

SPIRE User's Reference Manual

Herschel Data Processing

Version 3.0, Document Number: HERSCHEL-HSC-DOC-0935
05 October 2010



SPIRE User's Reference Manual: Herschel Data Processing

Table of Contents

I. Categorized view of Commands	ix
II. How to use this manual	xvi
II.1. Related documentation	xvi
1. DP Commands	1
1.1. Introduction	1
1.2. AbstractApodizeFunction	3
1.3. AbstractMapperTask	4
1.4. AddIfgmTask	6
1.5. addPointingParam	7
1.6. ApodizeIfgmTask	9
1.7. associateSkyPosition	13
1.8. AverageInterferogramTask	15
1.9. AverageSpectraTask	17
1.10. BandEdge	21
1.11. BaselineCorrectionTask	22
1.12. BathTempTimeline	25
1.13. BeamProf	26
1.14. BolometerPhysics	27
1.15. BolometerResponseCorrectionTask	28
1.16. BolPar	31
1.17. BsmAnglesTimeline	32
1.18. BsmFlagsExtract	33
1.19. BsmOps	34
1.20. BsmPos	35
1.21. BuildingBlockContext	36
1.22. BuildingBlockProduct	37
1.23. calcBathTemp	38
1.24. CalcBsmAnglesTask	40
1.25. calcBsmFlags	42
1.26. calcJfetVolt	44
1.27. calcOpticalPower	46
1.28. calcRmsVoltRes	47
1.29. CalibrationOdPlugin	50
1.30. CalibrationPlugin	51
1.31. ChanGain	53
1.32. ChanMask	54
1.33. ChannelCalibTable	55
1.34. ChanNoise	56
1.35. ChanNomRes	57
1.36. ChanNum	58
1.37. ChanTimeConst	59
1.38. ChanTimeOff	60
1.39. checkAdcLatchTrunc	61
1.40. CheckClip	63
1.41. ChopJiggTimeline	64
1.42. ClippingCorrectionTask	65
1.43. ComboPlotLayoutDirectorImp	67
1.44. concurrentGlitchDeglitcher	68
1.45. CreateIfgmTask	70
1.46. CreateOffsetHistTask	77
1.47. CreateResetHistTask	79
1.48. createSpirePointing	81
1.49. cutPhotDetTimelines	83
1.50. CWT	85
1.51. dbQueryMaker	86

1.52. DeglitchIfgmTask	89
1.53. DemodPhotProduct	92
1.54. demodulate	93
1.55. DeNoddingTask	95
1.56. DetAngOff	97
1.57. Detection	98
1.58. DetectorTimelineExplorerComponent	99
1.59. DetectorTimelineExplorer	100
1.60. DetectorTimeline	102
1.61. DivideIfgmTask	104
1.62. EdpBlockContext	105
1.63. EdpContext	106
1.64. EdpProduct	107
1.65. ElecCross	108
1.66. elecCross	109
1.67. engConversion	111
1.68. EngConvOdPlugin	117
1.69. eventLogger	118
1.70. Ext	120
1.71. ExtList	121
1.72. ExtRep	122
1.73. FactorFinder	123
1.74. FlatFielderTask	124
1.75. FlatField	125
1.76. formatConversion	126
1.77. FourierTransformTask	128
1.78. GaussianApodizeFunction	134
1.79. GenericSpireDataFrameImpl	135
1.80. GlitchFinder	136
1.81. GlitchRemoval	141
1.82. HammingApodizeFunction	143
1.83. HanningApodizeFunction	144
1.84. HolderList	145
1.85. HolderRep	146
1.86. HousekeepOdPlugin	147
1.87. IfgmTask	148
1.88. IfgmZeropadder	149
1.89. InstModeMask	150
1.90. JiggleAverageTask	151
1.91. JiggPhotProduct	152
1.92. JiggPointSourceFitProduct	153
1.93. joinNhkTimelines	154
1.94. joinPhotDetTimeline	156
1.95. Level0BlockContext	158
1.96. Level0Context	159
1.97. Level0OdPlugin	160
1.98. Level0Plugin	161
1.99. Level1Context	163
1.100. LinearRegression	164
1.101. LpfPar	165
1.102. LpfResponseCorrectionTask	166
1.103. MadScanMapperTask	168
1.104. maskBadChan	171
1.105. maskBadTmParam	173
1.106. MedianInterferogramTask	174
1.107. MexicanHat	175
1.108. Mosaic	176
1.109. MultiplyIfgmTask	178

1.110. NaiveApppMapperTask	179
1.111. NaiveScanMapperTask	181
1.112. NhkTimeline	183
1.113. NodAverageTask	184
1.114. NonLinCorr	186
1.115. NortonBeerApodizeFunctionOnePointEight	187
1.116. NortonBeerApodizeFunctionOnePointFive	188
1.117. NortonBeerApodizeFunctionOnePointFour	189
1.118. NortonBeerApodizeFunctionOnePointNine	190
1.119. NortonBeerApodizeFunctionOnePointOne	191
1.120. NortonBeerApodizeFunctionOnePointSeven	192
1.121. NortonBeerApodizeFunctionOnePointSix	193
1.122. NortonBeerApodizeFunctionOnePointThree	194
1.123. NortonBeerApodizeFunctionOnePointTwo	195
1.124. NortonBeerApodizeFunctionTwoPointZero	196
1.125. ObsData	197
1.126. OffsetHist	198
1.127. OptCross	199
1.128. Pcal	200
1.129. PcalOdPlugin	201
1.130. PcalTask	202
1.131. PhaseCorrectionTask	205
1.132. PhotBolPar	210
1.133. PhotBsmOps	211
1.134. PhotBsmPos	212
1.135. PhotChanGain	213
1.136. PhotChanMask	214
1.137. PhotChanNoise	215
1.138. PhotChanNomRes	216
1.139. PhotChanNum	217
1.140. PhotChanTimeConst	218
1.141. PhotChanTimeOff	219
1.142. PhotDetAngOff	220
1.143. PhotElecCross	221
1.144. PhotFlatField	222
1.145. PhotFluxConversionTask	223
1.146. PhotFluxConv	225
1.147. PhotInstModeMask	226
1.148. PhotLpfPar	227
1.149. PhotOffsetHist	228
1.150. PhotOptCrossCorrectionTask	229
1.151. PhotOptCross	231
1.152. PhotPcal	232
1.153. PhotRsrF	233
1.154. PhotTempDriftCorr	234
1.155. PointedPhotProduct	235
1.156. PointedPhotTimeline	236
1.157. PointedProduct	237
1.158. PointSourceFitTask	238
1.159. qIaDataConversion	240
1.160. QualAssessSpireLevel05Task	242
1.161. QualAssessSpireLevel1Task	244
1.162. QualAssessSpireLevel2Task	246
1.163. QualAssessSpireTask	248
1.164. QualityPlugin	250
1.165. rawDataExtraction	251
1.166. rawProductCreator	255
1.167. Reconstruction	258

1.168. removeBaseline	259
1.169. ResetHist	261
1.170. ScanContext	262
1.171. SecondDeglitching	263
1.172. SigmaKappaDeglitcherTask	264
1.173. SignalJumpDetectorTask	270
1.174. SingleBlockContext	272
1.175. SourceFluxTask	273
1.176. SpecBandEdge	274
1.177. SpecBeamParam	275
1.178. SpecBolPar	276
1.179. SpecBsmOps	277
1.180. SpecBsmPos	278
1.181. SpecChanGain	279
1.182. SpecChanMask	280
1.183. SpecChanNoise	281
1.184. SpecChanNomRes	282
1.185. SpecChanNum	283
1.186. SpecChanTimeConst	284
1.187. SpecChanTimeOff	285
1.188. SpecDetAngOff	286
1.189. SpecElecCross	287
1.190. SpecExplorer	288
1.191. SpecFlatField	290
1.192. SpecFluxConversionTask	291
1.193. SpecFluxConv	294
1.194. SpecInterRef	295
1.195. SpecLpfPar	296
1.196. SpecNlp	297
1.197. SpecNonLinCorr	298
1.198. SpecNonLinearityCorrectionTask	299
1.199. SpecOeOpd	301
1.200. SpecOffsetHist	302
1.201. SpecOptCrossCorrectionTask	303
1.202. SpecOptCross	305
1.203. SpecPcal	306
1.204. SpecPhaseCorrLim	307
1.205. SpecScalEm	308
1.206. SpecScalRsrF	309
1.207. SpecSmecStepFactor	310
1.208. SpecSmecZpd	311
1.209. SpecTempDriftCorr	312
1.210. SpectrometerDetectorInterferogram	313
1.211. SpectrometerDetectorSpectrum	318
1.212. SpectrometerPointSourceSpectrum	321
1.213. SpireCal	322
1.214. spireCal	324
1.215. SpireCalibrationProduct	326
1.216. SpireDataProduct	327
1.217. SpireInterferogramId	328
1.218. SpireInterferogramCompositeDataset	329
1.219. SpireListContext	330
1.220. SpireMapContext	331
1.221. SpireParameter	332
1.222. SpirePhotLargeMapPipelineTask	333
1.223. SpirePhotParallelPipelineTask	334
1.224. SpirePhotPointSourcePipelineTask	335
1.225. SpirePhotSmallMapPipelineTask	336

1.226. SpirePointingProduct	337
1.227. SpirePreprocessCubeTask	340
1.228. SpirePreprocessedCube	343
1.229. SpirePreprocessThread	344
1.230. SpirePreprocessTool	346
1.231. SpireProduct	347
1.232. SpireProductRule	348
1.233. SpireProductTypeRule	349
1.234. SpireSpecFullMapPipelineTask	350
1.235. SpireSpecPointSourcePipelineTask	351
1.236. SpireSpectrumId	352
1.237. SpireSpectrumCompositeDataset	354
1.238. SpireTmPacketConverter	355
1.239. splitRawData	356
1.240. SubtractIfgmTask	357
1.241. TelemMask	358
1.242. TelescopeScalSubtractionTask	359
1.243. TempDriftCorr	361
1.244. TemperatureCondition	362
1.245. TemperatureDriftCorrectionTask	364
1.246. timeConvReord	368
1.247. TimeCorrelationTask	371
1.248. TimeDomainPhaseCorrectionTask	373
1.249. TimelineSimplePlot	376
1.250. TimePos	380
1.251. translateMcuets	381
1.252. WaveletDeglitcherTask	382
1.253. Wavelet	386

Categorized view of Commands

This chapter provides a categorized view of all built-in DPfunctions, tasks and objects.




Calibration Product


-  [BsmOps](#)
-  [PhotFluxConv](#)
-  [SpecBeamParam](#)
-  [SpecFluxConv](#)
-  [SpecOeOpd](#)
-  [SpecSmecZpd](#)

Calibration product

-  [BandEdge](#)
-  [BeamProf](#)
-  [BolometerPhysics](#)
-  [BolPar](#)
-  [ChanGain](#)
-  [ChanMask](#)
-  [ChannelCalibTable](#)
-  [ChanNoise](#)
-  [ChanNomRes](#)
-  [ChanNum](#)
-  [ChanTimeConst](#)
-  [ChanTimeOff](#)
-  [DetAngOff](#)
-  [ElecCross](#)
-  [FlatField](#)
-  [InstModeMask](#)
-  [LpfPar](#)
-  [NonLinCorr](#)
-  [OffsetHist](#)
-  [OptCross](#)
-  [PhotBolPar](#)
-  [PhotBsmOps](#)
-  [PhotBsmPos](#)
-  [PhotChanGain](#)
-  [PhotChanMask](#)





- [J PhotChanNoise](#)
- [J PhotChanNomRes](#)
- [J PhotChanNum](#)
- [J PhotChanTimeConst](#)
- [J PhotChanTimeOff](#)
- [J PhotDetAngOff](#)
- [J PhotElecCross](#)
- [J PhotFlatField](#)
- [J PhotInstModeMask](#)
- [J PhotLpfPar](#)
- [J PhotOffsetHist](#)
- [J PhotOptCross](#)
- [J PhotPcal](#)
- [J PhotRsrF](#)
- [J ResetHist](#)
- [J SpecBandEdge](#)
- [J SpecBolPar](#)
- [J SpecBsmOps](#)
- [J SpecChanGain](#)
- [J SpecChanMask](#)
- [J SpecChanNoise](#)
- [J SpecChanNomRes](#)
- [J SpecChanNum](#)
- [J SpecChanTimeConst](#)
- [J SpecChanTimeOff](#)
- [J SpecDetAngOff](#)
- [J SpecElecCross](#)
- [J SpecFlatField](#)
- [J SpecLpfPar](#)
- [J SpecNlp](#)
- [J SpecNonLinCorr](#)
- [J SpecOffsetHist](#)
- [J SpecOptCross](#)
- [J SpecPcal](#)
- [J SpecPhaseCorrLim](#)
- [J SpecScalEm](#)

-  [SpecScalRsrF](#)
-  [SpecSmecStepFactor](#)
-  [TelemMask](#)

 **Calibration table product to be used with TemperatureDriftCorrectionTask**

-  [PhotTempDriftCorr](#)

 **Dataset**



-  [SpireInterferogram1d](#)
-  [SpireInterferogramCompositeDataset](#) - Herschel-SPIRE SpireInterferogramComposite-Dataset Dataset
-  [SpireSpectrum1d](#) - Herschel-SPIRE SpireSpectrum1d Dataset
-  [SpireSpectrumCompositeDataset](#) - Herschel-SPIRE SpireSpectrumCompositeDataset Dataset

 **PCAL Calibration product**




-  [Pcal](#)

 **Pipeline**

 **DeNodding**

-  [PointedPhotProduct](#) - This class creates an empty Pointed Photometer Product. One method is provided,
-  [PointedPhotTimeline](#) - Help







 **Map Making**

-  [MadScanMapperTask](#) - This task takes a Photometer DetectorTimeline and produces a Spire Photometer Map Product
-  [NaiveAppMapperTask](#) - This task takes a Averaged Photometer Pointing Product and produces a Spire Photometer Map Product
-  [NaiveScanMapperTask](#) - This task takes a Photometer DetectorTimeline and produces a Spire Photometer Map Product

 **User Products**


-  [JiggPhotProduct](#)
-  [JiggPointSourceFitProduct](#) - Jiggle Point Source Fit Product.

 **Product**

-  [SpecInterRef](#) - SpecInterRef.
-  [SpecTempDriftCorr](#)
-  [SpectrometerDetectorInterferogram](#) - SpectrometerDetectorInterferogram.
-  [SpectrometerDetectorSpectrum](#) - Herschel-SPIRE Spectrometer Detector Spectrum Product
-  [SpectrometerPointSourceSpectrum](#) - Herschel-SPIRE Spectrometer Point Source Spectrum Product.
-  [SpirePreprocessedCube](#) - Herschel-SPIRE Preprocessed Cube

 **SPIRE**


 [calcOpticalPower](#) - Task compute Optical Power falling on the bolometers

 [createSpirePointing](#) - Task to create the spire pipeline product

 **IA_GUI**

 [DetectorTimelineExplorerComponent](#) - Explorer for DetectorTimeline products.

 [DetectorTimelineExplorer](#) - Explorer for DetectorTimeline products.


 [TimelineSimplePlot](#) - Plotter for DetectorTimeline products


 **Pipeline**

 **Common Task**


 [addPointingParam](#) - Add Pointing Parameters Task.


 [calcBathTemp](#) - Bath temperature computation task.


 [calcJfetVolt](#) - Task to convert raw detector values into JFET voltages.


 [calcRmsVoltRes](#) - RMS bolometer voltage and resistance computation task.

 [checkAdcLatchTrunc](#) - Check ADC Latch-up and Truncation task.


 [cutPhotDetTimelines](#) - Task to cut a Pointed Photometer Timelines that were previously joined with the detector timelines


 [dbQueryMaker](#) - Database Query Maker Task.

 [engConversion](#) - Engineering Conversion Task.


 [eventLogger](#) - Event logger task.

 [formatConversion](#) - Format Conversion Task.

 [joinNhkTimelines](#) - Task to join scan line NHK timelines with the NHK timelines of the leading and trailing

 [joinPhotDetTimeline](#) - Task to join scan line detector timelines (both photometer and spectrometer) with the detector timelines of the leading and trailing turnaround building block in scan maps.


 [maskBadChan](#) - Mask bad channels task.

 [maskBadTmParam](#) - Mask bad telemetry parameters task.

 [qlaDataConversion](#) - QLA Data Conversion Task.

 [rawDataExtraction](#) - Raw Data Extraction Task.

 [rawProductCreator](#) - Raw Product Creator Task.


 [removeBaseline](#) - Task to remove the baseline from Pointed Photometer Timeline.


 [splitRawData](#) - Split Raw Data Task.

 [timeConvReord](#) - Time Conversion and Reordering Task.

 [translateMcuets](#) - Translate MCUET Task.

 **Spectrometer Task**

 [ApodizeIfgmTask](#) - Task to apply an apodization function to a set of interferograms.

 [AverageInterferogramTask](#) - Task that calculates the detector-by-detector and opd-by-opd median of the interferogram samples in a SPIRE SDI product.

- [AverageSpectraTask](#) - AverageSpectraTask
- [BaselineCorrectionTask](#) - Task to remove from a set of interferograms the offset and low frequency drift.
- [ClippingCorrectionTask](#) - Class reconstructs the signal where it is clipped in SDT product
- [CreateIfgmTask](#) - Task used to create SPIRE Spectrometer interferograms.
- [DeglitchIfgmTask](#) - Task to perform Second Level Deglitching on SPIRE Spectrometer Data.
- [FactorFinder](#) - FactorFinder
- [FourierTransformTask](#) - Task to apply the Fourier Transform to create a set of spectra from a set of interferograms.
- [GlitchFinder](#) - Class to identify glitches in SPIRE Spectrometer Data using various glitch detection algorithms.
- [GlitchRemoval](#) - Class to remove glitches that have been identified in SPIRE Spectrometer Data.
- [IfgmZeropadder](#) - IfgmZeropadder
- [MedianInterferogramTask](#) - Task that calculates the detector-by-detector and opd-by-opd median of the interferogram samples in a SPIRE SDI product.
- [ObsData](#) - Helper class that stores general SPIRE Spectrometer observation data for CreateIfgmTask.
- [PhaseCorrectionTask](#) - Task to correct for asymmetries in SPIRE Spectrometer interferograms.
- [SpecFluxConversionTask](#) - Flux Conversion for the SPIRE Spectrometer
- [SpecOptCrossCorrectionTask](#) - Task to apply Optical crosstalk correction to a Spectrometer Detector Spectrum
- [SpirePreprocessCubeTask](#) - SpirePreprocessCubeTask
- [TelescopeScalSubtractionTask](#) - The TelescopeScalSubtractionTask Work-package :
- [TemperatureCondition](#) - This task checks the sources temperature fluctuations. Its result is a boolean :
- [TimeDomainPhaseCorrectionTask](#) - Task used to correct SPIRE FTS science timelines for the delay induced by the readout electronics and the thermal time constant of the bolometers.
- [TimePos](#) - Helper class to store Time/Position doublets for the CreateIfgmTask.
- [WaveletDeglitcherTask](#) - WaveletDeglitcherTask

📁 **SPIRE task**

- [spireCal](#) - SPIRE Calibration Task.

📁 **TempDriftCorrCalibTable Calibration product**

- [TempDriftCorr](#)


📁 **algorithm**


- [BsmFlagsExtract](#) - This is a the version using the current of the chopper.

- class**
 - [Detection](#)
 - [Ext -](#)
 - [HolderList](#)
 - [Reconstruction](#)
 - [Wavelet](#)
 - herschel**
 - spire**
 - ia**
 - common**
 - timecorr**
 - [TimeCorrelationTask](#) - This module applies a time correction from on-board-time (OBT) to TAI. These times are also known as
 - pipeline**
 - phot**
 - bolorespcorr**
 - [BolometerResponseCorrectionTask](#) - This task takes a Photometer Detector Timeline (PDT) product as input and removes the Bolometer transient reponse.
 - elecfiltercorr**
 - [LpfResponseCorrectionTask](#) - This task takes a Photometer Detector Timeline (PDT) product as input
 - pointsource**
 - [PointSourceFitTask](#) - This task takes an Averaged Pointed Photometer Product and produces a
 - [SourceFluxTask](#) - This task takes a Jiggle Point Source Fit Product and produces a Jiggled Photometer Product. This task is part of the jiggle pipeline, between Level 0.5 and Level 1.
 - spire**
 - ia**
 - pipeline**
 - phot**
 - fluxconv task**
 - [PhotFluxConversionTask](#) - The task corrects detector voltage non-linearities and then converts the result to optical power.
- spire_ia_pipeline_common_deglitch task**
 - [SigmaKappaDeglitcherTask](#) - This task detects and removes cosmic ray glitches from a detectortimeline.
- spire_ia_pipeline_phot_pcal task**


 [PcalTask](#) - PCAL Processing Task


 **task**


 [AbstractMapperTask](#) - abstract class for the SPIRE mappers


 [associateSkyPosition](#) - Task to add RA and Dec in photometer detector timelines


 [CalcBsmAnglesTask](#) - This is the task which converts the BSMT from raw unit to degree


 [calcBsmFlags](#) - CalcBsmFlagsTask

 [demodulate](#) - This task follows


 [DeNoddingTask](#) - produces a Pointed Photometer Product.


 [elecCross](#) - Task to correct the electric crosstalk for the different bolometers.


 [FlatFielderTask](#) - This is a prototype of the correction task


 [JiggleAverageTask](#) - average the Demodulated Photometer Product for each jiggle positions


 [NodAverageTask](#) - produces an Averaged Pointed Photometer Product.


 [PhotOptCrossCorrectionTask](#) - Task to correct the Optical Crosstalk for the different bolometers.


 [QualAssessSpireLevel05Task](#) - Task for quality assessment of level 0.5 Products


 [QualAssessSpireLevel1Task](#) - Task for quality assessment of level 1 Products


 [QualAssessSpireLevel2Task](#) - Task for quality assessment of level 2 Products


 [QualAssessSpireTask](#) - Parent Task class for other quality assessment Tasks


 [SecondDeglitching](#) - This is a prototype of second level deglitcher


 [SignalJumpDetectorTask](#) - This task detects jumps in temperature and dark pixel detectors


 [SpirePhotLargeMapPipelineTask](#) - Task to execute the SPIRE Photometer Large Map Pipeline (POF5)

 [SpirePhotParallelPipelineTask](#) - Task to execute the SPIRE Photometer Parallel Pipeline (PARALLEL)

 [SpirePhotPointSourcePipelineTask](#) - Task to execute the SPIRE Photometer Point Source Pipeline (POF2)

 [SpirePhotSmallMapPipelineTask](#) - Task to execute the SPIRE Photometer Small Map Pipeline (either POF3 or POF10)

 [SpireSpecFullMapPipelineTask](#) - Execute the SPIRE Spectrometer Fully Sampled Map Pipeline (SOF2)

 [SpireSpecPointSourcePipelineTask](#) - Execute the SPIRE Spectrometer Point Source Pipeline (SOF1)

How to use this manual

The *User's Reference Manual* contains information about all the main tasks and classes that you can use within your scripts. There are four version of this manual: the *HCSS* version describes the functions provided by the core Herschel software, and is always shipped with HIPE; the three other versions describe functions provided by HIFI, PACS and SPIRE software. You may or may not have these additional manuals, depending on the software you installed.

This manual includes the following sections:

- [Categorized view of Commands](#). All the functions described in the manual, organised by category.
- [Section 1.1](#). This section lists all the available commands in alphabetical order. Each command has a description, usage instructions and examples.

II.1. Related documentation

The aim of this manual is to provide reference information for all the user-relevant aspects of the SPIRE software. Despite our efforts, you may find that some routines are missing, or have incomplete or inaccurate descriptions. In this case you are encouraged to consult the *Javadoc* developer's reference documentation. You can access the Javadoc by clicking on *SPIRE Developer's Reference Manual (API)* in the table of contents of the HIPE Help System.

Guidance on how to use the Javadoc is provided in the *Scripting and Data Mining* guide: ????

Some Javadoc pages may have links to more in-depth developer documentation. Be aware that these are *not* fully fledged help documents and are most useful to system developers or advanced users only.

Inspecting the source code

If you are an advanced user versed in Java and Jython, you might want to have a look at the source code of a task or class. You can include source code in your HIPE installation in the following ways:

- If you are installing a developer build via the Continuous Installation System, the `--src=yes` option will install source code. This is enabled by default if you use the `--developer` option.

With the `--unpack=yes` option, the source code will be unpacked into a `src` subdirectory in your HIPE installation. With the `--unpack=no` option, source files for each module will be kept as ZIP files in the repository (usually under `$HOME/.hcss.d/repository` in UNIX systems and under `%HOMEPATH%/.hcss.d/repository` in Windows).

- If you are using an installer, select the checkbox next to the question *Would you like to have the source code installed?* This is only available if you choose the *Advanced* installation. The source code will be in the `src` subdirectory of your HIPE installation.

Chapter 1. DP Commands

1.1. Introduction

This chapter is a complete listing of all built-in DP functions, task and objects, collectively referred to as commands.

How to use this chapter

All of Dp's commands are documented alphabetically in this chapter. A first section gives a general overview what a command is used for and how to use it. This includes a general description and some examples. The "API summary" section provides a quick overview of the available constructors, methods and properties of a command. It is intended as quick reference for users that just need to remember a certain functionality. The "API details" section provides more detailed information about the constructors, methods and properties. Furthermore it provides links to other references and information about the command's history.

Overview

A table gives the full name of the command, indicates if it is written in Java or Jython and if it is just an ordinary command or if it follows the HCSS Task specifications. The import statement in this table can be copied to a Jython console.

Description

General description of the command.

Examples

Most commands include one or more examples that demonstrates how the command is used. Most of the examples are just one or two lines of DP code that can be entered at the Jython prompt. Others are code fragments or routines designed to serve as an examples for your own programs.

Limitations and Miscellaneous

The "Limitations" and "Miscellaneous" sections describes the boundaries of applicability and other specialties of the command.

API Summary

The API Summary provides a quick summary of the commands constructor, method, and properties. Note that most of the arguments are positional parameters that must be supplied in the order indicated by the command's syntax. However arguments can be often supplied by specifying their names. If its the case then they can provided in any order.

Constructor

Constructors are used to create a command.

Method

Methods are functions of a command that can be executed.

Properties

Properties are fields of a command that can be read or written by a command. INPUT properties contain values that are required by a command, OUTPUT properties contain the result of a command, INOUT provide both functionalities at the same time.


See also

The "See also" section provides links to related commands, to other manuals, or to external resources in the Internet.

History

The "History" section describes the changes occurred to the command.

1.2. AbstractApodizeFunction

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.AbstractApodizeFunction
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import AbstractApodizeFunction

Description

Abstract class for SPIRE Spectrometer apodization functions.


See also

- [GaussianApodizeFunction](#)
- [HammingApodizeFunction](#)
- [HanningApodizeFunction](#)
- [NortonBeerApodizeFunctionOnePointOne](#)
- [NortonBeerApodizeFunctionOnePointTwo](#)
- [NortonBeerApodizeFunctionOnePointThree](#)
- [NortonBeerApodizeFunctionOnePointFour](#)
- [NortonBeerApodizeFunctionOnePointFive](#)
- [NortonBeerApodizeFunctionOnePointSix](#)
- [NortonBeerApodizeFunctionOnePointSeven](#)
- [NortonBeerApodizeFunctionOnePointEight](#)
- [NortonBeerApodizeFunctionOnePointNine](#)
- [NortonBeerApodizeFunctionTwoPointZero](#)

History

- 2009-07-24 - YFR: First version
- 2010-02-26 - TRF: [SPIRE-2275] Fixed Javadoc and URM issues.
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.3. AbstractMapperTask

Full Name:	herschel.spire.ia.pipeline.phot.scanmap.AbstractMapperTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.scanmap import AbstractMapperTask
Category:	task

Description

abstract class for the SPIRE mappers

API Summary

Properties
<code>Product; MANDATORY input [INPUT, No default value, default=no default value]</code>
<code>String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]</code>
<code>Double; OPTIONAL resolution [INPUT, Default output map resolution is 6", default=10" and 14" for PSW]</code>
<code>Wcs; OPTIONAL wcs [INPUT, Default map is North-oriented, default=with the default resolution>]</code>
<code>Boolean; OPTIONAL doMap [INPUT, , default=no default value]</code>
<code>Boolean; OPTIONAL storeTod [INPUT, , default=no default value]</code>
<code>SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]</code>
<code>Double; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]</code>
<code>Boolean; OPTIONAL doInterpolation [INPUT, , default=no default value]</code>
<code>Integer; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]</code>

API details

Properties


<code>Product; MANDATORY input [INPUT, No default value, default=no default value]</code>
PointedProduct or SpireListContext of PointedProduct timelines or a single PointedProduct, Tod or TodBuffer
<code>String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]</code>
Bolometer array to be processed ('PSW', 'PMW', 'PLW', 'SSW', or 'SLW')
<code>Double; OPTIONAL resolution [INPUT, Default output map resolution is 6", default=10" and 14" for PSW]</code>
Resolution of the output map in arcsec

Wcs; OPTIONAL wcs [INPUT, Default map is North-oriented, default=with the default resolution>]
World Coordinate System for the output map
Boolean; OPTIONAL doMap [INPUT, , default=no default value]
If set to false, the Tod product and the Invntt array will be created, but not the map
Boolean; OPTIONAL storeTod [INPUT, , default=no default value]
If set to true, the pre-processed Tod is made available with todBuffer = mapper.getTodBuffer()
SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]
Output map product
Double; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]
If set, reorder and merge the building blocks. The value is the maximum gap in seconds allowed for merging two adjacent building blocks
Boolean; OPTIONAL doInterpolation [INPUT, , default=no default value]
If set to true, NaN values in the timeline are interpolated
Integer; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]
If set, remove polynomial fit from start to end of the timeline. The value specifies the polynomial order

History

- March 2007: First version
- 19 Apr 2007: PC, use of temporary files
- 20 Apr 2007: PC, use of the new DetectorTimeline data structure and use of scan context
- 17 Sep 2007: PC, don't use preamble and postamble methods.
- 21 Nov 2008: PC, initial parameter check has become redundant + comment fixing in AbstractMapperTask


1.4. AddIfgmTask

Full Name:	herschel.spire.ia.dataset.AddIfgmTask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import AddIfgmTask

History

- 2009-04-21 - YFR: First Version
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions and fixed vague descriptions

1.5. addPointingParam

Full Name:	herschel.spire.ia.pipeline.common.engdata.AddPointingParamTask
Alias:	addPointingParam
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import AddPointingParamTask
Category:	SPIRE/Pipeline/Common Task

Description

Add Pointing Parameters Task.

This task is used by the EngConversionTask to add the pointing information metadata parameter in detector timelines.

API Summary

Properties
DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
Long step [INPUT, OPTIONAL, default=No default value.]
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value]
DetectorTimeline result [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
Input Detector Timeline.
Long step [INPUT, OPTIONAL, default=No default value.]
STEP value.
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value]
Nominal Housekeeping Timeline.
DetectorTimeline result [OUTPUT, OPTIONAL, default=No default value.]
Output Detector Timeline.

History

- 2006-06-11 - PPa: First version.
- 2006-06-29 - PPa: Use 2 bytes for scanLineId computation.

- 2006-09-14 - PPa: Compatible with optimized products.
- 2007-03-05 - PPa: Use EdpProduct for nhkt.
- 2007-10-03 - PPa: Update documentation.
- 2008-05-22 - PPa: Updated following the SPIRE AOT Implementation document.
- 2008-06-18 - PPa: Fix nodId and pointNum calculation. Better documentation.
- 2008-07-21 - PPa: Change class name. Add step parameter.
- 2008-10-08 - PPa: Change names of spectrometer modes (SPR-0713).
- 2008-12-15 - PPa: Add parameter descriptions (SPR-0985). Fix history format.
- 2009-01-26 - PPa: Support "PARALLEL" instMode [SPR-1168].
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2010-02-15 - PPa: Support POF10 instMode. Improve URM.
- 2010-05-17 - PPa: Correct category [SPIRE-2259].

1.6. ApodizeIfgmTask

Full Name:	herschel.spire.ia.pipeline.spec.apodize.ApodizeIfgmTask
Alias:	ApodizeIfgmTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.apodize import ApodizeIfgmTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to apply an apodization function to a set of interferograms.

Task Execution Steps: (from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

1. For each interferogram in each scan:
 - a. Multiply the interferogram by the specified apodization function

If the apodType is set to "prePhaseCorr" the resultant interferograms will be the same length as the input interferogram. If the apodType is set to "postPhaseCorr" the resultant interferograms will be restricted to OPDs ≥ 0 .

Examples

Example 1: Apply the Norton Beer 2.0 apodization function to the symmetric portion of a set of interferograms.

```
presdi = apodizeIfgm(sdi=sdi, apodType="prePhaseCorr", apodName="aNB_20")
```

Example 2: Apply the Norton Beer 1.5 apodization function to the right-hand side (OPD greater than or equal to 0) of a set of interferograms.

```
sdi = apodizeIfgm(sdi=sdi, apodType="postPhaseCorr", apodName="aNB_15")
```

API Summary

Properties
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]
String apodType [INPUT, MANDATORY, default="prePhaseCorr"]
String apodFunctionName [INPUT, OPTIONAL, default="aNB_15"]
SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=no default value]

API details

Properties

[SpectrometerDetectorInterferogram](#) **sdi** [INPUT, MANDATORY, default=no default value]

Input SDI whose contents are to be apodized

String apodType [INPUT, MANDATORY, default="prePhaseCorr"]
The type of apodization to be performed. Can be either "prePhaseCorr" or "postPhaseCorr". The "postPhaseCorr" apodization will result in interferograms whose OPDs are ≥ 0 .
String apodFunctionName [INPUT, OPTIONAL, default="aNb_15"]
Name of the apodization function. Valid names refer to the variants of the Norton-Beer functions: aNB_11, aNB_12, ..., aNB_20
SpectrometerDetectorInterferogram outSdi [OUTPUT, MANDATORY, default=no default value]
Resultant apodized SDI


History

- 2007-04-03 - TRF: Re-organized version of RegSampledApodization.java
- 2007-05-31 - AH: Class placed in independent package. JParameters updated.
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-10-02 - ZW: Added super() in constructor
- 2007-10-02 - ZW: Re-organized variables in execute().
- 2007-11-13 - TRF: Fixed creator metadata in output sdi
- 2008-04-04 - PK: Changed getDoubleSidedPortion to have new symmetry (extra point on RHS instead of LHS).
- 2008-04-08 - PK: Added apodType parameters to "prePhaseCorr" and "postPhaseCorr", deprecated "pre" and "post". Added checks to input.
- 2008-04-17 - PK: Added MetaData check for sdi input
- 2008-04-22 - PK: Added all required checks for sdi input using VerifyDataset.checkInput. Current implementation throws exceptions when input is bad.
- 2008-07-15 - TRF: Merging the changes made on 08 April 2008 in the user tree wrt to the apodType keyword with the code in the main tree. This change satisfies SPIRE SCR-0722.
- 2008-07-15 - TRF: Removed references to VerifyDataset as it sets up an unapproved dependency on the ft.util package.
- 2008-07-15 - TRF: Removed setCreator() step from MetaData modification to conform with Analysis in SPIRE SPR-0754.
- 2008-08-21 - TRF: New class named ApodizeIfgmTask to replace the misnamed ApodizeIfgms class (SPIRE SPR-0819)
- 2008-08-23 - TRF: Task no longer recognizes the apodType vlues "pre" and "post" [SPIRE SCR-0722]
- 2008-10-07 - TRF: Added descriptions to TaskParameters [SPIRE SPR-0902]. Made jparameter descriptions more descriptive. Added a reference to the SPIRE Pipeline Description Document.
- 2008-10-10 - TRF: Fixed bug in the manner in which setProgress() was being updated [SPIRE SPR-0935]
- 2008-11-18 - TRF: Task is now interruptible [SPIRE SPR-1043]

- 2008-11-21 - TRF: Suppressed unchecked, serial, javadoc warnings
- 2008-11-25 - TRF: Fixed getDoubleSidedPortion() to fix the shift in DS apodization [SPIRE SPR-1036]
- 2008-11-25 - TRF: Changed jparameter entries to suppress warnings [HCSS SPR-3232]
- 2008-11-25 - TRF: Added enums for apodNames and apodTypes [SPIRE SPR-1045]
- 2008-11-25 - TRF: Added checks for valid apodNames and apodTypes. Invalid selections now throw SignatureExceptions [SPIRE SPR-1045]
- 2008-11-25 - TRF: Added TaskParameter modifiers for apodName and apodType for nicer HIPE usage [SPIRE SPR-1045]
- 2008-11-26 - TRF: Fixed a bug for the default apodFunctionName case [SPIRE SPR-1045]
- 2009-01-29 - TRF: [SPIRE SPR-1165] Added serialVersionUID = 1L, removed SuppressWarnings, fixed unchecked warnings
- 2009-01-29 - TRF: [SPIRE SPR-1098] Added throws InterruptedException to javadoc tag to execute()
- 2009-02-04 - TRF: [SPIRE SPR-1204] keySet().iterator() -> getScanNames()
- 2009-05-05 - TRF: Task now allows for targeted logging.
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-06-10 - TRF: [SPIRE SCR-1454] Use propagateMeta() from SpireDataProduct
- 2009-06-10 - TRF: [SPIRE SPR-1511] Correctly set creator metadata in output sdi.
- 2009-08-04 - TRF: [SPIRE-1765] Import Apodize from herschel.spire.ia.pipeline.spec.apodize.util
- 2009-08-31 - YFR: Create apodize functions inside apodize package itself and use them instead of apodize function in ft package, removed redundant enum to keep adding new apodize function easy
- 2009-09-08 - TRF: [SPIRE-1656] Valid entries for TaskParameter apodType = "prePhaseCorr" | "postPhaseCorr"
- 2009-09-08 - TRF: [SPIRE-1656] TaskParameter apodType now Mandatory
- 2009-09-08 - TRF: [SPIRE-1656] Metadata apodType in output SDI now set to the value of the apodType TaskParameter
- 2009-09-08 - TRF: [SPIRE-1656] Output sdi is always a new product.
- 2009-09-08 - TRF: [SPIRE-1656] TaskParameter apodFunctionName -> apodName
- 2009-09-08 - TRF: [SPIRE-1656] Task now checks the phaseCorrectionApplied metadata in its input sdi and issues a WARNING if its values and the chosen apodType do not agree.
- 2009-09-11 - YFR: [SPIRE-1916] Added sorting to available apodize function names so that they display in order inside hipe
- 2010-01-25 - YFR: [SPIRE-2349] Removed deprecated method call in TaskParameter class, and overrided 'getCustomModifiers' method to set modifiers to 'apodName' and 'apodType' task parameters.
- 2010-01-25 - YFR: Removed usage of deprecated AbstractComposite() methods

- 2010-02-26 - TRF: [SPIRE-2275] Fixed Javadoc and URM issues.
- 2010-02-26 - TRF: [SPIRE-2407] Task now sets apodName metadata in output
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.7. associateSkyPosition

Full Name:	herschel.spire.ia.pipeline.common.pointing.AssociateSkyPositionTask
Alias:	associateSkyPosition
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.pointing import AssociateSkyPositionTask
Category:	task

Description

Task to add RA and Dec in photometer detector timelines

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
SpirePointingProduct spp [Input, Mandatory, default=No default value]
String type [Input, Optional, default="filt"]
PointedPhotTimeline result [Output, Optional, default=No default value]

API details

Properties


DetectorTimeline data [Input, Mandatory, default=No default value]
Input Photometer Detector Timeline
SpirePointingProduct spp [Input, Mandatory, default=No default value]
Spire Pointing Product
String type [Input, Optional, default="filt"]
Pointing type: filtered, gyropropagated or commanded
PointedPhotTimeline result [Output, Optional, default=No default value]
Result Pointed Photometer Timeline with the RA and Dec associated to each detector

History

- 19 Jun 2006: First version.
- 13 Nov 2006: Some channels are not pixels. Small refactoring.
- 15 Mar 2007: Use new product structure

- 16 Apr 2007: moved in main.
- 30 Aug 2007: PC, use of herschel.share.unit package, add default serialVersionUID
- 18 Sep 2007: Change from: herschel.spire.ia.pipeline.phot.scanmap to herschel.spire.ia.pipeline.common.pointing
- 29 Feb 2008: Gives a PointedPhotometerTimeline as output for the Photometer
- 21 Apr 2008: Add a Serial Version
- 28 Jan 2009: Remove the fileName metadata in output product
- 2009-05-10 - XX: Changes in the parameters for HIPE integration
- 2009-06-06 - PPa: Update documentation and parameter descriptions
- 2009-08-17 - PPa: Add sampleTime column in "ra" and "dec" tables [SPIRE-1086]. Refactoring and correct metadata propagation.
- 2009-09-09 - PPa: Fix type and description of output products for jggled observations [SPIRE-1927]
- 2009-09-13 - PPa: Add possibility to compute the sky position with gyropropagated or commanded pointing.
- 2010-02-13 - AP: Speed-up theTask by modifying the loops in execute() to use the new method SpirePointingProduct.getSkyPositions (String [], Double1d,String)
- 2010-03-31 - AP: SPIRE-2463. using spire.ia.util.InstMode to check if input product is from ScanMode.
- 2010-04-14 - AP: SPIRE-2482 Set Meta parameter level to 10

1.8. AverageInterferogramTask

Full Name:	herschel.spire.ia.pipeline.spec.average.AverageInterferogramTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.average import AverageInterferogramTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task that calculates the detector-by-detector and opd-by-opd median of the interferogram samples in a SPIRE SDI product.

