

spi_psd_sim2fits

User Manual

Version 1.1.0
30 September 2002

Jürgen Knödseder
Centre d'Etude Spatiale des Rayonnements
knodseder@cesr.fr
<http://www.cesr.fr/~jorgen/index.html>

Contents

1	Introduction	1
2	Getting started	1
3	Parameter file	2
4	Interface definition	4
5	Algorithm	4
6	Error codes	4

1 Introduction

The executable `spi_psd_sim2fits` is an ISDC compliant executable that transforms a file of simulated PSD pulse shapes into an ISDC compliant data format for analysis using the ISDC software system. For each run, the task generates a Science Window Group (SWG) that contains the simulated PSD pulse shapes in the `SPI.-xCRV-RAW` data structure and the corresponding PSD events in the `SPI.-xPSD-RAW` data structure. `x` is one of `O`, `E`, `C`, or `D` and indicates the SPI science mode for which the simulated data were generated (Operational, Emergency, Calibration, or Diagnostics mode).

The FITS data structure can be found in the file `spi_raw_xxxx.fits`, where `xxxx` is either `oper`, `emer`, `calb`, or `diag`. The file is located in the science window tree in the directory `spi/raw`.

The task also updated the photon block structures `SPI.-xBLK-HRW`, where `x` again indicates the science mode. The block structures are found in the file `spi_raw_block.fits` which is located in the science window tree in the directory `eng/raw`.

Finally, `spi_psd_sim2fits` also creates the Science Window Group grouping table `swg_raw.fits` which is found at the base of the Science Window Group directory. All other files that are found in the Science Window Group are empty (yes, there are so many files but they are all empty).

2 Getting started

Before installing `spi_psd_sim2fits`, make sure that the ISDC support platform 4.1 or higher is installed on your system, and that the libraries `spi_psdlib` (version 1.5.0 or higher), `spi_toolslib` (version 1.8.0 or higher), and `tools_lib` (version 2.2.4 or higher) are installed. The first two libraries may be downloaded at the web page <http://www.cesr.fr/~jurgen/isdc> while `tools_lib` can be obtained from ISDC (you can also get the `tools_lib` library included in the `sys-sw` package that you find on the author's web site).

After downloading the `spi_psd_sim2fits.tar.gz` file, step into a directory that should hold the distribution, move the `spi_psd_sim2fits.tar.gz` file into this directory and type:

```
$ gunzip spi_psd_sim2fits.tar.gz
$ tar xvf spi_psd_sim2fits.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
```

3 Parameter file

```
#####
#
#           Centre d'Etude Spatiale des Rayonnements           #
#           (in collaboration with ISDC)                       #
#
#           PSD PULSE SHAPE SIMULATIONS                       #
#
# -----#
#
# File:      spi_sim2fits.par                                  #
# Version:   1.1.0                                           #
# Component: sim                                             #
#
# Author:    Juergen Knoedlseder                             #
#            knodlseder@cesr.fr                              #
#            CESR                                            #
#
# Purpose:   Parameter file of the SPI pulse shape simulation #
#            executable                                       #
#
# History:   1.1.0 30-Sep-2002 First ISDC delivery (Rev. 1)  #
#
#####
#
# Input Simulation data file
#=====
infile, fr, ql, "psd_sim_sample_30_9_2002.txt",,, "PSD simulated pulse shapes input file name"
#
# Base paths
#=====
scwbase, s, hl, "${REP_BASE_PROD}/scw",,, "Science Window Base Path"
cfgbase, s, hl, "${ISDC_ENV}/templates",,, "Template Base Path"
#
# Science window definition
#=====
revid,    s, ql, "9999",,, "Revolution ID"
pointid,  s, ql, "0000",,, "Pointing ID"
subid,    s, ql, "000",,, "Subdivision ID (minimum is 001)"
scwtype,  s, ql, "0",0|1|2,, "Science Window Type (0=pointing, 1=slew, 2=other)"
version,  s, ql, "001",,, "Version number"
previd,   s, ql, "000000000000",,, "Previous Science Window ID"
#
# Task parameters
#=====
detid0,  i, h, 0, 0, 18, "Detector ID channel 0"
detid1,  i, h, 1, 0, 18, "Detector ID channel 1"
detid2,  i, h, 2, 0, 18, "Detector ID channel 2"
detid3,  i, h, 3, 0, 18, "Detector ID channel 3"
detid4,  i, h, 4, 0, 18, "Detector ID channel 4"
detid5,  i, h, 5, 0, 18, "Detector ID channel 5"
detid6,  i, h, 6, 0, 18, "Detector ID channel 6"
```

```

detid7, i, h, 7, 0, 18, "Detector ID channel 7"
detid8, i, h, 8, 0, 18, "Detector ID channel 8"
detid9, i, h, 9, 0, 18, "Detector ID channel 9"
detid10, i, h, 10, 0, 18, "Detector ID channel 10"
detid11, i, h, 11, 0, 18, "Detector ID channel 11"
detid12, i, h, 12, 0, 18, "Detector ID channel 12"
detid13, i, h, 13, 0, 18, "Detector ID channel 13"
detid14, i, h, 14, 0, 18, "Detector ID channel 14"
detid15, i, h, 15, 0, 18, "Detector ID channel 15"
detid16, i, h, 16, 0, 18, "Detector ID channel 16"
detid17, i, h, 17, 0, 18, "Detector ID channel 17"
detid18, i, h, 18, 0, 18, "Detector ID channel 18"
psdmode, s, h, "OPER", OPER|EMER|CALB|DIAG,, "PSD sub-assembly mode"
method, s, h, "PROCESS", SCAN|PROCESS,, "Working method (SCAN/PROCESS)"
#
# Standard parameters
#=====
clobber, b, h, yes,, "Overwrite existing output data ?"

