

If set, declination for the center of the output image, in decimal degrees.

Spectrum Extraction Parameters

spectrumset	no	string	`spectrum.fits`	file name

Name of the spectrum product file.

spectralbinsize	no	real	10	>= 1

Binning factor for spectrum creation.

specchannelmin	no	integer	0	>= 0

If set, the minimum channel number to consider for spectrum creation.

specchannelmax	no	integer	4095	>= 0

If set, the maximum channel number to consider for spectrum creation.

nonStandardSpec	no	boolean	false	true|false

Should be set to `true` if a spectrum with a non-standard event PATTERN or non-standard channel range is required.

Rate Extraction Parameters

rateset	no	string	`rate.fits`	file name

Name of the time series product file.

timecolumn	no	string	`TIME`	name of existing column

Name of column for time coordinate (to be used in rates files creation).

timebinsize	no	real	1	> 0

Size of time bins for the time series files.

timemin	no	real	0	

If set, the earliest time to consider in time series file creation.

timemax	no	real	1000	

If set, the latest time to consider in time series file creation.

maketimecolumn	no	boolean	false	true|false

If true, include a time column in the FITS table when creating a time series. If false, the table will only include a `COUNTS` column and the necessary keywords to deduce the time for each bin.

makeratecolumn	no	boolean	false	true|false

If true, produces a lightcurve containing a `RATE`, rather than a `COUNTS` column. An `ERROR` column is also produced.

Histogram Extraction Parameters

histogramset	no	string	`histo.fits`	file name

File name for the histogram.

histogramcolumn	no	string	`TIME`	column name

Name of the column from which to extract the histogram.

histogrambinsize	no	real	1	> 0

Binning factor for histogram extraction.

histogrammin	no	real	0	

If set, the lower limit for histogram extraction.

histogrammax	no	real	1000	

If set, the upper limit for histogram extraction.

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.

RangeType *(error)*
In the subroutine getRange, a column was accessed which is not of a supported data type.

ColType *(error)*
Column type conversion not supported. The column cannot be converted to the desired data type.

EmptyTableName *(error)*
The name of the input table was specified as an empty string.

DssException *(error)*
An error occurred while parsing the selection expression in order to produce a data subspace specification.

ImgBinVal *(error)*
Binning factor must be an integer for integer-valued columns. Error occurred while performing image extraction.

XRangeVal *(error)*
X ranges must be integers for integer-valued columns.

YRangeVal *(error)*
Y ranges must be integers for integer-valued columns.

ImageType *(error)*
Image type is not of a supported data type.

WcsTransform *(error)*
The program failed to transform the `raimgecenter` and `decimagecenter` coordinates into pixel coordinates. This is because and error was returned by the wcslib routine wcsfwd. The error code is listed in the error message.

RateBinVal *(error)*
Binning factor must be an integer for integer-valued columns. Error occurred while performing light curve extraction.

TimeRangeVal *(error)*
Time ranges must be integers for integer-valued columns.

HistoBinVal *(error)*
Binning factor must be an integer for integer-valued columns.

HistoRangeVal *(error)*
Ranges must be integers for integer-valued columns.

SpecIntColumn *(error)*
Energy column must be of an integer type in order to extract an OGIP compliant spectrum.

NonStandardSpectrum *(error)*
A spectrum with a non-standard PATTERN or PI range has been requested which is incompatible with the matrix generation tasks. This may be overridden by setting the parameter `nonStandardSpec`=true.

NotXmmSpectrum *(warning)*
The input data set was not recognized as an XMM data set. A weighted rate spectrum can only be produced for XMM data.
corrective action: A spectrum file is not produced.

NoAttrib *(warning)*
The Attribute does not exist in the input event list. No value was copied.
corrective action: No value for the given attribute is copied into the product file.

MinParamRange *(warning)*
Value of parameter outside allowed range for column. Setting to minimum value.
corrective action: The minimum value used for product extraction is set to the minimum allowed value.

MaxParamRange *(warning)*
Value of parameter outside allowed range for column. Setting to maximum value.
corrective action: The maximum value used for product extraction is set to the maximum allowed value.