Task Execution Steps:

For each detector in the input SDI product:

1. Calculate the average of the signal at each opd across all forward scans.
2. Calculate the average of the signal at each opd across all reverse scans.
3. Store the average signal for each scan direction into separate datasets in the output SDI.

An option may be selected whereby an outlier rejection algorithm can be applied before computing the average of the voltage at each OPD. Execution in this mode will mean that samples flagged as glitches or as unusable will not affect the final average.

Examples

Example 1: Nominal usage:

```
avgSdi = averageInterferogram(sdi=sdi)
```

Example 2: Do not filter glitches:

```
avgSdi = averageInterferogram(sdi=sdi, filter=Boolean.FALSE)
```

API Summary

Properties
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]
Boolean filter [INPUT, OPTIONAL, default=True]
SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=no default value]

API details

Properties

```
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]
```

Input Spectrometer Detector Interferogram Product

<code>Boolean filter [INPUT, OPTIONAL, default=True]</code>

Set this keyword to true to filter out glitches and unusable samples from the average computation.
--


<code>SpectrometerDetectorInterferogram outsdI [OUTPUT, MANDATORY, default=no default value]</code>

Average Spectrometer Detector Interferogram Product

History

- 2010-02-03 - TRF: [SPIRE-2369] Reject first level glitches
- 2010-02-03 - TRF: [SPIRE-2310] Misc fixes to URM information
- 2010-02-05 - TRF: Fixed progress counter
- 2010-05-18 - TRF: [SPIRE-2310] Misc fixes to URM information

1.9. AverageSpectraTask

Full Name:	herschel.spire.ia.pipeline.spec.average.AverageSpectraTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.average import AverageSpectraTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

AverageSpectraTask

Task Execution Steps:

For each detector in the input SDS product:

1. If "separateScanDirections" keyword set to False:
- 2.
3. If "separateScanDirections" keyword set to True:
- 4.

An option may be selected whereby an outlier rejection algorithm can be applied before computing the average and standard deviation of the flux for each spectral bin. Execution in this mode will mean that samples deemed to be outliers will therefore not affect either the final average or standard deviation values.

A spectral sample for a given detector, $L_{\{n-i\}}(wn_k)$, will be considered as an outlier if its value falls outside a range defined by the mean of the samples for that spectral bin plus or minus a number d times the standard deviation of the samples for that spectral bin. The quantity d is referred to as the "threshold factor" and its value will be determined dynamically based on the number of scans, n , in the input data product such that the module would consider valid data as an outlier 0.1% of the time.

By default, samples that fall outside 3.2905 standard deviations from the mean shall be considered as outliers.

See Appendix 4 of the Spire Spectrometer Pipeline Description document (SPIRE-BSS-DOC-002966) for further details.

Examples

Example 1: Nominal usage:

```
asds = averageSpectra(sds=ssds, bandEdge=obs.calibration.spec.bandEdge)
```

Example 2: Compute the average for forward/reverse scans separately:

```
asds = averageSpectra(sds=ssds, bandEdge=obs.calibration.spec.bandEdge,
    separateScanDirections=Boolean.TRUE)
```

Example 3: Include out-of-band spectral samples:

```
asds = averageSpectra(sds=ssds, bandEdge=obs.calibration.spec.bandEdge,
    INCLUDE_OOB=Boolean.TRUE)
```

API Summary

Properties
SpectrometerDetectorSpectrum sds [INPUT, MANDATORY, default=no default value]
SpecBandEdge bandEdge [INPUT, OPTIONAL, default=No default value]
Boolean separateScanDirections [INPUT, OPTIONAL, default=False]
String outlierType [INPUT, OPTIONAL, default=No default value]
Double thresholdFactor [INPUT, OPTIONAL, default=No default value]
Boolean INCLUDE_OOB [INPUT, OPTIONAL, default=False]
SpectrometerDetectorSpectrum outsd s [OUTPUT, MANDATORY, default=no default value]

API details

Properties


SpectrometerDetectorSpectrum sds [INPUT, MANDATORY, default=no default value]
Input Spectrometer Detector Spectrum Product
SpecBandEdge bandEdge [INPUT, OPTIONAL, default=No default value]
Optical passband as defined by the calibration product SCalSpecBandEdge
Boolean separateScanDirections [INPUT, OPTIONAL, default=False]
Set this keyword parameter to False to compute the average flux over all scans regardless of scan direction. The output SDS product in this scenario will contain a single spectral composite dataset. Set this keyword parameter to True to separately compute the average flux over all forward scans and all reverse scans in the input SDS product. The output SDS product in this scenario will contain two spectral composite datasets, one contains the average flux computed from the forward scans, the other containing the average flux computed from the reverse scans.
String outlierType [INPUT, OPTIONAL, default=No default value]
This keyword parameter controls which algorithm is to be used to identify outliers. If not set, no outlier rejection shall be performed. Acceptable values are: STD or MAD or 3Sigma
Double thresholdFactor [INPUT, OPTIONAL, default=No default value]
Set this keyword parameter as threshold factor for outlier finding
Boolean INCLUDE_OOB [INPUT, OPTIONAL, default=False]
This keyword parameter controls whether to remove out of band data from final spectrum
SpectrometerDetectorSpectrum outsd s [OUTPUT, MANDATORY, default=no default value]
Average Spectrometer Detector Spectrum Product

History

- 2007-11-06 - YH: Made `calculateAverageAndStdDev()`, `checkWaveNumbers()` and `getFluxesColumnAndWaveNumbersColumn()` to static methods.
- 2007-11-26 - YH: Changed `getFluxesColumnAndWaveNumbersColumn()` to `getFluxesColumn()` and `getWaveNumbersColumn()`.
- 2008-04-18 - PK: Added `MetaData`, `mask`, and real flux checks for `sds` input.
- 2008-05-22 - ES: Added `badInput()` method to log occurrences of bad input. Removed all `SignatureExceptions` and replaced them with a call to `badInput()`. If exceptions are desired in the future, all that is required is a simple change to `badInput`.
- 2008-07-02 - TRF: Added `serialVersionUID` to comply with SPR-0752
- 2008-07-02 - TRF: Added `VERSION`, `TASK_VERSION` attributes.
- 2008-07-02 - TRF: Modified `updateMetaData()` so that `creator` is set properly (SPR-0754)
- 2008-07-03 - TRF: Commented out the previous change while waiting for guidance from the Product Definition Working Group.
- 2008-07-03 - TRF: Modified behaviour so that only if spectra from the `_same_` pixel are of different lengths will an exception be thrown. (SPR-0755)
- 2008-07-03 - TRF: Code refactoring: deprecated `checkWaveNumbers()`, `getFluxesColumn()`
- 2008-07-16 - TRF: Changed logging level of `badInput()` to fine until a policy is developed for how to handle erroneous input.
- 2008-07-16 - TRF: Changed `obsMode` to `instMode` in `checkMetaData()`.
- 2008-07-16 - TRF: Deprecated `getWaveNumbersColumn()` as it is only called by `checkWaveNumbers()`, which has itself been deprecated.
- 2008-07-31 - ES: Changed accepted input flux types. `AverageSpectra` now accepts `Complex`, `Double`, or `Float` values for the flux. Changes were made to `calculateAverageAndStdDev()`, `RemoveOutliers()`. The method `checkRealFlux()` has been deprecated, and the method `checkFlux()` has been added, where `CheckFlux()` checks that flux is either `Double`, `Float`, or `Complex`.
- 2008-08-01 - ES: Fixed bug in `checkWavenumbers()` where `SpectrumId.getSize()` was being used to check wavenumber data length, when this method will always return 1. Replaced this call with `SpectrumId.getWave().length()`, as well as forced an exception to be thrown if bad wavenumber data is discovered, rather than simply logging it, as if no exception is thrown, `AverageSpectra` will crash at a later point.
- 2008-08-13 - ES: Removed unnecessary check in `checkMetaData()`
- 2008-08-21 - TRF: New class named `AverageSpectraTask` to replace the mis-named `AverageSpectra` class (SPIRE SPR-0820)
- 2008-08-23 - TRF: Removed deprecated methods: `checkRealFlux()`, `checkArrayHasZeroAndConstantDelta()`, `checkEquals()`, and `calculateAvgAndStdDev()`
- 2008-11-18 - TRF: Improved URM documentation [SPIRE SPR-0900]
- 2008-11-18 - TRF: Deprecated unused/redundant Task attributes
- 2008-11-28 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2008-12-02 - TRF: Added `checkInterrupted()` to `execute()` [SPIRE SPR-1053]

- 2008-12-02 - TRF: Fixed unchecked warnings
- 2008-12-02 - TRF: Complex1d flux no longer supported (Double1d or Float1d only)
- 2009-01-12 - YFR: [SPIRE SCR-1104] Modified execute method to add the new functionality for dual direction averaging
- 2009-01-12 - YFR: [SPIRE SPR-1104] Changed dir parameter's type from String to Boolean
- 2009-01-12 - YFR: [SPIRE SPR-1067] Added rejectOutlier method to reject outliers, also added static local variables: STD_ALGORITHM, MAD_ALGORITHM, DEFAULT_STD_THRESHOLD, DEFAULT_MAD_THRESHOLD, deglitchTypes
- 2009-03-13 - TRF: [SPIRE SPR-1293] Changed == conditional to .equals() in checkWavenumbers()
- 2009-03-17 - TRF: [SPIRE SPR-1245] Removed private updateMeta() and replaced with sds.propagateMeta(sds)
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-04-01 - YFR: Changed the logger message to show the correct scan the task is performing on
- 2009-04-01 - YFR: [SPIRE SPR-1336] Added code to propagate ra and dec values from input SpireSpectrum1ds to averaged SpireSpectrum1d
- 2009-04-29 - YFR: Added threshold checking when it is not null
- 2009-04-30 - YFR: Added codes to set mean and std to Double.NaN for dead channels
- 2009-05-06 - YFR: Changed badInput method to make 100% coverage possible
- 2009-06-10 - TRF: [SPIRE SCR-1454] Use propagateMeta() from SpireDataProduct
- 2009-06-10 - TRF: [SPIRE SPR-0754] Correctly set creator metadata in output sds.
- 2009-06-12 - TRF: [SPIRE SPR-1507] Cast sds as a Product in propagateMeta()
- 2009-07-20 - YFR: [SPIRE-1688] Removed deprecated methods, replaced with equivalent methods
- 2009-07-31 - TRF: [SPIRE-1497] Change default MAD threshold. Updated documentation.
- 2009-08-27 - TRF: [SPIRE-1661] "dir" -> "separateScanDirections"
- 2009-09-04 - TRF: [SPIRE-1913] Added a check for the case where outlier rejection rejects all samples at a given WN
- 2009-09-16 - DMT: [SPIRE-1366] Added Task parameter "bandEdge" in order to get optical passband for detectors to calculate QC metadata
- 2009-09-23 - DMT: [SPIRE-1366] Added metadata functionality (ratioSpectralOutliers and ratioSpectralDifference)
- 2009-02-03 - TRF: [SPIRE-2310] Misc fixes to URM information
- 2010-02-04 - TRF: [SPIRE-2003] All types of empty sds throw BadDataExceptions with meaningful error msgs.
- 2010-02-19 - YFR: [SPIRE-2155] Removed out of band data from final spectrum and added 'INCLUDE_OOB'


1.10. BandEdge

Full Name:	herschel.spire.ia.dataset.BandEdge
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BandEdge
Category:	Calibration product

History

- 2007-09-07 - TRF: First version (TRF)
- 2008-02-25 - PP: Use new makeTable method (PP)
- 2009-06-03 - PPa: Include termistors, resistors and dark [SPIRE-SPR-1107]
- 2009-06-04 - PPa: Remove N channels.
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.11. BaselineCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.baseline.BaselineCorrectionTask
Alias:	BaselineCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.baseline import BaselineCorrectionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to remove from a set of interferograms the offset and low frequency drift.

Task Execution Steps (from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

Task Execution Steps:

1. For each scan:
 - a. Fit low-order polynomial or Compute low frequency ifgm by way of FT;
 - b. Subtract polynomial from interferogram

Examples

Example 1: Remove low frequency interferogram baseline (frequencies less than 4 cm-1):

```
sdi=baselineCorrection(sdi=sdi, type= "fourier", threshold = 4)
```

Example 2: Remove low-order polynomial fit to interferogram baseline (order = 4):

```
sdi=baselineCorrection(sdi=sdi, type= "polynomial", degree = 4)
```

API Summary

Jython Syntax

Removes drift from interferograms.

Properties

[SpectrometerDetectorInterferogram](#) **sdi** [INPUT, MANDATORY, default=no default value]

Integer **degree** [INPUT, OPTIONAL, default=4]

Double **threshold** [INPUT, OPTIONAL, default=4.0]

String **type** [INPUT, OPTIONAL, default="polynomial"]

[SpectrometerDetectorInterferogram](#) **outsdi** [OUTPUT, MANDATORY, default=no default value]

[SpectrometerDetectorInterferogram](#) **outbaselinesdi** [IO, Optional, default=null]

API details

Properties


<code>SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]</code>
The input SDI product to be corrected
<code>Integer degree [INPUT, OPTIONAL, default=4]</code>
Degree of polynomial to fit to ifgm baseline
<code>Double threshold [INPUT, OPTIONAL, default=4.0]</code>
Threshold value for use in FT baseline correction
<code>String type [INPUT, OPTIONAL, default="polynomial"]</code>
Baseline correction algorithm to use. 'polynomial' 'fourier'
<code>SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=no default value]</code>
The corrected SDI.
<code>SpectrometerDetectorInterferogram outbaselinesdi [IO, Optional, default=null]</code>
Secondary SDI output containing the ifgm baselines removed from the input SDI.

History

- 2007-05-31 - AH: Class placed in independent package. JParameters updated.
- 2007-07-23 - AH: Added FT baseline algorithm
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-09-27 - TRF: Removed conditional that allows for a choice between DoubleId and FloatId in the non-mask columns of SpireInterferogramId.
- 2007-10-02 - ZW: Added super(); in constructor
- 2008-01-22 - PK: Refactored baseline correction into baseline calculation and baseline subtraction methods.
- 2008-01-22 - PK: Added outbaselinesdi parameter and changed default type to "polynomial" and default degree to 4.
- 2008-02-13 - PK: Added channeltable parameter and added logic for handling bad data.
- 2008-04-17 - PK: Added MetaData check for sdi input.
- 2008-05-05 - TRF: Removed imports to deprecated FFT_PACK classes in herschel.spire.ia.pipeline.spec.ft.util
- 2008-05-05 - TRF: Removed imports to plot and herschel.spire.ia.pipeline.spec.ft.gui
- 2008-07-15 - TRF: Removed references to VerifyDataset and ChannelDataFlags as they constituted unapproved dependencies on the ft.util package.

- 2008-07-15 - TRF: Removed setCreator() step from MetaData modification to conform with Analysis in SPIRE SPR-0754.
- 2008-07-17 - TRF: Fixed some javadoc.
- 2008-07-29 - TRF: Added a progress logging message (INFO) to execute() message.
- 2008-08-21 - TRF: [SPIRE SPR-0821] New class named BaselineCorrectionTask to replace the misnamed BaselineCorrection class
- 2008-08-25 - TRF: Fixed bug in polynomialBaselineCorrection()
- 2008-10-07 - TRF: [SPIRE SPR-0903] Added descriptions to TaskParameters. Made jparameter descriptions more descriptive. Added a reference to the SPIRE Pipeline Description Document.
- 2008-10-09 - TRF: [SPIRE SPR-0933] Fixed bug in the manner in which setProgress() was being updated
- 2008-12-02 - TRF: [HCSS SPR-3232] Fixed some jtag urm documentation
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2009-01-29 - TRF: [SPIRE SPR-1165] Added serialVersionUID = 1L, removed SuppressWarnings
- 2009-01-29 - TRF: [SPIRE SPR-1195] Added checkInterrupted() to execute()
- 2009-01-29 - TRF: Removed unused 'channelTable' TaskParameter
- 2009-02-04 - TRF: [SPIRE SPR-1204] keySet().iterator() -> getScanNames()
- 2009-05-05 - TRF: Task now allows for targeted logging.
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-05-11 - YFR: [SPIRE SPR-1065] Reject truncated points, warnings issued if a interferogram is removed. BadDataException if resultant SDI is empty
- 2009-06-11 - YFR: [SPIRE SPR-1511] Added code to handle "creator" metadata
- 2009-11-27 - DMT: [SPIRE-2205] Added quality control metadata for the task: wrongBaseliningSSW, wrongBaselineSLW
- 2010-01-15 - TRF: [SPIRE-2276] Misc fixes to URM information
- 2010-01-15 - TRF: Corrected some misapplications of == (should be .equals())
- 2010-05-18 - TRF: [SPIRE-2481] Task to ignore non-nominal detectors when setting QC metadata
- 2010-05-18 - TRF: [SPIRE-2276] Fixed URM and Javadoc


1.12. BathTempTimeline

Full Name:	herschel.spire.ia.dataset.BathTempTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BathTempTimeline

History

- 2007-05-30 - RG: from SinkTemperatureTimeline in proto herschel.spire.ia.modules.detresponse
- 2007-05-30 - RG: add a constructor with a detector timeline
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-04-06 - PPa: make _descNames private and transient


1.13. BeamProf

Full Name:	herschel.spire.ia.dataset.BeamProf
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BeamProf
Category:	Calibration product

History

- 2010-01-20 - SG: First version


1.14. BolometerPhysics

Full Name:	herschel.spire.ia.pipeline.common.detresponse.BolometerPhysics
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.detresponse import BolometerPhysics
Category:	Calibration product

History

- 18 Aug 2006: First version (RG)
- 28 March 2007 : public
- Fri 7 Sep 2007: vbolo amplitude and not rms, a lot of division and multiplication by 2.
- March 2009 the method computeVoltage has now a precision of 0.1 microVolt PP
- April, 29th 2009 the method computeVoltage has now a precision of nanoVolts
- October, 23rd 2009 merge BolometerPhysics2 R Gastaud

1.15. BolometerResponseCorrectionTask

Full Name:	herschel.spire.ia.pipeline.phot.bolorespcurr.BolometerResponseCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.bolorespcurr import BolometerResponseCorrectionTask
Category:	herschel/spire/ia/pipeline/phot/bolorespcurr

Description

This task takes a Photometer Detector Timeline (PDT) product as input and removes the Bolometer transient response.

This task is part of the photometer pipeline, between Level 0.5 and Level 1.

API Summary

Properties
DetectorTimeline PDT_IN [Input, Mandatory, default=No Default value]
PhotChanTimeConst PhotChanTimeConst [Input, Optional, default=No default value]
DetectorTimeline PDT_OUT [Output, Mandatory, default=No Default value]

Limitations

None

API details

Properties

DetectorTimeline PDT_IN [Input, Mandatory, default=No Default value]
PDT: Photometer timelines to be corrected
PhotChanTimeConst PhotChanTimeConst [Input, Optional, default=No default value]
Bolometer time constants calibration table
DetectorTimeline PDT_OUT [Output, Mandatory, default=No Default value]
PDT: Photometer timelines corrected for Low Pass Filter delay"


History

- 18 September 08.(AP) First attempt, by translating working Jython code.
- 19 Dec 08 (AP) Task removes "fileName" metadata from the output Product (Re: SPR-1077)

- 13 Jan 09 (AP) Fixed: Task was not propagating units of the columns in input table.
- 13 Jan 09 (AP) Added Message output ("Task complete")
- 13 Jan 09 (AP) Added default serialVersionUID
- 16 Jan 09 (AP) [rtag 0_8]RE-APPLIED CHANGE FROM D_SPIRE_IA_PIPELINE_PHOT_BOLORESPCORR_0_2 (V.1.2). MISSING FROM Versions 1.3 to 1.5: { 04 November 08. Replaced all instances (4) of getVoltageTable with getSignalTable. In response to SPIRE-0959 "The BolometerResponseCorrectionTask looks at the "voltage" dataset while it should look for the "signal" dataset because it works after the flux conversion." }
- 06 Mar 09 (AP) Modified makeTransferFunction: Changed 1./L to 1./(L-1) for even_length arrays
- 06 Mar 09 (AP) Removed second instance import herschel.ia.task.*;
- 06 Mar 09 (AP) Removed unused 'TimeArray_Sec = TimeArray.subtract(TimeArray.get(0));'
- 06 Mar 09 (AP) [rtag 0_9]MODIFIED to accept Optional PhotChanTimeConst
- 08 Mar 09 (AP) changed "@jparameter ChanTimeConst" to "@jparameter PhotChanTimeConst"
- 08 Mar 09 (AP) [rtag 1_0]Added Log message when a channel in the PhotChanTimeConst has NaN values
- 19 Mar 09 (AP) [rtag 1_1] Removed Bug: changed w.multiply(to w.copy().multiply(in makeTransferFunction
- 14 Apr 09 (AP) Added progress output on console
- 14 Apr 09 (AP) [rtag 1_2] Added PDT.fits file to test dir and changed TESTPATH pointer accordingly (SCR-1150)
- 14 Apr 09 (AP) [rtag 1_7]Uploading again to test Junit test
- 14 Apr 09 (AP) Fixed warning by Changing Arr=Arr.range(L-1); to Double1d Arr=Double1d.range(L-1); in method linspace
- 13 May 09 (AP) removed offset of 10⁻¹⁵ from the Transfer function
- 13 May 09 (AP) added an offset of +10000000 to every timelin in removeFilterFromPDT to avoid negative values in timelines and subsequent errors due to Fourier reconstruction
- 13 May 09 (AP) added Severe warning if timelines with negative values are used in deconvolution
- 09 Sep 09 (AP) No correction applied if fast bolo time const is NaN
- 09 Sep 09 (AP) Undo previous change (No correction applied if fast bolo time const is NaN)
- 09 Sep 09 (AP) [SPIRE-2296] FourierOffset reduced to 10000 to avoid artifacts on Thermistor timelines
- 01 Jan 10 (AP) SPIRE-2086 Task makes calculations in Double1d and at the end it converts to Float1d/Double1d depending on input
- 13 Feb 10 (AP) SPIRE-2267 FIXED Task has no class description.- makeTransferFunction has no description for method, parameters or return value
- 19 Feb 10 (AP) SPIRE-2086 remove "add double number to array after converting it to Float1d", since it converts it back to Double1d
- 22 Feb 10 (AP) SPIRE-2388 Process only bolometers with Bolometer.isSeeing()

- 23 Feb 10 (AP) SPIRE-2408 Added setProgress() method
- 10 May 10 (AP) SPIRE-2267 Added initials to history tags.
- 10 May 10 (AP)SPIRE-2267 changed jcategory tag to include entire path
- 10 May 10 (AP)SPIRE-2267 changed jlimitation from TBD to None
- 10 May 10 (AP)SPIRE-2267 changed jhelp tag to mention the location of the task in the pipeline.


1.16. BolPar

Full Name:	herschel.spire.ia.dataset.BolPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BolPar
Category:	Calibration product

History

- 2006-10-09 - PP: First version (PP)
- 2006-10-13 - PP: split Load resistance column into two.
- 2006-10-16 - PP: tempT0 now metadata parameter for each array.
- 2006-10-26 - PP: fix some descriptions
- 2008-02-25 - PP: use new makeTable method
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2008-12-01 - TRF: Fixed some more jtag urm documentation [HCSS SPR-3232]
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2009-11-18 - PPa: Set the units of condG0 [SPR-2113]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.
- 2010-04-21 - PPa: Fix the see link [SPIRE-2255]


1.17. BsmAnglesTimeline

Full Name:	herschel.spire.ia.dataset.BsmAnglesTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BsmAnglesTimeline

History

- 2006-03-06 - RG: standardization of Yangle and Zangle uppercase only for the first letter
- 2006-05-11 - PP: code cleanup
- 2007-05-11 - RG: add quality metrics numberOutPositions
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-04-21 - PPa: Fix getZangleError() method.

1.18. BsmFlagsExtract

Full Name:	herschel.spire.ia.pipeline.common.bsm.BsmFlagsExtract
Alias:	BsmFlagsExtract
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.bsm import BsmFlagsExtract
Category:	algorithm

Description


This is a the version using the current of the chopper.

It is called by the BsmFlagsExtractor. It fills the fields of CJT.

History

- Aug 2005: First version
- 9 Sep 2005: code rewritten using chopper.current (RG)
- 8 Dec 2005: code rewritten using chopper sensor (long1d) and new calibration
- 10 Feb 2006 : R Gastaud rewrite
- 14 Feb 2006 : R Gastaud quiet
- 14 Feb 2006 : R Gastaud remove hard coded parameter (noise and aotmode) see bsmOperationTable
- 15 March 2006 : R Gastaud created from MirrorMotion
- 06 March 2007 (RG) standardization of Yangle uppercase only for the first letter
- 23 July 2009 (RG) comment the
- 24 August 2009 RG simplified version
- 2009-09-10 - RG: use the new cjt with chopBeamId instead of chopFlag, fill the new fields:
- comChopSens comJiggSens and ChopCyclePositionUncertainty and chopUncertainty
- 21 october 2009 RG spr 2091 check for chopper which never stabilise.
- 24 November 2009 RG spr 2198 replace system.outprintln by logger


1.19. BsmOps

Full Name:	herschel.spire.ia.dataset.BsmOps
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BsmOps
Category:	Calibration Product

History

- 2006-09-19 - PP: Adapted from Rene's code see also PhotBsmOps, SpecBsmObs, PhotObsModes, SpecObsModes
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-03-16 - SG: SCR-0735: obsMode -> instMode
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.
- 2010-04-06 - PPa: Fix javadocs [SPIRE-2255]. Remove computeJiggId(long,long,String) [SPIRE-2255]. Fix computeBeam method [SPIRE-2479]
- 2010-04-21 - PPa: Removed bad jhelp text [SPIRE-2255]


1.20. BsmPos

Full Name:	herschel.spire.ia.dataset.BsmPos
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BsmPos

History

- 2006-03-06 - RG: change degrees in seconds_arc
- 2006-03-06 - RG: standardization of Yangle and Zangle uppercase only for the first letter
- 2006-09-04 - PP: Rewriting with inheritance see also PhotBsmPos, SpecBsmPos
- 2006-10-26 - PP: fix setYangleError method
- 2008-02-25 - PP: remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.21. BuildingBlockContext

Full Name:	herschel.spire.ia.dataset.context.BuildingBlockContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import BuildingBlockContext

History

- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2009-04-28 - PPa: Improve javadocs [SPR-1098]

1.22. BuildingBlockProduct

Full Name:	herschel.spire.ia.dataset.BuildingBlockProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import BuildingBlockProduct


Example

Example 1: creation:
<pre>\$ x=BuildingBlockProduct(description="This is my SPIRE data product",modelName="Flight") \$ x.creator="Me, myself and I" \$ x.obsMode="POF1" \$ x.obsid=0x3000C251 \$ x.subsystem="photometer" \$ x.bbId=1 \$ x["Spectrum"]=aSpectrum</pre>

History

- 2006-01-30 - PP: First version
- 2006-02-06 - PP: Add more constructors.
- 2006-02-07 - PP: Add getBbExecCounter(), getBbFullType(), getBbFullType() and getBbInst().
- 2006-03-31 - PP: Fix BBID keyword description. Add getBbIdHex().
- 2006-04-19 - PP: Start changing bbId into bbid, as agreed with the HDPG.
- 2006-04-24 - PP: Fix computation of bbType.
- 2006-06-29 - PP: Change getBbExecCounter() into getBbCount(). Remove deprecated methods.
- 2006-08-10 - PP: Add serialVersionUID
- 2007-09-18 - PP: Fix copy constructor
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fix some javadoc and compiler warnings [SPIRE SPR-0684]
- 2009-07-30 - PPa: getBbCount() changed to return int instead of long [SPIRE-SCR-1472]

1.23. calcBathTemp

Full Name:	herschel.spire.ia.pipeline.common.engdata.CalcBathTempTask
Alias:	calcBathTemp
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import CalcBathTempTask
Category:	SPIRE/Pipeline/Common Task

Description

Bath temperature computation task.

This task is used to compute the temperature of the thermistors from the value of their resistances and theirs parameters.

API Summary

Properties
DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
ChanNum chanNum [INPUT, MANDATORY, default=No default value.]
BolPar bolPar [INPUT, MANDATORY, default=No default value.]
DetectorTimeline output [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
Detector timeline to be processed, PDT or SDT.
ChanNum chanNum [INPUT, MANDATORY, default=No default value.]
Channel Number Mapping Table.
BolPar bolPar [INPUT, MANDATORY, default=No default value.]
Bolometer parameters table.
DetectorTimeline output [OUTPUT, OPTIONAL, default=No default value.]
Output detector timeline.

History

- 2008-11-07 - RG: first version.
- 2009-01-07 - PPa: Refactoring and inclusion in engdata package.

- 2009-01-28 - PPa: Add serialVersionUID.
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-10-18 - PPa: Add PTC channels to the temperature table [SPIRE-1981].
- 2009-12-02 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.24. CalcBsmAnglesTask

Full Name:	herschel.spire.ia.pipeline.common.bsm.CalcBsmAnglesTask
Alias:	CalcBsmAnglesTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.bsm import CalcBsmAnglesTask
Category:	task

Description

This is the task which converts the BSMT from raw unit to degree

API Summary

Properties
Product input [Input, Mandatory, default=No default value]
BsmPos bsmPos [Input, Mandatory, default=No default value]
BsmAngleTimeline bat [Output, Optional, default=No default value]

API details

Properties


Product input [Input, Mandatory, default=No default value]
BSMT or NHKT: beam steering mirror contains the 2 angles timelines (chopper and jiggle in raw unit. NHKT also contains these two timelines.
BsmPos bsmPos [Input, Mandatory, default=No default value]
SPIRE calibration product: contains tabulated values of Y, Z angles versus chopper and jiggle angles.
BsmAngleTimeline bat [Output, Optional, default=No default value]
BAT: Y and Z angles timelines in degrees

History

- 14 Feb 2006 : R Gastaud
- 06 March 2007 (RG) standardization of Yangle and Zangle uppercase only for the first letter
- 07 March 2007 RG use edpProduct, remove preamble and postamble and the static fields
- 20 March 2007 RG add meta keyword subsystem
- 04 Sept 2007 RG add nhkt as input instead of bsmt
- RG 5 June 2009 remove setResetOnExecute, setHistory(true);
- add package.html, add creator

- RG 25 Nov 2009 add bsmMovedDuringScanFlag spr SPIRE-2081

1.25. calcBsmFlags

Full Name:	herschel.spire.ia.pipeline.common.bsm.CalcBsmFlagsTask
Alias:	calcBsmFlags
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.bsm import CalcBsmFlagsTask
Category:	task

Description

CalcBsmFlagsTask

This task is the second task of the bsm tasks. It computes the times when the chopper mirror is at the rest position. This divides the time line in half chopper cycles. In each half chopper cycle the chopper should have a constant value, or up or down, and the jiggle should have a constant value, as given by the bsmOps calibration. As the chopper sensor and the jiggle sensor evolve toward their commanded values during the half chopper cycle, I choose the latest stabilised values of chopper and jiggle to compare to bsmOps. The chopper is stabilised when the speed (difference between 2 consecutive values) is lower than a threshold.

API Summary

Properties
EdpProduct bsmt [Input, Mandatory, default=No default value]
BsmOps bsmOps [Input, Mandatory, default=No default value]
Double delayTime [Input, Optional, default=0.090 seconds]
ChopJiggTimeline cjt [Output, Optional, default=No default value]

API details


Properties

EdpProduct bsmt [Input, Mandatory, default=No default value]
BSMT: Beam Steering Mirror timeline to be converted
BsmOps bsmOps [Input, Mandatory, default=No default value]
SPIRE calibration product: contains the chopper and jiggle angles of the nodes of the jiggle maps
Double delayTime [Input, Optional, default=0.090 seconds]
It is a time expressed in second. This is only used to compute the QUALITY factor. After the beginning motion of the BSM, wait for Delay Time before computing the distance to the commanded value.
ChopJiggTimeline cjt [Output, Optional, default=No default value]
CJT SPIRE Data Product: the start and end time and the jiggle map index of each node.

History

- Aug 2005: First version
- 1 Sep 2005: change name into MirrorMotionCalculator; code cleanup (PP)
- 10 Feb 2006 : R Gastaud
- 14 Feb 2006 : R Gastaud quiet
- 14 Feb 2006 : R Gastaud remove hard coded parameter (noise and aotmode) see bsmOperationTable
- 13 March 2006: change name into BsmFlagsExtractor
- 16 March 2007 RG add meta keyword subsystem
- 20 March 2007 RG remove underscore from cjt name cosmetic
- 24 august 2009 RG remove bsmpos
- 21 october 2009 RG spr 2091 check for chopper which never stabilise
- 24 November 2009 RG spr 2198 replace system.outprintln by logger

1.26. calcJfetVolt

Full Name:	herschel.spire.ia.pipeline.common.engdata.CalcJfetVoltTask
Alias:	calcJfetVolt
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import CalcJfetVoltTask
Category:	SPIRE/Pipeline/Common Task

Description

Task to convert raw detector values into JFET voltages.

This task is used by the Engineering Conversion to convert raw ADC values of detector signals into voltages.

The task computes the voltage measured by JFET using the equation:

$$V_JFET=[5/Gtot(w)]*[ADU-2^{14}+52428.8*OFF]/(2^{16}-1)$$

defined in DCU Design document p. 77.

API Summary

Properties
DetectorTimeline rawData [INPUT, MANDATORY, default=No default value.]
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value.]
Double biasFreq [INPUT, OPTIONAL, default=No default value.]
OffsetHist offsetHist [INPUT, MANDATORY, default=No default value.]
ChanGain chanGain [INPUT, MANDATORY, default=No default value.]
DeterctorTimeline converted [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties

DetectorTimeline rawData [INPUT, MANDATORY, default=No default value.]
Input Detector Timeline to be converted.
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value.]
Nominal housekeeping timeline.
Double biasFreq [INPUT, OPTIONAL, default=No default value.]
Bias frequency to be used, in Hz. If provided, housekeeping will be ignored.

OffsetHist offsetHist [INPUT, MANDATORY, default=No default value.]
--

Detector offset history calibration product.
--

ChanGain chanGain [INPUT, MANDATORY, default=No default value.]
--

Channel Gain Table calibration product.


DeterctorTimeline converted [OUTPUT, OPTIONAL, default=No default value.]
--

Converted Detector Timeline.

