spi_obs_hist

User Manual

Version 2.7.0
19 October 2006

Jürgen Knödlseder
Centre d'Etude Spatiale des Rayonnements
knodlseder@cesr.fr
http://www.cesr.fr./~jurgen/index.html

Note to the user

This software has been written to analyse data of the SPI telescope onboard INTEGRAL. Particular care has been taken in making the software user friendly and well documented. If you appreciated this effort, and if this software and User Manual were useful for your scientific work, the author would appreciate a corresponding acknowledgment in your published work.

Contents

1 Introduction

The executable `spi_obs_hist` is part of the SPI scientific analysis preparation software component (SAP). It's main purpose is the generation of event spectra that may be used for imaging or spectral analysis, but in the same time it also derives the Good Time Intervals and the Deadtime for each pointing and pseudo-detector. This optimises execution speed and allow for a rapid data preparation for scientific analysis.

In general, the SPI scientific analysis preparation (SAP) consists of the following steps

- `spi_obs_create` create an observation group
- `spi_obs_point` extract stable pointings for SPI
- `spibounds` define energy boundaries for spectral binning
- `spi_gain_corr` (optional) apply gain correction to photon/photon data and store them in `COR` data. `spi_obs_hist` allows for on-the-fly gain calibration which avoids storing gain corrections in `COR` data, hence this step may be avoided.
- `spi_psd_optimise` (optional) optimise onground PSD discrimination. This step is only needed if PSD pseudo-detectors are selected.
- `spi_psd_postprocess` (optional) perform onground PSD discrimination and store results in `COR` data. This step is only needed if PSD pseudo-detectors are selected and if `COR` data should be written. `spi_obs_hist` allows for on-the-fly PSD discrimination which avoids storing the corrected PSD flag in `COR` data.
- `spi_obs_hist` build event spectra, Good Time Intervals, and Deadtimes
- `spi_psd_respgen` (optional) generates PSD response. This step is only needed if PSD pseudo-detectors are selected.
- `spi_psd_effigen` (optional) generates PSD efficiencies. This step is only needed if PSD pseudo-detectors are selected.
- `spi_obs_back` generate background model

So in case that no `COR` data should be written (default) the analysis chain consists of `spi_obs_create` → `spi_obs_point` → `spibounds` → `spi_obs_hist` → `spi_obs_back` if no PSD pseudo-detectors should be used and of `og_create` → `spi_obs_point` → `spi_psd_optimise` → `spibounds` → `spi_obs_hist` → `spi_psd_respgen` → `spi_psd_effigen` → `spi_obs_back` if PSD pseudo-detectors are used.

`spi_obs_hist` is written in the ANSI C++ language and has been developped under ISDC support platform 6.3. It requires `spi_toolslib` version 4.10.7 or higher.

2 Getting started

Before installing `spi_obs_hist`, make sure that the ISDC support platform 6.3 or higher is installed on your system, and that you have installed the library `spi_toolslib` version 4.10.7 or higher.

After downloading the `spi_obs_hist.tar.gz` file, step into a directory that should hold the distribution, move the `spi_obs_hist.tar.gz` file into this directory and type after the UNIX prompt `$` (don't type this prompt):

```
$ gunzip spi_obs_hist.tar.gz
$ tar xvf spi_obs_hist.tar
```

The first command uncompresses the distribution file, the second unpacks the files.

Before configuration, the distribution needs to be reset to a clean state. To do this, type

```
$ make distclean
```

Then, configure the distribution. It is assumed here that you have previously installed the ISDC support platform, thus you should type

```
$ ~/bin/ac_stuff/configure
```

Finally, build the distribution by typing

```
$ make global_install
```

To perform a unit test, type

```
$ make test
```

Make sure that the test data `spi_test_data-1.0.tar.gz` are available at your site (they should reside in a directory whose name is defined by the `ISDC_TEST_DATA_DIR` environment variable).