```

The following parameters have to be specified:

- **infile** : specifies the file name (including the path) of the simulated PSD pulse shape data (i.e. your ASCII file)
- **scwbase** : specifies the base directory into which the science window groups are stored.
- **cfgbase** : specifies the directory where the ISDC template files from the support platform reside (they are needed to build the science window group)
- **revid** specifies the revolution identifier of the science window that should be generated. This parameter has to be a 4 character string.
- **pointid** specifies the pointing identifier of the science window that should be specified. This parameter has to be a 4 character string. The minimum pointing identifier is 0000.
- **subid** specifies the subdivision identifier of the science window that should be specified. This parameter has to be a 3 character string. The minimum subdivision identifier is 001.
- **scwtype** specifies the science window type. The following types are allowed: 0 is a pointing, 1 is a slew, and 2 is any other science window type.
- **version** specifies the science window version number. This parameter has to be a 3 character string.
- **previd** specifies the previous science window identifier (without version number). This identifier will be stored in the science window grouping table header, and allows for linking of the science windows.
- **detid nn** (where nn runs from 0 to 18) specifies the mapping on input data detector numbers to the detector identifier stored in the output data. **In the actual version it is assumed that all curves have an in put detector number of 0, hence only detid0 is used so far.**
- **psdmode** specifies the PSD sub-assembly mode for which the pulse data should be simulated. Possible choices are OPER, EMER, CALB, and DIAG. Note that in EMER mode, no PSD curves are saved.
- **method** specifies if the input data should only be scanned (SCAN) or if they should be scanned and subsequently also processed (PROCESS). The default is PROCESS.
- **clobber** : if set to **yes**, the output data structure will be deleted before a new one is created. This allows to overwrite old data structures. Be careful when using this option since you may loose some data in this case.

4 Interface definition

spi_psd_sim2fits produces an ISDC compliant data structure with the following FITS files and HDUs filled:

swg_raw.fits	[GNRL_SCWG_RAW]
eng/raw/spi_raw_block.fits	[SPI.-OBLK-HRW]
	[SPI.-EBLK-HRW]
	[SPI.-CBLK-HRW]
	[SPI.-DBLK-HRW]
spi/raw/spi_raw_oper.fits	[SPI.-OPSD-RAW]
	[SPI.-OCRV-RAW]
	[SPI.-OPPS-RAW]
spi/raw/spi_raw_emer.fits	[SPI.-EPSD-RAW]
	[SPI.-ECRV-RAW]
	[SPI.-EPPS-RAW]
spi/raw/spi_raw_calb.fits	[SPI.-CPSD-RAW]
	[SPI.-CCRV-RAW]
	[SPI.-CPPS-RAW]
spi/raw/spi_raw_diag.fits	[SPI.-DPSD-RAW]
	[SPI.-DCRV-RAW]
	[SPI.-DPPS-RAW]

The extension characters **O**, **E**, **C**, and **D** stand for Operation, Emergency, Calibration, and Diagnostic mode, and only the corresponding structures are filled depending on the `psdmode` parameter. Usually, for each curve also a PSD event is generated (hence the **PSD** and **CRV** structures are filled with one row per curve).

The input ASCII file contains one PSD curve per row. The first 2 integers in the row reflect the simulated time (**SCTIME**) and the detector identifier (**DETID**) as extracted from the Goddard simulations. The third number is a floating point number and gives the energy of the event in keV. The following 79 integers describe the pulse shape, and may take values between 0 and 511 (inclusive).

5 Algorithm

There is no algorithm implemented.

6 Error codes

No executable specific error code is defined. However, ISDC support errors or `tools_lib` error may be returned by the executable.