



merge

February 1, 2016

Abstract

This task merges EPIC event lists or additional files (attitude, orbit) from two exposures (even from different instruments), pointing in the same or an adjacent direction. Sky coordinates are (optionally) adjusted for attitude errors and then reprojected to a common reference point. The output files can be used by **evselect** and the higher level detection and time analysis tasks.

1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING
EPIC MOS	IMAGING

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

Two files from two exposures (even from different instruments) are taken as input. These may be EPIC event lists or two other additional files (attitude files, orbit files etc [see comments]). The files are checked via their extension names (**EVENTS**, **ATTHK**, **ORBIT**) for their type, and depending on the type, certain attributes are checked to see whether the files are compatible with each other. In the attitude and orbit cases, any files can be merged together (so long as they are both attitude or both orbit files), whereas in the events case, some care should be taken. For instance event files from different modes should not be merged, though perhaps event files from different instruments can (see comments). Warnings are given when the event files are deemed not entirely compatible (i.e. their **INSTRUME** or **FILTER** keywords are different). Here it is up to the user to interpret the output files sensibly.

Also, in the event files case, the two pointings need to be close to each other; the accuracy of the reprojection degrades with the offset between the two pointings, and warnings will be given when this offset becomes significantly large. Only if the input files pass all the checks does the merging proceed and an output file (event list, attitude file, orbit file) is produced. The individual blocks (e.g. **EVENTS**)



are merged in time order, as are the other relevant extensions (e.g. `EXPOSUnn`, `STDGTI`, `GTI`, etc [see comments]). In the event files case, the sky coordinates are deprojected and reprojected again onto a common sky position (given either by the user, via the parameters `withradec`, `ra` and `dec`, or as, by default, the mean of the two input pointings), and new WCS attributes are calculated (the size of the WCS ‘image’ is given by the parameter `imagesize`). It is possible also to adjust the individual event file sky coordinates prior to merging. This is useful when one sees offsets between sources in two datasets. Changes to RA, Dec and position angle can be given for each of the input event files (see examples).

Given here is a summary of the current conditions whereby event files (or parts thereof [blocks/columns etc]) are or are not merged. In each condition, the necessary warning/error messages are given.

Condition	Action
Different <code>INSTRUME</code>	continue (warning)
Different <code>INSTRUME</code>	<code>EXPOSUnn</code> extensions not merged (warning)
Different <code>INSTRUME</code>	<code>STDGTI*</code> extensions not merged (warning)
Different <code>FILTER</code>	continue (warning)
Different <code>DATAMODE</code>	stop (error)
No block (extension) in one/both file(s)	extensions not merged (warning)
No column in one/both extension(s)	columns not merged (warning)
<code>TIME</code> column missing in one/both file(s)	extensions merged in input order (warning)
No <code>X/Y</code> columns in one/both extension(s)	<code>X/Y</code> columns not reprojected (warning)

It is strongly recommended not to merge event files from different observations because of the difficulty of accurately using the merged file in downstream software. By default, an error is issued and the task aborted, if two input files with different observation numbers are used. However, expert users may override this behaviour by setting the command line parameter `mergedifferentobs="yes"`.

If present, the following event file columns will be merged: `TIME`, `RAWX`, `RAWY`, `DETX`, `DETY`, `X`, `Y`, `PHA`, `PI`, `FLAG`, `PATTERN`, `PAT_ID`, `PAT_SEQ`, `PAY_TYP`, `PAT_IND`, `PAT_ORG`, `CCDNR`, `WEIGHT`. Up to 12 additional columns may be specified by command line parameter.

The output files can be used by `evselect` and the higher level detection and time analysis tasks. See the caveats and limitations listed in the comments section.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
<code>set1</code>	yes	string		
first input file				
<code>set2</code>	yes	string		
second input file				
<code>outset</code>	yes	string		
output file				
<code>withradec</code>	no	boolean	N	Y/N
input user-defined RA/Dec (otherwise mean used)				
<code>ra</code>	no	real	0.0	0–360



celestial RA coordinate of new reference point (if withradec=Y) (deg)

dec	no	real	0.0	-90--+90
------------	----	------	-----	----------

celestial Dec coordinate of new reference point (if withradec=Y) (deg)

imagesize	no	real	0.36	>0
------------------	----	------	------	----

half-size of final merged image (deg) (see comments)

withset1adjustpos	no	boolean	N	Y/N
--------------------------	----	---------	---	-----

