epicproc

August 16, 2005

Abstract

epicproc is a package that contains two tasks, **epproc** and **emproc**, that can be used to process data from the EPIC instruments.

1 Instruments/Modes

Instrument	Mode
EPIC PN	IMAGING, TIMING, BURST
EPIC MOS	IMAGING, REDUCED IMAGING, TIMING, COMPRESSED TIMING

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

The SAS package **epicproc** contains the two tasks **epproc** and **emproc**, that can be used to process the data from the EPIC MOS and PN instruments.

The two tasks are structured in the same manner, and actually share most of the code and parameters. In the remainder of this document we describe the common parts of the tasks. For the instrument specific details, and the complete list of task parameters refer to the description of **epproc** and **emproc**.

3.0.0.1 Conventions Because the tasks **epproc** and **emproc** share many of the command line parameters, in the following, when we refer to `epicproc` we mean either **epproc** or **emproc**.

3.1 Exposure selection

3.1.1 Explicit exposure selection

You can select the exposures to process explicitly via the parameters `withinstexpid` and `instexpids`. For instance:

- `epicproc --withinstexpids=yes --instexpids="IIFnnn ..."`

where `instexpids` is a list of strings where:

- `II` is one of `M1`, `M2`, or `PN`.
- `F` is one of `S` or `U` (scheduled or unscheduled, respectively).
- `nnn` is the exposure number (`001 002` ...).

Note that the `IIFnnn` matches the analogous string found in the ODF file names. You may also want to look at the contents of the ODF with the **odfbrowser**, from which individual exposures can be selected, and the **epicproc** tasks can be configured and launched.

3.1.2 Implicit selection through the mode

If you opt to select the data modes that should be processed, then all the exposures in a given mode are processed. The relevant parameters are:

- `selectmodes`
- `imaging`, `rimaging`, `timing`, `ctiming` for **emproc**.
- `imaging`, `timing`, `burst` for **epproc**.

If both `selectmodes` and `withinstexpids` are given, the former is ignored.

By default all available exposures are processed.

3.1.2.1 Timing modes The processing of timing modes may require that the user specify additional parameters to the tasks **epframes** and **emframes**. You should use the parameters `withsrccoords`, `srcra`, and `srcdec`. For PN you can optionally use `timingsrcposition` in `RAWY` coordinates. In most cases the default values are adequate.

3.2 CCD selection

The processing of individual CCDs can be switched off via the parameters `selectccds` and `ccdn`, where `n` is an integer from 1 to 7 or 12, for **epproc** or **emproc** respectively.

By default, all CCDs are processed.

3.3 Attitude processing

In this processing stage the task **atthkgen** is run if requested. This is controlled via the parameter `runatthkgen` that is `true` by default.

If the output of **atthkgen** is found on disk, the task is not run again.

The output of **atthkgen**, if present, is used by the task **attcalc**.

3.4 Housekeeping processing

In this processing stage the task **hkgtigen** is run. This stage can be switched off by setting the parameter `runhkgtigen` to false.

If the output of **hkgtigen** is found on disk, the task is not run again.

hkgtigen creates a GTI file based on the **selectlib** conditions based on the contents of the housekeeping files. The housekeeping GTI is used together with any other external GTI file specified (see 3.5).

Additional parameters related to this processing stage are `withparameters`, `parameters`, `except`, `withoverrideparameters`, and `overrideparameters`. Normally **hkgtigen** is driven by the contents of the CCF, and there is no need to make use of these additional parameters.

3.5 External GTIs

An external GTI can be supplied to **epicproc** with the parameters `withgtiset` and `gtiset`. If the parameter `applygti` is further set to `true` (this is the default), the external GTI will be combined with the housekeeping GTI (if any, see 3.4) and applied to the event lists.

3.6 Intermediate files

Normally **epicproc** removes the many intermediate data files that are created during the data processing stages. The user can control which of the intermediate files are not deleted by setting the following parameters:

- `removeintermediategtis`: the intermediate GTIs created by the tasks **emframes** and **epframes** are not deleted if this parameter is set to `false`.
- `removeintermediateeventlists`: this parameter controls whether the intermediate events lists used to store data between tasks are removed.
- `removetemporaries`: all other intermediate files created by **epicproc** are removed if this parameter is set to `false`.

3.7 Bad pixels

By default **epicproc** will try to detect bad pixels for any imaging exposure. The new bad pixels are then used in the data reduction together with any other known (via the calibration files) bad pixels.

The parameter `searchforbadpixels` controls whether the task used for finding bad pixels has to be run or nor. There is a task called **badpixfind** search for bad pixels and can be used by both task: **epproc** and **emproc**.

Additionally, **emproc** has another specific task to search for bad pixels called **embadpixfind**. This task can be selected through the parameter `badpixfindalgo`, that can only be found in **emproc**. Please, refer to **emproc** for a detailed explanation of the parameter `badpixfindalgo`

Further, the parameter `searchforbadcolumns` controls whether bad columns are also searched for, if you have selected **badpixfind**. Both parameters are set to `true` by default.

A number of parameters are available to modify the way **badpixfind** operates: `thresholdlabel`, `lothresh`, `hithresh`, `columnsearchlabel`, `locolthresh`, `hicolthresh`, `flickertimesteps`, `flickerksthresh`, `flickerchisqthresh`, `backgroundrate`, `narrowerthanpsf`, `threshabovebackground`, `loenergythresh`, `hienergythresh`. Refer to **badpixfind** for a detailed explanation.

3.8 Filtering

epicproc will filter the events lists (`filterevents` is set to `true` by default) with the **selectlib** expression given in `filterexpression`. The filtering is destructive by default, that is to say the filtered events are physically removed from the event lists. This behavior can be changed by setting the parameter `flagfilteredevents` to `true`. In this case all events will be retained, and a flag column will be set to indicate what events would have been removed. Refer to **evselect** for details on the destructive and non-destructive filtering.

The default filtering expressions for **epicproc** make use of the XMM events attribute flags (see **evatt**), and should be suitable for most users.

4 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.

SubTaskError *(warning)*
epicproc has detected an error from one of the sub-tasks.
corrective action: The processing of the current data set is abandoned.

5 Output Files

All output data sets start with a string composed by one or more of the following parts, each separated by an underscore character (_):

- `rrrr`: the revolution number
- `oooooooooo`: the observation identifier

- `EPN` or `EMOS1` or `EMOS2`: the instrument name
- `eeee`: the exposure identifier. For instance: `S001`.
- `cc`: the CCD number
- `nn`: the node number

These parts are hierarchically structured, so that, say, any data set name that contains the exposure identifier will also contain the revolution number, the observation identifier, and the instrument name.

Additional strings indicate the contents of the data set:

- `AttHk`: attitude housekeeping
- `Gti`: GTI
- `AuxGti`: GTI based on the auxiliary file
- `FrmGti`: GTI created by **epframes** or **emframes**
- `HkGti`: housekeeping GTI
- `Evts`: the data set contains an event list
- `ImagingEvts`: imaging mode event list
- `TimingEvts`: timing mode event list
- `BurstEvts`: burst mode event list
- `Temp`: a temporary data set, usually removed

References