History

- 2007-03-26 - PPa: First version.
- 2007-03-28 - PPa: Use also offsets. Some javadoc improvements.
- 2007-05-10 - PPa: Add "offsetApp" metadata keyword when detector offset are applied.
- 2007-05-29 - PPa: Change DetectorTimeline structure: store the signal in the "voltage" table.
- 2007-09-27 - PPa: Change name.
- 2008-04-30 - PPa: Use new methods of DetectorTimeline.
- 2008-07-01 - PPa: Change housekeeping parameter name. Change the biasFreq logic. Add comments. biasFreq keyword is now saved here.
- 2008-07-02 - PPa: Refactoring. Use getBiasFreq().
- 2008-07-17 - PPa: Change name in the constructor. Check for time table presence. Change voltage values to Float1d (SPR-0748).
- 2008-07-18 - PPa: public constructor.
- 2009-01-26 - PPa: Add parameter descriptions [SPR-0985] and serialVersionUID.
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-04-01 - PPa: Set creator accordingly to policy.
- 2009-05-08 - PPa: Implement new requirements [SCR-1249].
- 2009-05-12 - PPa: Fix BadDataException message [SCR-1249].
- 2009-05-25 - PPa: Change name of Offset History parameter [SCR-1424].
- 2009-06-08 - PPa: Cope with N channels if not present in calibration product.
- 2009-06-12 - PPa: Do not throw exception when the offset doesn't come from an other observation, but add a metadata [SPR-1494].
- 2009-12-02 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.27. calcOpticalPower

Full Name:	herschel.spire.ia.pipeline.common.detresponse.CalcOpticalPowerTask
Alias:	calcOpticalPower
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.detresponse import CalcOpticalPowerTask
Category:	SPIRE

Description

Task compute Optical Power falling on the bolometers

from the value of the temperature of each sub array PMW, PSW, PLM, SSW, SLW

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
ChanNum chanNum [Input, Mandatory, default=No default value]
BolPar bolPar [Input, Mandatory, default=No default value]
DetectorTimeline output [Output, Optional, default=No default value]

API details


Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
Detector timeline to be processed, PDT or SDT
ChanNum chanNum [Input, Mandatory, default=No default value]
Channel Number Mapping Table
BolPar bolPar [Input, Mandatory, default=No default value]
Bolometer parameters table
DetectorTimeline output [Output, Optional, default=No default value]
Output detector timeline

History

- 2009-10-23 - RG: first version

1.28. calcRmsVoltRes

Full Name:	herschel.spire.ia.pipeline.common.engdata.CalcRmsVoltResTask
Alias:	calcRmsVoltRes
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import CalcRmsVoltResTask
Category:	SPIRE/Pipeline/Common Task

Description

RMS bolometer voltage and resistance computation task.

Task to reconstruct the detector voltage of each bolometer from JFET voltage, correcting for harness cable parasitic capacity.

API Summary

Properties
DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
NhkTimeline nhkt [INPUT, MANDATORY, default=No default value.]
ChanGain chanGain [INPUT, MANDATORY, default=No default value.]
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
BolPar bolPar [INPUT, OPTIONAL, default=No default value.]
ChanNomRes chanNomRes [INPUT, OPTIONAL, default=No default value.]
Boolean appRcRoll [INPUT, OPTIONAL, default=true.]
DetectorTimeline output [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties

DetectorTimeline data [INPUT, MANDATORY, default=No default value.]
Detector timeline to be processed.
NhkTimeline nhkt [INPUT, MANDATORY, default=No default value.]
Housekeeping timeline.
ChanGain chanGain [INPUT, MANDATORY, default=No default value.]
Channel gain table.
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
Channel number mapping table.
BolPar bolPar [INPUT, OPTIONAL, default=No default value.]
Bolometer parameters table.

ChanNomRes <code>chanNomRes</code> [INPUT, OPTIONAL, default=No default value.]
--

Channel nominal resistances table.

Boolean <code>appRcRoll</code> [INPUT, OPTIONAL, default=true.]
--

Flag to switch on/off the RC roll correction.

DetectorTimeline <code>output</code> [OUTPUT, OPTIONAL, default=No default value.]


Output detector timeline.

History

- 2006-11-07 - RG: first version
- 2007-03-22 - RG: rewritten following Matt Griffin draft
- 2007-03-26 - RG: remove unneeded getter methods now part of calibration products
- 2007-03-29 - PPa: Moved in main. Remove de-phasing correction. Refactoring.
- 2007-05-10 - PPa: Add "rcRollApp" metadata keyword to the result
- 2007-05-29 - PPa: Change DetectorTimeline structure: store the signal in the "voltage" table, compute also the resistance
- 2007-07-27 - PPa: Migrate to new Unit API
- 2007-08-03 - PPa: Fix logic in harness gain computation loop
- 2007-08-06 - PPa: Read the capacity from the calibration file
- 2007-08-07 - PPa: Fix bias voltage definition.
- 2008-02-15 - PPa: Do not perform correction for PTC channels.
- 2008-04-30 - PPa: Use new methods of DetectorTimeline
- 2008-05-30 - PPa: Add computation of Phase shift (SPR-749)
- 2008-07-02 - PPa: Use chanNum. Change input name. Fix table descriptions.
- 2008-07-10 - PPa: Refactoring. Add bias amplitudes in metadata.
- 2008-07-17 - PPa: Change name in the constructor. Check for time table presence. Change voltage values to Float1d (SPR-0748)
- 2008-08-29 - PPa: Partial implementation of SCR-0834. Now the BolPar is optional.
- 2008-09-03 - PPa: Add "appRcRoll" parameter (still not used).
- 2008-11-01 - PPa: Full usage of "appRcRoll" parameter SCR-0834. Small refactoring.
- 2008-11-10 - PPa: Set PTC amplitude in metadata (SCR-0730)
- 2009-01-28 - PPa: Add parameter descriptions [SPR-0985]. Fix history format. Add missing jparameter. Add serialVersionUID.
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true)

- 2009-04-01 - PPa: Set creator accordingly to policy.
- 2009-06-08 - PPa: Cope with N channels if not present in calibration product.
- 2009-12-07 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.29. CalibrationOdPlugin

Full Name:	herschel.spire.ia.pipeline.pg.CalibrationOdPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import CalibrationOdPlugin

Description


Calibration Operational Day Plugin.

This class implements the Product Generator plugin for the generation of the SPIRE Calibration context of the OpDayContext. It is used by the SPG Operational Day processing stage. It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-10-16 - PPa: First version.
- 2007-10-18 - PPa: Use ProductSink.
- 2008-01-17 - PPa: Move to new OpDayPlugin interface
- 2008-02-11 - PPa: Use the NHKT created by the HousekeepOdPlugin
- 2008-06-27 - PPa: Use getCount() instead of size()
- 2008-06-30 - PPa: More logging messages. Re-throw IO and general security exceptions.
- 2008-11-13 - PPa: Create the ResetHist in any case (due to SPR-1006)
- 2008-07-22 - PPa: Set the OD number in the ResetHist and OffsetHist [SPR-1599]
- 2009-08-24 - SG: Subclass AbstractCalibrationPlugin
- 2010-01-18 - PPa: Add constructor javadoc [SPIRE-2266]
- 2010-02-04 - PPa: Refactoring
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]

1.30. CalibrationPlugin

Full Name:	herschel.spire.ia.pipeline.pg.CalibrationPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import CalibrationPlugin

Description

Calibration Plugin.


This class implements the Product Generator plugin for the generation of the SPIRE Calibration context. It is used by the SPG during the preprocessing stage and by the SpireProprocessThread class (i.e. the obsExporter application). It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-03-21 - PPa: First version.
- 2007-04-03 - PPa: Complete implementation.
- 2007-04-05 - PPa: Some fix, tested
- 2007-04-16 - PPa: Extract reset and offset history only for the given observation. Remove code duplication.
- 2007-04-25 - PPa: Add some more logging info.
- 2007-05-09 - PPa: EngConverion output is now a EdpContext. Top-level try-catch now catches only IOException and GeneralSecurityException.
- 2007-06-14 - SG: Use FineTime in cal API
- 2007-09-18 - PPa: RawDataExtraction is now in rawdata package
- 2007-09-20 - PPa: Refactoring + use the ObjectStore
- 2007-10-29 - PPa: Now looks if the histories are already in the storage.
- 2007-11-19 - PPa: Add PcalList.
- 2008-06-27 - PPa: Use getCount() instead of size()
- 2008-06-30 - PPa: Change logic: if the OpDayContext is corrupted, recreate the missing calibration products (SPR-0738).
- 2008-06-30 - PPa: More logging messages. TBD: Need some code cleening
- 2008-11-13 - PPa: Check if the ResetHist got from the loaded calibration context is good (due to SPR-1006)
- 2008-11-13 - PPa: Do not get the offset histories from calibration context (workaround for SPR-1006)
- 2008-07-22 - PPa: Set the OD number in the ResetHist and OffsetHist [SPR-1599]
- 2009-08-24 - SG: subclass AbstractCalibrationPlugin [SG]
- 2009-09-11 - SG: set defaults for biasMode & aperture [SCR-1882] [SG]

- 2009-10-27 - PPa: Do not set the aperture from the uplink [SPIRE-2102]
- 2009-12-18 - PPa: Look only for SPIRE OpDayContext [SPIRE-2289]
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-02-04 - PPa: Refactoring + allow possibility to run efficiently when not navigating.
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]
- 2010-04-16 - PPa: Refactoring + fix retrieval of SpecOffsetHist from the OpDayContext [SPIRE-2450]


1.31. ChanGain

Full Name:	herschel.spire.ia.dataset.ChanGain
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanGain
Category:	Calibration product

History

- 2006-08-18 - PP: First version (PP)
- 2007-03-27 - PP: Now stores the LIA plus amplifier gain.
- 2007-12-06 - PP: Add units for metadata
- 2008-02-25 - PP: use new makeTable method
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-05-26 - PPa: change name of the frequency dependency parameter and allow for a specific PTC value [SPR-1389]
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.32. ChanMask

Full Name:	herschel.spire.ia.dataset.ChanMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanMask
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2006-10-09 - PP: Small refactoring.
- 2006-11-02 - PP: fix isDead and isNoisy methods
- 2007-11-14 - PP: Add isPixel column and methods.
- 2007-12-12 - PP: Add isConnected column and methods.
- 2008-02-25 - PP: Use new makeTable method
- 2008-03-28 - PP: Remove isPixel and isConnected (moved in ChanNum).
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-03-06 - PPa: Introduce isSlow column [SPIRE SCR-1266].
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.33. ChannelCalibTable

Full Name:	herschel.spire.ia.dataset.ChannelCalibTable
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChannelCalibTable
Category:	Calibration product

History

- 2006-08-08 - PP: First version
- 2007-03-16 - PP: Enable multidimensional data
- 2007-09-07 - TRF: Fixed some javadoc (TRF)
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: New makeTable method. Remove serialVersionUID.
- 2008-05-27 - PP: Use isSeeing() instead of isVisible()
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-06-03 - PPa: Add hasName(String) method [SCR-1222]. Add method to create product without disconnected channels [SPR-1376]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.


1.34. ChanNoise

Full Name:	herschel.spire.ia.dataset.ChanNoise
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanNoise
Category:	Calibration product

History

- 2008-01-28 - PP: First version
- 2008-03-13 - EP: added new metadata
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.35. ChanNomRes

Full Name:	herschel.spire.ia.dataset.ChanNomRes
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanNomRes
Category:	Calibration product

History

- 2008-06-23 - PP: First version (PP)
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.36. ChanNum

Full Name:	herschel.spire.ia.dataset.ChanNum
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanNum
Category:	Calibration product

History

- 2008-01-29 - PPa: First version (PP)
- 2008-02-25 - PPa: Use new makeTable method
- 2008-03-12 - EP: Add all set methods and correct doc strings for others
- 2008-05-19 - PPa: Add "isPtc" columns and missing methods. Update Javadoc.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-02-17 - PPa: Add adcChannel column [SPIRE SCR-1134]
- 2009-05-21 - PPa: Add jfetMembrane column [SPIRE SCR-1379]
- 2009-06-08 - PPa: Change jfetMembrane into StringId
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.37. ChanTimeConst

Full Name:	herschel.spire.ia.dataset.ChanTimeConst
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanTimeConst
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-09-21 - TRF: Fixed bug in setError()
- 2007-10-23 - ZW: Fixed bug in timeConst()
- 2008-02-25 - PP: Use new makeTable method
- 2008-10-17 - TRF: Added four new columns and associated accessor methods: slowTimeConst, slowTimeConstError, amplitude, amplitudeError. See SPIRE SPR-0913 for details.
- 2008-11-18 - TRF: Deprecated get/setAmplitudeUnit() and get/setAmplitudeErrorUnit() [SPIRE SCR-1007]. These methods will simply be removed and not replaced.
- 2008-11-27 - TRF: Removed deprecated methods [SPIRE SCR-1007]
- 2009-06-03 - PPa: Remove N channels [SPR-1376]. Add units at creation.
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.38. ChanTimeOff

Full Name:	herschel.spire.ia.dataset.ChanTimeOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChanTimeOff
Category:	Calibration product

History

- 2006-08-22 - PP: First version
- 2006-10-09 - PP: Small refactoring.
- 2008-02-25 - PP: Use new makeTable method
- 2008-10-03 - PP: workaround for java 1.6 bug (HCSS SPR-5135)
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.39. checkAdcLatchTrunc

Full Name:	herschel.spire.ia.pipeline.common.engdata.CheckAdcLatchTruncTask
Alias:	checkAdcLatchTrunc
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import CheckAdcLatchTruncTask
Category:	SPIRE/Pipeline/Common Task

Description

Check ADC Latch-up and Truncation task.

Task to flag detector signals where the an ADC latch-up happen and/or the detector values were out of range.

API Summary

Properties
DetectorTimeline rawData [INPUT, MANDATORY, default=No default value.]
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
DetectorTimeline checked [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


DetectorTimeline rawData [INPUT, MANDATORY, default=No default value.]
Input Detector Timeline to be checked.
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
Channel number mapping table.
DetectorTimeline checked [OUTPUT, OPTIONAL, default=No default value.]
Checked Detector Timeline.

History

- 2007-03-27 - PPa: Replaces AdcFlagger, implements also the check for out of range values.
- 2007-03-28 - PPa: Do not process offset detector timelines.
- 2007-07-27 - PPa: Migrate to new Unit API.
- 2007-10-03 - PPa: Change name. Add adcErr metadata. Use SpireMask.
- 2007-11-14 - PPa: truncation column changed to Float1d.

- 2008-02-25 - PPa: Use new ArrayEnum.
- 2008-04-30 - PPa: Use new methods of DetectorTimeline.
- 2008-07-21 - PPa: Set also TRUNCATED_UNCORR flag.
- 2008-07-22 - PPa: Refactoring.
- 2008-11-10 - PPa: Add "adcErr" metadata also in the case that no error was found (SPR-0978).
- 2008-11-23 - PPa: Fix number of channels per ADC for spectrometer data (SPR-1040). Add parameters descriptions (SPR-0985).
- 2009-01-28 - PPa: Fix history format. Add serialVersionUID.
- 2009-02-09 - PPa: Add truncation metadata [SPR-1215].
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-05-11 - PPa: Implement new requirements [SCR-1355].
- 2009-05-22 - PPa: Change name [SCR-1149].
- 2009-06-09 - PPa: Change name of QC metadata [SCR-1485]. Add masks in metadata.
- 2009-09-17 - PPa: Fix name of QC metadata ratioTruncated [SPIRE-1964].
- 2009-11-18 - PPa: Split ratioTruncated per array [SPIRE-2177].
- 2009-12-15 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.40. CheckClip

Full Name:	herschel.spire.ia.pipeline.spec.clip.CheckClip
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.clip import CheckClip


Description

unknown

History

- 2008-01-11 - DB: first java version
- 2009-01-23 - DB: corrected jhistory
- 2009-03-11 - DB: not taking into account as clipped glitched sample
- 2009-05-19 - DB: parametrization vector
- 2009-07-20 - DB: change name to ClippingCorrectionTask and TestClippingCorrectionTask cf SPIRE SPR 1653
- 2010-01-28 - DB: SPIRE-2311 update javadoc add class, method description


1.41. ChopJiggTimeline

Full Name:	herschel.spire.ia.dataset.ChopJiggTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ChopJiggTimeline

History

- 2006-03-06 - RG: standardization of Yangle and Zangle uppercase only for the first letter
- 2006-06-25 - PP: Code cleanup
- 2006-07-04 - PP: Code cleanup
- 2007-03-15 - RG: add quality metrics numberMissingPositions
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.

1.42. ClippingCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.clip.ClippingCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.clip import ClippingCorrectionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Class reconstructs the signal where it is clipped in SDT product

API Summary

Properties
DetectorTimeline input [Input, Mandatory, default=no default value]
Integer fitPoints [Input, Optional, default=Default value]
DetectorTimeline result [Output, MANDATORY, default=no default value]

API details

Properties


DetectorTimeline input [Input, Mandatory, default=no default value]
Input DetectorTimeline product.
Integer fitPoints [Input, Optional, default=Default value]
half of the number of points used for polynomial fitting.
DetectorTimeline result [Output, MANDATORY, default=no default value]
Output corrected DetectorTimeline product.

History

- 2008-01-11 - DB: first java version
- 2008-04-29 - DB: corection warning Trevor Fulton version build 814
- 2008-07-02 - DB: public -> private SpireMask trunk
- 2008-07-10 - DB: add description to task Parameters add mask TRUCATED_UNCORR flag
- 2008-11-20 - DB: ClippingTasks becomes interruptible
- 2008-11-21 - DB: implement progress bar
- 2009-01-23 - DB: corrected jhistory + protected->private + supress @SuppressWarnings("unchecked") + nbpoint < 9

- 2009-01-26 - DB: no correction if nbpoint < 9 + corrected baseline calculation on 1000 points or s.length()
- 2009-01-28 - DB: add description @jparameter tag
- 2009-03-10 - DB: add max and min value for number, warning message for not default parameter usage, explicit messages
- 2009-03-13 - DB: when clipped samples are glitched too, no reconstruction + initialisation TRUNCATED_UNCORRECTED for clipped samples + bug 2perform of method apply
- 2009-04-23 - DB: add Meta data for QC SCR 1361
- 2009-06-01 - DB: modified setCreator
- 2009-06-02 DB add setMeta for uncorrected clipped
- 2009-06-10 - DB: modified setCreator
- 2009-04-23 - DB: correction Meta data for QC SCR 1361 NUMBER of DETECTOR
- 2009-07-20 - DB: change name to ClippingCorrectionTask and number to fitPoints of SPIRE SPR 1653
- 2009-08-06 - DB: change script name to ClippingCorrection in constructor
- 2009-10-19 - DB: SPIRE-2063 perform input parameter check before making a copy of the input detector timeline
- 2009-10-19 - DB: SPIRE-2097 update mask/flag to product metadata for TRUNCATED_UNCORRECTED
- 2009-11-04 - DB: SPIRE-2131 change the degree of the polynomial fitting go back to 8
- 2010-01-28 - DB: SPIRE-2311 update javadoc add class, method description
- 2010-04-27 - DB: SPIRE-2477 ClippingCorrectionTask too verbose , update info message
- 2010-05-11 - DB: add "@jcategory"
- 2010-05-11 - DB: add ParameterValidator
- 2010-04-27 - DB: re-open SPIRE-2477 ClippingCorrectionTask too verbose , update info message , misunderstanding on message

1.43. ComboPlotLayoutDirectorImp

Full Name:	herschel.spire.ia.gui.ComboPlotLayoutDirectorImp
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import ComboPlotLayoutDirectorImp

Description

ComboPlotLayoutDirectorImp


This class is the director of the layout for the mosaic. When this class is set to direct a layout, its `addPlot()`, `getPlotBounds()` and `Layout()` are called for every plot to be added. After `comboplot.setLayoutDirector(this class)` is called this class should give the comboplot the grounds on which to layout the plots on the chart. no method here needs invoking, comboplot will invoke what it deems necessary. This class implements the `ComboPlotLayoutDirector` interface.

1. The `ComboPlotLayoutDirectorImp` is used with the `Mosaic` class in the following way:
 - a. Create an instance of the implementing class `ComboPlotLayoutDirectorImp`. *(Ex: `ComboPlotLayoutDirectorImp layoutDirector = new ComboPlotLayoutDirectorImp()`)
 - b. Create an instance of the `ComboPlot` class
(Ex: `ComboPlot _combo = new ComboPlot()`)
 - c. Set the `_combo` layout director to be the new `layoutDirector`
(Ex: `_combo.setLayoutDirector(layoutDirector)`).
 - d. The `ComboPlot` class will call all the methods its needs.
2. A single `Mosaic` with all the plots of the selected detectors displayed in the honeycomb arrangement.

History

- 2008-09-10 - KZ: Class created.
- 2008-09-11 - KZ: Implemented all methods.
- 2008-09-16 - KZ: Altered `getChartBounds` to allow for plotting plots at the optimum space.
- 2008-09-16 - KZ: Altered `calcContents` to check for detectors that are at positions such as `x.5` or `y.5`.
- 2008-12-01 - TRF: First Release in main branch
- 2008-12-02 - TRF: Cleaned up some javadoc and jtag warnings -- [SPIRE SPR-0678], [HCSS SPR-3232] Deleted commented code
- 2010-02-03 - TRF: [INFR-365] Migrated from `herschel.spire.ia.dataset.gui`

1.44. concurrentGlitchDeglitcher

Full Name:	herschel.spire.ia.pipeline.common.deglitch.ConcurrentGlitchDeglitcherTask
Alias:	concurrentGlitchDeglitcher
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.deglitch import ConcurrentGlitchDeglitcherTask

Description

Concurrent Glitch Deglitcher Task.

This task can be used to remove concurrent glitches from SPIRE Detector Timelines.

Example

Example 1: Run the task on a PDT

```
HIPE>> pdt=concurrentGlitchDeglitcher(pdt,chanNum=chanNum,kappa=2.0)
```

API Summary

Properties
DetectorTimeline dt [INPUT, MANDATORY, default=No default value]
ChanNum chanNum [INPUT, MANDATORY, default=No default value]
Float kappa [INPUT, OPTIONAL, default=2.5]
Integer size [INPUT, OPTIONAL, default=15]
Boolean correctGlitches [INPUT, OPTIONAL, default=true]
DetectorTimeline output [OUTPUT, OPTIONAL, default=No default value]

API details

Properties

DetectorTimeline dt [INPUT, MANDATORY, default=No default value]
Input Detector Timeline to be deglitched.
ChanNum chanNum [INPUT, MANDATORY, default=No default value]
Channel Number Mapping product.
Float kappa [INPUT, OPTIONAL, default=2.5]
Kappa value.
Integer size [INPUT, OPTIONAL, default=15]
Median filter half size.
Boolean correctGlitches [INPUT, OPTIONAL, default=true]
Flag to perform the reconstruction (true) or not (false).


<code>DetectorTimeline output [OUTPUT, OPTIONAL, default=No default value]</code>

Deglitched Detector Timeline.

History

- 2010-04-12 - PPa: First version.

1.45. CreateIfgmTask

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.CreateIfgmTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm import CreateIfgmTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task used to create SPIRE Spectrometer interferograms.

Task Execution Steps (from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

The process by which interferograms are created involves two steps, each of which is described below.

1. **Interpolation of the SMEC timeline.** This step converts the spectrometer mechanism timeline from one that is non-uniform in position to one that is uniform in position.

- a. **Establish an OPD position vector.** This step creates a common vector of OPD positions that will be the basis for the interferograms for all of the spectrometer detector channels and for all of the scans in the observation. This common position vector will contain samples that are uniformly-spaced in terms of position. A common set of OPD positions is desired as it will allow for easier scan-to-scan and detector-to-detector comparisons of the interferograms of the particular SPIRE spectrometer observation.

In order to maintain the independence of the spectrometer detector signal samples, the step size of the common OPD vector is chosen in such a way as to maintain the independence of the spectrometer detector signal samples. For an SDT sampling rate s [Hz] and a SMEC scanning speed v_{SMEC} [cm/s MPD], the position step size, ΔMPD in units of cm is given by: $\Delta\text{MPD} = v_{\text{SMEC}} / s$. This step is then converted such that it is in terms of OPD by the following relation $\Delta\text{OPD} = \text{FLOOR}[4 * \Delta\text{MPD}]$ where $\text{FLOOR}[]$ denotes that the step size is rounded down to the nearest integer in units of microns OPD and the factor of four is the nominal conversion between MPD and OPD for a Mach-Zehnder FTS. Using the nominal SPIRE spectrometer settings, this results in an OPD step size of 25 microns.

- b. **Map the common OPD position vector to a SMEC position vector.** This step maps the common OPD positions established in the preceding step to SMEC mechanical path difference positions. This step takes into account the position of ZPD and the step-size conversion factor. As such this step is performed separately for each detector channel. Since the step-size conversion factor could be different for each channel, this operation could have the effect of making the intrinsic resolution slightly different for each detector channel.
 - c. **Parse the measured SMEC timeline into discrete scans.** This step splits the actual SMEC timeline ($z(t_{\text{SMEC}})$) into a series of discrete timelines, each of which represents one scan. The delineation of the measured SMEC timeline is accomplished by comparing consecutive samples and finding those samples where the motion of mirror mechanism changed direction.
 - d. **Interpolate the measured SMEC timelines onto the mapped SMEC timelines.** The next step is to determine, on a channel-by-channel and scan-by-scan basis, the times when the spectrometer mechanism reached the mapped SMEC positions. Since, for each detector channel, there is a 1:1 relationship between the mapped SMEC positions and the regularly-spaced OPD positions, this step effectively determines the times when the SMEC reached the regularly-spaced OPD positions.
2. **Merge the spectrometer detector and the mapped SMEC timelines.** This step combines the signal samples from the timeline of a given spectrometer detector channel with the mapped SMEC timeline.

- a. **Interpolation of the spectrometer detector timelines.** The spectrometer detector signal samples are mapped onto the times corresponding to the SMEC positions by way of interpolation. Since there is a 1:1 relationship between the mapped SMEC positions and their sample times, this interpolation effectively maps the spectrometer detector signal sampled to the regularly-spaced SMEC positions.

Examples

Example 1: Create interferograms using a calibration context (Default usage).

```
sdi=createIfgm(sdt=sdt, smect=smect, nhkt=nhkt, spp=spp, cal=obs.calibration,
interpolType= -"spline")
```

Example 2: Override a given calibration product.

```
# Create a copy of the SpecSmecStepFactor (obliquity) calibration product
mySmecStepFactor = SpecSmecStepFactor(obs.calibration.spec.smecStepFactor)
# Override the entry for channel -"SLWB3" (was 4.0)
mySmecStepFactor.setStepFactorChannel("SLWB3", 3.9)
# Use the new calibration product
sdi=createIfgm(sdt=sdt, smect=smect, nhkt=nhkt, spp=spp, cal=obs.calibration,
smecStepFactor=mySmecStepFactor, interpolType= -"spline")
```

API Summary

Properties
DetectorTimeline sdt [INPUT, MANDATORY, default=no default value]
EdpProduct smect [INPUT, MANDATORY, default=no default value]
NhktTimeline nhkt [INPUT, MANDATORY, default=no default value]
SpirePointingProduct spp [INPUT, OPTIONAL, default=no default value]
SpecSmecZpd smecZpd [INPUT, OPTIONAL, default=0.785cm]
SpecSmecStepFactor unknown [INPUT, OPTIONAL, default=4.0]
SpecChanTimeOff chanTimeOff [INPUT, OPTIONAL, default=no default value]
String interpolType [INPUT, OPTIONAL, default="SPLINE"]
SpectrometerDetectorInterferogram result [OUTPUT, MANDATORY, default=no default value]

API details

Properties

DetectorTimeline sdt [INPUT, MANDATORY, default=no default value]
Spectrometer Detector timeline used to generate Interferogram.
EdpProduct smect [INPUT, MANDATORY, default=no default value]
Spetrometer Mechanism Timeline used to generate resultant Interferogram.
NhktTimeline nhkt [INPUT, MANDATORY, default=no default value]
Housekeeping product which will be used to generate the Interferogram.

SpirePointingProduct spp [INPUT, OPTIONAL, default=no default value]
Pointing Timeline that will be used to register the RA/DEC for all detectors for all interferograms created.
SpecSmecZpd smecZpd [INPUT, OPTIONAL, default=0.785cm]
SpecSmecZpd calibration product. This calibration product contains the position of ZPD in units of cm MPD for each Spectrometer Detector. If not provided, use 0.785cm for all Spectrometer Detectors.
SpecSmecStepFactor unknown [INPUT, OPTIONAL, default=4.0]
SpecSmecStepFactor calibration product. This calibration product contains the obliquity factor for each Spectrometer Detector. If not provided, use 4.0 for all Spectrometer Detectors.
SpecChanTimeOff chanTimeOff [INPUT, OPTIONAL, default=no default value]
SpecChanTimeOff calibration product that contains the time offsets for each detector between the frametime given in the SDT and the actual sample time.
String interpolType [INPUT, OPTIONAL, default="SPLINE"]
Type of interpolation to perform. Valid entries are 'SPLINE' or 'OVERSAMPLED' (Sinc interpolation via the FT). If not set, SPLINE will be used.
SpectrometerDetectorInterferogram result [OUTPUT, MANDATORY, default=no default value]
The resultant SDI product that will contain interferograms for all detectors for all scans of the building block.

History

- 2007-01-24 - TRF: method addFineValues() now checks to see if the coarse OE and fine OE columns in the SMECT contains converted values or not. The discriminating parameter is whether the value returned by getQuantity() is "null" or not. If it is not, the units are checked that they are in centimetres and if not, a SEVERE message is sent and an error will result.
- 2007-03-02 - TRF: Removed all references to the deprecated SCalOeZpdProduct calibration product.
- 2007-03-09 - TRF: Complete overhaul of IfgmCreation.
- 2007-03-13 - TRF: No longer use sdt.keySet(). Instead, the channel names are retrieve via sdt.getChannelNames(). Changed some logging levels to suppress debugging output.
- 2007-03-20 - TRF: Changed casting of some sdt and smect parameters from LongId to DoubleId to avoid class cast exceptions.
- 2007-03-20 - TRF: Cleaned up some JavaDoc
- 2007-03-20 - TRF: Removed dormant setSdiMeta() method, moved this functionality to the execute() method.
- 2007-03-21 - TRF: Changed dX to a DoubleParameter (value + unit)
- 2007-03-22 - TRF: Added Quantity and description to XofT and ZofT TD columns

- 2007-04-12 - YuW: Finished testing complete functionality however a new method convertMPDtoZPD has been added.
- 2007-04-16 - TRF: Added a method to set the direction MetaData parameter in the output SDI CDs
- 2007-05-31 - AH: Class placed in independent package. JParamaters updated.
- 2007-06-28 - AH: Added time offset calibration product
- 2007-07-20 - AH: MPD grid is computed individually per channel which establishes a common OPD grid across all channels.
- 2007-07-31 - AH: Migrate to new Unit API
- 2007-08-28 - TRF: Migrate to new Unit API
- 2007-09-07 - TRF: Added SpecSmecStepFactor argument
- 2007-10-02 - ZW: Re-organized variables in execute().
- 2007-11-13 - TRF: Fixed metaData (type, description, creator) in output SDI.
- 2007-11-13 - TRF: Added a quick hack to mask scans that do not contain the default position of ZPD (0.7850cm).
- 2008-04-28 - TRF: Changed deprecated getChannelSignal() to getSignal()
- 2008-05-12 - ES: Propagate metaData correctly
- 2008-05-15 - ES: Update propagation of metaData to reflect new product methods
- 2008-05-16 - ES: Changed the type of Parameter used when setting numScans from DoubleParameter to LongParameter
- 2008-07-15 - TRF: Fixed the top-level task description javadoc to conform with the SPIRE Spectrometer Pipeline Description document (SPIRE-BSS-DOC-002966).
- 2008-07-15 - TRF: Removed setCreator() step from MetaData modification to conform with Analysis in SPIRE SPR-0754.
- 2008-07-18 - TRF: Modified readIt() so that it no longer assumes that DetectorTimeline signal data is of type Double1d (SPIRE SPR-0781)
- 2008-08-08 - ES: Added SpirePointingProduct task parameter as per input requirements.
- 2008-08-08 - ES: Added checkInput() method, for checking input is proper.
- 2008-08-08 - ES: Added checkMask() method, for checking that input masks are properly set.
- 2008-08-08 - ES: Deprecated removeSmallScans() method, replaced it with new removeBadScans() method, which removes scans that are either too short or too long (outside of 95-105% of the commanded range).
- 2008-08-08 - ES: [SPIRE SPR-0835] Added getCommandedResolution() method that calculates what the output SDI resolution should be, based on the SCANSTART and SCANEND commanded values of the input nhkt.
- 2008-08-12 - ES: Deprecated checkMask(), as its purpose is no longer an input requirement.
- 2008-08-18 - ES: Change removeBadScans() to only mark scans as bad if their minimums are larger than the standard minimums (instead of larger OR smaller), or if their maximums are smaller


than the standard maximums (instead of larger OR smaller). As well, the commanded SCANSTART and values are now used in determining acceptable ranges for inputs that are not HR, MR, or LR.

- 2008-08-21 - TRF: Fixed bug in readIt(). The method now checks to see if calSpecChanTime-Off is null. If it is, the assigned time offset is 0 seconds.
- 2008-08-21 - TRF: [SPIRE SPR-0818] New class named CreateIfgmTask to replace the mis-named CreateIfgm class
- 2008-08-23 - TRF: Removed deprecated methods checkMask() and removeSmallScans().
- 2008-10-07 - TRF: [SPIRE SPR-0918] Added descriptions to TaskParameters. Made jparameter descriptions more descriptive. Modified some of the HTML tags in the description.
- 2008-10-09 - TRF: [SPIRE SCR-931] SpireInterferogramId TDs in the output SDI are now sorted by name.
- 2008-11-06 - TRF: [SPIRE SPR-1032] Added checkInterrupted() to execute()
- 2008-11-12 - TRF: Minor code clean-up, removing some commented lines.
- 2008-11-12 - TRF: [SPIRE SPR-0948] Added a maskInterpolator() method that blindly propagates the SpireMasks from the input SDT to the output SDI
- 2008-11-21 - TRF: Fixed unchecked, serial, javadoc warnings
- 2008-11-21 - TRF: [HCSS SPR-3232] Changed jhistory entries based on note in
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2008-12-10 - TRF: [SPRIE SPR-0948] Deprecated calSpecChanMask TaskParameter. Removed all usage of the chanMask product in the task
- 2008-12-12 - TRF: [SPRIE SPR-0948] Added channelIsUnusable(), createUnusableIx() to propagate Unusable channels to the output product, Depreacted isChannelDead()
- 2008-12-12 - TRF: [SPRIE SPR-1063] Added getCommandedResolution(nhkt) in anticipation of the implementation of SCR-0605. Until then, commandedResolution will be derived from nhkt.SCANSTART, nhkt.SCANEND
- 2008-12-17 - TRF: [SPRIE SPR-1068] Added modified determineDx(), determineDopd() so that the SMEC speed and detector sampling rate are taken into account.
- 2009-01-15 - TRF: [SPRIE SPR-1120] Removed dependency on herschel.binstruct.util.Sorting in findUniqScanNumber(). Using herschel.ia.numeric.toolbox.basic.Sort instead
- 2009-01-15 - TRF: Removed @SuppressWarnings(serial), replaced with serialVersionUID
- 2009-01-19 - TRF: [SPRIE SPR-1120] Had originally forgot to remove the associate import statement. It has now been removed.
- 2009-01-19 - TRF: Removed deprecated private methods checkConstant() and isChannelDead()
- 2009-01-19 - TRF: Changed logging levels to conform with agreed upon logging policy.
- 2009-01-19 - TRF: [SPRIE SPR-1068] Fixed (potential) infinite loop in determineDx()
- 2009-03-11 - YFR: [SPIRE SPR-1062] Pointing information is now propagated to individual interferograms

- 2009-03-17 - TRF: [SPIRE SPR-1107] Set warning messages when non-nominal channels are not found in cal products to config level
- 2009-05-05 - TRF: [SPIRE SPR-1063] Modified the setting of the commandedResolution metadata keyword to accomodate PFM4 data and the CR case.
- 2009-05-06 - TRF: [SPIRE SPR-1182] Added a note on the OPD grids being slightly different for each channel to the documentation.
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-05-05 - TRF: [SPIRE SPR-1063] Modified removeBadScans() to trap off the coammnded scan start/end.
- 2009-05-12 - YFR: [SPIRE SPR-1064] Added methods to remove truncated uncorrect points for a given scan from beginning and end
- 2009-05-12 - YFR: [SPIRE SCR-1064] Added methods to remove truncated points for a given scan from beginning and end
- 2009-05-21 - YFR: [SPIRE SCR-xxxx] Deprecated methods getCommonEncoderMinimum, getCommonEncoderMaximum, filterScans and removeBadScans as we moved the interferogram length check from before merging with signal time to after
- 2009-05-21 - YFR: [SPIRE SCR-1283] Added method to shape SDI to make sure scans have identical OPD after mpds have been truncated because sdt times have been truncated by mask bits
- 2009-05-21 - YFR: [SPIRE SCR-1270] Throw BadDataException when the output sdi is empty and remove empty scans from sdi
- 2009-06-12 - TRF: [SPIRE SPR-1507] Cast sdt as a Product in propagateMeta()
- 2009-06-12 - TRF: [SPIRE SPR-1511] Task now sets Creator as per SPIRE Pipeline Policy
- 2009-06-12 - YFR: [SPIRE SPR-1181] Added method to sort scan numbers based on scan time
- 2009-06-13 - TRF: [SPIRE SPR-1107] Task now checks for valid channel names and removes unexpected entries.
- 2009-06-13 - TRF: [SPIRE SPR-1107] Task no longer catches exceptions if valid channels are not found in a calibration product
- 2009-06-13 - TRF: [SPIRE SCR-1489] Task can now accept a calibration context in lieu the long list of calibration products.
- 2009-06-13 - TRF: [SPIRE SCR-1489] Deprecated task parameters that violate the SPIRE Pipeline policy.
- 2009-06-25 - TRF: [SPIRE SPR-1526] WARNINGS are issued whenever an interferogram is removed/omitted.
- 2009-07-16 - TRF: [SPIRE SPR-1667] Scan direction is now set in the output sdcids
- 2009-07-16 - TRF: [SPIRE SPR-1635] maskInterpolator() uses NearestNeighbor instead of Linear Interpolation.
- 2009-07-16 - TRF: [SPIRE SPR-1526] WARNINGS are now issued whenever points/scans are truncated.
- 2009-07-17 - YFR: [SPIRE SPR-1526] Added check to where we find the valley and peak, when the smect is truncated, we don't want to include points don't belong to the given scan

- 2009-07-19 - TRF: Removed deprecated methods
- 2009-07-31 - TRF: [SPIRE SPR-1182] Moved the note on OPD grids to the JavaDoc.
- 2009-07-31 - TRF: Removed deprecated TaskParameters.
- 2009-08-04 - TRF: Task no longer checks for non-existent TaskParameters.
- 2009-08-04 - TRF: [SPIRE-1705] Removed empty catch statement.
- 2009-08-27 - TRF: [SPIRE-1814] Corrected the TaskParameter names.
- 2009-09-02 - TRF: [SPIRE-1905] level metadata in output sdi is now set to "10"
- 2009-09-04 - TRF: [SPIRE-1914] Fixed the nhkt.SCANS input check
- 2009-09-11 - YFR: [SPIRE-1947] Fixed the exception when the user's locale is not CANADA
- 2009-09-15 - TRF: [SPIRE-1941] Allowed extrapolate when using Regrid
- 2009-09-15 - TRF: [SPIRE-1063] Fixed resolution setting when SCANRES has been set in NHKT
- 2009-10-15 - DMT: [SPIRE-1362] Added quality control metadata collection to the task
- 2009-10-16 - TRF: [SPIRE-2015] Added another short scan check
- 2009-11-04 - TRF: [SPIRE-2133] Removed reference to "lvdtdcsignal"
- 2009-11-10 - TRF: [SPIRE-2145] Ignore missing scans when computing dMPD
- 2010-01-15 - TRF: Removed usage of deprecated EXTRAPOLATION keyword in Regrid()
- 2010-01-15 - TRF: Corrected some misapplications of == (should be .equals())
- 2010-01-15 - TRF: Removed redundant code
- 2010-01-15 - TRF: [SPIRE-2280] Misc fixes to URM information
- 2010-01-15 - TRF: [SPIRE-2329] setRaAndDec() now uses new getSkyPositions() method
- 2010-02-22 - WT: [SPIRE-2309] Reimplemented getMissedFringes() method
- 2010-03-01 - TRF: [SPIRE-2376], [SPIRE-2342] Task now checks to see if SMEC was in OPEN LOOP mode
- 2010-04-01 - WT: [SPIRE-2445] Changed LR/MR/HR/CR start and end values
- 2010-05-18 - TRF: [SPIRE-2280] Misc fixes to Javadoc/URM information

1.46. CreateOffsetHistTask

Full Name:	herschel.spire.ia.pipeline.pg.CreateOffsetHistTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.pg import CreateOffsetHistTask

Description

Task to create a Offset History calibration product

API Summary

Properties
EdpContext timelines [Input, Mandatory, default=No default value]
OffsetHist offsetHist [Output, Optional, default=No default value]

API details

Properties


EdpContext timelines [Input, Mandatory, default=No default value]
An EdpContext containing offset timelines
OffsetHist offsetHist [Output, Optional, default=No default value]
The resulting Offset History product

History

- 2006-11-27 - PPa: First version
- 2006-12-01 - PPa: Use only first and last frame of the building blocks.
- 2007-03-09 - PPa: Use EdpProduct class and replace deprecated method of DetectorTimeline
- 2007-03-28 - PPa: Now the OffsetHist stores values in raw values.
- 2007-05-07 - PPa: EngConverion output is now a EdpContext, so change the class of "timelines" parameter
- 2007-08-06 - PPa: Add modelName calculation.
- 2007-11-13 - PPa: Fix modelName and add Suppress.
- 2008-04-30 - PPa: Use new methods of DetectorTimeline
- 2008-08-27 - PPa: Fix SPR-0846.
- 2009-01-21 - PPa: Implement SCR-0939
- 2009-01-26 - PPa: Better handling of empty input.