Adjust sky positions of set 1 events before merge

set1deltara	no	real	0.0	-5--+5
--------------------	----	------	-----	--------

Change in RA (set 1) (deg) (if withset1adjustpos=Y)

set1deltadec	no	real	0.0	-5--+5
---------------------	----	------	-----	--------

Change in Dec (set 1) (deg) (if withset1adjustpos=Y)

set1deltaposangle	no	real	0.0	-180--+180
--------------------------	----	------	-----	------------

Change in position angle (set 1) (deg) (if withset1adjustpos=Y)

withset2adjustpos	no	boolean	N	Y/N
--------------------------	----	---------	---	-----

Adjust sky positions of set 2 events before merge

set2deltara	no	real	0.0	-5--+5
--------------------	----	------	-----	--------

Change in RA (set 2) (deg) (if withset2adjustpos=Y)

set2deltadec	no	real	0.0	-5--+5
---------------------	----	------	-----	--------

Change in Dec (set 2) (deg) (if withset2adjustpos=Y)

set2deltaposangle	no	real	0.0	-180--+180
--------------------------	----	------	-----	------------

Change in position angle (set 2) (deg) (if withset2adjustpos=Y)

columns	no	string		
----------------	----	--------	--	--

optional event list columns to be merged (up to 12 column names may be specified)

mergedifferentobs	no	boolean	N	Y/N
--------------------------	----	---------	---	-----

Allow event files from different observations to be merged ?

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.

filesNonCompatible (*error*)

Input files are non-compatible (file type, DATAMODE)

differentObservations (*error*)

Two input event files from different observations have been used. This is strongly discouraged



because of the subsequent difficulty of accurately using downstream software on the merged event file. Experts can force the task to merge these files by setting the command line parameter `mergedifferentobs="yes"`

MergeGeneralWarning (*warning*)

Merged eventsets may result in incorrect results of some downstream tasks
corrective action: See the task description!

filesNonCompatible (*warning*)

Input files not entirely compatible (`INSTRUME`, `FILTER`)
corrective action: Input files are merged - care should be taken as to subsequent use

noExposureMerge (*warning*)

Input files are from different instruments - `EXPOSURE` extensions not merged
corrective action: Different instrument Exposure extensions not merged

noGTIMerge (*warning*)

Input files are from different instruments - `GTI` extensions not merged
corrective action: Different instrument `GTI` extensions not merged

largePointingOffset (*warning*)

Pointings are offset by a significantly large amount
corrective action: Reprojection is performed - care should be taken as to subsequent use

noExtension (*warning*)

extension does not exist in one or both files - not merging
corrective action: Particular extension is not merged into output file

noTimeColumn (*warning*)

`TIME` column missing in one file - cannot establish order
corrective action: Files are merged in the user-input order

noColumn (*warning*)

Column 'Column name' missing in one file - not merging
corrective action: Particular column is not merged into output file

noXYColumns (*warning*)

`X` and/or `Y` columns do not exist in one or both input files - no reprojecting
corrective action: Coordinate reprojection is not performed

fileTimeOverlap (*warning*)

Files overlap in time - some `TIME` keywords cannot be correctly calculated
corrective action: Calculation of some keywords not performed - `ONTIME`, `LIVETIME` etc

tooManyColumns (*warning*)

Too many columns given on command line
corrective action: Only first 12 user-specified columns will be copied

unMatchedGTI (*warning*)

Flare `GTI` extension missing in one dataset: will result in incorrect exposure times!
corrective action: Create matching input datasets and rerun task

unMatchedSTDGTI (*warning*)

`STDGTI` extension missing in one dataset: will result in incorrect exposure times!
corrective action: Create matching input datasets and rerun task



6 Input Files

One of the following: Either

1. Two **evlistcomb** EPIC output event files, or
2. Two **atthkgen** output files, or
3. Two **orbit** output files

Keywords read are (depending on input files): TELESCOP, INSTRUME, FILTER, DATAMODE, TCRVL, TCDLT, TCRPX.

7 Output Files

1. Merged file of the two input files. In the events case, the X, Y sky coordinates are reprojected to a new common reference point, and new WCS attributes are calculated.