MaxDataRange *(warning)*
Data values found outside of allowed range for column; will be ignored.
corrective action: Values greater than the maximum allowed value will not be used for product extraction

NoInputEvents *(warning)*
The input event table contains no events (zero rows).
corrective action: None - empty product files may be produced

NoFilteredEvents *(warning)*
No events have been selected during the filtering process.
corrective action: None - empty product files may be produced

BadHistoLimits *(warning)*
The column used for histogram extraction has its upper limit equal to its lower limit, or the column has the same value throughout.
corrective action: An empty histogram is created.

BadHistoLimits *(error)*
The column used for histogram extraction has its upper limit less than its lower limit.

HistoColumnName *(warning)*
The input column for accumulating a histogram is called COUNTS, which causes a name clash in the histogram output file.
corrective action: The X axis column in the histogram file is renamed to OLD_COUNTS.

XlimitEpsilon *(warning)*
X axis upper limit adjusted by an epsilon amount. This is to guarentee that all events which fall into the range of the image get accumulated.
corrective action: X axis upper limit adjusted by an epsilon amount.

YlimitEpsilon *(warning)*
Y axis upper limit adjusted by an epsilon amount. This is to guarentee that all events which fall into the range of the image get accumulated.
corrective action: Y axis upper limit adjusted by an epsilon amount.

BadImgLimits *(warning)*
One of the columns used for image extraction has its upper limit less than or equal to its lower limit, or the column has the same value throughout.
corrective action: No image is created.

NoWCS *(warning)*
No WCS information is available for an image column.
corrective action: Pixel based WCS information is written to the output image.

BinVsFrameTime *(warning)*
Time bin size is less than the mean frame readout time.
corrective action: No action taken.

RateEnergy *(warning)*
A data subspace energy filter has been found which cannot be incorporated into the calculation of `E_MIN` and `E_MAX` values for a rate curve. `E_MIN` and `E_MAX` values may be incorrect.
corrective action: Energy filter is ignored.

BadRateLimits *(warning)*
The column used for light curve extraction has its upper limit equal to its lower limit, or the column has the same value throughout.
corrective action: An empty light curve is created.

BadRateLimits *(error)*
The column used for light curve extraction has its upper limit less than its lower limit

BadSpecLimits *(warning)*
The column used for spectral extraction has its upper limit equal to its lower limit, or the column has the same value throughout.
corrective action: An empty spectrum file is created.

BadSpecLimits *(error)*
The column used for spectral extraction has its upper limit less than its lower limit

NotXMM *(warning)*
Can only update exposure information on XMM data. No updating of exposure information is performed.
corrective action: Exposure information is not updated.

RgsTimeFilter *(warning)*
You have used the `TIME` column to filter an RGS events list. This will invalidate any exposure information in the produced products.
corrective action: Execution continues.

NoTargetOntime *(warning)*
The `ONTIME`*nn* attribute for the target CCD could not be found in the event list. The global `ONTIME` will be set to zero.
corrective action: `ONTIME` is set to zero.

NoTargetLivetime *(warning)*
The `LIVETI`*nn* attribute for the target CCD could not be found in the event list. The global `LIVETIME` will be set to zero.
corrective action: `LIVETIME` is set to zero.

ExpMultiCcdFilters *(warning)*
A data subspace component was found which contains more than one CCDNR filter, ie more than one expression specifying which CCD events should be selected from. The exposure correction code is not designed to handle this, I will in most cases produce wrong results when updating exposure information. Usually having more than one CCDNR filter in a component results in the exposure time (on time) not being uniform for all events in the CCD.
corrective action: Execution continues

ExpBadCcdType *(warning)*
The CCDNR column was used in a filtering expression for which it cannot be guaranteed that the selection criteria will result in a uniform exposure time (on time) for each CCD. This part of the filtering expression is thus ignored in the updating of the exposure information.
corrective action: Filter ignored in updating of exposure information