- 2009-01-28 - PPa: Add jparameter tags and parameter descriptions [SPR-0985].
- 2009-03-10 - PPa: Don't use EdpContext.get(int).
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true) and addRow.
- 2009-05-22 - PPa: Change package and name [SCR-0961]. Add parameter descriptions. Fix creator.
- 2009-06-09 - PPa: Tentative setting of start-end date.
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]

1.47. CreateResetHistTask

Full Name:	herschel.spire.ia.pipeline.pg.CreateResetHistTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.pg import CreateResetHistTask

Description

Task to extract the DPU Counter Reset History

API Summary

Properties
Date startDate [Input, Optional, default=No default value]
Date endDate [Input, Optional, default=No default value]
FineTime startTime [Input, Optional, default=No default value]
FineTime endTime [Input, Optional, default=No default value]
Long obsid [Input, Optional, default=No default value]
Long bbid [Input, Optional, default=No default value]
Boolean tmPack [Input, Optional, default=true]
NhkTimeline nhkt [Input, Optional, default=No default value]
String dbName [Input, Optional, default=No default value]
ObjectStore store [Input, Optional, default=No default value]
ResetHist resetHist [Output, Optional, default=No default value]

API details

Properties


Date startDate [Input, Optional, default=No default value]
Start date of the time range.
Date endDate [Input, Optional, default=No default value]
End date of the time range.
FineTime startTime [Input, Optional, default=No default value]
Start time of the time range.
FineTime endTime [Input, Optional, default=No default value]
End time of the time range.
Long obsid [Input, Optional, default=No default value]
Observation ID (OBSID).
Long bbid [Input, Optional, default=No default value]
Building Block ID (BBID).

Boolean tmPack [Input, Optional, default=true]
Flag to use TmSourcePackets (tmPack=True) instead of SpireDataFrames (tmPack=False)
NhkTimeline nhkt [Input, Optional, default=No default value]
Nominal Housekeeping Timeline. If used, the ResetHist will be created using the TRESET data of this product.
String dbName [Input, Optional, default=No default value]
Database name. It is used to override the configuration property and access a DB that differ from the one specified i configuration files.
ObjectStore store [Input, Optional, default=No default value]
ObjectStore for direct connection with the versant database.
ResetHist resetHist [Output, Optional, default=No default value]
The output ResetHist calibration product

History

- 2005-08-17 - PPa: First (not complete) version.
- 2006-06-06 - PPa: Completed implementation
- 2006-09-29 - PPa: Refactoring
- 2006-11-10 - PPa: Add tmPack parameter
- 2006-12-14 - PPa: Implement extraction of dataframes
- 2007-04-12 - PPa: If no data is found, throw a RuntimeException. Get source also for Spire-DataFrame.
- 2007-07-27 - PPa: Migrate to new Unit API
- 2007-10-11 - PPa: Fix usage of ObjectStore
- 2007-10-15 - PPa: Add start/endTime and some documentation.
- 2008-02-11 - PPa: Allow possibility to create the ResetHist from a NHKT
- 2008-03-27 - PPa: Select the appropriate parameter table (SPR-497).
- 2008-07-17 - PPa: Fix name
- 2009-05-19 - PPa: remove setResetOnExecute, add serialVersionUID
- 2009-05-22 - PPa: Change package and name [SCR-0961]. Add parameter descriptions. Fix creator.
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]

1.48. createSpirePointing

Full Name:	herschel.spire.ia.pipeline.common.pointing.CreateSpirePointingTask
Alias:	createSpirePointing
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.pointing import CreateSpirePointingTask
Category:	SPIRE

Description

Task to create the spire pipeline product

Example

Example 1: generation of a SpirePointingProduct:

```
$
spp=createSpirePointing(hpp=obs.auxiliary.pointing,siam=obs.auxiliary.siam,detAngOff=obs.calibr
```

API Summary

Properties
PointingProduct hpp [Input, Mandatory, default=No default value]
SiamProduct siam [Input, Mandatory, default=No default value]
DetAngOff detAngOff [Input, Mandatory, default=No default value]
BsmAnglesTimeline bat [Input, Optional, default=No default value]
SpirePointingProduct spp [Output, Optional, default=No default value]

API details


Properties

PointingProduct hpp [Input, Mandatory, default=No default value]
Herschel Pointing Product
SiamProduct siam [Input, Mandatory, default=No default value]
SIAM Product
DetAngOff detAngOff [Input, Mandatory, default=No default value]
Detector Angle Offset calibration product
BsmAnglesTimeline bat [Input, Optional, default=No default value]
BSM Angles Timeline
SpirePointingProduct spp [Output, Optional, default=No default value]
Spire Pointing Product

History

- 2009-07-30 - PPa: New task to create a spire pointing product. [SPIRE-1648]
- 2009-07-31 - PPa: fix parameters
- 2009-09-13 - PPa: Add example [SPIRE-1860]. Make "bat" optional

1.49. cutPhotDetTimelines

Full Name:	herschel.spire.ia.pipeline.common.util.CutPhotDetTimelinesTask
Alias:	cutPhotDetTimelines
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.util import CutPhotDetTimelinesTask
Category:	SPIRE/Pipeline/Common Task

Description

Task to cut a Pointed Photometer Timelines that were previously joined with the detector timelines of the trailing and following turnaround building block in scan maps.

Examples

Example 1: How to use the task from HIPE command line.

```
my_output = cutPhotDetTimelines(pdt=my_ppt,extend=True)
```

Example 2: How to get the CVS number version of the task.

```
print cutPhotDetTimelines.VERSION
```

API Summary

Properties
PointedPhotTimeline pdt [Input, Mandatory, default=No default value]
Boolean extend [Input, Optional, default=Boolean.FALSE]
PointedPhotTimeline output [Output, Optional, default=No default value]

API details

Properties

PointedPhotTimeline pdt [Input, Mandatory, default=No default value]
Input (mandatory): Pointed Photometer Timelines Product to be cut.
Boolean extend [Input, Optional, default=Boolean.FALSE]
Input (optional): Flag to keep half of turnaround data.
PointedPhotTimeline output [Output, Optional, default=No default value]
Output (optional): Pointed Photometer Timelines Product.

History

- 2009-08-28 - PPa: First version
- 2009-09-08 - PPa: Update javadoc and parameter descriptions. Correct metadata propagation.
- 2010-05-11 - GM: Add version to creator

1.50. CWT

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.CWT
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import CWT


Description

unknown

History

- 2006-04-24 - CO: first java version
- 2006-10-01 - CO: add noiseEstimation method
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching ->herschel.spire.ia.pipeline.common.deglitch
- 2008-04-29 - DB: correction warning Trevor Fulton version build 814
- 2009-01-23 - DB: correction jhistory
- 2009-01-29 - DB: correction coeff /a convolution

1.51. dbQueryMaker

Full Name:	herschel.spire.ia.pipeline.common.rawdata.DbQueryMakerTask
Alias:	dbQueryMaker
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import DbQueryMakerTask
Category:	SPIRE/Pipeline/Common Task

Description

Database Query Maker Task.

This task is a front-end task to access SpireDataFrames or TmSourcePackets stored in a telemetry database.

The task translates the user inputs in a database query to access SpireDataFrames or TmPackets. The task output is a ProductReader, i.e. a connection with the database. To access the dataframe/packets one should use the openStream() method which returns a herschel.access.ProductStream that will be used by the user to download the data with the stream.next() method.

The user can specify the range of data to be extracted specifying the start date/time, the end date/time, the OBSID, the BBID, the BBFULLTYPE, one or more APID, the TYPE value, and/or the mnemonic name of the TM packet. It is also possible for the user to specify if telemetry data should be downloaded from the database in form of SpireDataFrames or TM packets. The latter form will be used to access event reporting packets and other packets that are not converted into dataframes. The connection type and the database name have to be specified in the properties. It is possible to override the database name with the "dbName" keyword.

Note: this task is intended only for ICC members to access telemetry data, or usage in SGP. It is not for general users.

Example

Example 1: Make a query to extract NHK telemetry packets of a specific obsid from operational database.

```
HIPE>>
connection=dbQueryMaker(obsid=0x50001234L,packetType="NHK",database="ops_1",tmPack=True)
```

API Summary

Properties
Date startDate [INPUT, OPTIONAL, default=No default value.]
Date endDate [INPUT, OPTIONAL, default=No default value.]
FineTime startTime [INPUT, OPTIONAL, default=No default value.]
FineTime endTime [INPUT, OPTIONAL, default=No default value.]
Long obsid [INPUT, OPTIONAL, default=No default value.]
Long bbid [INPUT, OPTIONAL, default=No default value.]
Long bbfulltype [INPUT, OPTIONAL, default=No default value.]
Integer apid [INPUT, OPTIONAL, default=No default value.]
Integer[] apids [INPUT, OPTIONAL, default=No default value.]

Properties
Integer type [INPUT, OPTIONAL, default=No default value.]
String packetType [INPUT, OPTIONAL, default=No default value.]
Boolean tmPack [INPUT, OPTIONAL, default=false.]
String dbName [INPUT, OPTIONAL, default=No default value.]
ObjectStore store [INPUT, OPTIONAL, default=No default value.]
ProductReader connection [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


Date startDate [INPUT, OPTIONAL, default=No default value.]
Start date of the time range of the query.
Date endDate [INPUT, OPTIONAL, default=No default value.]
End date of the time range of the query.
FineTime startTime [INPUT, OPTIONAL, default=No default value.]
Start time of the time range of the query.
FineTime endTime [INPUT, OPTIONAL, default=No default value.]
End time of the time range of the query.
Long obsid [INPUT, OPTIONAL, default=No default value.]
Observation ID (OBSID).
Long bbid [INPUT, OPTIONAL, default=No default value.]
Building Block ID (BBID).
Long bbfulltype [INPUT, OPTIONAL, default=No default value.]
Building Block Full Type (BBFULLTYPE), i.e. the 16 most significant bits (bbid >>> 16).
Integer apid [INPUT, OPTIONAL, default=No default value.]
Application Program ID (APID).
Integer [] apids [INPUT, OPTIONAL, default=No default value.]
Array of APIDs (override apid input).
Integer type [INPUT, OPTIONAL, default=No default value.]
TM packet TYPE value.
String packetType [INPUT, OPTIONAL, default=No default value.]
Mnemonic name of dataframe/TM packet type. Override apid and apids input.
Boolean tmPack [INPUT, OPTIONAL, default=false.]
Flag to use TmSourcePackets (tmPack=True) instead of SpireDataFrames (tmPack=False).

<code>String dbName [INPUT, OPTIONAL, default=No default value.]</code>
Database name. It is used to override the configuration property and access a DB that differ from the one specified in configuration files.
<code>ObjectStore store [INPUT, OPTIONAL, default=No default value.]</code>
ObjectStore for direct connection with the versant database.
<code>ProductReader connection [OUTPUT, OPTIONAL, default=No default value.]</code>
Database connection for a SpireDataFrame or TmPacket stream.

History

- 2005-03-00 - PPa: First version
- 2005-05-00 - PPa: Added all useful parameters to build a complete query. Now handle DataFrames and TmPackets.
- 2005-06-00 - PPa: Moving to the new documentation schema and to the new package. Merged in the main tree.
- 2005-07-05 - PPa: Added TM packet TYPE.
- 2005-08-16 - PPa: Added database name.
- 2005-08-23 - PPa: Add SpireDataFrame classname setting in dataframe access
- 2005-09-22 - PPa: Better exception handling; added logFailed() method
- 2005-11-16 - PPa: Change "dfFlag" into "tmPack".
- 2006-04-22 - PPa: Use new TelemetryAcces.setBbid() method.
- 2006-05-31 - PPa: Add call of setResetOnExecute(true).
- 2006-08-11 - PPa: Refactoring.
- 2006-09-13 - PPa: Return a connection instead of a stream.
- 2007-04-12 - PPa: When asked for a specific packetName, it asks for both normal and redundant APID
- 2007-04-16 - PPa: Add the possibility to use an ObjectStore. Access workaround
- 2007-10-15 - PPa: Update documentation.
- 2008-03-27 - PPa: Select the appropriate parameter table (SPR-497).
- 2008-07-17 - PPa: Fix name
- 2009-04-21 - PPa: Remove setResetOnExecute(true).
- 2009-06-02 - PPa: Add description [SPR-1457].
- 2010-05-11 - PPa: Improve Javadoc and URM [SPIRE-2262]
- 2010-05-12 - PPa: Update URM [SPIRE-2262]

1.52. DeglitchIfgmTask

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.deglitch.DeglitchIfgmTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm.deglitch import DeglitchIfgmTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to perform Second Level Deglitching on SPIRE Spectrometer Data.

Task Execution Steps: This task performs deglitching on a single SDI product. The task offers four statistical deglitching algorithms to identify glitches in an SDI and reconstruct the flagged samples. The STD and MAD algorithms require at least 3 scans in the input SDI, WINDOW_STD and WINDOW_MAD require at least 2. The task performs in the following manner: - If the total number of scans in the SDI is less than the minimum number required for the selected deglitching algorithm, a message is logged and the input SDI is returned unchanged. - If there are enough scans for the selected deglitching algorithm for a given direction, the task performs regular deglitching. - If there is at least one scan, but there are not enough scans for a given direction to perform deglitching, helper scans are added from the other direction. - If there are exactly 2 scans for a given direction and the WINDOW_MAD or WINDOW_STD algorithm is selected, the neighbour method is used to evaluate the signal just before and after the flagged sample. - Helper scans are removed if any were introduced. - Samples, flagged as glitches, are replaced by the average value of samples not affected by a glitch for the same OPD.

1. Performs glitch detection with given input parameters or suitable defaults
2. Performs glitch correction with given input parameters or suitable defaults

Examples

Example 1: Use as a detection threshold a Windowed Median Absolute Deviation of size 33 samples.

```
sdi=deglitchIfgm(sdi=sdi, deglitchType="MAD_WINDOW", windowSize=33)
```

Example 2: Use as a detection threshold Median Absolute Deviation with the default settings.

```
sdi=deglitchIfgm(sdi=sdi, deglitchType="MAD")
```

Example 3: Use as a 3-sigma STDDEV detection threshold but do not correct the glitches.

```
sdi=deglitchIfgm(sdi=sdi, deglitchType="STD", correctGlitches=Boolean.FALSE)
```

API Summary

Properties
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]
String deglitchType [INPUT, OPTIONAL, default=No default value]
Double thresholdFactor [INPUT, OPTIONAL, default=No default value]
Integer windowSize [INPUT, OPTIONAL, default=No default value]

Properties
Boolean identifyGlitches [INPUT, OPTIONAL, default=true]
Boolean correctGlitches [INPUT, OPTIONAL, default=true]
SpectrometerDetectorInterferogram outsdI [OUTPUT, MANDATORY, default=None]

API details

Properties


SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=None default value]
Input Spectrometer Detector Interferogram product to be deglitched.
String deglitchType [INPUT, OPTIONAL, default=None default value]
The deglitching method used, can be either "STD", "STD_WINDOW", "MAD", "MAD_WINDOW". The default deglitchType is determined by GlitchFinder.
Double thresholdFactor [INPUT, OPTIONAL, default=None default value]
The factor that defines what is considered to be a glitch. The default thresholdFactor is determined by GlitchFinder.
Integer windowSize [INPUT, OPTIONAL, default=None default value]
The running window that will be used to look for outliers in the secondary statistics. The default windowSize is determined by GlitchFinder.
Boolean identifyGlitches [INPUT, OPTIONAL, default=true]
The option of identifying glitches before glitch correction. The default is true.
Boolean correctGlitches [INPUT, OPTIONAL, default=true]
The option of removing glitches following glitch identification. The default is true.
SpectrometerDetectorInterferogram outsdI [OUTPUT, MANDATORY, default=None]
The resultant deglitched SDI.

History

- 2007-05-31 - AH: Class placed in independent package. JParamaters updated.
- 2008-08-25 - TRF: [SPIRE SPR-0823] New class named DeglitchIfgmTask to replace the misnamed DeglitchIfgm class
- 2008-11-13 - TRF: [SPIRE SPR-0811] Added new task parameters thresholdFactor and windowSize.
- 2008-11-13 - TRF: [SPIRE SCR-0811] Improved the user/developer documentation.
- 2008-11-13 - TRF: [SPIRE SPR-1030] Implement masking using the SpireMask interface.
- 2008-11-13 - TRF: [SPIRE SPR-1031] Task is now interruptible

- 2008-12-02 - TRF: [SPIRE SPR-0811] Improving User documentation
- 2008-12-12 - TRF: [HCSS SPR-3232] Fixed some jtag urm documentation
- 2008-12-12 - TRF: Fixed unchecked warnings
- 2008-12-12 - TRF: [SPIRE SPR-1081] Task no longer propagates "fileName" metadata
- 2008-12-15 - JL: [SPIRE SCR-1095] Added identifyGlitches and correctGlitches boolean input parameters.
- 2008-12-15 - JL: [SPIRE SCR-1095] Added getExpectedFalsePositivesRatio method.
- 2008-12-17 - TRF: [SPIRE SPR-0811] Improving User documentation
- 2009-01-29 - TRF: [SPIRE SPR-1165] Removed SuppressWarnings annotation
- 2009-02-04 - TRF: [SPIRE SPR-1204] keySet().iterator() -> getScanNames()
- 2009-04-16 - TRF: [HCSS SPR-6051], [SPIRE SPR-1135]: Changed jcategory tag from Task to task
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-05-22 - TRF: Modified the call to getLogger() so that the messages printed by this task could be controlled
- 2009-06-12 - TRF: [SPIRE SPR-1511] Task now sets Creator as per SPIRE Pipeline Policy
- 2009-11-19 - DMT: [SPIRE SPR-2044] Implemented the code for 2 or fewer scans per direction observations
- 2010-03-01 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.
- 2010-05-18 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.

1.53. DemodPhotProduct

Full Name:	herschel.spire.ia.dataset.DemodPhotProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import DemodPhotProduct

Description


Demodulated Photometer Product.

Class implementing Demodulated Photometer Product

History

- 2008-02-01 - RG: created from DemodPhotProduct
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2008-12-01 - TRF: Fixed some more jtag urm documentation [HCSS SPR-3232]
- 2009-08-24 - PPa: Remove "wcs" metadata [SPIRE-1854]
- 2009-09-10 - PPa: numSamplesOn, numSamplesOff replace nsample
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-04-21 - PPa: Remove useless public field. Fix jhelp [SPIRE-2255]

1.54. demodulate

Full Name:	herschel.spire.ia.pipeline.phot.demodulation.DemodulateTask
Alias:	demodulate
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.demodulation import DemodulateTask
Category:	task

Description

This task follows

SPIRE-UCF-DOC-002890, The SPIRE Analogue Signal Chain and Photometer Detector Data Processing Pipeline Matt Griffin. The points of the detector timeline are labelled up if they belong to a time window around the chopper up, and labelled down if they belong to a time window around chopper down. * the "good" points are selected in a time window around the chopper motion ($t_{zero}[k] + \text{transition time} < \text{time} < (t_{zero}[k+1] + \text{delay time})$) Then a mean of the points up is computed, a mean of the points down, and the difference of means for one chopper cycle.

API Summary

Properties
PointedPhotTimeline ppt [Input, Mandatory, default=No default value]
ChopJiggTimeline cjt [Input, Mandatory, default=No default value]
Double delayTime [Input, Optional, default=0.05 seconds]
Double transitionTime [Input, Optional, default=0.1 seconds]
PhotDemodProduct output [Output, Mandatory, default=No default value]

API details

Properties

PointedPhotTimeline ppt [Input, Mandatory, default=No default value]
PPT: Photometer Bolometer Timelines (with pointing information) to demodulate
ChopJiggTimeline cjt [Input, Mandatory, default=No default value]
Chopper Jiggle Timeline: beam steering mirror two angles timelines
Double delayTime [Input, Optional, default=0.05 seconds]
after the beginning motion of the BSM, the data are still good during the Delay Time, unit is seconds
Double transitionTime [Input, Optional, default=0.1 seconds]
after the beginning motion of the BSM, the data are not stabilised during Transition Time, unit is seconds


PhotDemodProduct output [Output, Mandatory, default=No default value]
--

Demodulated Detector Timeline: the demodulated photometer timeline
--

History

- July 2007: First version
- 16 March 2007 RG add meta keyword subsystem
- 04 December 2007 RG code reorganised, change the name from JiggAverager
- 13 December 2007 RG code reorganised one point per chopper cycle and not per jiggle state
- 01 February 2008 RG reorganise with new input/output products PointedPhotTimeline PhotDemodProduct
- 2008-10-21 - RG: - SPR 815 correct the "average" of pointings, now calls DemodLibrary.averageAstro(),
- to be replaced by the correct library when it will exist
- accept as input both float1d and double1d in signal
- 03-June-2009 SPR-1458 RG correct the jparameter and setDescription
- 28-august-2009 spire spr-1878 propagate mask ISDEAD, ISNOISY, ISNOCHOPSKY, ISSLOW, NO_RESP_DATA

1.55. DeNoddingTask

Full Name:	herschel.spire.ia.pipeline.phot.denodding.DeNoddingTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.denodding import DeNoddingTask
Category:	task

Description

produces a Pointed Photometer Product.

This task takes averaged demodulated Timelines produced by DemodulateTask and produces a Pointed Photometer Product.

Example

Example 1: A Jython example

```
from herschel.spire.ia.pipeline.phot.denodding import *
task = DeNoddingTask()
```

API Summary

Properties
DenodInput input [Input, Mandatory, default=No default value]
denodding output [Output, Optional, default=No default value]

Limitations

This class assumes that an ABBA nod sequence is identified via a 0 and 1 in nodId where the values are set by the parameters nodA and nodB in the code.

API details

Properties


DenodInput input [Input, Mandatory, default=No default value]
Takes Level 1 averaged demodulated timelines from DemodulateTask as input
denodding output [Output, Optional, default=No default value]
Provides output in the form of a Pointed Photometer Product, which serves as input to the level 1 task NodAverageTask.

History

- 2005-12-01 - TB: v0.1 First "empty" version
- 2006-01-01 - TB: v0.2 First "in anger" version
- 2006-04-01 - TB: v0.3 Rewritten to take multiple DPPs and different observation sequence. AB rather than ABBA.

- 2006-05-01 - TB: v0.4 Now the requirement is ABBA again.
- 2006-06-01 - TB: v0.5 compiles. Altered to remove IOS assumptions.
- 2006-06-10 - TB: v0.6 Altered to remove composite datasets (DPP no longer has them).
- 2006-10-01 - TB: v0.7 Need to handle AB observations (partic. for 64 point jiggle).
- 2007-03-01 - TB: Various compile warnings and deprecated functions removed.
- 2007-03-14 - TB: New data structure for DPPs and PPPs means DeNodding has been changed
- 2007-04-03 - TB: pre- and postambles merged into execute
- 2007-05-01 - TB: now PPP is called PointedPhotProduct
- 2008-06-25 - TB: Replaced depreciated methods for DetectorTimeline class.
- 2008-10-14 - BOH: Renamed task, and edited to follow task naming convention
- 2008-11-03 - BOH: changed NullPointerException to MissingDataException in the case of when input size is 0.
- 2008-11-04 - BOH: temporarily revert to NullPointerException.
- 2009-01-21 - BOH: edited jtags for compliance with recommended structure
- 2009-04-08 - BOH: refracted code. Fixed the formulae for the flux and errors for numDpps==4. Added a NullPointerException in the case of a ABB pointing.
- 2009-05-21 - BOH: Added units for PPP. Removed useless input.
- 2009-06-12 - BOH: Descriptions set.[SPR-1459]. Added metadata 'spireNodPointingUncertainty' [SPR-1467].
- 2009-08-12 - BOH: Added quality control metrics [SPIRE SPR-1468, 1713]
- 2009-08-22 - BOH: Major refractoring [SPIRE SPR-932]
- 2009-08-26 - BOH: fixed java.lang.IndexOutOfBoundsException crash related to [SPIRE-1787].
- 2009-09-10 - BOH: removed wcsType and wcsReference from metadata [SPIRE-1939].
- 2009-09-11 - BOH: added QC flag if numDpps==3 [SPIRE-1417].
- 2009-09-11 - BOH: added metadata for output DPP [SPIRE-1713, SPIRE-1877].
- 2009-10-09 - BOH: additional refraction [SPIRE-1977].
- 2009-10-21 - PPa: Fix metadata propagation [SPIRE-1877].
- 2009-11-18 - BOH: Fix for fluxes [SPIRE-2096].
- 2009-11-23 - BOH: Refactoring as per SPIRE-1997.
- 2010-03-01 - BOH: updated for Javadoc/URM [SPIRE-2251].
- 2010-04-15 - SG: Refactoring [SPIRE-1997].


1.56. DetAngOff

Full Name:	herschel.spire.ia.dataset.DetAngOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import DetAngOff
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2006-11-13 - PP: Use PixAngOffsetTable
- 2007-02-25 - PP: Implement ChannelCalibTable
- 2008-05-19 - PP: Change name.
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2009-03-13 - SG: Add aperture [SPIRE SCR-1241]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.57. Detection

Full Name:	herschel.spire.ia.pipeline.common.deglitch.detection.Detection
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.detection import Detection
Category:	class

History

- 2006-04-24 - CO: first java version
- 2007-04-09 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2008-04-29 - DB: corection warning Trevor Fulton version build 814
- 2008-05-13 - DB: correction error build 838
- 2009-01-23 - DB: correction jhistory
- 2009-02-03 - DB: correct index kept in computeHolderAnalysis()
- 2009-05-19 - DB: parametrization vector
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag

1.58. DetectorTimelineExplorerComponent

Full Name:	herschel.spire.ia.gui.DetectorTimelineExplorerComponent
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import DetectorTimelineExplorerComponent
Category:	SPIRE/IA GUI

Description

Explorer for DetectorTimeline products.

This explorer allows to visualize SPIRE DetectorTimeline products within HIPE (right-click on the product to be visualized and choose "DetectorTimelineViewer" from the menu). This explorer allow to:

1. Display all the Detectors (both spectrometer and photometer) in the honeycomb arrangement.
2. Plotting and Overplotting the timelines with single click on the desired detector.
3. Using TablePlotter to visualize the data.


See also

- [DetectorTimelineExplorer](#)
- [User's Manual](#)
- [DetectorTimelineExplorer presentation](#)

History

- 2007-05-04 - PPa: First version
- 2008-08-08 - GM: New layout, add new scale color (heat)
- 2010-01-07 - GM: New layout, added TablePlotter
- 2010-02-03 - TRF: [INFR-365] Migrated from herschel.spire.ia.dataset.gui
- 2010-05-11 - GM: Update the JavaDoc

1.59. DetectorTimelineExplorer

Full Name:	herschel.spire.ia.gui.DetectorTimelineExplorer
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import DetectorTimelineExplorer
Category:	SPIRE/IA GUI

Description

Explorer for DetectorTimeline products.

This explorer allows to visualize SPIRE DetectorTimeline products within DatasetInspector (right-click on the product to be visualized and choose "DetectorTimelineExplorer" from the menu) or HIPE command line. This explorer allow to:

1. Display all the Detectors (both spectrometer and photometer) in the honeycomb arrangement.
2. Plotting and Overplotting the timelines with single click on the desired detector.
3. Using TablePlotter to visualize the data.

In HIPE the same features are available also using Detetctor Timeline Viewer. To do this: identify the product for visualization from Variables list and right-click the timeline-type product, follow the "Open With" menu entry and select Detector Timeline Viewer from the drop-down menu.

Example

Example 1: How to launch DTE from HIPE command line or script.

```
from herschel.spire.ia.gui import DetectorTimelineExplorer
dte=DetectorTimelineExplorer()
p=dte.createFrame(yourProduct)
```

API Summary

Method
createFrame(Object ppp)

API details

Method

<code>createFrame(Object ppp)</code>
<p>Argument</p> <p><code>Object ppp</code> [INPUT, MANDATORY]</p> <p>Example</p> <p>Public method to launch DTE from HIPE command line.</p> <pre>from herschel.spire.ia.gui import DetectorTimelineExplorer dte=DetectorTimelineExplorer() p=dte.createFrame(yourProduct)</pre>


See also

- [DetectorTimelineExplorerComponent](#)
- [TimelineSimplePlot](#)
- [User's Manual](#)
- [DetectorTimelineExplorer presentation](#)

History

- 2007-05-04 - PPa: First version
- 2008-08-08 - GM: New layout, add new scale color (heat)
- 2010-01-07 - GM: New layout, added TablePlotter, new public method to create a standalone version of DTE
- 2010-02-03 - TRF: [INFR-365] Migrated from herschel.spire.ia.dataset.gui

1.60. DetectorTimeline

Full Name:	herschel.spire.ia.dataset.DetectorTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import DetectorTimeline

Description

Specialization of EdProduct for Detector Timeline products


like PDT, SDT etc. It provides: -- methods to access the signal and mask data for channel timelines. -- method to access the sampleTime (which is the same for all channels). -- method for compute the corrected readout time. -- the "source" metadata keyword and convenient methods to access it. -- methods to get the key set for pixel timelines, as Set or as String array.

History

- 2006-02-28 - PP: First version
- 2006-03-08 - PP: Add some javadoc.
- 2006-03-15 - PP: Fix getPixelNames(), add getPixelKeySet() and getPixelIds().
- 2006-03-17 - PP: Use int for pixelIds.
- 2006-03-20 - PP: Improve documentation. Merged in the main tree.
- 2006-05-31 - PP: Use enhanced for loop.
- 2006-06-01 - PP: Fix getPixelKeySet() and add getPixelError() and getPixelMask()
- 2006-06-20 - PP: Add get/setPixelSignalUnit, add set methods for signal, error, sampleTime and mask, add get/setPixelTable methods.
- 2007-02-20 - PP: Use generics.
- 2007-03-02 - PP: Move toward new structure requested by Bernhard Shulz: add new methods and deprecate old ones. Now the class should support both old and new structure.
- 2007-03-05 - PP: Now extends the new class EdpProduct. Some methods moved in EdpProduct
- 2007-03-08 - PP: Add method to compute the corrected readout time.
- 2007-03-15 - PP: Remove some deprecated methods.
- 2007-03-16 - PP: Remove more deprecated methods. Improve some javadoc.
- 2007-03-23 - PP: Add isPhotometer() and isSpectrometer() methods
- 2007-03-28 - PP: Remove more deprecated methods. Deprecate getPixelKeySet.
- 2007-05-24 - PP: Remove getPixelKeySet. Now supports only new structure.
- 2007-05-28 - PP: Add support for voltage and resistance tables
- 2007-05-29 - PP: More support for voltage tables
- 2007-07-27 - PP: Start migration to new Unit API

- 2007-08-10 - PP: Complete migration to the new Unit API.
- 2007-09-06 - TRF: Added a removeColumn(columnName) method
- 2007-09-07 - TRF: Added removeSignalColumn(), removeMaskColumn(), and removeQualityRow() methods
- 2007-09-18 - PP: Fix copy constructor
- 2007-11-14 - PP: Fix removeColumn method.
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-04-28 - PP: introduce new getVoltage, getSignal, getResistance, getVoltageUnit... (SPR-0660)
- 2008-05-08 - ES: Added initMeta(), getCommandedResoluton(), and setCommandedResoluton() methods
- 2008-06-26 - PP: Remove deprecated methods.
- 2008-07-01 - PP: Add biasFreq as metadata keyword, fix copy constructor.
- 2008-11-10 - PP: Remove commandedResolution [SPR-0757]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2008-12-01 - TRF: Fixed some more jtag urm documentation [HCSS SPR-3232]
- 2009-02-06 - PPa: Make removeColumn(String) compatible with history dataset [SPIRE-SPR-1211]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.


1.61. DividelfgmTask

Full Name:	herschel.spire.ia.dataset.DivideIfgmTask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import DivideIfgmTask

History

- 2009-04-21 - YFR: First Version
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions and fixed vague descriptions


1.62. EdpBlockContext

Full Name:	herschel.spire.ia.dataset.context.EdpBlockContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import EdpBlockContext

History

- 2007-01-08 - PPa: add Test Facility products
- 2007-03-07 - PPa: use EdpProduct
- 2007-03-21 - PPa: change name into EdpBlockContext. Refactoring.
- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-05-07 - PPa: add SOT and POT products.
- 2007-10-12 - PPa: Refactoring
- 2008-02-11 - PPa: NHKT is NhkTimeline and PDT & SDT are DetectorTimeline instances
- 2008-02-25 - PPa: remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-11-11 - PPa: add MCUET entry (SPR-1023)
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098].
- 2010-01-18 - PPa: Add "level" metadata keyword [SPIRE-2304].
- 2010-01-25 - PPa: Set type [SPIRE-2304].


1.63. EdpContext

Full Name:	herschel.spire.ia.dataset.context.EdpContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import EdpContext

History

- 2007-03-21 - PPa: Re-written from SG prototype
- 2007-10-12 - PPa: Refactoring
- 2006-02-25 - PPa: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-10-15 - PPa: Add getBbids methods (SCR-0942)
- 2008-10-20 - PPa: Fix getBbids(int) method
- 2009-03-06 - PPa: Remove get(int) method [SPIRE-SCR-1116]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098], add some useful methods.
- 2010-01-18 - PPa: Add "level" metadata keyword [SPIRE-2304].
- 2010-01-25 - PPa: Set type [SPIRE-2304].

1.64. EdpProduct

Full Name:	herschel.spire.ia.dataset.EdpProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import EdpProduct


Example

Example 1: creation:
<pre>\$ x=EdpProduct(description="This is my SPIRE data product",modelName="Flight")</pre>

History

- 2007-03-05 - PP: First version (PP)
- 2007-03-08 - PP: Add get/setSignalUnit methods
- 2007-03-12 - PP: Fix get/setSignalUnit methods
- 2007-03-13 - PP: change to get/setSignalUnits. Workaround for the FitsArchive
- 2007-05-28 - PP: remove the support to old product structure
- 2007-08-20 - PP: Migration to the new Unit API: remove setSignalUnits and deprecate getSignalUnits
- 2007-08-29 - PP: Complete migration to new Unit API.
- 2007-09-18 - PP: Fix copy constructor
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.


1.65. ElecCross

Full Name:	herschel.spire.ia.dataset.ElecCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ElecCross
Category:	Calibration product

History

- 2008-02-04 - PP: First version
- 2008-02-25 - PP: Use new makeTable method
- 2008-04-18 - RG: debug
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-05-23 - RG: add a set method SCR-1416
- setCrossTalk(String channel1, String channel2, double data)
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-01 - PPa: Remove usage of deprecated dataset methods.
- 2010-02-05 - YFR: [SPIRE-2255] Fixed typo 'channles'
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.66. elecCross

Full Name:	herschel.spire.ia.pipeline.common.eleccross.ElecCrossCorrectionTask
Alias:	elecCross
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.eleccross import ElecCrossCorrectionTask
Category:	task

Description

Task to correct the electric crosstalk for the different bolometers.