3 Parameter file

```
#############################################################################
#                                                                           #
#               Centre d'Etude Spatiale des Rayonnements                    #
#                     (in collaboration with ISDC)                          #
#                                                                           #
#                  SPI observation histogram creation                       #
#                                                                           #
# ------------------------------------------------------------------------- #
#                                                                           #
#  File:      spi_obs_hist.par                                              #
#  Version:   2.7.0                                                         #
#  Component: SAP                                                           #
#                                                                           #
#  Author:    Juergen Knoedlseder                                           #
#             knodlseder@cesr.fr                                            #
#             CESR                                                          #
#                                                                           #
#  Purpose:   Parameter file of the SPI observation histogram               #
#             executable                                                    #
#                                                                           #
#############################################################################
#
# The input DOLs/filenames
#=========================
inOgDOL,   s,q,                "",,,"Input Group DOL (SWG/OG/IDX)"
inEbdsDOL, s,q,                "",,,"Histogram energy boundaries DOL"
inPntDOL,  s,q,                "",,,"Pointing definition DOL"
inCoeffDOL,s,q,"spi_gain_idx.fits",,,"Gain correction DOL (File/IDX)"
inDiscDOL, s,q,    "psd_disc.fits",,,"PSD discrimination DOL"
#
# The output DOLs/filenames
#==========================
outOgDOL, s,q,       "og_spi.fits",,,"Observation output group DOL or filename"
outGtiDOL,s,q,          "gti.fits",,,"Good Time Interval DOL or filename"
outDspDOL,s,q,"evts_det_spec.fits",,,"Spectral histograms DOL or filename"
outDtiDOL,s,q,    "dead_time.fits",,,"Deadtime DOL or filename"
#
# Task parameters
#================
pdets,       s,q,  "0-18",,,"List pseudo detector indices (0-141)"
histoUnit,   i,h,     0,0,0,"Histogram unit (0=counts)"
histoType,   i,h,     0,0,0,"Histogram energy type (0=PI)"
doGainCalib, b,q,     yes,,,"Perform on-the fly gain correction?"
doPsdDisc,   b,q,     yes,,,"Perform on-the-fly PSD discrimination?"
doDeadtime,  b,q,     yes,,,"Perform on-the-fly deadtime calculation?"
randomise,   b,h,     yes,,,"Randomise energies?"
psdOnground, b,q,     yes,,,"Use PSD onground discrimination?"
shiftPha1,   b,h,     yes,,,"Shift PHA1 onboard spectra by 3282 channels?"
blankDead,   b,h,     yes,,,"Blank spectra after GeD failure?"
useSE,       b,h,     yes,,,"Use SE to build spectra?"
usePE,       b,h,     yes,,,"Use PE to build spectra?"
```

```
specStartTol,r,h,180.0,0.0,,"Spectrum start tolerance (seconds)"
specStopTol, r,h,180.0,0.0,,"Spectrum stop tolerance (seconds)"
#
# Standard parameters
#====================
clobber,b,h,yes,,,"Overwrite existing output data?"
verbose,i,h,4,0,4,"Information logging level"
```

Instead of specifying complete DOLs (Data Object Locations), which are composed of a filename plus the data structure extension (HDU), `spi_obs_hist` accepts also simple filenames and adds the appropriate data structure extensions. This means that **specified data structure extensions are ignored**.

The parameters have the following meaning:

- `ingrpDOL` (optional) specifies the DOL or filename of the input Observation Group (HDU `[GROUPING]`) for which event spectra should be built.

  If an output Observation Group has been specified (parameter `outgrpDOL`), the specification of this parameter is optional. If the parameter is left blank, the output Observation Group will then be used as input Observation Group. Otherwise, the input Observation Group will be copied into the output Observation Group.

- `inebdsDOL` (optional) specifies the DOL or filename of an energy boundary definition (HDU `[SPI.-EBDS-SET]`). This data structure specifies the energy boundaries of the binned data.

  If a `[SPI.-EBDS-SET]` element exists already in the input Observation Group, this element will be replaced by the specified DOL in the output Observation Group. Otherwise, the specified DOL will be attached to the output Observation Group. If left blank, it is assumed that a `[SPI.-EBDS-SET]` element exists already in the input Observation Group. If no such element is found, however, the task execution is aborted with an error message.