ExpBadCcdComp *(warning)*
The CCDNR column was used in a filtering expression for which it cannot be guaranteed that the selection criteria will result in a uniform exposure time (on time) for each CCD. This part of the filtering expression is thus ignored in the updating of the exposure information.
corrective action: Filter ignored in updating of exposure information

ExpBadTimeType *(warning)*
The TIME column was used in a filtering expression for which it cannot be guaranteed that the selection criteria will result in a uniform exposure time (on time) for each CCD. This part of the filtering expression is thus ignored in the updating of the exposure information.
corrective action: Filter ignored in updating of exposure information

ExpBadTimeComp *(warning)*
The TIME column was used in a filtering expression for which it cannot be guaranteed that the selection criteria will result in a uniform exposure time (on time) for each CCD. This part of the filtering expression is thus ignored in the updating of the exposure information.
corrective action: Filter ignored in updating of exposure information

ExpBadGlobalComp *(warning)*
The filtering expression used may not result in a uniform exposure time (on time) to be applied to all events for a given CCD. This is because the expression applies one time criteria to one set of events, and another time criteria to a different set of events.
corrective action: Execution continues

NotXmmProduct *(warning)*
Not XMM data. Cannot calculate EXPOSURE keyword for output products.
corrective action: Exposure calculation aborted. Execution continues.

NoProductExp *(warning)*
No live time information available. Cannot write `EXPOSURE` keyword for output products.
corrective action: `EXPOSURE` keyword not written

NoRGSExposure *(warning)*
No EXPOSURE keyword in RGS event list. Cannot write exposure value to the product file.
corrective action: Exposure calculation aborted. Execution continues.

RGSprodAttributes *(warning)*
Could not find requested RGS attributes to copy to the product extension.
corrective action: Execution continues.

NoCCDcolumn *(warning)*
Event list does not have a CCDNR column. Cannot calculate EXPOSURE keyword for EPIC products.
corrective action: Exposure calculation aborted. Execution continues.

NoOutput *(warning)*
Input parameters have been set such that no output will be produced and no input files modified.
corrective action: Immediately exit program.

ExpNoDss *(warning)*
The user has requested that exposure information be updated without the writing of the data subspace (`writedss` set to false). This means it is likely that all filter information will not be taken into account when updating the exposure information. In general, `writedss` should be set to true when `updateexposure` is set to true.
corrective action: Execution continues.

ExpBadDss *(warning)*
An error occurred while parsing and processing the data subspace information. This means it is likely that all filter information will not be taken into account when updating the exposure information.
corrective action: Execution continues.

FrameTimeBiggerThanPeriod *(warning)*
Check phase expression.
corrective action: Execution continues.

NoFrameTimeKeywordFound *(warning)*
Check event list.
corrective action: Execution continues.

NoExpEmpty *(warning)*
The exposure calculation cannot be performed for an empty event list, as it is a weighting of the number of photons in the various CCDs.
corrective action: Exposure calculation aborted. Execution continues.

Please see the **selectlib** documentation for information on errors and warnings having to do with event selection and filtering.

Please see the **dsslib** documentation for information on errors and warnings having to do with the data subspace.

6 Input Files

1. event list file to filter/extract products from
2. Good Time Interval files (optional)
3. Mask image filter file (optional)
4. Region specification file (optional)

7 Output Files

1. filtered event list (optional)
2. (OGIP-compliant) product files (optional)

8 Comments

- For interactive usage in a SAS session there is a Graphical-User-Interface to **evselect xmmselect**, which supports
  - the specification of the filtering expression
  - specification of all mandatory and optional parameters

  in a convenient and user-friendly manner. In addition, to simplify repeated selection-extraction-viewing cycles the **evselect** GUI controls the automatic spawning of viewers for the inspection of products.

9 Future developments

None.

9.1 Implementation Status

All envisaged functionality implemented.

9.2 TODO

Nothing.

References

[1] J. McDowell. A Data Model for Astrophysical Data Analysis. Technical report, Chandra Science Center, 1997. Found at the URL: http://hea-www.harvard.edu/∼jcm/asc/.