Extra/new keywords written (depending on input files): DATE-END, EQUINOX, RADECSYS, TCRVL, TCDLT, TCRPX, TCTYP, TLMIN, TLMAX, TDMIN, TDMAX, TCUNI.

8 Algorithm

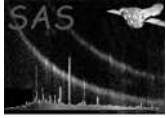
```
subroutine merge
```

- ```
* get file types, and depending thereupon, call particular
 setup subroutine
* depending on type, check for file consistency. Get time-ordering of
 input files
* merge files and main extensions, using common merge subroutine, merge
 individual columns only when they exist in both datasets.
* if type=events, call routines to (optionally) correct individual event
 file coordinates for small changes in RA, Dec and PA, then reproject
 sky positions onto single reference point on sky and calculate new
 WCS parameters.
* if type=events, merge EXPOSURE and GTI extensions, using common merge
 subroutine, again checking for availability of extensions and columns
* change header attributes where necessary, add history
```

```
end subroutine merge
```

## 9 Comments

- Care should be taken in using any output. Cases have been found whereby new output file attributes need to be calculated on the basis of input file attributes, and no doubt, these cases will continue to crop up. In many cases, care will be needed as regards any subsequent use of a particular merged file.



- The task is now able to correct both individual input event files for small errors in attitude, prior to merging. Very thorough testing of this feature has not yet been performed. Position angle is measured clockwise on the sky.
- Problems can occur when one of the input files contains GTI extensions (not STDGTI) and the other one doesn't. Everything seems ok but `evselect` can't handle this case and will calculate the wrong exposure time for files extracted from the merged event file.
- The GTI merge assumes that only one set of GTIs exist in each input file. If more than one set exist then the final result will probably be wrong.
- In the event file merging, the BADPIX extensions are not merged.
- It may be that the default `imagesize` may change, as merged images are likely to cover a large area of the sky than single event file images.
- Out-of-time events modeling in task `esplinemap` does not work with merged eventsets.
- When merging two files (C, A say) from different exposures, the two files will be automatically ordered so that they are merged in time order (AC). If one then wants to merge a third file (B) to the first two, this third file data being from a time in-between the first two, then problems *may* exist with subsequent tasks performed on the resultant merged file (ACB) as the data will not be time sorted. It is advised therefore to merge all data in time order (i.e. first A with B, then AB with C).
- It is advisable also, because of the large file sizes involved, the time-consuming nature of the reprojections and the fact, in merging e.g. four event files together, that three `merge` runs are necessary (A+B, AB+C, ABC+D, done preferably, remember, in time order, and not for instance in file size order), that the event files are first filtered to only the good events one is interested in.

## 10 Future developments

- In addition to the comments, many questions exist regarding merging files over from different exposures, and the handling of EXPOSURE, GTI and BADPIX extensions.
- In merging event files from different EPIC instruments (i.e. MOS and pn), the event attribute columns should not be merged, as different instruments assign different meanings to the same bits. Merging across different instruments should be done after good events have been selected (this is related also to the point above in the comments).

## 11 Examples

- `merge set1=ev1.fits set2=ev2.fits outset=evout.fits` (Merging two event files, reprojecting X/Y coordinates to mean reference point)
- `merge set1=ev1.fits set2=ev2.fits outset=evout.fits withradec=Y ra=1.1 dec=-0.7` (Merging two event files, reprojecting X/Y coordinates to user reference point)
- `merge set1=ev1.fits set2=ev2.fits outset=evout.fits withset1adjustpos=Y set1deltara=0.012 set1deltadec=-0.003 set1deltaposangle=0.001` (Merging two event files, reprojecting X/Y coordinates to mean reference point - set1's events are corrected for small attitude corrections prior to merging)



- *merge set1=att1.fits set2=att2.fits outset=attout.fits* (Merging two attitude files, i.e. atthkgen output files)
- *merge set1=orb1.fits set2=orb2.fits outset=orbout.fits* (Merging two orbit files, i.e. orbit output files)

Note: it is possible in many cases to treat the output files as ordinary (non-merged) files. For example, creation of an exposure map is possible for merged data: First **merge** the relevant event files and associated atthkgen output files, then form a sky image (via **evselect**) from the merged event file, then run **eexpmap** as normal using this image and the merged event and attitude files.

Just to reiterate, do take care with any merged output.

## References