- `inPntDOL` (optional) specifies the DOL or filename of a Pointing definition (HDU `[SPI.-OBS.-PNT]`). This data structure specifies the pointings that should be used for spectra building.

  If a `[SPI.-OBS.-PNT]` element exists already in the input Observation Group, this element will be replaced by the specified DOL in the output Observation Group. Otherwise, the specified DOL will be attached to the output Observation Group. If left blank, it is assumed that a `[SPI.-OBS.-PNT]` element exists already in the input Observation Group. If no such element is found, however, the task execution is aborted with an error message.

- `inCoeffDOL` (optional) specifies the DOL or filename of a gain coefficient index or file (HDU `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]`). This data structure specifies the gain calibration coefficients that should be used if on-the-fly energy correction should be performed (see parameter `doGainCalib`).

  If a `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` element exists already in the input Observation Group, this element will be replaced by the specified DOL in the output Observation Group. Otherwise, the specified DOL will be attached to the output Observation Group. If left blank, it is assumed that a `[SPI.-COEF-CAL]` or `[SPI.-COEF-CAL-IDX]` element exists already in the input Observation Group. If no such element is found, however, the task execution is aborted with an error message.

- `inDiscDOL` (optional) specifies the DOL or filename of a PSD discrimination parameters (HDU `[SPI.-DISC-PSD]`). This data structure specifies the PSD discrimination parameters that should be used if on-the-fly PSD discrimination should be performed (see parameter `doPsdDisc`).

  If a `[SPI.-DISC-PSD]` element exists already in the input Observation Group, this element will be replaced by the specified DOL in the output Observation Group. Otherwise, the specified DOL will

be attached to the output Observation Group. If left blank, it is assumed that a `[SPI.-DISC-PSD]` element exists already in the input Observation Group. If no such element is found, however, the task execution is aborted with an error message.

- `outgrpDOL` (optional) specifies the DOL or filename of the output Observation Group (HDU `[GROUPING]`). The output Observation Group will be a copy of the input Observation Group plus the resulting data structures `[SPI.-OBS.-GTI]`, `[SPI.-OBS.-DSP]`, and `[SPI.-OBS.-DTI]` (optionally) attached.

  If an input Observation Group has been specified (parameter `ingrpDOL`), the specification of this parameter is optional. If the parameter is left blank, the input Observation Group will then be used as output Observation Group.

- `outGtiDOL` specifies the DOL or filename of the result Good Time Intervals (HDU `[SPI.-OBS.-GTI]`).

- `outDspDOL` specifies the DOL or filename of the result Detector Spectra (HDU `[SPI.-OBS.-DSP]`).

- `outDtiDOL` specifies the DOL or filename of the result Deadtime (HDU `[SPI.-OBS.-DTI]`).

- `pdets` list of pseudo-detectors that should be used for the analysis. The list may be composed of intervals, specified by a minimum and maximum value (e.g. `0-18`), or specific, comma separated pseudo-detector indices (e.g. `0-5,7,9,12-18,123-141`). Recall that pseudo-detector indices run from `0` to `141`.

- `histoUnit` unit of event spectra (always `0` which corresponds to `COUNTS`).

- `histoType` type of event spectra (always `0` which corresponds to correct energy channels, i.e. `PI`).

- `doGainCalib` specifies if on-the-fly energy correction should be performed (default = `yes`).

- `doPsdDisc` specifies if on-the-fly PSD discrimination should be performed (default = `yes` if PSD pseudo-detectors have been specified; in this case a `SPI.-DISC-PSD` data structure is required in the input group which is created by the task `spi_psd_optimise`).

- `doDeadtime` specifies if on-the-fly deadtime calculation should be performed (default = `yes`).

- `randomise` specifies if energy randomisation should be performed (default = `yes`).

- `psdOnground` specifies if onground or inflight PSD discrimination should be used (default = `yes`, corresponding to onground discrimination).

- `shiftPha1` shift onboard PHA1 spectra by 3282 channels. This is needed since the first 3282 channels of the onboard spectra are not transmitted to ground, and the data are stored by ISDC as they are.

