

spi_obs_point

User Manual

Version 3.1.0
30 March 2006

Jürgen Knödseder
Centre d'Etude Spatiale des Rayonnements
knodseder@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	1
2	Getting started	1
3	Parameter file	2
4	Interface definition	4
5	Algorithm	6
6	Error codes	6

1 Introduction

The executable `spi_obs_point` is part of the SPI scientific analysis preparation software component (SAP). It is a more versatile equivalent to the `spipoint` task, and produces a pointing definition structure of HDU type `SPI.-OBS.-PNT` for an observation group. In contrast to `spipoint`, `spi_obs_point` also accepts missing pointing information in an observation group, and constructs pointing information from an auxiliary ASCII text file.

`spi_obs_point` is written in the ANSI C++ language and has been developed under ISDC support platform 6.3. It requires `spi_toolslib` version 4.8.1 or higher.

2 Getting started

Before installing `spi_obs_point`, 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 4.8.1 or higher.

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

```
$ gunzip spi_obs_point.tar.gz
$ tar xvf spi_obs_point.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 OBS pointing definition                         #
#
# -----#
#
# File:      spi_obs_point.par                                #
# Version:   3.1.0                                           #
# Component: SAP                                             #
#
# Author:    Juergen Knoedlseder                             #
#            knodlseder@cesr.fr                              #
#            CESR                                             #
#
# Purpose:   Parameter file of the SPI pointing definition   #
#
#####
#
# The input DOLs/filenames
#=====
ingrpDOL, s,q,"og_spi.fits",,, "Input Observation Group DOL or filename"
intxtDOL, s,q,      "",,, "Pointing definition ASCII file"
filterDOL,s,q,      "",,, "Filter DOL or filename"
#
# The output DOLs/filenames
#=====
outgrpDOL,s,q,"og_spi_pnt.fits",,, "Output Observation Group DOL or filename"
outpntDOL,s,q, "pointings.fits",,, "Pointings DOL or filename"
#
# Task parameters
#=====
useAUX,  b,h, yes,, "Try using pointing definition from AUX data ?"
useHist, b,h, yes,, "Try using pointing definition from historic AUX data ?"
useSWG,  b,h, yes,, "Try using pointing definition from SWG ?"
useSlew, b,h, no,,  "Use slew data ?"
xtol,    r,h, 1.0,, "Tolerance on the stability of X axis (arcmin)"
ztol,    r,h, 1.0,, "Tolerance on the stability of Z axis (arcmin)"
#
# Standard parameters
#=====
clobber,b,hl,yes,, "Overwrite existing output data ?"
verbose,i,hl,2,0,3, "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_point` 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 the pointing definition should be derived. The input group has to be of level COR.
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.
- **intxtDOL** (optional) specifies an ASCII file that provides pointing information in case that no pointing information is available in the input Observation Group. This file will only be used if the **useTXT** is set to **yes**. If left blank, set **useTXT = no**.
- **filterDOL** (optional) specifies a pointing filter file. Two types of filter files are supported: **SPI.-OBS.-FLT** (filter using the SPI pointing identifier) and **SPI.-GNRL-BTI** (filter pointings on basis of SPI bad-time-intervals). All pointings or bad-time-intervals that occur in the filter file will be excluded from the output pointing file. This allows to apply a pointing based data filter.
- **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 pointing definition data structure **SPI.-OBS.-PNT** 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.
- **outpntDOL** specifies the DOL of the pointing definition data structure (HDU [SPI.-OBS.-PNT]) that will be attached to the output Observation Group.
- **useAUX** specifies if **spi_obs_point** should attempt to find pointing information from auxiliary data that is attached to the Observation Group. If no such data will be found, **spi_obs_point** emits a warning but continues execution. Setting **useAUX=yes** does not harm in the case that no auxiliary data is available. However, if auxiliary data have been found, **spi_obs_point** will in this case not use the pointing information from the Science Window Groups or the ASCII input file.
- **useHist** specifies if historic pointing information should be used or not. This option may allow to force omitting historic pointing information, i.e. in the case that the file is corrupted.
- **useSWG** specifies if **spi_obs_point** should attempt to find pointing information from the Science Window Groups. In this case, **spi_obs_point** will read all relevant keywords for each Science Window Group, and will extract pointing information from these keywords.
- **useTXT** specifies if **spi_obs_point** should use pointing information from the ASCII file. This interpretation of the ASCII file depends on the value specified for the keyword **useSWG**.
If **useSWG=yes**, the pointing information for each science window will be extracted from the ASCII file. If a science window has no corresponding entry in the ASCII file (i.e. if there is no line with a corresponding science window identifier), the information contained in the science window group header will be used. If such information cannot be extracted from the header, the pointing will be set to 0 degrees in Right Ascension and Declination.
If **useSWG=no**, the ASCII file specifies the pointings that should be created. If the pointing information from the AUX data should be overwritten make sure that **useAUX=no**.
- **useSlew** specifies if slew Science Window Groups should be used in case of **useSWG=yes**. The Science Window Group type will be determined from the Science Window identifier, and slew Science Windows will be omitted if **useSlew=no**. This is reasonable since we cannot specify a single pointing for a slew in an ASCII file.

- `xtol` specifies the maximum pointing fluctuation tolerance of the INTEGRAL X-axis in units of arcmin that is accepted for a pointing to be considered as stable. This parameter is only used if pointing information is extracted from auxiliary data.
- `ztol` specifies the maximum pointing fluctuation tolerance of the INTEGRAL Z-axis in units of arcmin that is accepted for a pointing to be considered as stable. This parameter is only used if pointing information is extracted from auxiliary data.
- `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 or pointing definition structure), the task will exit with an error message.
- `verbose` specifies the verbose level of the executable during preparation of the output Observation Group. 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.

4 Interface definition

`spi_obs_point` takes the information that is available in either the Observation Group or an auxiliary ASCII file to generate a pointing definition data structure of type `SPI.-OBS.-PNT`.

From the input Observation Group, the following HDUs are required:

- `GNRL-IDX.-GRP` science window group index that groups all science window groups of level PRP.

The input Observation Group is copied into the output Observation Group, with the following additional HDUs attached:

- `SPI.-OBS.-PNT` pointing definition.

If no output Observation Group is specified, `spi_obs_point` attempts to make the input Observation Group to the output Observation Group, which is only possible if the `clobber` parameter is set to `yes`.

The ASCII pointing definition file can have the two following formats, depending on the value of `useSWG`.

If `useSWG=yes`, the format is:

```
900001010022 1000.000 1000.100 90.000 +10.000 90.000 -80.000
900001010032 1000.100 1000.300 90.000 +10.000 90.000 -80.000
900001010042 1000.300 1000.400 90.000 +10.000 90.000 -80.000
...
```

where the first 12 digit integer field specifies the Science Window Identifier, the following 2 floating-point parameters specify the pointing start and stop time (in IJD), and the last 4 floating-point parameters specify the pointing in degrees, with the meaning

- `RA_X` : SPI pointing axis Right Ascension ([0, 360])
- `DEC_X` : SPI pointing axis Declination ([-90, 90])
- `RA_Z` : SPI secondary axis Right Ascension ([0, 360])
- `DEC_Z` : SPI secondary axis Declination ([-90, 90])

Note that in the actual version the user has to assure himself that the angular separation of the X and Z axis definition amounts to 90 degrees. Note also that the start and stop time information is only used if no start and stop time has been found in the Science Window Header.

If useSWG=no, the format is:

```
04000001.000001 1000.000 1000.100 90.000 +10.000 90.000 -80.000
04000001.000002 1000.100 1000.300 90.000 +10.000 90.000 -80.000
04000001.000003 1000.300 1000.400 90.000 +10.000 90.000 -80.000
...
```

where the Science Window Identifier is replaced by the SPI Pointing Identifier (the remaining fields are identical to the case useSWG=yes).

The pointing filter `filterDOL` is a FITS file that specifies the pointings that should be excluded from the data. Two formats are supported, `SPI.-OBS.-FLT` (filter using the SPI pointing identifier) and `SPI.-GNRL-BTI` (filter pointings on basis of SPI bad-time-intervals).

A `SPI.-OBS.-FLT` table is comprised of two columns:

- `PTID.SPI` : SPI pointing identifier that should be excluded. The general SPI pointing identifier has the format `RRRRPPPP.SSSSSS` where `RRRR` is the revolution number, `PPPP` is the pointing number, and `SSSSSS` is the subdivision number. The pointing identifier is read from left to right, and all pointings that match the specified string will be excluded. For example, to exclude from revolution 0123 all pointings between 0010 and 0019 one may specify 0123001 in the filter file. Or to exclude all subdivision of the pointing 0093 in revolution 0047 one may specify 00470093.
- `REASON` : an ASCII string that specifies the reason why the pointing should be filtered (e.g. high count rate, solar flare, invalid instrument configuration, ...)

A `SPI.-GNRL-BTI` table is comprised of 9 columns:

- `START` : IJD start time of the time interval that should be excluded from the data. This time is used by `spi_obs_point` to define bad-time-intervals.
- `STOP` : IJD stop time of the time interval that should be excluded from the data. This time is used by `spi_obs_point` to define bad-time-intervals.
- `OBS.START` : OBTime that corresponds to `START`. Not used by `spi_obs_point`.
- `OBS.END` : OBTime that corresponds to `STOP`. Not used by `spi_obs_point`.
- `UTC.START` : UTC time string that corresponds to `START`. Not used by `spi_obs_point`.
- `UTC.END` : UTC time string that corresponds to `STOP`. Not used by `spi_obs_point`.
- `POINTING_ID` : Pointing identifier of the time interval that is excluded. Not used by `spi_obs_point`.
- `BTI.TYPE` : Type of bad-time-interval. Not used by `spi_obs_point`.
- `COMMENTS` : an ASCII string that specifies the reason why the pointing should be filtered (e.g. high count rate, solar flare, invalid instrument configuration, ...)

`spi_obs_point` calculates the overlapping time duration between each pointing and each row of the `SPI.-GNRL-BTI` table. If this overlapping duration exceeds 10 seconds, the corresponding pointing is excluded by `spi_obs_point`.

5 Algorithm

Please refer to the DAL3AUX User Manual for more information about INTEGRAL attitude determination.

6 Error codes

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

<code>SPI_OBS_POINT_ERROR_MEM_ALLOC</code>	-233100
<code>SPI_OBS_POINT_ERROR_INVALID_LIST</code>	-233101
<code>SPI_OBS_POINT_ERROR_BAD_FILE</code>	-233102
<code>SPI_OBS_POINT_ERROR_BAD_PAR</code>	-233103

They have the following meaning:

- `SPI_OBS_POINT_ERROR_MEM_ALLOC` : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task.
- `SPI_OBS_POINT_ERROR_INVALID_LIST` : an invalid list of pointings has been encountered. This error should never happen and indicates that the executable has somewhere a bug. Please report this to the author.
- `SPI_OBS_POINT_ERROR_BAD_FILE` : `spi_obs_point` was unable to open the specified ASCII file given by the parameter `intxtDOL`. Probably, the specified file name was wrong.
- `SPI_OBS_POINT_ERROR_BAD_PAR` : no valid pointing determination method has been specified.

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.