It works for any bolometer (spectrometer or photometer), and uses a calibration matrix which stores the coefficient of the crosstalk. This effect is small (less than 1%), so any method is good.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
ElecCross table [Input, Mandatory, default=No default value]
DetectorTimeline output [OUTPUT, Optional, default=No default value]

API details

Properties


DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT: bolometer timelines to be corrected for electrical crosstalk
ElecCross table [Input, Mandatory, default=No default value]
SPIRE calibration product: contains the coefficients of the electrical crosstalk
DetectorTimeline output [OUTPUT, Optional, default=No default value]
PDT or SDT: bolometer timelines corrected for electrical crosstalk

History

- 21-January-2008 : R Gastaud creation
- 14-January-2009 RG rewritten to correct SPR-1118, now works when some detectors are missing in the detector timeline
- 28-January-2009: RG fix SPR 1082 ElecCrossTask should not propagate the fileName metadata
- 28-January-2009: RG fix SPR-1165 Use serialVersionUID=1L instead of @SuppressWarnings("serial")
- 2009-05-23 - RG: SPR-1415 SciVal-Jiggle:Electrical Crosstalk matrix should be transposed

- now use the method `crossTalk(String channel1, String channel2)` as `SpecOptCrossCorrectionTask`
- 18-January-2010: RG SPR Spire 2025 add the comment for the flag `electricalCrosstalkCorrection-Done`
- (I hope that the comment is as clear as the name??)

1.67. engConversion

Full Name:	herschel.spire.ia.pipeline.common.engdata.EngConversionTask
Alias:	engConversion
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import EngConversionTask
Category:	SPIRE/Pipeline/Common Task

Description

Engineering Conversion Task.

This task implement the pipeline stage between level 0 and level 0.5.

Its purpose is to: 1) rearrange input raw Level 0 products into EDP products 2) convert raw values in engineering data, i.e. voltages, currents, etc. 3) re-process the sample time.

Examples

Example 1: Normal usage of engConversion in SPIRE pipelines:

```
HIPE>> level0_5 = engConversion(level0=obs.level0,cal=obs.calibration)
HIPE>> obs.level0_5=level0_5
```

Example 2: Use engConversion with temporary storage:

```
HIPE>>
tmppool=TemporalPool.createTmpPool("tempPool",TemporalPool.CloseMode.DELETE_ON_CLOSE)
HIPE>> ProductSink.getInstance().productStorage=ProductStorage(tmppool)
HIPE>> level0_5 =
engConversion(level0=obs.level0,cal=obs.calibration,tempStorage=True)
HIPE>> obs.level0_5=level0_5
```

Example 3: Run engConversion on the level 0 products of an observation context with a special Bolometer Parameter table

```
HIPE>> bolPar=FitsArchive().load("MyBolPar.fits")
HIPE>> level0_5 =
engConversion(level0=obs.level0,cal=obs.calibration,bolPar=bolPar)
```

API Summary

Properties
Level0Context level0 [INPUT, OPTIONAL, default=no default value.]
Level0BlockContext level0block [INPUT, OPTIONAL, default=No default value.]
Map rawData [INPUT, OPTIONAL, default=No default value.]
ResetHist resetHist [INPUT, OPTIONAL, default=No default value.]
ChanGain chanGain [INPUT, OPTIONAL, default=No default value.]
BolPar bolPar [INPUT, OPTIONAL, default=No default value.]
ChanMask chanMask [INPUT, OPTIONAL, default=No default value.]

Properties
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
ChanNomRes chanNomRes [INPUT, OPTIONAL, default=No default value.]
InstModeMask instModeMask [INPUT, OPTIONAL, default=No default value.]
SpireCalContex cal [INPUT, OPTIONAL, default=No default value.]
Boolean pixConv [INPUT, OPTIONAL, default=true.]
Boolean appRcRoll [INPUT, OPTIONAL, default=true.]
Boolean zeroOffset [INPUT, OPTIONAL, default=false.]
Boolean tempStorage [Input, OPTIONAL, default=false.]
Double offset [INPUT, OPTIONAL, default=0.079.]
Double drift [INPUT, OPTIONAL, default=-0.39.]
EdpContext converted [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties

Level0Context level0 [INPUT, OPTIONAL, default=no default value.]
Input Level-0 SPIRE Data Products wrapped in a Level0Context.
Level0BlockContext level0block [INPUT, OPTIONAL, default=No default value.]
Input Level-0 SPIRE Data Products wrapped in a Level0BlockContext.
Map rawData [INPUT, OPTIONAL, default=No default value.]
A map of Level-0 SPIRE Data Products.
ResetHist resetHist [INPUT, OPTIONAL, default=No default value.]
DPU reset history calibration product. Used if no HK product is provided in rawData.
ChanGain chanGain [INPUT, OPTIONAL, default=No default value.]
Channel Gain calibration product.
BolPar bolPar [INPUT, OPTIONAL, default=No default value.]
Bolometer parameters calibration product.
ChanMask chanMask [INPUT, OPTIONAL, default=No default value.]
Channel Mask calibration product.
ChanNum chanNum [INPUT, OPTIONAL, default=No default value.]
Channel Number Mapping calibration product.
ChanNomRes chanNomRes [INPUT, OPTIONAL, default=No default value.]
Channel Nominal Resistances calibration product.

InstModeMask instModeMask [INPUT, OPTIONAL, default=No default value.]
Instrument Mode Mask product.
SpireCalContex cal [INPUT, OPTIONAL, default=No default value.]
Spire Calibration Context.
Boolean pixConv [INPUT, OPTIONAL, default=true.]
Flag to use or not QLA conversion tables for detector values.
Boolean appRcRoll [INPUT, OPTIONAL, default=true.]
Flag to switch on/off the RC roll correction.
Boolean zeroOffset [INPUT, OPTIONAL, default=false.]
Flag for apply dummy 0 offsets for detector voltages.
Boolean tempStorage [Input, OPTIONAL, default=false.]
Flag to use or not ProductSink.
Double offset [INPUT, OPTIONAL, default=0.079.]
Time constant to correct the offset between detectors and pointing timing, in seconds.
Double drift [INPUT, OPTIONAL, default=-0.39.]
Time constant to correct the drift between detectors and pointing timing, in milliseconds per minute.
EdpContext converted [OUTPUT, OPTIONAL, default=No default value.]
Output level 0_5 products in a context.

History


- 2005-11-08 - PPa: First version, copied from EngDataProcess.
- 2006-02-23 - PPa: Nearly complete rewriting.
- 2006-03-08 - PPa: Add first obsMode computation.
- 2006-04-20 - PPa: Remove the use of deprecated getObsId() and getBbid() methods. Use enhanced loop.
- 2006-05-17 - PPa: Fix map usage bug. Code cleaning. Remove a SignatureException catching.
- 2006-05-22 - PPa: Use ArrayList and generics.
- 2006-05-23 - PPa: Fix housekeeping time conversion.
- 2006-05-31 - PPa: Add call of setResetOnExecute(true).
- 2006-06-22 - PPa: Use java.util.LinkedHashMap as inut and output. Refactoring.
- 2006-08-03 - PPa: Changed name from DataProcessing.java into EngConversion.java. Refactoring.
- 2006-08-17 - PPa: Refactoring. Fix conversion of test packets.

- 2006-09-14 - PPa: Compatible with optimized products.
- 2006-11-10 - PPa: Support POT and SOT products.
- 2006-11-21 - PPa: Add resetHist. Refactoring.
- 2007-01-08 - PPa: Add telemetry sorting task.
- 2007-03-06 - PPa: Use EdpProduct.
- 2007-03-13 - PPa: Use new product structure.
- 2007-03-29 - PPa: Use dedicated detector conversion. Add RC roll-off correction.
- 2007-05-09 - PPa: Add biasFreq metadata keyword to detector timelines. Use EdpContext as output. Refactoring. Version 0.7.
- 2007-05-29 - PPa: Adapt to the new DetectorTimeline structure. Refactoring. Version 0.8.
- 2007-07-27 - PPa: Migrate to new Unit API.
- 2007-07-30 - PPa: Fix offset setting bug: timeConversion has to be done before offset!
- 2007-08-06 - PPa: Set the unit of bias frequency.
- 2007-09-27 - PPa: Change name of some tasks.
- 2007-11-14 - PPa: Add task to remove dead channels.
- 2007-12-05 - PPa: Change name. Accept as input also Level0Context and Level0BlockContext.
- 2008-02-01 - PPa: Include AddNodRasterId. Change name of RemoveDeadChanTask.
- 2008-03-20 - PPa: Add the spire calibration context as input to simplify the syntax.
- 2008-03-25 - PPa: Use ChanNum to remove not connected channels in MaskBadChanTask.
- 2008-03-25 - PPa: Initialize parameter Tables using the start date of the first product (SPR-497).
- 2008-05-22 - PPa: Set the instMode. Remove some duplicated code.
- 2008-06-30 - PPa: Add usage of ProductSink (SPR-0745).
- 2008-07-01 - PPa: Change CalcJfetVoltTask usage (should fix SPR-0697).
- 2008-07-02 - PPa: Add ChanNomRes. BiasFreq now set by the CalcJfetVoltTask.
- 2008-07-04 - PPa: Use new TimeConvReordTask and new logic for time conversion.
- 2008-07-17 - PPa: Change name in constructor. Code cleaning.
- 2008-07-18 - PPa: Make order equal to PDD.
- 2008-07-21 - PPa: Use new AddPointingParamTask.
- 2008-07-22 - PPa: Set instMode in all timelines.
- 2008-07-23 - PPa: Unload product references when using the sink.
- 2008-08-29 - PPa: Partial implementation of SCR-0834. Now the BolPar is OPTIONAL for CalcRmsVoltResTask.

- 2008-09-25 - PPa: Make EngConversion more robust respect to missing telemetry (SPR-0860).
- 2008-09-25 - PPa: Some refactoring. Add controls on interruptions.
- 2008-10-14 - PPa: Fix crash in case of missing NHK.
- 2008-11-01 - PPa: Add "appRcRoll" parameter to switch off RC roll correction (SCR-0834).
- 2008-11-01 - PPa: Move MaskBadChanTask before DetFlaggerTask (SCR-0934). Some small refactoring.
- 2008-11-01 - PPa: Add few parameter descriptions.
- 2008-11-14 - PPa: Add all parameter descriptions (SPR-0985).
- 2009-01-07 - PPa: Add CalcBathTempTask [SCR-0874].
- 2009-01-28 - PPa: Fix history format and add serialVersionUID.
- 2009-02-23 - PPa: Fix how blocks are attached to the EdpContext [SPIRE SPR-1252].
- 2009-03-10 - PPa: Copy metadata from input Level0Context into output EdpContext [SPIRE SPR-1259].
- 2009-03-20 - PPa: Add "biasMode" metadata in all detector timelines [SPIRE SCR-1310].
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-04-01 - PPa: Exclude type and description keyword from copy [SPIRE SPR-1331]. Set creator accordingly to policy.
- 2009-04-01 - PPa: Improve logging for "empty" blocks.
- 2009-05-07 - PPa: Add new flag for dummy offset [SCR-1249]. Implement usage of InstModeMask.
- 2009-05-11 - PPa: Pass ChanNum to DetFlaggerTask [SCR-1355].
- 2009-05-25 - PPa: Change Offset History parameter of CalcJfetVoltTask from "offset" to "offsetHist" [SCR-1424].
- 2009-05-27 - PPa: Use InstModeMask from calibration context, if not specified as input.
- 2009-05-29 - PPa: Use InstModeMask only for photometer data [SPR-1448]. Check also that the InstModeMask is present.
- 2009-06-08 - PPa: FormatConversion returns a DetectorTimeline when RPDT/RSdT is processed.
- 2009-06-11 - PPa: Set the progress [SPR-1503].
- 2009-06-22 - PPa: Set the biasMode metadata only if it was not set by the Level0Plugin [SCR-1310].
- 2009-07-22 - PPa: Assume that MCUENG packets have frames [SPR-1699].
- 2009-08-04 - PPa: Add the metadata parameter "level" to output products [SPIRE-1596].
- 2009-08-07 - PPa: Add "tempStorage" as future replacement of "useSink" [SPIRE-1646].
- 2009-08-25 - PPa: Remove useSink [SPIRE-1646].

- 2009-09-05 - PPa: Add processing of MCU [SPIRE-519]. Avoid unnecessary deep copies of products.
- 2009-09-29 - PPa: Read calibration products only when needed [SPIRE-1990].
- 2009-12-07 - PPa: Use MaskBadTmParamTask [SPIRE-2079].
- 2009-12-16 - PPa: Add drift and offset [SPIRE-2293 and 2292].
- 2009-12-16 - PPa: offset shall be passed also for HK data [SPIRE-2292].
- 2010-02-15 - PPa: Use new format for Level 0 products [SPIRE-2139].
- 2010-02-18 - PPa: Support new format of level 0 products stored in a Level0Context [SPIRE-2139].
- 2010-02-19 - PPa: Use Level0Context as input [SPIRE-2139].
- 2010-02-24 - PPa: Update offset value [SPIRE-2412].
- 2010-04-07 - PPa: Fix metadata propagation into level 0.5 context [SPIRE-2486].
- 2010-05-10 - PPa: Set creator of level 0.5 context [SPIRE-2520].
- 2010-05-11 - PPa: Fix metadata propagation.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.68. EngConvOdPlugin

Full Name:	herschel.spire.ia.pipeline.pg.EngConvOdPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import EngConvOdPlugin

Description


Engineering Conversion Operational Day Plugin.

This class implements the Product Generator plugin for the generation of level 0.5 products of the OpDayContext. It is used by the SPG Operational Day processing stage. It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-10-16 - PPa: First experimental version.
- 2007-10-25 - PPa: Complete version
- 2007-11-21 - PPa: Add resteHist in conversion (E2E test fix)
- 2007-11-22 - PPa: Catch exception from performing engineering conversion task.
- 2007-12-13 - PPa: Use new EngConversionTask
- 2008-01-17 - PPa: Move to new OpDayPlugin interface
- 2009-06-01 - PPa: Refactoring
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]

1.69. eventLogger

Full Name:	herschel.spire.ia.pipeline.common.rawdata.EventLoggerTask
Alias:	eventLogger
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import EventLoggerTask
Category:	SPIRE/Pipeline/Common Task

Description

Event logger task.

Task to extract and log in a product events occurred in the given range.

Note: this task can be used only for ICC members with access to telemetry data. It is not intended for general users.

API Summary

Properties
Date startDate [INPUT, OPTIONAL, default=No default value]
Date endDate [INPUT, OPTIONAL, default=No default value]
Long obsid [INPUT, OPTIONAL, default=No default value]
Long bbid [INPUT, OPTIONAL, default=No default value]
Long bbfulltype [INPUT, OPTIONAL, default=No default value]
String dbName [INPUT, OPTIONAL, default=No default value]
SpireProduct events [OUTPUT, OPTIONAL, default=No default value]

API details

Properties

Date startDate [INPUT, OPTIONAL, default=No default value]	Start date of the time range of observation
Date endDate [INPUT, OPTIONAL, default=No default value]	End date of the time range of observation
Long obsid [INPUT, OPTIONAL, default=No default value]	Observation ID (OBSID)
Long bbid [INPUT, OPTIONAL, default=No default value]	Building Block ID (BBID)
Long bbfulltype [INPUT, OPTIONAL, default=No default value]	Building Block Full Type (BBFULLTYPE)

<code>String dbName [INPUT, OPTIONAL, default=No default value]</code>
--

Database name. It is used to override the configuration property and access a DB that differ from the one specified i configuration files.
--


<code>SpireProduct events [OUTPUT, OPTIONAL, default=No default value]</code>

Event Log product

History

- 2005-06-30 - PPa: First version.
- 2005-07-06 - PPa: More developement. Still not complete.
- 2005-07-07 - PPa: Fix errors.
- 2005-08-03 - PPa: Complete development.
- 2005-08-17 - PPa: Add super.preamble().
- 2005-11-16 - PPa: Use "tmPack" instead of "dfFlag". Add logFailed() method. Better handling of exceptions. Add documentation. Add "dbName"
- 2005-05-31 - PPa: Add call of setResetOnExecute(true).
- 2006-08-03 - PPa: Removed preamble.
- 2006-08-11 - PPa: Refactoring. The output is now a SpireProduct.
- 2006-09-13 - PPa: DbQueryMaker now returns a ProductReader instead of ProductStream.
- 2007-07-27 - PPa: Migrate to new Unit API
- 2007-09-19 - PPa: Change package.
- 2009-05-19 - PPa: remove setResetOnExecute, add serialVersionUID
- 2010-05-12 - PPa: Improve URM [SPIRE-2262]. Set the creator.

1.70. Ext

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.Ext
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import Ext
Category:	class


Description

This class gives a representation of successive maxima as a chained structure

History

- 2006-04-24 - CO: first java version
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag


1.71. ExtList

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.ExtList
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import ExtList

History

- 2006-03-21 - CO: first java version
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag


1.72. ExtRep

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.ExtRep
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import ExtRep

History

- 2006-03-13 - CO: first java version
- 2007-03-07 - CO: interpolate maxima line coordinate in computeExtList
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching -> herschel.spire.ia.pipeline.common.deglitch
- 2008-04-29 - DB: correction warning Trevor Fulton version build 814
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag

1.73. FactorFinder

Full Name:	herschel.spire.ia.pipeline.spec.ft.util.FactorFinder
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.ft.util import FactorFinder
Category:	SPIRE/Pipeline/Spectrometer Task

Description


FactorFinder

Used to determine prime factors of integers, and to determine ideal zeropad for a given interferogram length. This class has been changed back and forth between finding the next 2^n , and zeropadding to the next "nice" number. The Herschel FFT requires a 2^n number, while JNT operates well on nice factors. Also, the next "nice" number is a threshold which has not been completely defined. Currently, `findBestSize()` just returns the next 2^n .

History

- 2005-06-06 TRF: First implementation.
- 2007-09-04 TRF: Migrated to `herschel.spire.ia.pipeline.spec.ft.util`
- 2010-02-26 TRF: [SPIRE-2279] Fixed Javadoc and URM issues.
- 2010-05-18 TRF: [SPIRE-2279] Fixed some Javadoc and URM issues.

1.74. FlatFielderTask

Full Name:	herschel.spire.ia.pipeline.common.flatfield.FlatFielderTask
Alias:	FlatFielderTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.flatfield import FlatFielderTask
Category:	task

Description

This is a prototype of the correction task

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
FlatField table [INPUT, Mandatory, default=no default value]
DetectorTimeline corrected [OUTPUT, Optional, default=No default value]

API details

Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT: photometer or spectrometer bolometer timelines to be corrected
FlatField table [INPUT, Mandatory, default=no default value]
calibration product, table of flatfields
DetectorTimeline corrected [OUTPUT, Optional, default=No default value]
PDT or SDT: photometer or spectrometer bolometer timelines corrected for flatfield

History

- 14 Feb 2006 : R Gastaud
- 15 March 2007 RG use getChannel methods
- RG 5 June 2009 remove setResetOnExecute, setHistory(true);
- add package.html, correct spr 1099, remove filename from meta, add creator


1.75. FlatField

Full Name:	herschel.spire.ia.dataset.FlatField
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import FlatField
Category:	Calibration product

History

- 2006-08-24 - PP: First version
- 2008-02-25 - PP: Extends ChannelCalibTable
- 2008-11-27 - TRF: Fixed javadoc and compiler warnings [SPIRE SPR-0684]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.76. formatConversion

Full Name:	herschel.spire.ia.pipeline.common.engdata.FormatConversionTask
Alias:	formatConversion
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import FormatConversionTask
Category:	SPIRE/Pipeline/Common Task

Description

Format Conversion Task.

This task is used by the EngConversionTask to convert the format of raw (Level-0) data products into the Level 0.5 product formats.

API Summary

Properties
BuildingBlockProduct rawData [INPUT, MANDATORY, default=No default value.]
EdpProduct converted [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


BuildingBlockProduct rawData [INPUT, MANDATORY, default=No default value.]
Input raw SPIRE Data Product to be converted.
EdpProduct converted [OUTPUT, OPTIONAL, default=No default value.]
Converted SPIRE Data Product.

History

- 2005-12-31 - PPa: First draft version.
- 2006-01-06 - PPa: First completed version.
- 2006-01-09 - PPa: Make the output format more consistent with the EDP product formats.
- 2006-02-21 - PPa: Use BuildingBlockProduct.
- 2006-05-31 - PPa: Add call of setResetOnExecute(true).
- 2006-06-15 - PPa: Do not create a mask column for HK products.
- 2006-08-11 - PPa: Small refactoring.
- 2007-03-09 - PPa: "source" now is a level 0 product metadata keyword and not in the table. Remove logger.

- 2007-03-13 - PPa: Change output product structure.
- 2007-03-29 - PPa: Use EdpProduct as output.
- 2007-10-03 - PPa: Change name.
- 2007-10-04 - PPa: Fix bug.
- 2007-10-25 - PPa: Workaround for versioning info.
- 2008-07-18 - PPa: Remove time table.
- 2008-11-14 - PPa: Propagate the seqCount column [SPR-1005].
- 2008-12-29 - PPa: Add check on the structure of the input product [SPR-0980].
- 2008-12-29 - PPa: Remove check on version track. Add parameter descriptions (SPR-0985). Add more comments.
- 2009-01-28 - PPa: Fix history format and add serialVersionUID.
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-04-23 - PPa: Add check of units [SPR-0981].
- 2009-06-08 - PPa: Return a DetectorTimeline when the output is a PDT or a SDT.
- 2009-08-04 - PPa: Use propagateMeta().
- 2009-09-05 - PPa: Fix class of output products.
- 2009-12-15 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.77. FourierTransformTask

Full Name:	herschel.spire.ia.pipeline.spec.ft.FourierTransformTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.ft import FourierTransformTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to apply the Fourier Transform to create a set of spectra from a set of interferograms.

The purpose of the Fourier Transform module is to transform the set of interferograms from a SPIRE spectrometer observation into a set of spectra. This processing module is capable of transforming both double-sided and single-sided interferograms.

Task Execution Steps (from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

Double-sided Transform. For the double-sided transform, each interferogram in the SDI is examined and only the double-sided portion of the interferogram is used to compute the resultant spectrum. The resultant spectra will contain both real and imaginary components.

In this case, the discrete fourier transform that is used to compute the spectral components takes the form shown in.

Single-sided Transform. In the case of the single-sided transform, only those interferogram samples to one side of the position of zero path difference are considered. The spectra that result from the single-sided transform therefore contain only real components.

The discrete fourier transform that is used to compute the spectral components for single-sided interferograms takes the form shown in.

Wavenumber Grid. For both the single-sided and double-sided transforms the wavenumber grid onto which the spectrum is registered is calculated based on the interferogram sampling rate ($\Delta(\text{OPD})$) and on the maximum OPD displacement from the position of ZPD, L .

The Nyquist frequency (ω_{Nyquist}), the maximum independent frequency in the output spectrum, is given by:

$$\omega_{\text{Nyquist}} = 1/(2 * \Delta(\text{OPD}))$$

The spacing between independent spectral samples ($\Delta(\omega)$) is given by:

$$\Delta(\omega) = 1/(2 * L)$$

The spacing between spectral samples can be modified by padding the interferogram with zeroes. This procedure does not add any information to the spectrum but allows for an easier comparison between observations. In this case, a zero-padded interferogram ($V_{\text{ZP-i}}$) is given by:

$$V_{\text{ZP-i}}(x) = V(x) \{0 \leq x \leq L\}, 0 \{L < x \leq L_{\text{ZP}}\}$$

The corresponding spectral sampling interval is given by:

$$\Delta(\omega_{\text{ZP}}) = 1/(2 * L_{\text{ZP}})$$

Standard Padding. By default, this task in single-sided mode will pad the incoming interferograms to a length such that the output spectra will have a standard spectral sampling interval. This standard spectral sampling interval is based on the commanded spectral resolution (either low-resolution, medium-resolution, or high-resolution). These standard spectral sampling intervals are given below:

1. **Low Resolution:** 0.25 cm⁻¹
2. **Medium Resolution:** 0.05 cm⁻¹
3. **high Resolution:** 0.01 cm⁻¹

2ⁿ Padding. Another option available is to pad the incoming interferograms such that the number of samples is equal to a power of two. This increases the precision and efficiency of the FFT, but does result in spectral sampling intervals that are not nice round numbers as with standard padding.

Examples

Example 1: Double-sided transform, no padding

```
dsds = fourierTransform(sdi=presdi, ftType="prePhaseCorr", zeroPad="None")
```

Example 2: Single-sided transform, standard padding

```
ssds = fourierTransform(sdi=sdi, ftType="postPhaseCorr", zeroPad="standard")
```

Example 3: Single-sided transform, pad to next power of 2

```
ssds = fourierTransform(sdi=sdi, ftType="postPhaseCorr", zeroPad="2**n")
```

API Summary

Properties
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]
String ftType [INPUT, MANDATORY, default=No default value]
String zeroPad [INPUT, OPTIONAL, default=standard]
SpectrometerDetectorSpectrum sds [OUTPUT, MANDATORY, default=No default value]

API details

Properties

SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]
Input SDI product
String ftType [INPUT, MANDATORY, default=No default value]
Set this keyword 'postPhaseCorr' to perform singlesided transformation (using data from OPD >=0 and producing real output flux) or to 'prePhaseCorr' to perform doublesided transformation (using data from OPDs symmetric about ZPD and producing Complex flux).
String zeroPad [INPUT, OPTIONAL, default=standard]
Set this keyword to the type of zero-padding desired. Accepted values are: standard, for AOT standard padding (default); None, for no zero-padding; and 2**n, for padding to the next power of 2

<code>SpectrometerDetectorSpectrum sds [OUTPUT, MANDATORY, default=No default value]</code>

Output SDS product

History


- 2005-02-10 - AR: Removed SDS creation code from constructor and placed in new method for overriding.
- 2005-02-10 - AR: Removed zero padding.
- 2005-02-10 - AR: Added runFFT() method.
- 2005-02-14 - AR: Added option to zeropad single sided ifgm to the next 2^n
- 2005-02-28 - AR: Added algorithm which uses IfgmZeropadder and FactorFinder classes to zeropads interferograms up to next reasonable number.
- 2005-02-28 - AR: Replaced the herschel.ia.numeric.function.FFT with the JNT FFT. JNT FFT is currently being stored in herschel.spire.ia.pipeline.spec.ft. Eventually the Herschel FFT should be replaced by JNT as it is much faster.
- 2005-06-06 - AR: Currently zeropads to 2^n (IfgmZeropadder, FactorFinder) and uses the Herschel FFT. Now supports double and floating point precision SDIs. One TODO is to implement double and floating point precision Complex1ds in the SDS product.
- 2005-06-07 - AR: Now for a single-sided FFT of floating precision numbers, does a real-to-real transform using JNT.
- 2005-06-20 - TRF: herschel.ia.numeric.function no longer exists as per HCSS 673. Shift is now in herschel.ia.numeric.toolbox.basic,
- 2006-01-30 - TRF: FFT_PACK now used to compute forward transform.
- 2006-03-07 - TG: Removed the execute method since the call to super.execute() is done automatically by java.
- 2006-05-18 - TRF: Removed the need for a checked exception in the postamble as the result is now set with setDefaultValue().
- 2006-10-04 - TRF: Added a check to make sure that SS Ifgm contains an odd number of points as required by FFTPACK.
- 2006-12-14 - TRF: Removing preambles/postambes, added execute() method
- 2006-12-15 - TRF: Set `_sds` to null at the end of execute to conserve memory.
- 2006-12-15 - TRF: Removed setSds() and setSdsMeta() methods.
- 2007-03-27 - TRF: Units now set in columns of output sdi/sds.
- 2007-03-27 - TRF: Deprecated overloaded constructor
- 2007-03-28 - TRF: Changed a few logging levels
- 2007-05-31 - AH: Class placed in independent package. JParamaters updated.
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-10-02 - ZW: Added super() in constructor

- 2007-10-02 - ZW: Re-organized variables in execute().
- 2007-11-13 - TRF: Changed FFT package import from herschel.spire.ia.pipeline.spec.ft.util to herschel.ia.numeric.toolbox.xform.util
- 2007-11-13 - TRF: Fixed metaData (type, description, creator) in output SDS.
- 2008-01-07 - PK: Added optional scantype parameter describing the resolution of the input SDI (low, medium, high)
- 2008-01-09 - PK: Implemented zeropadding to a standard size specified by the scantype parameter.
- 2008-04-16 - PK: Added MetaData check for input sdi. Fixed sdi output MetaData.
- 2008-05-15 - ES: Removed need for scanType, this is now obtained via SDI's getCommandedResolution method
- 2008-05-15 - ES: Masks now propagate through FourierTransform rather than being removed
- 2008-05-15 - ES: Fixed single sided output flux so that it is now a Double1d rather than Complex1D
- 2008-05-16 - ES: Removed dependencies on ft.util. Truncation now occurs before zeropadding, rather than after.
- 2008-05-20 - ES: Added calculateResolution() method. This allows the actual resolution to be calculated in the correct location: after truncation, and before zeropadding. As a result, this value must be passed through some methods as an additional parameter. As well, setSpectrum() was modified so that this module now correctly determines spectral sampling interval depending on the commandedResolution of the input.
- 2008-07-10 - ES: Fixed some bugs with output size. "Deprecated" the "side" parameter and replaced it with "fftType". Both will work now, but "side" will produce log warnings.
- 2008-07-15 - TRF: Removed setCreator() step from MetaData modification to conform with Analysis in SPIRE SPR-0754.
- 2008-07-17 - TRF: [SPIRE SPR-0776] Fixed bug in checkMask(). Still waiting for the implementation of DEAD and NOISY in SpireMask().
- 2008-07-22 - ES: Reworked checkInput() to make use of the new SDI methods getUniqueDetectors and getScanNames(String). These methods replace getAnyIfgm()'s functionality. As a result, getAnyIfgm() has been deprecated. These changes were made as per SPR-0783.
- 2008-07-23 - ES: Fixed some javadoc issues.
- 2008-07-24 - ES: Fixed default value for the zeropad parameter- it is no longer read from Configuration.
- 2008-07-24 - ES: Fixed mask length bug in setSpectrum() by now using the same value as the other data set in that method.
- 2008-07-25 - ES: Changed the API. Parameters used to control zeropadding are now IA and ZP. The old parameters zeropad and nozeropad has been deprecated. Now, if IA is false, standard zeropadding will be performed. If IA is true and ZP is false, no zeropadding will be performed. If IA is false and ZP is true, 2ⁿ padding will be applied. The defaults for IA and ZP are both false.
- 2008-07-28 - ES: Fixed output vector length problems for all columns, most notably for the mask column. Fixes were achieved by making changes to the length calculations for ds cases where the RHS is longer than the LHS, and by modifying the mask vector creation to take into account the new length rather than simply copying the old mask.

- 2008-07-30 - ES: Fixed ss 2^n padding length, wavenumber, and nyquist issues. The factor by which we must apply to the $\Delta\sigma$ of the output is now calculated when determining the appropriate 2^n size in `prepareIfgm()`, and stored in the new global variable `_padded_factor`. Multiplying this with the actual resolution provides the desired results in the output wavenumber column. Moved the internal class `IfgmZeroPadder` into `ft.util`.
- 2008-08-13 - ES: Deprecated `checkMask()`, the input requirement it was designed to meet has been removed.
- 2008-08-13 - ES: Removed unnecessary metadata check in `checkMetaData()`.
- 2008-08-13 - ES: Changed wavenumber calculation in `setSpectrum` to produce some relevant data for any `commandedResolution` value, not only for "HR", "MR", "LR", or "".
- 2008-08-21 - TRF: [SPIRE SPR-0822] New class named `FourierTransformTask` to replace the misnamed `FourierTransform` class
- 2008-08-22 - TRF: Fixed typo in constructor
- 2008-08-23 - TRF: [SPIRE SCR-0721] Removed the deprecated task parameter `side` and `zeropad`
- 2008-08-27 - TRF: [SPIRE SPR-0851] Changed `setSpectrum` so that `spec` size and Nyquist determine `deltaSigma`
- 2008-08-29 - TRF: [SPIRE SPR-0852] Fixed bug in `prepareIfgm`. It now deals with ifgms whose LHS > RHS
- 2008-10-07 - TRF: [SPIRE SPR-0909] Added descriptions to `TaskParameters`. Made `jparameter` descriptions more descriptive.. Added a reference to the SPIRE Pipeline Description Document.
- 2008-10-10 - TRF: [SPIRE SPR-0936] Fixed bug in the manner in which `setProgress()` was being updated
- 2008-10-30 - TRF: [SPIRE SPR-1019] `execute()` now throws `InterruptedException`
- 2008-11-21 - TRF: Fixed serial, javadoc warnings
- 2008-12-02 - TRF: [HCSS SPR-3232] Fixed some `jtag urm` documentation
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2009-01-23 - TRF: [SPIRE SPR-1165] Added `serialVersionUID = 1L`, removed `SuppressWarnings`
- 2009-01-23 - TRF: [SPIRE SPR-1098] Added `throws InterruptedException` to javadoc tag to `execute()`
- 2009-01-23 - TRF: [SPIRE SPR-1162] Removed deprecated `nozerpad` `TaskParameter`, changed "results" to "sds"
- 2009-02-04 - TRF: [SPIRE SPR-1204] `keySet().iterator() -> getScanNames()`
- 2009-03-10 - YFR: [SPIRE SPR-1192] Moved `isDoubleSided` check from outside to inside double-sided implementation. So it checks `side` parameter first. If the parameter is set to be double-sided, it will call `isDoubleSided` method to check if the data is symmetric to determine if it should take the short path.
- 2009-03-17 - TRF: [SPIRE SPR-1062] `SpireInterferogram1d` metadata now propagated to `SpireSpectrum1d` via overloaded constructor.
- 2009-05-06 - TRF: [SPIRE SCR-1343] `pixel->channel`

- 2009-06-12 - TRF: [SPIRE SPR-1507] Cast sdi as a Product in propagateMeta()
- 2009-06-12 - TRF: [SPIRE SPR-1511] Task now sets Creator as per SPIRE Pipeline Policy
- 2009-06-14 - TRF: [SPIRE SPR-1181] INFO message now has scan numbers increasing
- 2009-07-20 - YFR: [SPIRE-1690] Removed deprecated methods, replaced with equivalent methods
- 2009-07-31 - TRF: [SPIRE-1690] Removed deprecated methods
- 2009-07-31 - TRF: [SPIRE-1657] Removed deprecated methods
- 2009-09-12 - TRF: [SPIRE-1657] Accepted values of ftType are now "prePhaseCorr" and "postPhaseCorr"
- 2009-09-12 - TRF: [SPIRE-1657] IA and ZP TaskParameters replaced with a zeroPad TaskParameter
- 2009-09-12 - TRF: [SPIRE-1657] Task now checks the sdi's apodType and phaseCorrApplied metadata for compliance.
- 2009-10-16 - TRF: [SPIRE-2017] Call the modified getStandardZeroPaddedSize() method which takes the Nyquist frequency into account
- 2009-11-17 - YFR: [SPIRE-2125] Added implementation to round up wave number grid
- 2010-02-26 TRF: [SPIRE-2279] Fixed Javadoc and URM issues.
- 2010-02-26 TRF: Removed usage of deprecated AbstractComposite() methods and various unused methods/variables.
- 2010-05-18 TRF: [SPIRE-2279] Fixed URM issues.

1.78. GaussianApodizeFunction

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.GaussianApodizeFunction
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import GaussianApodizeFunction

Description


Implementation of the Gaussian apodization function (width equivalent to 1.9 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2009-08-31 - YFR: [SPIRE SPR-1066] Gaussian apodization now tapers to 0.01 at the extrema
- 2009-09-13 - YFR: [SPIRE SPR-1191] Changed param name
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.79. GenericSpireDataFrameImpl

Full Name:	herschel.spire.ia.pipeline.common.rawdata.GenericSpireDataFrameImpl
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import GenericSpireDataFrameImpl

Description


Generic SPIRE data frame implementation.

This is a class used to hold data of a single telemetry frame. Note: this class is not intended to be used by normal users since only ICC members have access to the telemetry database.

History

- 2010-05-12 - Update: URM entry [SPIRE-2262]

1.80. GlitchFinder

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.deglitch.GlitchFinder
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm.deglitch import GlitchFinder
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Class to identify glitches in SPIRE Spectrometer Data using various glitch detection algorithms.

Task Execution Steps:

Check for sufficient scans in the input sdi.

Get the required parameters or default values for both scan directions.

Note: The values are computed while iterating through every detector based on the number of up and down scans in that detector. If any defaults are chosen, these may be subject to change if the scan sizes vary across detectors.

For each channel:

1. Perform glitch detection separately on up scans and down scans -- separate scans according to direction.

Start either : standard deviation glitch detection, mad glitch detection, windowed standard deviation glitch detection, windowed mad glitch detection, depending on the deglitchType parameter setting.

2. For standard deviation glitch detection:
 - a. Calculate the standard deviation across scans for each position element (i.e. the StdDev interferogram)
 - b. Calculate the mean across scans for each position element (i.e. the Mean interferogram)
 - c. For each position:
 - i. Get column of all the scans at position (ignoring samples with master bits and second level glitch bits set)
 - ii. For each indice in column:
 - A. A glitch is detected if $|\text{column.get(indice)} - \text{mean.get(position)}| > \text{stddev.get(position)} * \text{threshold}$
 - B. If a glitch is detected:
 - I. Flag a glitch in the interferogram at the given position, scan and channel using the SpireMask

Note: This detection method requires at least 3 scans in each scan direction to function properly.

For mad algorithm detection:

1. Calculate the mad across scans for each position element (i.e. the Mad interferogram)
2. Calculate the median across scans for each position element (i.e. the Median interferogram)

3. For each position:
 - a. Get column of all the scans at position (ignoring samples with master bits and second level glitch bits set)
 - b. For each indice in column:
 - i. A glitch is detected if $|\text{column.get(indice)} - \text{median.get(position)}| > \text{mad.get(position)} * \text{threshold} * 1.4826$
 - ii. If a glitch is detected:
 - A. Flag a glitch in the interferogram at the given position, scan and channel using the SpireMask

Note: This detection method requires at least 3 scans in each scan direction to function properly.