- `blankDead` set all entries after detector failures to 0. This allows to generate clean datasets using data before and after a detector failure in combination.

- `useSE` use SE for spectra building. By default specify `yes`. If `no` is specified, a special model is entered where SE are excluded for spectra building. This mode maybe useful to analyse single detector event data in the 700-1700 keV energy range which is affected by an electronic noise feature. Due to the exclusion of SE the detection efficiency of SPI is reduced to about 85%, hence resulting fluxes should be multiplied by roughly ~ 1.2. Note also that exclusion of SE will result only in events within the energy range $\sim 400 - 2300$ keV. Below and above this range, the number of events smoothly drops to zero. For later revolutions, the lower threshold may be raised up to 800 keV, resulting in useful events only within $800 - 2300$ keV.

- `usePE` use PE for spectra building. By default specify `yes`. If `no` is specified, a special model is entered where PE are excluded for spectra building. This mode is only useful for diagnostics and should not be used to create datasets for science analysis.

- `specStartTol` specifies the maximum amount of time an onboard spectrum may start before the start of the pointing (units of seconds). This parameter allows to gather also event spectra which start slightly earlier than the official start time of a given pointing.
- `specStopTol` specifies the maximum amount of time an onboard spectrum may stop after the stop of the pointing (units of seconds). This parameter allows to gather also event spectra which stop slightly later than the official stop time of a given pointing.
- `clobber` specifies if existing output data structures should be overwritten or not. If `yes` is specified, the executable will notify the user about the deletion of any file. If `no` is specified and the executable attempts to overwrite existing data (e.g. an existing output Observation Group), the task will exit with an error message.
- `verbose` specifies the verbose level of the executable. For `verbose=0`, no information will be logged in case of an error. For `verbose=1`, only errors will be logged, while for `verbose=2` also actions (such as DOL detachments and attachments and DOL deletion) will be logged. `verbose=3` provides a report about the histogram building. `verbose=4` provides a very detailed report about the histogram building, including event statistics.

4 Interface definition

to be written ...

5 Algorithm

to be written ...

6 Error codes

The executable `spi_obs_hist` may stop with the following error codes:

```
SPI_OBS_HIST_ERROR_MEM_ALLOC                -233200
SPI_OBS_HIST_ERROR_BAD_PARAMETER            -233201
SPI_OBS_HIST_ERROR_INVALID_PDETE            -233202
SPI_OBS_HIST_ERROR_UNKNOWN_SPEC             -233203
SPI_OBS_HIST_ERROR_SPEC_NOT_FOUND           -233204
SPI_OBS_HIST_ERROR_EBDS_EMPTY               -233205
```

They have the following meaning:

- `SPI_OBS_HIST_ERROR_MEM_ALLOC` : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task.
- `SPI_OBS_HIST_ERROR_BAD_PARAMETER` : an invalid task parameter has been specified. More information is found in the log file.
- `SPI_OBS_HIST_ERROR_INVALID_PDETE` : an invalid pseudo-detector has been specified in the pseudo-detector list (parameter `pdets`).

- `SPI_OBS_HIST_ERROR_UNKNOWN_SPEC` : during the access of the onboard spectra, one of the requested spectra has not been found. This error should never occur and may indicate a possible problem in some low-level routine (i.e. `spi_toolslib`). Please report this kind of problem to the author.
- `SPI_OBS_HIST_ERROR_SPEC_NOT_FOUND` : during the access to a set of spectra, the requested pseudo-detector has not been found. This error should never occur and may indicate a possible problem in some low-level routine (i.e. `spi_toolslib`). Please report this kind of problem to the author.
- `SPI_OBS_HIST_ERROR_EBDS_EMPTY` : energy boundary data structure `SPI.-EBDS-SET` is empty. Please provide a valid data structure for spectral binning definition.

In addition, all errors that may occur in calls to ISDC support functions (such as for example DAL, RIL or PIL) are forwarded. Please consult the ISDC web pages for getting information about these error codes.