For windowed standard deviation glitch detection:

1. Calculate the standard deviation for each position element across the scans (i.e. compute the StdDev interferogram)
2. Calculate the mean of the standard deviation array of a window, centered about each position element
3. Calculate the standard deviation of the standard deviation array of a window, centered about each position element
4. For position elements where the window extends beyond the standard dev. array, use the closest element
5. For each position:
 - a. If $\text{stddevArray.get(pos)} - \text{meanWindowedArray.get(pos)} > \text{stddevWindowedArr.get(pos)} * \text{threshold}$ (i.e. a glitch is detected):
 - i. Find the scan with the glitch by finding the scan with the maximum deviation from the median at pos
 - ii. Flag the glitch by setting the GLITCH_SECOND_LEVEL, GLITCH_SECOND_LEVEL_UNCCOR (SpireMask) in the mask column
 - iii. Remove the glitched scan from consideration
 - iv. While there are enough non-glitched scans and there is a glitch at this position (using same criteria as above):
 - A. Recompute the StdDevInterferogram, meanWindowedArray, and stddevWindowedArray about the pos
 - B. Find the scan with the glitch (using same criteria as above)
 - C. Flag the glitch
 - D. Remove glitched scan from consideration

Note: This detection method requires at least 3 scans in each scan direction to function properly.

For windowed mad glitch detection:

1. Calculate the standard deviation for each position element across the scans (i.e. compute the StdDev interferogram)

2. Calculate the median of the standard deviation array of a window, centered about each position element
3. Calculate the mad of the standard deviation array of a window, centered about each position element
4. For position elements where the window extends beyond the standard dev. array, use the closest element
5. For each position:
 - a. If $\text{stddevArray.get(pos)} - \text{medianWindowedArray.get(pos)} > \text{madWindowedArr.get(pos)} * \text{threshold}$ (i.e. a glitch is detected):
 - b.

Note: This detection method requires at least 3 scans in each scan direction to function properly.

API Summary

Jython Syntax
Flags glitches from interferograms by comparing the ifgm signals at each flagged point for a given channel across the scans within the observation that are of the same direction. Glitches are detected using either standard deviation or median absolute deviation techniques.
Properties
<code>SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]</code>
<code>String deglitchType [INPUT, OPTIONAL, default="MAD_WINDOW"]</code>
<code>Double thresholdFactor [INPUT, OPTIONAL, default=No default value]</code>
<code>Integer windowSize [INPUT, OPTIONAL, default=No default value]</code>
<code>SpectrometerDetectorInterferogram outsdI [OUTPUT, MANDATORY, default=No default value]</code>

API details

Properties

<code>SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]</code>
Input Spectrometer Detector Interferogram product to be deglitched.
<code>String deglitchType [INPUT, OPTIONAL, default="MAD_WINDOW"]</code>
The deglitching method used, can be either "STD", "MAD", "STD_WINDOW", "MAD_WINDOW"
<code>Double thresholdFactor [INPUT, OPTIONAL, default=No default value]</code>
The factor that defines what is considered to be a glitch. The default thresholdFactor depends on the deglitchType and the number of scans.

Integer <code>windowSize</code> [INPUT, OPTIONAL, default=No default value]
--

The running window that will be used to look for outliers in the secondary statistics. The default <code>windowSize</code> depends on the <code>deglitchType</code> .

SpectrometerDetectorInterferogram <code>outsdi</code> [OUTPUT, MANDATORY, default=No default value]
--


The resultant deglitched SDI.

History

- 2007-04-20 - TRF: Reworked old RemoveGlitches Task.
- 2007-05-31 - AH: Class placed in independent package. JParameters updated.
- 2007-10-02 - ZW: Made `getDoublesColumnFromArray()`, `calculateStdDevAcrossSignals()`, `getScansByDirection()`, `getSignals()` to static methods.
- 2007-10-02 - ZW: Re-organized variables in `execute()`.
- 2007-10-22 - TRF: Fixed Javadoc error.
- 2007-10-24 - TRF: Changed behavior of `checkForSufficientScans()`. Now returns gracefully if # of scans too few for `deglitchType` chosen.
- 2008-04-28 - PK: Specified `ArrayList` type to removed unchecked warnings.
- 2008-08-26 - TRF: [SPIRE SPR-0848] Modified call to `super()` in constructor to `super("GlitchFinder")` as `TaskTools` was generating errors
- 2008-11-13 - TRF: [SPIRE SCR-0811] Improved user interaction
- 2008-11-13 - TRF: [SPIRE SPR-1031] Task is now interruptible
- 2008-11-13 - TRF: [SPIRE SPR-1030] Task now implements the `SpireMask` Interface (`SpireMask.GLITCH_SECOND_LEVEL` and `SpireMask.GLITCH_SECOND_LEVEL_UNCORR`). `SpireMask` also used to update `MetaData` in output product.
- 2008-12-02 - TRF: [HCSS SPR-3232] Changed `jhistory` entries based on note in
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2008-12-15 - JL: [SPIRE SCR-1095] Refactored `execute` method. Modified flag glitches.
- 2008-12-15 - JL: [SPIRE SCR-1095] Implemented mad clipping, stddev clipping, windowed mad and windowed stddev.
- 2008-12-15 - JL: [SPIRE SCR-1095] Updated documentation. Moved the update meta data for `SpireMask` to the end of `execute`.
- 2008-12-15 - JL: [SPIRE SCR-1095] Added accessor methods for default/minimum parameter values.
- 2008-12-15 - JL: [SPIRE SCR-1095] Implemented hooks to set sdi QC meta data.
- 2008-12-15 - JL: [SPIRE SCR-1095] Added hooks to deal with finding the glitched scan in 2 scans.
- 2009-03-17 - TRF: Modified the call to `getLogger()` so that the messages printed by this task could be controlled

- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-06-12 - TRF: [SPIRE SPR-1511] Task now sets Creator as per SPIRE Pipeline Policy
- 2009-10-25 - DMT: [SPIRE SCR-1364] Added quality control metadata collection
- 2009-11-16 - YFR: [SPIRE-xxxx] Fixed the bug that throws an IndexOutOfBoundsException when all scans are rejected by mask
- 2009-11-16 - YFR: [SPIRE-xxxx] Fixed deprecated method calls
- 2009-12-11 - DMT: [SPIRE SPR-2247] Fixed log messages to not spam "Not enough scan" if there are no scans in that direction
- 2009-12-11 - DMT: [SPIRE SPR-2044] For 2 scans for a direction neighbour deglitching method is now used by default
- 2010-03-01 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.
- 2010-04-07 - WT: [SPIRE-1364] getNumElements() no longer ignores non-nominal detectors
- 2010-05-18 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.

1.81. GlitchRemoval

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.deglitch.GlitchRemoval
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm.deglitch import GlitchRemoval
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Class to remove glitches that have been identified in SPIRE Spectrometer Data.

Task Execution Steps:

1. For each channel:
 - a. Find interferogram samples where GLITCH_SECOND_LEVEL bit has been set (See Spire-Mask)
 - b. Replace these samples with the average of the samples for that channel from the other scans (but same scan direction) at that sample position, ignoring scans where the GLITCH_SECOND_LEVEL or MASTER bit is set.
 - c. Unset the GLITCH_SECOND_LEVEL_UNCORR in the mask for the corrected sample (See SpireMask).

API Summary

Jython Syntax
Removes glitches from interferograms by comparing the ifgm signals at each flagged point for a given channel across the scans within the observation that are of the same direction.
Properties
<code>SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]</code>
<code>SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=No default value]</code>

API details


Properties

<code>SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]</code>
Input Spectrometer Detector Interferogram Product that has 2nd level glitches identified.
<code>SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=No default value]</code>
Corrected Spectrometer Detector Interferogram.

History

- 2007-04-20 - TRF: Reworked old RemoveGlitches Task.
- 2007-05-31 - AH: Class placed in independent package. JParamaters updated.
- 2007-07-31 - AH: Migrate to new Unit API
- 2007-09-26 - TRF: Removed conditional that checks for null input/output sdi since SignatureException will not allow this.
- 2008-08-26 - TRF: [SPIRE SPR-0848] Modified call to super() in constructor to super("GlitchRemoval") as TaskTools was generating errors
- 2008-11-13 - TRF: [SPIRE SPR-1031] Task is now interruptible
- 2008-11-13 - TRF: [SPIRE SPR-1030] Task now implements the SpireMask Interface (SpireMask.GLITCH_SECOND_LEVEL and SpireMask.GLITCH_SECOND_LEVEL_UNCORR). SpireMask also used to update MetaData in output product.
- 2008-11-13 - TRF: [SPIRE SCR-0811] Improved user interaction
- 2008-12-02 - TRF: [HCSS SPR-3232] Changed jhistory entries based on note in
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2008-12-15 - JL: [SPIRE SCR-1095] Added hooks to set sdi QC meta data
- 2009-02-04 - TRF: [SPIRE SPR-1200, 1204] Changed sdi.keySet() to sdi.getScanNames()
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-05-06 - TRF: [SPIRE SPR-1165] Removed SuppressWarnings annotation
- 2009-05-22 - TRF: Modified the call to getLogger() so that the messages printed by this task could be controlled
- 2009-06-12 - TRF: [SPIRE SPR-1511] Task now sets Creator as per SPIRE Pipeline Policy
- 2009-12-11 - DMT: [SPIRE SPR-1634] Logging messages were updated to better describe the glitch removal process
- 2010-05-18 - DMT: [SPIRE SPR-1634] Logging messages were updated to better describe the glitch removal process
- 2010-05-18 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.

1.82. HammingApodizeFunction

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.HammingApodizeFunction
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import HammingApodizeFunction

Description


Implementation of the Hamming apodization function.

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2009-09-13 - YFR: [SPIRE SPR-1191] Changed param name
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.83. HanningApodizeFunction

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.HanningApodizeFunction
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import HanningApodizeFunction

Description


Implementation of the Hanning apodization function.

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2009-09-13 - YFR: [SPIRE SPR-1191] Changed param name
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.84. HolderList

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.HolderList
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import HolderList
Category:	class

History

- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch@jhistory
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag


1.85. HolderRep

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.HolderRep
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import HolderRep

History

- 2006-03-23 - CO: first java version
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag

1.86. HousekeepOdPlugin

Full Name:	herschel.spire.ia.pipeline.pg.HousekeepOdPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import HousekeepOdPlugin

Description


Housekeeping Operational Day Plugin.

This class implements the Product Generator plugin for the generation of housekeeping products of the OpDayContext. It is used by the SPG Operational Day processing stage. It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2008-02-08 - PPa: First version.
- 2008-06-30 - PPa: More logging messages. Re-throw IO and general security exceptions.
- 2009-04-01 - PPa: fix CHKT bbTypeName metadata removal
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]


1.87. IfgmTask

Full Name:	herschel.spire.ia.dataset.IfgmTask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import IfgmTask

History

- 2009-04-21 - YFR: First Version
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions and fixed vague descriptions

1.88. IfgmZeropadder

Full Name:	herschel.spire.ia.pipeline.spec.ft.util.IfgmZeropadder
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.ft.util import IfgmZeropadder
Category:	SPIRE/Pipeline/Spectrometer Task

Description


IfgmZeropadder

Pads a non-butterflied interferogram with zeroes. Can handle either double-sided or single-sided interferograms. Uses the FactorFinder class to find the number of points to use. This class can also an interferogram such that the spectrum of the is of a specific size.

History

- 2007-09-04 TRF: Migrated to herschel.spire.ia.pipeline.spec.ft.util
- 2008-01-09 PK: Added zeropadding to size n
- 2008-01-25 PK: Moved information about standard sizes of ifgms to separate enum ScanResolution
- 2008-01-28 PK: Modified zeropadding to 2^n so that it uses the zeroPad to n method. Also changed zeroPad so that it does not change the argument array.
- 2008-02-12 PK: Simplified 2^n size logic in zeroPad(Numeric1dData ifgm, boolean doublesided)
- 2008-07-29 ES: Revamped. This is now split up into ds and ss methods, as was the case in the implementation within FourierTransform.
- 2009-10-16 TRF: [SPIRE-2017] Modified the getStandardZeroPaddedSize() method to take the Nyquist frequency into account
- 2010-02-26 TRF: [SPIRE-2279] Fixed Javadoc and URM issues. Removed a long-since deprecated method.


1.89. InstModeMask

Full Name:	herschel.spire.ia.dataset.InstModeMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import InstModeMask
Category:	Calibration product

History

- 2009-04-29 - PPa: First version
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added description for method 'getInstModes'
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.90. JiggleAverageTask

Full Name:	herschel.spire.ia.pipeline.phot.demodulation.JiggleAverageTask
Alias:	JiggleAverageTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.demodulation import JiggleAverageTask
Category:	task

Description

average the Demodulated Photometer Product for each jiggle positions

API Summary

Properties
DemodPhotProduct input [Input, MANDATORY, default=no default value]
DemodPhotProduct output [Output, MANDATORY, default=no default value]

API details


Properties

DemodPhotProduct input [Input, MANDATORY, default=no default value]
Mandatory, No default value Demodulated Detector Timeline SPIRE Data Product to be averaged
DemodPhotProduct output [Output, MANDATORY, default=no default value]
Mandatory, No default value Averaged Detector Timeline

History

- RG 7 september 2009 SPR: SPIRE-1919 JiggleAverageTask doesn't
- propagate mask information RG 23 october 2009 SPR: spire-2110
- management of NaN via nsample
- RG 20 April 2010 add parameter skipFirst SCR SPIRE-2488

1.91. JiggPhotProduct

Full Name:	herschel.spire.ia.dataset.JiggPhotProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import JiggPhotProduct
Category:	Pipeline/User Products


Example

Example 1: A Jython example
<pre>from herschel.spire.ia.dataset import * myProduct = JiggPhotProduct()</pre>

History

- 2006-01-01 - DR: First version
- 2006-07-01 - DR: Simplified product structure
- 2007-05-03 - PP: change class name from Jpp.java; move in dataset package; refactoring.
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-09-08 - PPa: Remove useless metadata, add "level" metadata. [SPIRE-1546]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed descriptions
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods

1.92. JiggPointSourceFitProduct

Full Name:	herschel.spire.ia.dataset.JiggPointSourceFitProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import JiggPointSourceFitProduct
Category:	Pipeline/User Products

Description

Jiggle Point Source Fit Product.

This class provides an empty Jiggle Point Source Fit Product.

Example


Example 1: A Jython example

```
from herschel.spire.ia.modules.userproducts import *
myProduct = JiggPointSourceFitProduct()
```

History

- 2006-01-01 - DR: First version
- 2006-07-01 - DR: Simplified product structure
- 2006-10-01 - DR: Removed array-level table datasets.
- 2006-10-01 - DR: Added constructor to create empty output dataset.
- 2007-03-01 - DR: Added some convenience methods
- 2007-05-03 - PP: change class name from Jpsfp.java; move in dataset package; small refactoring.
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2008-12-01 - TRF: Fixed jhelp urm documentation [HCSS SPR-3232]
- 2009-06-11 - PPa: Remove usage of deprecated addRaw method. Change members to transient.
- 2010-02-05 - YFR: [SPIRE-2255] Fixed descriptions
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods.

1.93. joinNhkTimelines

Full Name:	herschel.spire.ia.pipeline.common.util.JoinNhkTimelinesTask
Alias:	joinNhkTimelines
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.util import JoinNhkTimelinesTask
Category:	SPIRE/Pipeline/Common Task

Description

Task to join scan line NHK timelines with the NHK timelines of the leading and trailing turnaround building block in scan maps.

Examples

Example 1: How to use in a script or in HIPE command line.

```
my_output =
  joinNhkTimelines(nhkt=my_nhkt, trailing=my_trail, following=my_follow)
```

Example 2: How to get the CVS number version of the task.

```
print joinNhkTimelines.VERSION
```

API Summary

Properties
NhkTimeline nhkt [Input, Mandatory, default=No default value]
NhkTimeline leading [Input, Optional, default=No default value]
NhkTimeline trailing [Input, Optional, default=No default value]
NhkTimeline output [Output, Optional, default=No default value]

API details


Properties

NhkTimeline nhkt [Input, Mandatory, default=No default value]
Input (mandatory): NHK timeline of the scan line.
NhkTimeline leading [Input, Optional, default=No default value]
Input (optional): NHK timeline of the leading turnaround building block
NhkTimeline trailing [Input, Optional, default=No default value]
Input (optional): NHK timeline of the trailing turnaround building block
NhkTimeline output [Output, Optional, default=No default value]
Output (optional): joined NHK timeline

History

- 2009-08-28 - PPa: First version
- 2009-09-08 - PPa: Update javadoc. Correct metadata propagation. Don't throw exception when trailing and following are null.
- 2009-09-15 - PPa: Apply a reordering on sample time to fix problems with telemetry [SPIRE-1950]
- 2009-10-18 - PPa: Remove not unique sampleTime samples [SPIRE-2059]
- 2010-05-11 - GM: add VERSION to creator

1.94. joinPhotDetTimeline

Full Name:	herschel.spire.ia.pipeline.common.util.JoinPhotDetTimelinesTask
Alias:	joinPhotDetTimeline
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.util import JoinPhotDetTimelinesTask
Category:	SPIRE/Pipeline/Common Task

Description

Task to join scan line detector timelines (both photometer and spectrometer) with the detector timelines of the leading and trailing turnaround building block in scan maps.

Examples

Example 1: How to use in HIPE command line.

```
my_output =
joinPhotDetTimelines(pdt=my_pdt, trailing=my_trail, following=my_follow)
```

Example 2: How to get the CVS number version of the task.

```
print joinPhotDetTimelines.VERSION
```

API Summary

Properties
DetectorTimeline pdt [Input, Mandatory, default=No default value]
DetectorTimeline leading [Input, Optional, default=No default value]
DetectorTimeline trailing [Input, Optional, default=No default value]
DetectorTimeline output [Output, Optional, default=No default value]

API details

Properties

DetectorTimeline [pdt](#) [Input, Mandatory, default=No default value]

Input (mandatory): Detector Timeline of the scan line.

DetectorTimeline [leading](#) [Input, Optional, default=No default value]

Input (optional): Detector timeline of the leading turnaround building block.

DetectorTimeline [trailing](#) [Input, Optional, default=No default value]

Input (optional): Detector Timeline of the trailing turnaround building block.


<code>DetectorTimeline output [Output, Optional, default=No default value]</code>

Output (optional): joined Detector Timeline.
--

History

- 2009-09-01 - PPa: First version
- 2009-09-08 - PPa: Correct metadata propagation. Don't throw exception when trailing and following are null.
- 2010-05-11 - GM: Add version to creator


1.95. Level0BlockContext

Full Name:	herschel.spire.ia.dataset.context.Level0BlockContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import Level0BlockContext

History

- 2007-01-24 - PPa: Add Test Facility products
- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-04-28 - PPa: Fix description (SPR-0597)
- 2008-11-10 - PPa: Add entry for RMCUET (SPR-0992)
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]
- 2010-01-17 - PPa: make RULE private [SPIRE-2255]. Add "level" metadata keyword [SPIRE-2304].


1.96. Level0Context

Full Name:	herschel.spire.ia.dataset.context.Level0Context
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import Level0Context

History

- 2007-03-21 - PPa: Re-written form SG prototype.
- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-04-28 - PPa: Add constructor with modelName; use addProduct to fix start/endDate (SPR-0597).
- 2008-09-23 - PPa: implement SPR-0611.
- 2009-03-06 - PPa: Remove get(int) method [SPIRE-SCR-1116]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]
- 2010-01-18 - PPa: Add "level" metadata keyword [SPIRE-2304].
- 2010-02-17 - PPa: Support new level 0 product format [SPIRE-2139]

1.97. Level0OdPlugin

Full Name:	herschel.spire.ia.pipeline.pg.Level0OdPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import Level0OdPlugin

Description


Level 0 Operational Day Plugin.

This class implements the Product Generator plugin for the generation of level 0 products of the Op-DayContext. It is used by the SPG Operational Day processing stage. It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-10-16 - PPa: First experimental version.
- 2007-11-21 - PP: Do not extract telemetry packets between one BB and the other in the same observation.
- 2007-12-04 - PPa: Use ProductStorage provided by ProcessEnvironment instead of a sink.
- 2008-01-17 - PPa: Move to new OpDayPlugin interface
- 2008-02-08 - PPa: Comment-out extraction of data from between observations.
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]

1.98. Level0Plugin

Full Name:	herschel.spire.ia.pipeline.pg.Level0Plugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import Level0Plugin

Description

Level 0 Plugin.


This class implements the Product Generator plugin for the generation of the SPIRE level 0 products. It is used by the SPG during the preprocessing stage and by the SpireProprocessThread class (i.e. the obsExporter application). It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-04-03 - PPa: First version.
- 2007-04-05 - PPa: add some log
- 2007-09-18 - PPa: RawDataExtraction is now in rawdata package
- 2008-01-29 - PPa: Pass ObjectStore to RawDataExtractionTask. Use new ObservationContext API.
- 2008-04-29 - PPa: Use modelName when create the Level0Context (SPR-0597)
- 2008-06-27 - PPa: Pass the ResetHist to RawDataExtractionTask (SPR-0747)
- 2008-07-13 - PPa: Rethow exceptions (SPR-0758)
- 2008-07-18 - PPa: Hack for SPR-0777 and 0778.
- 2008-09-23 - PPa: Fix SPR-0869.
- 2008-10-06 - PPa: Propagate metadata from ObservationContext to level 0 products (SPR-0895)
- 2009-06-09 - PPa: Use the MissingTm product to check the drops [SCR-0940]
- 2009-06-22 - PPa: Look in the uplink product which is the biasMode [SCR-1310]
- 2009-08-02 - PPa: Check if the AuxiliaryContext is empty or not. [SPIRE-1767]
- 2009-08-10 - PPa: Optimize look on compulsory metadata and exclude the propagation of formatVersion.
- 2009-09-11 - SG: Use UplinkReader [SCR-1882]
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-02-05 - PPa: Allow possibility to run efficiently when not navigating.
- 2010-02-16 - PPa: Use new level 0 format [SPIRE-2139], improve URM [SPIRE-2266]
- 2010-02-17 - PPa: Fix SPIRE-2393. Handle correctly biasMode for new level 0 products.
- 2010-02-19 - PPa: Use a Level0Context class with new level 0 products [SPIRE-2139]

- 2010-03-01 - PPa: Get the commandedResolution from uplink and add it in the Observation-Context metadata [SPIRE-2415]. Improve URM.

1.99. Level1Context

Full Name:	herschel.spire.ia.dataset.context.Level1Context
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import Level1Context

History

- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-05-03 - PPa: change level 1 product classes names
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: remove serialVersionUID: Context doesn't implement Serializable anymore
- 2009-03-06 - PPa: Remove get(int) method [SPIRE-SCR-1116]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]. Remove setting of creator.
- 2010-01-18 - PPa: Add "level" metadata keyword [SPIRE-2304].


1.100. LinearRegression

Full Name:	herschel.spire.ia.pipeline.common.deglitch.util.LinearRegression
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.util import LinearRegression

History

- 2006-10-27 - CO: first java version
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching ->herschel.spire.ia.pipeline.common.deglitch
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag
- 2010-01-25 - DB: SPIRE-2257 update constructor description, and method description


1.101. LpfPar

Full Name:	herschel.spire.ia.dataset.LpfPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import LpfPar
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-10-23 - ZW: Deleted Constructor LpfPar(String Desc)
- 2007-10-23 - ZW: Changed getTable to an abstract method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-21 - TRF: Removed redundant cast in R1(), R2(), R3(), R4(), C1(), C2() [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fix some compiler warnings [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed descriptions and added associated tasks' links.

1.102. LpfResponseCorrectionTask

Full Name:	herschel.spire.ia.pipeline.phot.elecfiltcorr.LpfResponseCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.elecfiltcorr import LpfResponseCorrectionTask
Category:	herschel/spire/ia/pipeline/phot/elecfiltcorr

Description

This task takes a Photometer Detector Timeline (PDT) product as input

and removes the Electrical Filter transient reponse effect This task is part of the photometer pipeline, between Level 0.5 and Level 1.

API Summary

Properties
DetectorTimeline PDT_IN [Input, Mandatory, default=No Default value]
PhotLpfPar lpfPar [Input, Optional, default=No default value]
DetectorTimeline PDT_OUT [Output, Mandatory, default=No Default value]

Limitations

None

API details

Properties


DetectorTimeline PDT_IN [Input, Mandatory, default=No Default value]
PDT: Photometer timelines to be corrected
PhotLpfPar lpfPar [Input, Optional, default=No default value]
Low Pass Filter paramter table
DetectorTimeline PDT_OUT [Output, Mandatory, default=No Default value]
PDT: Photometer timelines corrected for Low Pass Filter delay"

History

- 18 September 08. (AP) First attempt, by translating working Jython code.
- 7 October (AP) 08 renamed task output parameter from "result" to "PDT_OUT"
- 18 Dec 08 (AP) Task removes "fileName" metadata from the output Product (Re: SPR-1080)
- 13 Jan 09.(AP) Fixed: Task was not propagating units of the columns in input table.

- 13 Jan 09 (AP) Added Message output ("Task complete")
- 13 Jan 09 (AP) Added default serialVersionUID
- 13 Jan 09 (AP) Added Javadoc entries for methods fftshift and linspace
- 13 Jan 09 (AP) [rtag 0_5] Changed fftshift input variable from Array to inputArray (and related instances)
- 06 Mar 09 (AP) Modified makeTransferFunction: Changed 1./L to 1./(L-1) for even_length arrays
- 06 Mar 09 (AP) Removed second instance import herschel.ia.task.*;
- 06 Mar 09 (AP) Removed unused 'TimeArray_Sec = TimeArray.subtract(TimeArray.get(0));'
- 08 Mar 09 (AP) Added Log message: Calculated values of LPF Transfer function coefficients
- 08 Mar 09 (AP) [rtag0_6] Updated to accept PhotLpfPar as optional input
- 14 Apr 09 (AP) Added progress output on console
- 14 Apr 09 (AP) [rtag 0_7] Added PDT.fits file to test dir and changed TESTPATH pointer accordingly (SCR-1150)
- 14 Apr 09 (AP) Fixed warning by Changing Arr=Arr.range(L-1); to Double1d Arr=Double1d.range(L-1); in method linspace
- 04 Jun 09 (AP) changed jparameter photLpfPar to lpfPar and 2 instances in code
- 23 Nov 09 (AP) added VERSION variable
- 23 Nov 09 (AP) SPIRE-2043 added creator metadata
- 23 Nov 09 (AP) SPIRE-2086 Task makes calculations in Double1d and at the end it converts to Float1d/Double1d depending on input
- 13 Feb 10 (AP) SPIRE-2269 FIXED: "No class description." & "The constructor description is wrong."
- 13 Feb 10 (AP) SPIRE-2269 PROPERLY FIXED: "The constructor description is wrong."
- 22 Feb 10 (AP) SPIRE-2389 Process only bolometers with Bolometer.isSeeing()
- 23 Feb 10 (AP) SPIRE-2409 Add setProgress() method
- 26 Feb 10 (AP) SPIRE-2389 revert back to processing all bolometers
- 10 May 10 (AP) SPIRE-2269 Added initials tou jhistory tags
- 10 May 10 (AP) SPIRE-2269 changed jcategory tag to include full path of task
- 10 May 10 (AP) SPIRE-2267 changed jlimitation from TBD to None
- 10 May 10 (AP) SPIRE-2267 changed jhelp tag to mention the location of the task in the pipeline.

1.103. MadScanMapperTask

Full Name:	herschel.spire.ia.pipeline.phot.scanmap.MadScanMapperTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.scanmap import MadScanMapperTask
Category:	Pipeline/Map Making

Description

This task takes a Photometer DetectorTimeline and produces a Spire Photometer Map Product by performing a maximum likelihood estimation of the map.

Example

Example 1: task = MadScanMapperTask()
<pre>mapProduct=task(timelineProduct)</pre>

API Summary

Properties
SpireListContext input [INPUT, Tod or TodBuffer; MANDATORY, default=No default value]
String ; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]
Double ; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]
Wcs ; OPTIONAL wcs [INPUT, , default=with the default resolution>]
Integer ; OPTIONAL doMap [INPUT, , default=no default value]
Boolean ; OPTIONAL storeTod [INPUT, , default=no default value]
SimpleImage ; No default output [OUTPUT, MANDATORY, default=no default value]
Double ; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]
Integer ; OPTIONAL doInterpolation [INPUT, , default=no default value]
Integer ; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]
ChanNoise ; OPTIONAL chanNoise [INPUT, , default=no default value]
Integer ; OPTIONAL correlationLength [INPUT, , default=no default value]
Double ; OPTIONAL maxIterationPcg [INPUT, , default=no default value]
Double ; OPTIONAL maxRelativeErrorPcg [INPUT, , default=no default value]
Invntt ; OPTIONAL invntt [INPUT, , default=no default value]
SimpleImage ; MANDATORY output [OUTPUT, No default value, default=no default value]

API details

Properties

SpireListContext input [INPUT, Tod or TodBuffer; MANDATORY, default=No default value]
SpireListContext of detector timelines, Tod or TodBuffer
String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]
Bolometer array to be processed ('PSW', 'PMW' or 'PLW')
Double; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]
Resolution of the output map in arcsec
Wcs; OPTIONAL wcs [INPUT, , default=with the default resolution>]
World Coordinate System for the output map
Integer; OPTIONAL doMap [INPUT, , default=no default value]
If set to False, the Tod product and the Invntt array will be created, but not the map
Boolean; OPTIONAL storeTod [INPUT, , default=no default value]
If set to true, the pre-processed Tod is made available with <code>todBuffer = mapper.getTodBuffer()</code>
SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]
Output map product
Double; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]
If set, reorder and merge the building blocks. The value is the maximum gap in seconds allowed for merging two adjacent building blocks
Integer; OPTIONAL doInterpolation [INPUT, , default=no default value]
If set to true, NaN values in the timeline are interpolated
Integer; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]
If set, remove polynomial fit from start to end of the timeline. The value specifies the polynomial order
ChanNoise; OPTIONAL chanNoise [INPUT, , default=no default value]
Calibration table of the detector noise power spectrum
Integer; OPTIONAL correlationLength [INPUT, , default=no default value]
Correlation length

Double; OPTIONAL maxIterationPcg [INPUT, , default=no default value]

Maximum number of iterations in the Preconditioned Conjugate Gradient solver
--

Double; OPTIONAL maxRelativeErrorPcg [INPUT, , default=no default value]

Maximum relative error in the Preconditioned Conjugate Gradient solver
--

Invntt; OPTIONAL invntt [INPUT, , default=no default value]
--

First row of the inverse time-time noise correlation matrix


SimpleImage; MANDATORY output [OUTPUT, No default value, default=no default value]

Output map product

History

- March 2007: First version
- 19 Apr 2007: PC, use of temporary files
- 20 Apr 2007: PC, use of the new DetectorTimeline data structure and use of scan context
- 13 Nov 2008: PPa, fix compiling problems
- 21 Nov 2008: PC, task names should be registered in HIPE with a name starting with a lower case letter [SPIRE-SPR 989]

1.104. maskBadChan

Full Name:	herschel.spire.ia.pipeline.common.engdata.MaskBadChanTask
Alias:	maskBadChan
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import MaskBadChanTask
Category:	SPIRE/Pipeline/Common Task

Description

Mask bad channels task.

This task is used to set the mask for dead, noisy, slow channels and channels that are not chopped on sky.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value.]
ChanMask chanMask [Input, Optional, default=No default value.]
ChanNum chanNum [Input, Optional, default=No default value.]
InstModeMask instModeMask [Input, Optional, default=No default value.]
DetectorTimeline output [Output, Optional, default=No default value.]

API details


Properties

DetectorTimeline data [Input, Mandatory, default=No default value.]
Detector timeline to be processed.
ChanMask chanMask [Input, Optional, default=No default value.]
Channel Mask product.
ChanNum chanNum [Input, Optional, default=No default value.]
Channel Number Mapping product.
InstModeMask instModeMask [Input, Optional, default=No default value.]
Instrument Mode Mask product.
DetectorTimeline output [Output, Optional, default=No default value.]
Output detector timeline.

History

- 2007-11-14 - PPa: First version.
- 2008-02-02 - PPa: Change name.
- 2008-03-25 - PPa: Use ChanNum to remove not connected channels.
- 2008-07-17 - PPa: Implement new masking policy.
- 2008-10-24 - PPa: Fix SPR-0963.
- 2009-01-28 - PPa: Add parameter descriptions [SPR-0985]. Fix history format. Add serialVersionUID.
- 2009-02-20 - PPa: Throw SignatureException instead of RuntimeException [SPR-1248].
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-04-24 - PPa: Set isSlow mask [SPR-1266]. Add mask definitions to metadata. Add comments.
- 2009-05-08 - PPa: Implement usage of InstModeMask. Implement new requirements.
- 2009-06-08 - PPa: Cope with N channels if not present in calibration product.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.105. maskBadTmParam

Full Name:	herschel.spire.ia.pipeline.common.engdata.MaskBadTmParamTask
Alias:	maskBadTmParam
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import MaskBadTmParamTask
Category:	SPIRE/Pipeline/Common Task

Description

Mask bad telemetry parameters task.

This task is used to set the mask for TM parameters that are unreliable.

API Summary

Properties
EdpProduct data [Input, Mandatory, default=No default value.]
TelemMask chanMask [Input, Mandatory, default=No default value.]
EdpProduct output [Output, Optional, default=No default value.]

API details


Properties

EdpProduct data [Input, Mandatory, default=No default value.]
Timeline to be processed.
TelemMask chanMask [Input, Mandatory, default=No default value.]
Telemetry Mask product.
EdpProduct output [Output, Optional, default=No default value.]
Output timeline.

History

- 2009-11-17 - PPa: First version.
- 2009-12-07 - PPa: use reserved bit.
- 2009-12-15 - PPa: Correctly handle case of packet types not included in the TelemMask.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.106. MedianInterferogramTask

Full Name:	herschel.spire.ia.pipeline.spec.average.MedianInterferogramTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.average import MedianInterferogramTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task that calculates the detector-by-detector and opd-by-opd median of the interferogram samples in a SPIRE SDI product.

Task Execution Steps:

For each detector in the input SDI product:

1. Calculate the median of the voltage at each opd across all forward scans.
2. Calculate the median of the voltage at each opd across all reverse scans.
3. Store the median voltage for each scan direction into separate datasets in the output SDI.

Example

Example 1: Nominal usage:

```
medSdi = medianInterferogram(sdi=sdi)
```

API Summary

Properties
SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]
SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=no default value]

API details

Properties

SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=no default value]
Input Spectrometer Detector Interferogram Product
SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=no default value]
Median Spectrometer Detector Interferogram Product

History

- 2010-02-03 - TRF: [SPIRE-2310] Misc fixes to URM information


1.107. MexicanHat

Full Name:	herschel.spire.ia.pipeline.common.deglitch.wavelet.MexicanHat
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.wavelet import MexicanHat

History

- 2006-03-20 - CO: first definition
- 2007-02-22 - CO: admissibiliy factor added in definition
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2008-04-29 - DB: corection warning Trevor Fulton version build 814
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag

1.108. Mosaic

Full Name:	herschel.spire.ia.gui.Mosaic
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import Mosaic

Description

Mosaic:

Displays a mosaic of detectors plots arranged in the honeycomb arrangement.


1. The Mosaic performs the following functionalities:
 - a. Plot all the SWL detectors with user selected scans and plot type.
(Ex: plotting all Phases of all SWL with all scans.
 - b. Plot all the SSL detectors with user selected scans and plot type.
(Ex: plotting all Imaginary of all SSL with all scans.
 - c. Plot all the Co-aligned detectors with user selected scans and plot type.
(Ex: plotting all Real part of all Co-aligned with all scans.
 - d. Plot all the Unvignetted detectors with user selected scans and plot type.
(Ex: plotting all Absolute of all Unvignetted with all scans.
 - e. Plot all the Nominal detectors with user selected scans and plot type.
(Ex: plotting all Absolute of Nominal detectors all with all scans.
2. A single Mosaic with all the plots of the selected detectors displayed in the honeycomb arrangement.

History

- 2008-06-22 - RB: Created
- 2008-06-27 - RB: Modified to work with changes to the SpecExplorer engine.
- 2008-07-02 - RB: Mosaic now loads scan selections and makes use of a new method in the SpecExplorer engine to fix a bug that caused multiple confirmation windows to pop up.
- 2008-09-10 - KZ: Changed addOverplot method to allow printing/saving of Mosaic by adding the co-aligned detectors to the comboplot.
- 2008-09-16 - KZ: Changed getXYLocations so that it checks for null values.
- 2008-09-22 - KZ: Changed the font size in the addPlot and addOverPlot from 9 to 11
- 2008-09-23 - KZ: Fixed some bugs in the Layout for unvignetted Co-aligned detectors by creating a new function to contains the unvignetted layout map (getCoalignedUnvignetted).
- 2008-09-23 - KZ: Disabled the title and subtitle by removing the calls to setTitle() function and other calls in the displayMosaicFor functions.

- 2008-09-23 - KZ: Allowed plots in the comboplot to contain tick marks by changing the `removeNoncenterPlotterFeatures()` so that it does not call the function `setAuxiliaryAxesVisibility()` and by removing the if statement (`if(fixedYAxis)`) in the for loop in the `displayMosaicFor` functions...etc
- 2008-12-01 - TRF: First Release in main branch
- 2008-12-02 - TRF: Cleaned up some javadoc and jtag warnings -- [SPIRE SPR-0678], [HCSS SPR-3232] Deleted commented code
- 2010-02-03 - TRF: [INFR-365] Migrated from `herschel.spire.ia.dataset.gui`


1.109. MultiplyIfgmTask

Full Name:	herschel.spire.ia.dataset.MultiplyIfgmTask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import MultiplyIfgmTask

History

- 2009-04-21 - YFR: First Version
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions and fixed vague descriptions

1.110. NaiveAppMapperTask

Full Name:	herschel.spire.ia.pipeline.phot.scanmap.NaiveAppMapperTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.scanmap import NaiveAppMapperTask
Category:	Pipeline/Map Making

Description

This task takes a Averaged Photometer Pointing Product and produces a Spire Photometer Map Product

by performing a standard coaddition of the samples.

Example

Example 1: task = NaiveAppMapperTask()
<pre>mapProduct=task(Appp)</pre>

API Summary

Properties
<code>PointedPhotProduct; MANDATORY input [INPUT, No default value, default=no default value]</code>
<code>String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]</code>
<code>Double; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]</code>
<code>Wcs; OPTIONAL wcs [INPUT, , default=with the default resolution>]</code>
<code>Integer; OPTIONAL doMap [INPUT, , default=no default value]</code>
<code>Boolean; OPTIONAL storeTod [INPUT, , default=no default value]</code>
<code>SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]</code>

API details

Properties

<code>PointedPhotProduct; MANDATORY input [INPUT, No default value, default=no default value]</code>	Input averaged photometer pointed product
<code>String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]</code>	Bolometer array to be processed ('PSW', 'PMW' or 'PLW')
<code>Double; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]</code>	Resolution of the output map in arcsec

Wcs; OPTIONAL wcs [INPUT, , default=with the default resolution>]

World Coordinate System for the output map
--

Integer; OPTIONAL doMap [INPUT, , default=no default value]
--

If set to False, the Tod product will be created, but not the map

Boolean; OPTIONAL storeTod [INPUT, , default=no default value]

If set to true, the pre-processed Tod is made available with todBuffer = mapper.getTodBuffer()
--


SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]

Output map product

History

- March 2007: First version
- 19 Apr 2007: PC, use of temporary files
- 20 Apr 2007: PC, use of the new DetectorTimeline data structure and use of scan context
- 21 Nov 2008: PC, task names should be registered in HIPE with a name starting with a lower case letter [SPIRE-SPR 989]

1.111. NaiveScanMapperTask

Full Name:	herschel.spire.ia.pipeline.phot.scanmap.NaiveScanMapperTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.scanmap import NaiveScanMapperTask
Category:	Pipeline/Map Making

Description

This task takes a Photometer DetectorTimeline and produces a Spire Photometer Map Product by performing a standard coaddition of the samples.

Example

Example 1: `task = NaiveScanMapperTask()`

```
map = task(timelineProduct, array="PSW")
```

API Summary

Properties
SpireListContext input [INPUT, Tod or TodBuffer; MANDATORY, default=No default value]
String ; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]
Double ; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]
Wcs ; OPTIONAL wcs [INPUT, , default=with the default resolution>]
Integer ; OPTIONAL doMap [INPUT, , default=no default value]
Boolean ; OPTIONAL storeTod [INPUT, , default=no default value]
SimpleImage ; No default output [OUTPUT, MANDATORY, default=no default value]
Double ; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]
Integer ; OPTIONAL doInterpolation [INPUT, , default=no default value]
Integer ; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]

API details

Properties

```
SpireListContext input [INPUT, Tod or TodBuffer; MANDATORY, default=No default value]
```


SpireListContext of detector timelines, Tod or TodBuffer

String; OPTIONAL array [INPUT, Default bolometer array is PSW, default=no default value]
Bolometer array to be processed ('PSW', 'PMW' or 'PLW')
Double; OPTIONAL resolution [INPUT, , default=10" and 14" for PSW]
Resolution of the output map in arcsec
Wcs; OPTIONAL wcs [INPUT, , default=with the default resolution>]
World Coordinate System for the output map
Integer; OPTIONAL doMap [INPUT, , default=no default value]
If set to False, the Tod product will be created, but not the map
Boolean; OPTIONAL storeTod [INPUT, , default=no default value]
If set to true, the pre-processed Tod is made available with <code>todBuffer = mapper.getTodBuffer()</code>
SimpleImage; No default output [OUTPUT, MANDATORY, default=no default value]
Output map product
Double; No default mergingMaximumGap [INPUT, MANDATORY, default=no default value]
If set, reorder and merge the building blocks. The value is the maximum gap in seconds allowed for merging two adjacent building blocks
Integer; OPTIONAL doInterpolation [INPUT, , default=no default value]
If set to true, NaN values in the timeline are interpolated
Integer; No default baselinePolynomialOrder [INPUT, MANDATORY, default=no default value]
If set, remove polynomial fit from start to end of the timeline. The value specifies the polynomial order

History

- March 2007: First version
- 19 Apr 2007: PC, use of temporary files
- 20 Apr 2007: PC, use of the new DetectorTimeline data structure and use of scan context
- 21 Nov 2008: PC, task names should be registered in HIPE with a name starting with a lower case letter [SPIRE-SPR 989]


1.112. NhkTimeline

Full Name:	herschel.spire.ia.dataset.NhkTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import NhkTimeline

History

- 2007-03-23 - PP: First version
- 2007-03-27 - PP: Fix substrings
- 2007-08-20 - PP: Migrate to the new Unit API
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-07-11 - PP: Add biasAmpl and biasAmplUnit, deprecate biasVoltage and biasVoltageUnit
- 2008-11-04 - PP: Fix biasAmpl and biasAmplUnit for PTC channels. Remove biasVoltage and biasVoltageUnit.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-05-27 - PPa: Fix conversion formula for bias frequency [SPR-1442]

1.113. NodAverageTask

Full Name:	herschel.spire.ia.pipeline.phot.denodding.NodAverageTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.denodding import NodAverageTask
Category:	task

Description

produces an Averaged Pointed Photometer Product.

This task takes a Pointed Photometer Product from DeNoddingTask and produces an Averaged Pointed Photometer Product, which is the final Level 1 product for the POF2 pipeline.

Example

Example 1: A Jython example
<pre>from herschel.spire.ia.pipeline.phot.denodding import * task = NodAverageTask() output = task([ppp1, ppp2], operator="mean")</pre>

API Summary

Properties
Mandatory input [Input, No default value, default=no default value]
Optional operator [Input, No default value, default=no default value]
Optional deglitch [Input, No default value, default=no default value]
Optional output [Output, No default value, default=no default value]

API details

Properties

Mandatory input [Input, No default value, default=no default value]
Takes a pointed photometer product (PPP) produced by DeNoddingTask as input to the task.
Optional operator [Input, No default value, default=no default value]
Optional deglitch [Input, No default value, default=no default value]
Takes the optional deglitch parameter as input


Optional output [Output, No default value, default=no default value]

Provide an averaged pointed photometer product (APPP) as a final output product.
--

History

- 2005-12-01 - TB: First "empty" version
- 2006-01-01 - TB: First version of the algorithm
- 2006-06-01 - TB: Major update.
- 2007-03-01 - TB: Completely changed the structure of input and output products
- 2007-04-01 - TB: Merged preamble() and postamble() into execute(). Fixes for 64-point jiggle
- 2008-06-25 - TB: Replaced depreciated methods for DetectorTimeline class.
- 2008-10-14 - BOH: Renamed task, and edited to follow task naming convention.
- 2009-01-19 - BOH: edited AveragingTask.java to add warning to empty catch statement
- 2009-01-21 - BOH: edited jtags for compliance with recommended structure
- 2009-04-08 - BOH: Fixed jexample.
- 2009-05-18 - PPa: remove useless imports
- 2009-05-21 - BOH: Added units to output TableDataset [SPIRE SPR-1418]
- 2009-07-22 - BOH: Added quality control metrics [SPIRE SPR-1468]
- 2009-08-12 - BOH: Added quality control metrics [SPIRE SPR-1468, 1713]
- 2009-08-22 - BOH: Added mask to appp [SPIRE-1844].
- 2009-08-26 - BOH: Fixed mask size [SPIRE-1858].
- 2009-09-12 - BOH: Added level metadata keyword [SPIRE-1951]
- 2009-11-18 - BOH: fixed errors for Weighted Mean [SPIRE-1606].
- 2009-11-23 - BOH: revert fix for SPIRE-2178 in order to fix SPIRE-2192
- 2009-12-09 - BOH: fixed error calculation in the APPP product [SPIRE-2241].
- 2010-03-01 - BOH: updated for Javadoc/URM [SPIRE-2251].


1.114. NonLinCorr

Full Name:	herschel.spire.ia.dataset.NonLinCorr
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import NonLinCorr
Category:	Calibration product

History

- 2008-06-25 - AS: Rewritten following Ed's description.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-02-19 - AS: Modified spectrometer units method (K1/K1E coefficients).
- 2009-06-05 - PPa: Renamed.
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods.
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.115. NortonBeerApodizeFunctionOnePointEight

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointEight
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointEight

Description


Implementation of the Norton Beer 1.8 apodization function (Hanning with width equivalent to 1.8 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.116. NortonBeerApodizeFunctionOnePointFive

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointFive
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointFive

Description


Implementation of the Norton Beer 1.5 apodization function (Hanning with width equivalent to 1.5 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.117. NortonBeerApodizeFunctionOnePointFour

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointFour
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointFour

Description


Implementation of the Norton Beer 1.4 apodization function (Hanning with width equivalent to 1.4 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.118. NortonBeerApodizeFunctionOnePointNine

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointNine
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointNine

Description


Implementation of the Norton Beer 1.9 apodization function (Hanning with width equivalent to 1.9 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.119. NortonBeerApodizeFunctionOnePointOne

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointOne
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointOne

Description


Implementation of the Norton Beer 1.1 apodization function (Hanning with width equivalent to 1.1 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.120. NortonBeerApodizeFunctionOnePointSeven

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointSeven
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointSeven

Description


Implementation of the Norton Beer 1.7 apodization function (Hanning with width equivalent to 1.7 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.121. NortonBeerApodizeFunctionOnePointSix

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointSix
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointSix

Description


Implementation of the Norton Beer 1.6 apodization function (Hanning with width equivalent to 1.6 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.122. NortonBeerApodizeFunctionOnePointThree

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointThree
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointThree

Description


Implementation of the Norton Beer 1.3 apodization function (Hanning with width equivalent to 1.3 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.123. NortonBeerApodizeFunctionOnePointTwo

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionOnePointTwo
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionOnePointTwo

Description


Implementation of the Norton Beer 1.2 apodization function (Hanning with width equivalent to 1.2 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.124. NortonBeerApodizeFunctionTwoPointZero

Full Name:	herschel.spire.ia.pipeline.spec.apodize.util.NortonBeerApodizeFunctionTwoPointZero
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.apodize.util import NortonBeerApodizeFunctionTwoPointZero

Description


Implementation of the Norton Beer 2.0 apodization function (Hanning with width equivalent to 2.0 times a Sinc).

See Apodization section of SPIRE-BSS-DOC-002966 for further details.

History

- 2009-07-24 - YFR: First version
- 2010-05-18 - TRF: [SPIRE-2275] Fixed URM issues.

1.125. ObsData

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.ObsData
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm import ObsData
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Helper class that stores general SPIRE Spectrometer observation data for CreateIfgmTask.


See also

- [CreateIfgmTask](#)

History

- 2007-09-07 - TRF: First version
- 2007-11-13 - TRF: Added and implemented the setShortScanMask() method
- 2007-11-13 - TRF: Added a bit mask for short scans (00000010)
- 2008-04-28 - PK: Changed type of HashMap scanNumberToIndex to HashMap<Integer,Integer> to remove unchecked conversion warning.
- 2008-08-14 - ES: Added toString() method.
- 2009-01-19 - TRF: [SPIRE SPR-1133] Fixed URM warnings.
- 2009-04-16 - TRF: [HCSS SPR-6051], [SPIRE SPR-1135]: Changed jcategory tag from Task to task
- 2010-03-01 - TRF: [SPIRE-2280]: Misc URM/Javadoc fixes.
- 2010-05-18 - TRF: [SPIRE-2280]: Misc URM fixes.


1.126. OffsetHist

Full Name:	herschel.spire.ia.dataset.OffsetHist
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import OffsetHist
Category:	Calibration product

History

- 2006-11-17 - PP: First version (PP)
- 2007-03-28 - PP: Store raw values and not converted ones.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-01-21 - PPa: Add obsid column [SPIRE SCR-0939]
- 2009-02-17 - PPa: Add method to get the applicable obsid [SPIRE SCR-0939]
- 2009-07-23 - PPa: Add subsystem metadata [SPIRE-SPR-1599]
- 2009-09-07 - PPa: Set version to "1" [SPIRE-1902]
- 2009-10-02 - PPa: Fix offset method.
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.127. OptCross

Full Name:	herschel.spire.ia.dataset.OptCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import OptCross
Category:	Calibration product

History

- 2008-05-20 - PP: First version
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.128. Pcal

Full Name:	herschel.spire.ia.dataset.Pcal
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import Pcal
Category:	PCAL Calibration product

History

- 2007-11-11 - AS: Add methods to get/set data by column; rename pixel get/set methods.
- 2007-11-14 - AS: Modify get/set by column to get/set by ArrayData type
- 2008-02-08 - PP: Remove useless imports and copy constructors.
- 2008-02-25 - PP: Extend ChannelCalibTable
- 2008-11-27 - TRF: Fixed some javadoc warnings [SPIRE SPR-0684]
- 2009-09-07 - PPa: Set version to "1" [SPIRE-1902]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.129. PcalOdPlugin

Full Name:	herschel.spire.ia.pipeline.pg.PcalOdPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import PcalOdPlugin

Description


PCAL Operational Day Plugin.

This class implements the Product Generator plugin for processing PCAL flashes products of the OpDayContext. It is used by the SPG Operational Day processing stage. It is not intended for use by the general user who doesn't have access to telemetry databases.

History

- 2007-11-19 - PPa: First experimental version.
- 2007-11-21 - PPa: Make it more robust against exceptions
- 2008-01-17 - PPa: Move to new OpDayPlugin interface
- 2008-02-08 - PPa: Fix SPR-0640.
- 2008-10-28 - PPa: comment out Pcal processing for SPR-0972
- 2009-04-01 - PPa: Re-insert Pcal processing (SPR-0972)
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]

1.130. PcalTask

Full Name:	herschel.spire.ia.pipeline.phot.pcal.PcalTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.pcal import PcalTask
Category:	spire_ia_pipeline_phot_pcal_task

Description

PCAL Processing Task

Task for processing PCAL Flash Building Blocks. The PCAL is an internal calibration source inside the SPIRE instrument, which is flashed to illuminate the detectors. A Flash Building Block includes several on-off cycles of the PCAL with a constant flash-on level and typically a period of a few seconds. The PcalTask processes the detector timelines after Level 0.5 processing and first-level deglitching, to measure statistics about the flash-off signals, the flash-on signals, the difference between flash-on and flash-off, the length of the stable plateaus, and the time constants of the detectors after each toggling of the PCAL.

The PcalTask currently takes two inputs. The first is the {@link DetectorTimeline} containing the detector signals, which can be for either the Photometer or the Spectrometer. The second input is the Subsystem Control Unit Timeline (SCUT) which contains the measured current and voltage timelines for the PCAL source. The PcalTask uses the current timeline to determine the starting and ending times for each flash-on and flash-off period. Then for each detector, the time boundaries are refined further to delimit the stable intervals of each flash-on or flash-off plateau. The output of the PcalTask is a {@link Pcal} product (either {@link PhotPcal} or {@link SpecPcal} depending on the input timeline), which contains the measured statistics of the plateaus and the detector time constants. As of April 2009 the detector time constants are not yet populated.

The following statistical parameters are calculated and written to the output Pcal history product:

- **baseMean** : the ensemble mean value of all base mean values computed from the stable part of the base sequence.
- **baseStdDev** : the standard deviation of all base mean values.
- **signalDiffMean** : the mean value of the ensemble of signal differences. The signal difference for a flash is the difference between the mean of the base level following the flash, and the mean of the stable flash.
- **signalDiffStdDev** : the standard deviation of the signal differences.
- **flashMean** : the ensemble mean value of the flash mean values.
- **flashStdDev** : the standard deviation of the flash mean values.
- **baseNoise** : the square root of sum of squares of the baseStdDev divided by the number of base intervals.
- **stdDevOfMeanOfSignalDifferences** : the standard deviation of the signal differences divided by the square root of the number of flashes. This gives an error bound on the estimate of the mean of the pcal signal difference (assuming stability of the pcal source).
- **flashStableSamples** : mean number of samples in the stable part of the flash intervals.
- **baseStableSamples** : mean number of samples in the stable part of the base intervals.
- **settleTimeOnMean** : mean of duration of start of flash intervals prior to stability.

- settleTimeOnStdDev : standard deviation of start of flash intervals prior to stability.
- settleTimeOffMean : mean of duration of start of base intervals prior to stability.
- settleTimeOffStdDev : standard deviation of start of base intervals prior to stability.

To run the PcalTask, only a DetectorTimeline and a Subsystem Control Unit Timeline are needed:

```
pcalTask = PcalTask()
pcalProd = pcalTask(pdt, scut)
```

The PcalTask is also available in the Tasks view in HIPE.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
EdpProduct scut [Input, Mandatory, default=No default value]
String detectorPlot [Input, Optional, default=No default value]
Pcal out [OUTPUT, Mandatory, default=No default value]

API details

Properties


DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT: Input detector timeline.
EdpProduct scut [Input, Mandatory, default=No default value]
Housekeeping product containing pcal flash current.
String detectorPlot [Input, Optional, default=No default value]
Optional string requesting diagnostic plots (e.g.PSWD16(detector) or 'ENSEMBLE' for all detectors.
Pcal out [OUTPUT, Mandatory, default=No default value]
Pcal history product.

History

- 2007-11-11 - AS: Add test data.
- 2008-1-17 - AS: Analyze Pcal and detector data, and populate calib product.
- 2008-2-27 - AS: Accept and process either photometer or spectrometer timeline.
- 2008-11-13 - AS: Provide capability to accept Float1D input detector timeline
- internal calculations are done in Double1d:
- copy selected metadata from input to Pcal product:

- (obsid, bbid, odNumber, startDate, endDate)
- 2009-3-3 - AS: 1. [SPR-972] Modify code to initialize global class variables
- in methods rather than constructor. This will allow an
- instantiation of the task to be run correctly multiple times.
- 2. Correct error in pcalAnalysis method that results in a
- spurious extra flash being detected at the end of the sequence.
- 3. Use a slightly coarser digitization of the pcal current to
- avoid outliers in the histogram aggregate of the flash.
- 2009-3-5 - AS: 1. [SPR-1268] Modify code to correct error in calculation of
- signal difference mean.
- 2009-3-9 - AS: 1. Modify code to make task known to hype ("hype").
- 2009-3-9 - AS: 1. Add description of parameters using addDescription for hype.
- 2009-3-30 - AS: 1. [SCR-1201] Add capability to process fewer than maximum
- number of detectors in an array. Set hasData field.
- 2. [SPR-972] As part of workaround, reset all global objects
- to empty objects, so that memory usage may be minimized.
- 2009-3-22 - AS: 1. [SCR-1264] Revise flash search algorithm to take advantage
- of knowledge of operation of pcal radiator. Add additional
- statistical parameters to output history product.
- 2009-7-17 - AS: [SPR-1665] Reset isphotometer/isspectrometer flags at end of task.
- 2009-8-5 - AS: [SCR-1605] Modify algorithm to utilize first base level/flash level
- in calculation of signal difference.
- 2009-8-12 - AS: [SCR-1666] Propagate metadata as appropriate.
- [SCR-1279] Implement hype progress indicator.

1.131. PhaseCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.phase.PhaseCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.phase import PhaseCorrectionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to correct for asymmetries in SPIRE Spectrometer interferograms.

Task Execution Steps (from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

1. From the double-sided spectra in the input SDS product
2. Calculate weighted fit to phase, $j(s)$, within the observation band to a 4th order polynomial.
3. Calculate phase function $e^{-ij(s)}$
4. If the resolution mode for the building block is LR or MR:
 - a. Multiply the spectra in the SDS with phase function.
 - b. If apodization is requested
 - c. Compute FFT^{-1} of phase corrected spectra to produce phase-corrected interferograms.
5. If the resolution mode for the building block is HR:
 - a. Butterfly the computed phase function, making imaginary portion anti-symmetric.
 - b. Compute FFT^{-1} of phase function.
 - c. Truncate phase function to create PCF.
 - d. If PCF apodization is requested, apodize the PCF.
 - e. Convolve the PCF with the interferograms in the input SDI to correct the phase.
 - f. Replace interferograms in the input SDI with these phase-corrected interferograms.

Example

Example 1: The following jython script shows an example of the PhaseCorrectionTask.

```
#Compute the double-sided Fourier Transform of all interferograms
dsds = fourierTransform(sdi=presdi, ftType="prePhaseCorr", zeroPad="None")
#Correct for asymmetries in the interferograms by fitting a second-order
polynomial, using a convolution kernel of width 127 points.
sdi = phaseCorrection(sdi=sdi, sds=dsds, polyDegree=2, pcfSize=127, nlp=nlp,
phaseCorrLim=phaseCorrLim)
```

API Summary

Properties

[SpectrometerDetectorInterferogram](#) **sdi** [[INPUT, MANDATORY, default=No default value](#)]

Properties
SpectrometerDetectorSpectrum sds [IO, MANDATORY, default=No default value]
SCalSpecNlp nlp [INPUT, OPTIONAL, default=no default value]
SpecPhaseCorrLim phaseCorrLim [INPUT, OPTIONAL, default=no default value]
Integer polyDegree [INPUT, OPTIONAL, default=spire.ia.pipeline.spec.ft.phasecorrect.polydegree property's value]
Integer pcfSize [INPUT, OPTIONAL, default=spire.ia.pipeline.spec.ft.phasecorrect.pcfsize property's size]
String convolApodName [INPUT, OPTIONAL, default=No Default value]
SpectrometerDetectorInterferogram outsdi [OUTPUT, MANDATORY, default=No default value]

API details

Properties

SpectrometerDetectorInterferogram sdi [INPUT, MANDATORY, default=No default value]
Input interferogram product to be phase-corrected
SpectrometerDetectorSpectrum sds [IO, MANDATORY, default=No default value]
Input Double-sided SDS product. The spectra in this product should be derived from the interferograms in the input SDI product. The phase of the interferograms is derived from these spectra. These spectra will also be phase-corrected during task execution.
SCalSpecNlp nlp [INPUT, OPTIONAL, default=no default value]
Product that contains the Non linear (optical) phase for each channel.
SpecPhaseCorrLim phaseCorrLim [INPUT, OPTIONAL, default=no default value]
Product that contains phase fitting limits (i.e. Band edges) for each spectrometer channel. These edges provide a guide to the fitting routine that will be used during phase correction.
Integer polyDegree [INPUT, OPTIONAL, default=spire.ia.pipeline.spec.ft.phasecorrect.polydegree property's value]
Degree of the polynomial that will be fitted to the in-band phase of the input interferograms.
Integer pcfSize [INPUT, OPTIONAL, default=spire.ia.pipeline.spec.ft.phasecorrect.pcfsize property's size]
Width (in points) of the phase-correction function (PCF) that will be convolved with the input interferograms. NB this is only valid for HR observations, for MR/LR observations, convolution is not invoked.

<code>String convolApodName [INPUT, OPTIONAL, default=No Default value]</code>
--

The name of the apodization function that is to be applied to the PCF. NB this is only valid for HR observations, for MR/LR observations, convolution is not invoked.

<code>SpectrometerDetectorInterferogram outsdI [OUTPUT, MANDATORY, default=No default value]</code>

Output SDI product that contains the phase-corrected interferograms

History

- 2007-05-03 - AH: Cleaned up implementation, separate methods, added Javadoc
- 2007-05-04 - AH: First implementation taken from RegSampledPhaseCorrection
- 2007-05-31 - AH: Class placed in independent package. JParamaters updated.
- 2007-07-31 - AH: Migrate to new Unit API
- 2007-09-07 - TRF: Added a temp import to herschel.spire.ia.pipeline.spec.ft.util until FFT_PACK code is moved into HCSS
- 2007-09-07 - TRF: Changed calBandEdge to calSpecBandEdge and this arg now expects a SCalSpecBandEdge cal product. This change removes the need for SpecBand and for the getBandEdges method which has been removed.
- 2007-09-24 - TRF: Fixed bug in getFittedPhase. Method now works even if SCalSpecBandEdge is not provided.
- 2007-09-27 - TRF: Removed plots.
- 2007-09-27 - TRF: Removed conditional that allows for a choice between Double1d and Float1d in the non-mask columns of SpireInterferogram1d and SpireSpectrum1d.
- 2007-10-02 - ZW: Added super() in constructor
- 2007-11-13 - TRF: Changed FFT package import from herschel.spire.ia.pipeline.spec.ft.util to herschel.ia.numeric.toolbox.xform.util
- 2007-11-14 - TRF: Added isBadChannel() method to filter out erroneous channels
- 2008-04-18 - TRF: Fixed javadoc warning by changing version: annotation to version.
- 2008-05-26 - ES: Added truncateChannelColumns() method to ensure that pixels defined as bad channels will have correct column lengths in the output SDI.
- 2008-05-26 - ES: Added badInput() method to change thrown exceptions to logger messages and to facilitate a change back in the future.
- 2008-05-26 - ES: Removed dependencies on herschel.spire.ia.pipeline.spec.ft.util.
- 2008-05-27 - TRF: Added some more info to the checkEquals() call (pixelNames)
- 2008-05-27 - TRF: Changed logging level for badInput() to fine added array size check to checkEquals()
- 2008-06-13 - TRF: Fixed some Javadoc
- 2008-07-16 - TRF: Modified top-level Javadoc to conform with the information in the SPIRE Spectrometer Pipeline Description document (SPIRE-BSS-DOC-002966)

- 2008-08-14 - ES: Added methods `getDsRange()` and `setDsRange()` in order to properly truncate columns in the "LR"/"MR"/ds case.
- 2008-08-18 - ES: Refactored `checkCompatibleMetaData()`.
- 2008-08-18 - ES: Deprecated `setIfgm()`, `checkArrayHasZeroAndConstantDelta()`, `checkArrayHasPositiveAndNegativeValuescheckEquals()`, and `checkEquals()`, as they are either not used, or not actually needed.
- 2008-08-21 - TRF: [SPIRE SPR-0824] New class named `PhaseCorrectionTask` to replace the mis-named `PhaseCorrection` class
- 2008-08-21 - TRF: Removed deprecated methods: `checkArrayHasPositiveAndNegativeValues()`, `checkArrayHasZeroAndConstantDelta()`, `checkEquals()`, `setIfgm()`
- 2008-08-27 - TRF: [SPIRE SPR-0850] Fixed bug that left some phase-corrected ifgms with columns of unequal length.
- 2008-10-07 - TRF: [SPIRE SPR-0917] Added descriptions to `TaskParameters`. Made `jparameter` descriptions more descriptive. Added a reference to the SPIRE Pipeline Description Document.
- 2008-10-10 - TRF: [SPIRE SPR-0938] Fixed bug in the manner in which `setProgress()` was being updated
- 2008-10-15 - TRF: [SPIRE SPR-0871] Implemented a test QC parameter for `phaseWrap`.
- 2008-11-13 - TRF: [SPIRE SPR-1028] Added `checkInterrupted()` to `execute()`
- 2008-11-13 - TRF: [SPIRE SPR-1029] Suppressed the QC warning for `phaseWrap`
- 2008-11-21 - TRF: Fixed unchecked, serial, and javadoc warnings
- 2008-12-09 - TRF: [HCSS SPR-3232] Fixed some `jtag urm` documentation
- 2008-12-09 - TRF: [SPIRE SPR-1029] Deprecated `isBadChannel()`
- 2008-12-09 - TRF: Removed usage of `Iterator`. Added private `getDefaultLow/HighBandEdge()` accessor methods.
- 2008-12-10 - TRF: [HCSS SPR-3232] Fixed some `drm` documentation
- 2008-12-10 - TRF: [SPIRE SPR-1029] Removed deprecated `isBadChannel()`, `setDsRange()`
- 2009-01-23 - TRF: [SPIRE SPR-1165] Added `serialVersionUID = 1L`, removed `SuppressWarnings`
- 2009-01-23 - TRF: [SPIRE SPR-1133] Fixed DRM warning
- 2009-01-23 - TRF: [SPIRE SPR-1163] Added new `TaskParameters` to agree with SPIRE Pipeline policy, deprecated old `TaskParameters`
- 2009-01-23 - TRF: [SPIRE SPR-1098] Added `throws javadoc tag` to `execute()`
- 2009-01-29 - TRF: [SPIRE SPR-1163] Removed deprecated `TaskParameters`
- 2009-02-04 - TRF: [SPIRE SPR-1200, 1204] Changed `sdi.keySet()/sds.keySet()` to `sdi.getScanNumbers()/sds.getScanNumbers()`
- 2009-02-17 - YFR: [SPIRE SPR-1189] Fixed the incorrect scales for the output interferogram in `inverseFourierTransform` method
- 2009-03-16 - TRF: Fixed the package to which the logger is assigned

- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-05-25 - YFR: [SPIRE SPR-1513] Remove interferogram and spectrum when can't get correct phase while spectrum has too few data to process
- 2009-06-12 - YFR: [SPIRE SPR-1513] A `BadDataException` will be thrown when there are no scans left in the output sdi
- 2009-06-12 - YFR: [SPIRE SPR-1511] Task now sets Creator data in accordance with SPIRE Pipeline policy
- 2009-06-12 - TRF: [SPIRE SPR-1187] Task now accepts a non-linear phase calibration product task parameter
- 2009-06-13 - TRF: [SPIRE SPR-1029] If a data channel is not found in the calibration product, the exception is allowed to propagate.
- 2009-06-13 - TRF: [SPIRE SCR-1378] Use `PhaseCorrLim` calibration product instead of `BandEdges` calibration product
- 2009-06-14 - TRF: [SPIRE SCR-1270] Change INFO message to count scans in ascending order
- 2009-07-20 - YFR: [SPIRE-1691] Removed deprecated methods, replaced with equivalent methods
- 2009-09-12 - TRF: [SPIRE-1656] Task now sets the `phaseCorrectionApplied` metadata to true in sdi and dsds upon completion
- 2009-10-20 - DMT: [SPIRE-1365] Deprecated `qcPhaseWrap`, added `phaseWrapSSW` and `phaseWrapSLW`
- 2009-11-06 - TRF: [SPIRE-2116] Corrected the MR/LR implementation
- 2009-11-30 - TRF: [SPIRE-2220] Fixed the case when spec or ifgm contain NaNs
- 2010-01-26 - TRF: [SPIRE-1365] Fixed phase wrap QC metadata to ignore non-nominal and unusable channels
- 2010-01-26 - TRF: [SPIRE-2283] Fixed URM entries
- 2010-01-26 - TRF: [SPIRE-1658] Removed deprecated `SpecBandEdge` task parameter
- 2010-01-26 - TRF: [SPIRE-1365] Removed deprecated `fixPhaseWrap(sdi, inBandPhase)`
- 2010-01-26 - TRF: Removed usage of deprecated `AbstractComposite()` methods
- 2010-02-01 - TRF: [SPIRE-2336] Inverted ifgms to be flipped prior to phase correction
- 2010-03-15 - TRF: [SPIRE-2448] `truncatePcf()` no longer assumes that input data are of a specific size.
- 2010-05-18 - TRF: [SPIRE-2283] Fixed Javadoc/URM entries


1.132. PhotBolPar

Full Name:	herschel.spire.ia.dataset.PhotBolPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotBolPar
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.133. PhotBsmOps

Full Name:	herschel.spire.ia.dataset.PhotBsmOps
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotBsmOps
Category:	Calibration product

History

- 2006-09-20 - PP: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.
- 2010-04-06 - PPa: Don't use PhotObsModes [SPIRE-2255]


1.134. PhotBsmPos

Full Name:	herschel.spire.ia.dataset.PhotBsmPos
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotBsmPos
Category:	Calibration product

History

- 2006-08-30 - PP: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.135. PhotChanGain

Full Name:	herschel.spire.ia.dataset.PhotChanGain
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanGain
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.136. PhotChanMask

Full Name:	herschel.spire.ia.dataset.PhotChanMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanMask
Category:	Calibration product

History

- 2008-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.137. PhotChanNoise

Full Name:	herschel.spire.ia.dataset.PhotChanNoise
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanNoise
Category:	Calibration product

History

- 2008-01-28 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.138. PhotChanNomRes

Full Name:	herschel.spire.ia.dataset.PhotChanNomRes
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanNomRes
Category:	Calibration product

History

- 2008-06-25 - PP: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.139. PhotChanNum

Full Name:	herschel.spire.ia.dataset.PhotChanNum
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanNum
Category:	Calibration product

History

- 2008-01-29 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.140. PhotChanTimeConst

Full Name:	herschel.spire.ia.dataset.PhotChanTimeConst
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanTimeConst
Category:	Calibration product

History

- 2007-09-07 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.141. PhotChanTimeOff

Full Name:	herschel.spire.ia.dataset.PhotChanTimeOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotChanTimeOff
Category:	Calibration product

History

- 2006-08-22 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.142. PhotDetAngOff

Full Name:	herschel.spire.ia.dataset.PhotDetAngOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotDetAngOff
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-05-20 - PP: Change name.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.143. PhotElecCross

Full Name:	herschel.spire.ia.dataset.PhotElecCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotElecCross
Category:	Calibration product

History

- 2008-02-04 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.144. PhotFlatField

Full Name:	herschel.spire.ia.dataset.PhotFlatField
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotFlatField
Category:	Calibration product

History

- 2006-08-18 - PP: First version (PP)
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.145. PhotFluxConversionTask

Full Name:	herschel.spire.ia.pipeline.phot.fluxconv.PhotFluxConversionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.fluxconv import PhotFluxConversionTask
Category:	spire/ia/pipeline/phot/fluxconv task

Description

The task corrects detector voltage non-linearities and then converts the result to optical power.

The non-linearity correction function is

$$S1 = K1 * (V3 - V0) + K2 * \ln(V3 - K3) - K2 * \ln(V0 - K3)$$

after which the flux correction

$$J = S1 * F$$

is applied.

API Summary

Properties
DetectorTimeline pdt [Input, Mandatory, default=No default value]
PhotFluxConv fluxConv [Input, Mandatory, default=No default value]
DetectorTimeline out [OUTPUT, Optional, default=No default value]

API details

Properties


DetectorTimeline pdt [Input, Mandatory, default=No default value]
PDT: Input detector timeline.
PhotFluxConv fluxConv [Input, Mandatory, default=No default value]
Input calibration product of correction coefficients.
DetectorTimeline out [OUTPUT, Optional, default=No default value]
Output product converted to flux units.

History

- 24 Jan 2008 : R Gastaud & A.Schwartz (convert prototype)
- 15 Feb 2008 AS Incorporate calibration tables into calculations.

- 04 Apr 2008 AS Modify correction equation to be consistent with new version of pipeline requirements doc
- "The SPIRE Analogue Signal Chain and Photometer Detector Data Processing Pipeline", Issue 4
- dated: Feb 20 2008, by Matt Griffin.
- 23 Jul 2008 AS Add tests for out of range input data and
- ill conditioned input data (data that results in negative argument to log).
- 19 Feb 2009 AS Remove info logging for bad/out-of-range detectors.
- 24 Mar 2009 AS [SCR-1201] Add capability to process fewer than the
- maximum number of detectors in an array
- (if the product has fewer).
- 24 Apr 2009 AS 1. [SCR-1281] Generate quality control metadata for
- samples out of range, and below K3 voltage.
- 24 Apr 2009 AS 2. [SCR-1047] Handshaking between FluxConv and
- TempDriftCorr calibration products.
- 08 May 2009 [SCR-1141] Tooltips and URM cleanup.
- 12 Aug 2009 [SPR-1141] Tooltips and URM cleanup.
- 21 Aug 2009 AS [SCR-1280] Set mask bits for data below K3 voltage
- and data for which V0flag or Kflag is NOT set
- in the calibration product. Do not process
- these data.
- 04 Sep 2009 AS [SCR-1141] More tooltip cleanup.
- [SCR-1825] Rename calibration product input parameter.
- 10 May 2010 AS Modify javadoc/URL; add package.html


1.146. PhotFluxConv

Full Name:	herschel.spire.ia.dataset.PhotFluxConv
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotFluxConv
Category:	Calibration Product

History

- 2006-02-25 - PPa: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2006-02-25 - PPa: Use new ArrayEnum
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-01-15 - PPa: Add bias amplitudes in metadata [SPIRE SPR-1044]


1.147. PhotInstModeMask

Full Name:	herschel.spire.ia.dataset.PhotInstModeMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotInstModeMask
Category:	Calibration product

History

- 2009-04-29 - PP: First version
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.148. PhotLpfPar

Full Name:	herschel.spire.ia.dataset.PhotLpfPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotLpfPar
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-09-18 - TRF: This product contains only the phot table
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.149. PhotOffsetHist

Full Name:	herschel.spire.ia.dataset.PhotOffsetHist
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotOffsetHist
Category:	Calibration product

History

- 2006-11-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.150. PhotOptCrossCorrectionTask

Full Name:	herschel.spire.ia.pipeline.phot.optcross.PhotOptCrossCorrectionTask
Alias:	PhotOptCrossCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.optcross import PhotOptCrossCorrectionTask
Category:	task

Description

Task to correct the Optical Crosstalk for the different bolometers.

It works for any bolometer (spectrometer or photometer), and uses a calibration matrix which stores the coefficient of the crosstalk. This effect is small (less than 1%), so any method is good.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
OptCross optCross [Input, Mandatory, default=No default value]
DetectorTimeline output [OUTPUT, Optional, default=No default value]

API details

Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
PDT, SDT, PPT or PPP: bolometer timelines to be corrected for optical crosstalk
OptCross optCross [Input, Mandatory, default=No default value]
SPIRE calibration product: contains the coefficients of the optical crosstalk
DetectorTimeline output [OUTPUT, Optional, default=No default value]
PDT, SDT, PPT or PPP: bolometer timelines corrected for optical crosstalk, same class than the input

See also


- [???](#)

History

- 04 July 2008 : R Gastaud
- 28-January-2009 RG rewritten to correct SPR-1179, now works when some detectors are missing in the detector timeline

- same than SPR-1118
- 28-January-2009: RG fix SPR 1076 PhotOptCrossCorrectionTask should not propagate the file-Name metadata
- 28-January-2009: RG fix SPR-1165 Use serialVersionUID=1L instead of @SuppressWarnings("serial")
- 2009-05-23 - RG: same changement than ElecCrossTalk
- SPR-1415 SciVal-Jiggle:Electrical Crosstalk matrix should be transposed
- now use the method crossTalk(String channel1, String channel2) as SpecOptCrossCorrectionTask


1.151. PhotOptCross

Full Name:	herschel.spire.ia.dataset.PhotOptCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotOptCross
Category:	Calibration product

History

- 2008-05-20 - PP: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.152. PhotPcal

Full Name:	herschel.spire.ia.dataset.PhotPcal
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotPcal
Category:	Calibration product

History

- 2008-02-08 - AS: Fix description
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.153. PhotRsrf

Full Name:	herschel.spire.ia.dataset.PhotRsrf
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotRsrf
Category:	Calibration product

History

- 2009-10-21 - PPa: First version [SPIRE-2045]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.154. PhotTempDriftCorr

Full Name:	herschel.spire.ia.dataset.PhotTempDriftCorr
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PhotTempDriftCorr
Category:	Calibration table product to be used with TemperatureDriftCorrectionTask

History

- 2006-02-28 - AS: Initial Release
- 2008-10-07 - AS: Change descriptive comments.
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]

1.155. PointedPhotProduct

Full Name:	herschel.spire.ia.dataset.PointedPhotProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PointedPhotProduct
Category:	Pipeline/DeNodding


Description

This class creates an empty Pointed Photometer Product. One method is provided, `getProduct()`, which returns the empty `PointedPhotProduct` product.

History

- 2005-12-01 - TB: First version
- 2006-04-02 - TB: Cycle ID not needed
- 2006-06-05 - TB: CompositeDatasets no longer in use, nor is IOS
- 2007-03-14 - TB: Altered due to new ADT and PointedPhotProduct structures
- 2007-05-03 - TB: Change class name from Ppp.java; move in dataset package; small refactoring. (PP)
- 2007-09-06 - TRF: Fixed some Javadoc
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-10-07 - PP: Remove `getSignal()` overloading method (SPR-0888)
- 2008-11-27 - TRF: Fix some javadoc warnings
- 2009-09-11 - PPa: Remove `wcsType`, `wcsReference`, and `equinox` [SPIRE-1939]
- 2009-10-09 - BOH: Add compulsory metadata [SPIRE-1877]
- 2009-10-22 - PPa: Remove some unnecessary methods, cleanup, refactoring, remove unneeded "bbid", fix javadoc.
- 2009-10-27 - BOH: Restored signal methods [SPIRE-2096]
- 2010-02-05 - YFR: Removed usage of deprecated `AbstractComposite` methods
- 2010-05-10 - PPa: Remove "rasterId" [SPIRE-2538], do not extend `DetectorTimeline` [SPIRE-2504], fix javadoc [SPIRE-2255], refactoring.
- 2010-05-12 - PPa: Revert back to extend `DetectorTimeline` [SPIRE-2504].

1.156. PointedPhotTimeline

Full Name:	herschel.spire.ia.dataset.PointedPhotTimeline
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PointedPhotTimeline
Category:	Pipeline/DeNodding

Description


Help

This class creates an empty Pointed Photometer Timeline

History

- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-04-28 - PP: fix getSignal and getMask methods
- 2008-05-22 - PP: Update: now equinox is added by the SpireDataProduct class.
- 2008-10-07 - PP: Remove getSignal overloading method (SPR-0888)
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-09-07 - PPa: Remove "wcsType" and "wcsReference" [SPIRE-1926]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.157. PointedProduct

Full Name:	herschel.spire.ia.dataset.PointedProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import PointedProduct

History

- 2008-11-27 - TRF: Fixed some javadoc warnings
- 2009-04-20 - PPa: Add getChannelNames and getChannelKeySet methods [SCR-1340]

1.158. PointSourceFitTask

Full Name:	herschel.spire.ia.pipeline.phot.pointsource.PointSourceFitTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.pointsource import PointSourceFitTask
Category:	herschel/spire/ia/pipeline/phot/pointsource

Description

This task takes an Averaged Pointed Photometer Product and produces a

Jiggle Point Source Fit Product. This task is part of the jiggle pipeline, between Level 0.5 and Level 1.

Example

Example 1: A Jython example
<pre>from herschel.spire.ia.modules.userproducts import * task = PointSourceFit() outputProduct = task.perform(inputProduct)</pre>

Limitations


The class is just an empty container for the time being. It takes the input product, performs a few basic checks to ensure that all the needed datasets are present, then builds an empty output product.

History

- July 2005: First "empty" version
- July 2006: Major update
- October 2006: Adapted to changes in JPSFP product
- March 2007: Adapted to changes in the APPP product
- 14 Apr 2008: (TRF) Replaced calls to deprecated Double2d.appendColumn(slice) with Double2d.append(slice, 1)
- 28 Nov 2008 (BOH): integrated task into HIPE, edited names of tasks within HIPE.
- 09 Mar 2009 (AP): fixed error: Estimation of fitted gaussian Sigma from $2.3548 * \text{fwhm}$ to $\text{fwhm}/2.3548$
- 09 Mar 2009 (AP): Temporarily commented out calculation in widthDifference and _fitQualityControl in processPixel(). It seems to be causing exceptions.
- 09 [tag 1_17] Mar 2009 (AP) :Changed order to append longitude(RA) first and then latitude (Dec) in the Coords variable in processPixel().
- 04 Jun 2009 (AP):modified processPixel method to ignore signal with NaN (was freezing the Task)
- 06 Sep 2009 (AP):change fitter from Amoeba to LevenbergMarguadt
- 06 Sep 2009 (AP): re-enabled "keep fwhm fit parameter fixed"

- 06 Sep 2009 (AP): changed so that fit result uses NO weights.
- 06 Sep 2009 (AP): Replaced instances of longitude/latitude with ra/dec
- 06 Sep 2009 (AP): Workaround to get spire.ia.data.JiggxxxProducts to compile
- 02 Oct 2009 (AP): Changed default PSW FWHM to 18.5"
- 02 Oct 2009 (AP): Fixed bug in fitting.
- 12 Nov 2009 (AP): SPIRE-2137 - Module does not crash when NaNs are present in ra/dec during fitting
- 18 Nov 2009 (AP): SPIRE-2137 - Moved fitter initialisation in try statement.
- 13 Jan 2010 (AP): Changed default beam sizes, values from ScanMap AOT release note Issue 2
21 October 2009
- 10 Feb 2010 (AP): Change all ra and dec values relative to commanded ra/dec. Apply Cos factor to ra values. Revert fit ra.dec to original Commanded values
- 10 May 2010 (AP)SPIRE-2273 changed jcategory tag to include entire path
- 10 May 2010 (AP)SPIRE-2273 changed jhelp tag to mention the location of the task in the pipeline.
- 13 May 2010 (AP) changed fitter default tolerance
- 13 May 2010 (AP) remove raJigPos1 and decJigPos1, instead of removing raNominal and decNominal

1.159. qlaDataConversion

Full Name:	herschel.spire.ia.pipeline.common.engdata.QlaDataConversionTask
Alias:	qlaDataConversion
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import QlaDataConversionTask
Category:	SPIRE/Pipeline/Common Task

Description

QLA Data Conversion Task.

This task is used by the Engineering Conversion to convert non-detector data from ADU to engineering units. It converts raw ADC values extracted from the telemetry into engineering values, i.e. voltages, currents, etc. using the QLA conversion tables.

API Summary

Properties
EdpProduct rawData [INPUT, MANDATORY, default=No default value.]
Boolean pixConv [INPUT, OPTIONAL, default=true.]
EdpProduct converted [Output, OPTIONAL, default=No default value.]

API details

Properties


EdpProduct rawData [INPUT, MANDATORY, default=No default value.]
Input raw SPIRE Data Product to be converted.
Boolean pixConv [INPUT, OPTIONAL, default=true.]
Flag to use QLA conversion tables for detector values. If set to "false", detector readouts will be not converted.
EdpProduct converted [Output, OPTIONAL, default=No default value.]
Converted SPIRE Data Product.

History

- 2005-01-00 - PPa: First version.
- 2005-08-00 - PPa: Moved to the new doc system, add interruption control. Added flag for detector values. Added the call of super.preamble(). Make conversion to double even if there is no conversion table.
- 2005-09-00 - PPa: Code cleanup.
- 2005-10-00 - PPa: Do not make conversion to double if there is no conversion table again.
- 2005-12-07 - PPa: Comment out "error" column.

- 2006-02-21 - PPa: Use new product formats; use `BuildingBlockProduct`.
- 2006-06-01 - PPa: Add `setResetOnExecute(true)`.
- 2006-06-15 - PPa: Add computation of the how many saturated samples are present.
- 2006-08-10 - PPa: Refactoring.
- 2006-09-14 - PPa: Compatible with optimized products.
- 2006-10-04 - PPa: Use `ChanGain` instead of `PixelGainTable`.
- 2007-01-24 - PPa: Use centimeters for length quantities (SMEC).
- 2007-03-13 - PPa: Use the new product structure.
- 2007-03-15 - PPa: Re-implement truncation, adding also 0.
- 2007-03-26 - PPa: Check for truncation even if `pixConv=false`.
- 2007-03-28 - PPa: Truncation is now done in `DetFlagger`. Do not convert anymore offset values.
- 2007-07-27 - PPa: Migrate to new Unit API.
- 2007-10-03 - PPa: Change name.
- 2008-07-17 - PPa: Change name in constructor.
- 2008-07-18 - PPa: Do not crash if a column is not recognized.
- 2008-09-25 - PPa: Remove unused `chanGain` parameter (SPR-0872). Update jtags.
- 2008-12-01 - PPa: Do not try to convert the `seqCount` column (SPR-1046).
- 2008-12-29 - PPa: Use new lookup methods of `ParameterMapper` instead of column descriptions. Add parameter descriptions (SPR-0985).
- 2009-01-28 - PPa: fix again `jhistory` format. Add `serialVersionUID`.
- 2009-03-30 - PPa: Remove deprecated `setResetOnExecute(true)`.
- 2009-09-05 - PPa: Handle MCUET products.
- 2009-12-16 - PPa: Use new `AbstractComposite` methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.160. QualAssessSpireLevel05Task

Full Name:	herschel.spire.ia.pipeline.common.quality.QualAssessSpireLevel05Task
Alias:	QualAssessSpireLevel05Task
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.quality import QualAssessSpireLevel05Task
Category:	task

Description

Task for quality assessment of level 0.5 Products

API Summary

Properties
<code>Product input [Input, Mandatory, default=No default value]</code>
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>

API details

Properties


<code>Product input [Input, Mandatory, default=No default value]</code>
the input Product that contains the metadata information from the pipeline
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>
the predefined SpireQualityProduct in which the quality control data will be stored

History

- 22 Oct 07: Created the task from QualAssessSpireLevel1Task.java.
- 15 Nov 07: Made child of QualAssessSpireTask; removed redundant lines;
- incorporated javadoc formatting into comments.
- 12 May 08: Adapted to use SpireQualityEnum
- 27 Aug 08: Added fix for SPIRE-SPR 0691; changed method for handling
- Boolean flags in input.
- 22 May 09: Constructor fixed (see SPIRE-SPR 1425).
- 15 Jul 09: Added new variables to be assessed by module.
- 17 Jul 09: Finished adding new quality control metrics.
- 12 Aug 09: Shifted assessments of level 0.5 spectrometer-specific metadata
- to QualAssessSpireLevel1Task.java (see SPR SPIRE-1377).

- 23 Nov 09: Changed acceptable range for ratioInvalidTimes (SPR SPIRE-2193).
- 11 Mar 10: Updated limits (SCR SPIRE-2425).

1.161. QualAssessSpireLevel1Task

Full Name:	herschel.spire.ia.pipeline.common.quality.QualAssessSpireLevel1Task
Alias:	QualAssessSpireLevel1Task
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.quality import QualAssessSpireLevel1Task
Category:	task

Description

Task for quality assessment of level 1 Products

API Summary

Properties
<code>Product input [Input, Mandatory, default=No default value]</code>
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>

API details

Properties


<code>Product input [Input, Mandatory, default=No default value]</code>
the input Product that contains the metadata information from the pipeline
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>
the predefined SpireQualityProduct in which the quality control data will be stored

History

- 07 Sep 07: Created the task from original QualAssessSpire.java.
- 22 Oct 07: Added more metadata parameters.
- 13 Nov 07: Added more metadata parameters.
- 15 Nov 07: Made child of QualAssessSpireTask; removed redundant lines;
- incorporated javadoc formatting into comments.
- 12 May 08: Adapted to use SpireQualityEnum.
- 27 Aug 08: Added fix for SPIRE-SPR-0691; changed method for handling
- Boolean flags in input.
- 28 Aug 08: Removed assessment of glitchNumber.
- 10 Nov 08: Shortened number of parameters used to call checkGreater
- and checkLess following modifications to QualAssessSpireTask.

- 02 Apr 09: Updated module to work with new metadata.
- 20 Apr 09: Made small spelling change to evaluation of qcPhaseWrap
(formerly qcPhaseWraps).
- 22 May 09: Constructor fixed (see SPIRE-SPR 1425).
- 15 Jul 09: Added new variables to be assessed by module; also shifted some
assessments to other modules.
- 17 Jul 09: Finished adding new quality control metrics.
- 12 Aug 09: Shifted assessments of level 0.5 spectrometer-specific metadata
to QualAssessSpireLevel1Task.java (see SPR SPIRE-1377).
- 23 Nov 09: Added assessment of bsmMovedDuringScanFlag (SPR SPIRE-2082).
- Changed acceptable range for ratioPhotFirstLevelGlitchesSXR
(SPR SPIRE-2193).
- 22 Feb 10: Added assesment of smecOpenLoopCount (SPR SPIRE-2342).
- 11 Mar 10: Updated limits (SCR SPIRE-2425).
- 17 Mar 10: Made minor changes to limits of quality control flags
(SCR SPIRE-2425; see comments from 16 Mar 10).

1.162. QualAssessSpireLevel2Task

Full Name:	herschel.spire.ia.pipeline.common.quality.QualAssessSpireLevel2Task
Alias:	QualAssessSpireLevel2Task
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.quality import QualAssessSpireLevel2Task
Category:	task

Description

Task for quality assessment of level 2 Products

API Summary

Properties
<code>Product input [Input, Mandatory, default=No default value]</code>
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>

API details

Properties


<code>Product input [Input, Mandatory, default=No default value]</code>
the input Product that contains the metadata information from the pipeline
<code>SpireQualityProduct output [Input/Output, Mandatory, default=No default value]</code>
the predefined SpireQualityProduct in which the quality control data will be stored

History

- 07 Sep 07: Created the task from QualAssessSpireLevel2Task.java.
- 15 Nov 07: Made child of QualAssessSpireTask; removed redundant lines;
- incorporated javadoc formatting into comments.
- 12 May 08: Adapted to use SpireQualityEnum.
- 27 Aug 08: Added fix for SPIRE-SPR-0691.
- 10 Nov 08: Shortened number of parameters used to call checkGreater
- and checkEqual following modifications to QualAssessSpireTask.
- 22 May 09: Constructor fixed (see SPIRE-SPR 1425).
- 15 Jul 09: Added new variables to be assessed by module.
- 17 Jul 09: Finished adding new quality control metrics.
- 23 Nov 09: Added assessment of mapmakingConvergenceFailurePwvFlag

- metadata (SPR SPIRE-2168).

1.163. QualAssessSpireTask

Full Name:	herschel.spire.ia.pipeline.common.quality.QualAssessSpireTask
Alias:	QualAssessSpireTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.quality import QualAssessSpireTask
Category:	task

Description

Parent Task class for other quality assessment Tasks

API Summary

Properties
Product input [Input, Mandatory, default=No default value]
QualityContext output [Input/Output, Mandatory, default=No default value]

API details

Properties


Product input [Input, Mandatory, default=No default value]
the input Product that contains the metadata information from the pipeline
QualityContext output [Input/Output, Mandatory, default=No default value]
predefined QualityContext in which the quality control data will be stored

History

- 15 Nov 07: Created the task from QualAssessSpireLevel05Task.java.
- 12 May 08: Adapted to use SpireQualityEnum.
- 27 Aug 08: Added checkBoolean to check Boolean flags.
- 28 Aug 08: Changed logic of checkGreater and checkLess methods to handle multiple inputs.
- 08 Oct 08: Added parameter descriptions. Altered some descriptions to indicate that output should be QualityContext.
- 10 Nov 08: Modified methods that set flags to use the QualityFlagsHelper to add messages to QualityContext. Disabled software to set metadata (including Boolean flags) in the QualityContext.
- 09 Dec 08: Commented out deprecated code that sets metadata in the

- QualityContext.
- 22 May 09: Constructor fixed (see SPIRE-SPR 1425).
- 22 Feb 10: Fixed issue with children of QualAssessSpireTask being
- associated with non-SPIRE Products in HIPE (SCR SPIRE-2350).

1.164. QualityPlugin

Full Name:	herschel.spire.ia.pipeline.pg.QualityPlugin
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import QualityPlugin

Description


Quality Plugin.

This class implements the Product Generator plugin for the population of the quality context. It is used by the SPG during the postprocessing stage. It is not intended for use by the general user.

History

- 2007-04-20 - PPa: First version.
- 2007-04-25 - XX: don't run quality on level0 products
- 2007-10-15 - XX: updated for latest QC tasks
- 2008-03-18 - XX: SPR-0639; changes to ObservationContext API
- 2008-07-24 - XX: Shifted context creation with null values to populate method; removed confusing redundancy in runQC.
- 2008-11-10 - XX: Disabled code that adds metadata with null values to a new QualityContext.
- 2008-12-09 - XX: Removed additional lines that add metadata parameters to the QualityContext.
- 2010-03-01 - PPa: Improve URM [SPIRE-2266]

1.165. rawDataExtraction

Full Name:	herschel.spire.ia.pipeline.common.rawdata.RawDataExtractionTask
Alias:	rawDataExtraction
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import RawDataExtractionTask
Category:	SPIRE/Pipeline/Common Task

Description

Raw Data Extraction Task.

The purpose of this task is to extract telemetry from the database and to arrange data in Level-0 SPIRE Data Products. It is used by the CalibrationPlugin to obtain detector offset data and by the Level0Plugin to generate level 0 products for the processed observation.

It uses the DbQueryMakerTask to connect and query the database, and uses the RawProductCreatorTask to generate the level 0 products.

Note: this task is intended only for ICC members to access telemetry data, or usage in SGP. It is not for general users.

Examples

Example 1: Extract telemetry packets with OBSID=3000C251 from database ops_1.

```
# Note that this will produce level 0 products in old format, i.e. splitted by
# telemetry type
level0products=rawDataExtraction(obsid = 0x3000C251L, tmPack = True,
dbName="ops_1")
```

Example 2: Same as previous example but putting all telemetry types in a single product.

```
level0products=rawDataExtraction(obsid = 0x3000C251L, tmPack = True,
dbName="ops_1", multiType=True)
```

API Summary

Properties
Date startDate [INPUT, OPTIONAL, default=No default value]
Date endDate [INPUT, OPTIONAL, default=No default value]
FineTime startTime [INPUT, OPTIONAL, default=No default value]
FineTime endTime [INPUT, OPTIONAL, default=No default value]
Long obsid [INPUT, OPTIONAL, default=No default value.]
Long bbid [INPUT, OPTIONAL, default=No default value.]
Boolean tmPack [INPUT, OPTIONAL, default=false.]
String dbName [INPUT, OPTIONAL, default=No default value]
Long maxSize [INPUT, OPTIONAL, default=No default value]
ResetHist resetHist [INPUT, OPTIONAL, default=No default value]
String packetType [INPUT, OPTIONAL, default=No default value]

Properties
<code>ObjectStore store [INPUT, OPTIONAL, default=No default value]</code>
<code>Boolean joined [INPUT, OPTIONAL, default=Boolean.FALSE]</code>
<code>Boolean multiType [INPUT, OPTIONAL, default=Boolean.FALSE]</code>
<code>LinkedHashMap rawData [OUTPUT, OPTIONAL, default=No default value]</code>
<code>Level0Context level0 [OUTPUT, OPTIONAL, default=No default value .]</code>

API details

Properties

<code>Date startDate [INPUT, OPTIONAL, default=No default value]</code>
Start date of the time range.
<code>Date endDate [INPUT, OPTIONAL, default=No default value]</code>
End date of the time range.
<code>FineTime startTime [INPUT, OPTIONAL, default=No default value]</code>
Start date of the time range, in FineTime.
<code>FineTime endTime [INPUT, OPTIONAL, default=No default value]</code>
End date of the time range, in FineTime.
<code>Long obsid [INPUT, OPTIONAL, default=No default value.]</code>
Observation ID (OBSID).
<code>Long bbid [INPUT, OPTIONAL, default=No default value.]</code>
Building Block ID (BBID)
<code>Boolean tmPack [INPUT, OPTIONAL, default=false.]</code>
Flag to use TmSourcePackets (tmPack=True) instead of SpireDataFrames (tmPack=False)
<code>String dbName [INPUT, OPTIONAL, default=No default value]</code>
Database name. It is used to override the configuration property and access a DB that differ from the one specified in property files.
<code>Long maxSize [INPUT, OPTIONAL, default=No default value]</code>
Max number of SpireDataFrame (to avoid out of memory errors)
<code>ResetHist resetHist [INPUT, OPTIONAL, default=No default value]</code>
DPU reset history calibration product.
<code>String packetType [INPUT, OPTIONAL, default=No default value]</code>
Telemetry packet type name
<code>ObjectStore store [INPUT, OPTIONAL, default=No default value]</code>
ObjectStore for direct connection with the versant database.


Boolean <code>joined</code> [INPUT, OPTIONAL, default=Boolean.FALSE]
Flag to have joined timelines. If true, products are not splitted in building blocks.
Boolean <code>multiType</code> [INPUT, OPTIONAL, default=Boolean.FALSE]
Flag to have timelines with multiple telemetry types. If true, products are not splitted by type.
LinkedHashMap <code>rawData</code> [OUTPUT, OPTIONAL, default=No default value]
Array of Level-0 SPIRE Data Products.
Level0Context <code>level0</code> [OUTPUT, OPTIONAL, default=No default value .]
Level-0 SPIRE Data Products wrapped as a Level0Context

History

- 2005-11-03 - PPa: 3 First version, extracted form EngDataProcess.java
- 2005-11-09 - PPa: Code clean-up
- 2005-11-14 - PPa: Added possibility to use a stream of TmSourcePackets
- 2006-02-20 - PPa: use BuildingBlockProduct.
- 2006-03-09 - PPa: Change "sdfStream" into "stream" for RawProductCreator.
- 2006-06-14 - PPa: Add maxSize parameter (Max number of SpireDataFrame to be downloaded to avoid out of memory errors). Remove preamble.
- 2006-06-22 - PPa: Use java.util.LinkedHashMap as output. Refactoring.
- 2006-09-13 - PPa: DbQueryMaker now returns a ProductReader instead of ProductStream.
- 2006-09-14 - PPa: Compatible with optimized products
- 2006-11-21 - PPa: Refactoring. Add resetHist.
- 2007-03-27 - PPa: Add a first dialog.
- 2007-06-11 - PPa: Adapt to new JInputLongParamField.java
- 2007-08-09 - PPa: Add bbTypeName metadata to products
- 2007-09-18 - PPa: Remove the Dialog since it will be implemented in the hcss. Add Object-Store usage. Update documentation.
- 2007-10-15 - PPa: Add start/endTime. Version 0.8.
- 2008-02-07 - PPa: Add "joined" parameter. Add as second output the Level0Context. Version 0.9.
- 2008-07-17 - PPa: Fix name.
- 2009-04-21 - PPa: Remove setResetOnExecute. Fix "creator" metadata [SPR-1354].
- 2009-05-04 - PPa: Implement workaround for Test Facility telemetry having wrong modelName [SPR-1370]
- 2009-06-02 - PPa: Add description [SPR-1457]

- 2010-02-01 - PPa: Start implementation of SPIRE-2139
- 2010-05-12 - PPa: Update URM [SPIRE-2262]

1.166. rawProductCreator

Full Name:	herschel.spire.ia.pipeline.common.rawdata.RawProductCreatorTask
Alias:	rawProductCreator
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import RawProductCreatorTask
Category:	SPIRE/Pipeline/Common Task

Description

Raw Product Creator Task.

This task translates SPIRE data frames or TM packets in an array of SPIRE Level 0 Products. SPIRE data frames can be provided as an array or as a ProductStream stream. If no input is provided, the task output is a list of all supported products.

Note: this task is intended only for ICC members to access telemetry data, or usage in SGP. It is not for general users.

Example

Example 1: Generate level 0 products from a stream, using a ResetHist product.

```
HIPE>> dbConnection=dbQueryMaker(....)
HIPE>> resetHist=....
HIPE>> level0prods=rawProductCreator(stream=dbConnection,resetHist=resetHist)
```

API Summary

Properties
<code>SpireDataFrame[] sdfArray</code> [INPUT, OPTIONAL, default=No default value.]
<code>ProductStream stream</code> [INPUT, OPTIONAL, default=No default value.]
<code>Long maxSize</code> [INPUT, OPTIONAL, default=Long.MAX_VALUE.]
<code>Boolean joined</code> [INPUT, OPTIONAL, default=Boolean.FALSE.]
<code>Boolean multiType</code> [INPUT, OPTIONAL, default=Boolean.FALSE.]
<code>ResetHist resetHist</code> [INPUT, OPTIONAL, default=No default value.]
<code>BuildingBlockProduct[] rawData</code> [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties

<code>SpireDataFrame[] sdfArray</code> [INPUT, OPTIONAL, default=No default value.]
SpireDataFrame array to be converted.
<code>ProductStream stream</code> [INPUT, OPTIONAL, default=No default value.]
SpireDataFrame or TM packet stream.


Long maxSize [INPUT, OPTIONAL, default=Long.MAX_VALUE.]
Max number of SpireDataFrame (to avoid out of memory errors).
Boolean joined [INPUT, OPTIONAL, default=Boolean.FALSE.]
Flag to have joined timelines. If true, products are not splitted in building blocks.
Boolean multiType [INPUT, OPTIONAL, default=Boolean.FALSE.]
Flag to have joined timelines. If true, products are not splitted in building blocks.
ResetHist resetHist [INPUT, OPTIONAL, default=No default value.]
DPU Reset History Table
BuildingBlockProduct[] rawData [OUTPUT, OPTIONAL, default=No default value.]
Array of Level-0 SPIRE Data Product.

History

- 2005-12-28 - PPa: New class, copied from ProductCreator.
- 2005-12-31 - PPa: Use isNeededParam method from DfCreator. Add sdfTime, packetTime and seqCount column.
- 2006-01-06 - PPa: Fix bug.
- 2006-02-20 - PPa: Use BuildingBlock products.
- 2006-03-09 - PPa: Fix creation of empty products. Change "sdfStream" into "stream".
- 2006-04-22 - PPa: Remove usage of deprecated setObsId() and setBbId() of BuildingBlock-Product.
- 2006-05-23 - PPa: Remove usage of deprecated methods of SpireDataFrames
- 2006-06-06 - PPa: Reset the SpireTmPacketProcessor at the start of the extraction process.
- 2006-08-03 - PPa: Use checkInterrupted() method. Remove preamble and postamble.
- 2006-08-09 - PPa: Fix computation of modelName.
- 2006-08-10 - PPa: Add Test Facility Control TM packets
- 2006-09-14 - PPa: Optimize size of level-0 products
- 2006-10-31 - PPa: "Workaround" for redundant electronic telemetry.
- 2006-11-10 - PPa: Uses the ResetHist calibration product when provided.
- 2007-03-09 - PPa: "source" now is a level 0 product metadata keyword and not in the table.
- 2007-03-20 - PPa: use getSourceName() instead of getModelName()
- 2008-01-10 - PPa: Add flag to produce joined timelines.
- 2008-03-27 - PPa: Select the appropriate parameter table (SPR-497).
- 2008-05-22 - PPa: remove BBTYPE from NHKT (SPR-0563)

- 2008-07-17 - PPa: Fix name. SeqCount now is Int1d.
- 2008-09-04 - PPa: Add setting of "source" metadata in the getAllProducts method.
- 2008-11-11 - PPa: Add RMCUET (SPR-0992)
- 2009-01-07 - PPa: Add electronic side metadata [SCR-1003]. Improve documentation and insert parameter descriptions.
- 2009-02-17 - PPa: Add check on the electronic side for each frame [SCR-1003].
- 2009-05-19 - PPa: Do not set the creator [SPR-1354]
- 2009-06-02 - PPa: Add description [SPR-1457]
- 2009-08-04 - PPa: Add the metadata parameter "level" [SPIRE-1596]
- 2009-08-10 - PPa: Use new GenericSpireDataFrameImpl and SpireTmPacketConverter to generate products from TM packets [SPIRE-1369]
- 2010-02-01 - PPa: Start implementation of SPIRE-2139
- 2010-05-12 - PPa: Update URM [SPIRE-2262]


1.167. Reconstruction

Full Name:	herschel.spire.ia.pipeline.common.deglitch.reconstruction.Reconstruction
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.reconstruction import Reconstruction
Category:	class

History

- 2006-04-28 - CO: first java version (ancient classes now removed)
- 2007-02-28 - CO: new java version (new class)
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herschel.spire.ia.pipeline.common.deglitch
- 2008-12-08 - DB: flag as glitched and/or not uncorrected all the sample affected by the reconstruction
- 2008-12-15 - DB: add widthmax and widthmin and modification of valueb add widthmax and widthmin get method for flag mask
- 2008-12-16 - DB: correct calculation of wCoef
- 2009-01-23 - DB: correction jhistory
- 2009-03-10 - DB: prepare reconstruction noisy add newCWT_noisy()
- 2009-04-22 - DB: new version for reconstruction with noise on CWT
- 2009-04-30 - DB: corrected reconstruction, bug when interacted glitches
- 2009-09-10 - DB: SPIRE-1935 new reconstruction by linear interpolation, new parameters
- 2009-11-12 - DB: Reconstruction Polynomiale
- 2009-12-18 - DB: Update method getfitPoint() {
- 2010-01-25 - DB: SPIRE-2257 update javadoc setmethod -> setMethod
- 2010-03-01 - DB: SPIRE-2422 numeric.toolbox.basic.Rms removal replace Bsic.RMS by Basic.STDDEV

1.168. removeBaseline

Full Name:	herschel.spire.ia.pipeline.common.util.RemoveBaselineTask
Alias:	removeBaseline
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.util import RemoveBaselineTask
Category:	SPIRE/Pipeline/Common Task

Description

Task to remove the baseline from Pointed Photometer Timeline.

Use this task to remove the baseline. The baseline corresponds to the median value.

Examples

Example 1: How to use the task from HIPE command line.

```
my_output = removeBaseline(pdt=my_ppt, chanNum=my_chanNum)
```

Example 2: How to get the CVS number version of the task.

```
print removeBaseline.VERSION
```

API Summary

Properties
PointedPhotTimeline pdt [Input, Mandatory, default=No default value]
ChanNum chanNum [Input, Mandatory, default=No default value]
PointedPhotTimeline output [Output, Optional, default=No default value]

API details


Properties

PointedPhotTimeline pdt [Input, Mandatory, default=No default value]
Input (mandatory): Pointed Photometer Timeline
ChanNum chanNum [Input, Mandatory, default=No default value]
Input (mandatory): Channel Number Mapping Table
PointedPhotTimeline output [Output, Optional, default=No default value]
Output (optional): Pointed Photometer Timeline

History

- 2009-09-08 - PPa: First version
- 2009-10-18 - PPa: Set output signal with the correct class. [SPIRE-2085]
- 2010-05-11 - GM: Add VERSION to creator.


1.169. ResetHist

Full Name:	herschel.spire.ia.dataset.ResetHist
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import ResetHist
Category:	Calibration product

History

- 2006-10-04 - PP: First version
- 2007-05-24 - PP: Remove usage of deprecated methods.
- 2007-10-26 - PP: Remove setting of start and end dates.
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-07-23 - PPa: Add subsystem metadata [SPIRE-SPR-1599]
- 2009-09-07 - PPa: Set version to "1" [SPIRE-1902]
- 2009-11-04 - PPa: Set version to "1", this time for real! [SPIRE-1902]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods.


1.170. ScanContext

Full Name:	herschel.spire.ia.dataset.context.ScanContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import ScanContext

History

- 2007-04-16 - PPa: First prototype.
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: remove serialVersionUID: Context doesn't implement Serializable anymore
- 2009-03-06 - PPa: Remove get(int) method [SPIRE-SCR-1116]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]. Remove setting of creator.

1.171. SecondDeglitching

Full Name:	herschel.spire.ia.pipeline.phot.demodulation.SecondDeglitchingTask
Alias:	SecondDeglitching
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.demodulation import SecondDeglitchingTask
Category:	task

Description

This is a prototype of second level deglitcher

API Summary

Properties
DemodPhotProduct dpp [Input, Mandatory, default=No default value]
Double threshold [Input, Optional, default=default value = 5.0]
DemodPhotProduct dpp_out [Output, Optional, default=No default value]

API details


Properties

DemodPhotProduct dpp [Input, Mandatory, default=No default value]
DemodPhotProduct to be masked
Double threshold [Input, Optional, default=default value = 5.0]
values which are more distant than threshold*sigma are masked
DemodPhotProduct dpp_out [Output, Optional, default=No default value]
BEWARE COPY INPUT CHANGED Algorithm: As in the spire pipeline description spire-ral-doc-002437 paragraph 4.2.9 It does not work, but it is written as specified. WARNING: this does not detect glitches, but the motion of the BSM !!!!! anyway it does not detect anything because with 8 points the standard deviation is not a good choice, should be replaced by Median-AbsoluteDeviation

History

- 30 Janu 2008 : R Gastaud
- 24/07/2009 R Gastaud to talkative SPR-1516
- 10 December 2009 R Gastaud SPR SPIRE-1466 metadata for the quality
- 18 December 2009 R Gastaud back to standard deviation to stick to the documentation

1.172. SigmaKappaDeglitcherTask

Full Name:	herschel.spire.ia.pipeline.common.deglitch.SigmaKappaDeglitcherTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.deglitch import SigmaKappaDeglitcherTask
Category:	spire_ia_pipeline_common_deglitch_task

Description

This task detects and removes cosmic ray glitches from a detectortimeline.

The purpose of this function is to provide a simple, robust algorithm that will detect cosmic ray glitches and optionally remove them. The algorithm is as follows: [We will provide a concise description of the overall algorithm, and then will describe the details of each step.]

The Deglitcher Reconstructor algorithm operates individually on all of the detector timelines in the input product. The level one mask is read and all samples that have been previously identified as unusable, like glitches, saturation, etc. (either by a previous execution of this task or by other means) are treated by the module as glitches. The algorithm has three parts: 1) a preconditioning sequence that preprocesses the timeline to remove the astronomical signal from the timeline, 2) a glitch detection sequence that searches the preprocessed signal for statistical outliers by a Sigma Kappa Algorithm and flags them, and 3) a reconstruction sequence which removes the glitches that were previously detected either by the current task or any other task, and then replaces the removed glitches by statistically #neutral# data. In addition the algorithm provides a #Large Loop Iteration# which allows the user to specify a number of subsequent complete executions of the task.

Each of the detection and reconstruction steps may be disabled individually via the command line. So the task may be run as a detector-only algorithm (in which case the preprocessing and detection algorithms will be run, the glitch mask will be updated, but no reconstruction will be performed), or as a reconstruction-only algorithm (where the input glitch mask will be used to specify a set of glitches or otherwise unusable readouts to be reconstructed). The algorithm is parametrically driven and the user may specify a subset of input parameters, and accept defaults for other input parameters. In the future an optional parameter calibration product may be specified, in which case the parameters will be taken from the calibration product, or may individually be overridden by explicitly specifying input parameters. The task would then take optimized default parameters from the calibration product, based on the signal sample rate, telescope slew rate, detector array name, and instrument.

DETECTION OF GLITCHES (STEP 1)

1. DETRENDING SIGNAL

- a. Fit a line to ends of the timeline , and subtract the fitted line from the input. This will result in a signal which can be extended cyclically for filtering (call result S2)

2. FILTER CASCADE SEQUENCE

- a. Filter the fitted timeline (S2) with a boxcarfilter (Box).
- b. Subtract the boxcarfiltered signal from the fitted timeline $S3=(S2-Box)$. This results in a high-pass filtered version of the input signal.
- c. Repeat the boxcar filter "N" times (parameter specified), with the second iteration filtering the highpass filter from the first sequence (i.e. boxcar on the fitted timeline, followed by subtraction of the boxcar from the fitted timeline), and each subsequent iteration filtering the output from the previous filter. This filter cascade sequence generates a frequency response function which is

the product of the frequency response function of the original high pass filter multiplied by itself "N" times. So this filter cascade results in the further depression of low and midlevel frequencies relative to the highest spatial frequencies. So

```
Highpass1= S2 # BoxCar (S2)
Highpass2 = Highpass1 # BoxCar (Highpass1)
S4 = Highpassn = Highpassn-1 # BoxCar (Highpassn-1)
```

3. GLITCH DETECTION AND CHARACTERIZATION

- a. Glitch Detection: Run a "sigma-kappa" algorithm on S4 iteratively to detect "glitches" (SigKap). The algorithm first calculates the standard deviation of S4, and uses a value of kappa standard deviations about the mean (the mean being very close to zero) as a threshold to locate glitches (so a point whose absolute value is greater than (threshold # mean) is considered above the threshold).
- b. Iterate the algorithm by first recalculating the standard deviation after omitting all flagged points from consideration. Then apply this new sigma-kappa to locate points above the threshold and flag them. This iterative process continues until no new points are flagged. [Optionally stop here]
- c. If any two glitches are within a small parameter specified $(=(\text{boxFilterWidth}+1)/2)$ number of readouts of each other, this step aggregates the two glitches into one.
- d. Glitch Recording: Flag all detected glitches in mask array for the detector. [Optionally stop here]
- e. Large Glitch Characterization: Uses a simple model of the bolometer response function to a detect and flag the large glitches. This step uses a two parameter model of the bolometer response function to detect and flag the large glitches. It uses a large glitch discriminator time constant (Td), and a large glitch removal time constant (Tr) to select a consecutive sequence of readouts marked as glitches ($>Td$) and to extend (and flag) the sequence ($+Tr$) if it meets the selection criterion. The reason that we take this approach is to remove the entire effect of a glitch, not just the portion that lies above the sigma-kappa threshold. [Optionally stop here]

4. REMOVAL OF GLITCHES:

- The user may select either the DIRECT or LOWPASSNOTCH reconstruction option.
- a. DIRECT RECONSTRUCTION
 - i. Fit a line to the averaged signal just preceding and following a glitch for DIRECT reconstruction, or
 - ii. Replace the glitch by the randomized version of the fitted line (i.e. a "noisy" version of the line, where the noise statistics match the noise characteristics of the final iteration of the sigma-kappa algorithm) for DIRECT reconstruction.
 - b. LOWPASSNOTCH FILTER RECONSTRUCTION
 - i. Separate the signal into background (LowPassN) and LowPassNotch (Input - LowPassN) when using LOWPASSNOTCH reconstruction.
 - ii. Generate Low Pass Notch Filter by
 - A. Run n successive iterations of a low pass boxcar on the input detector time line:

```
(so 1st iteration LowPass1 = Boxcar(Input)
2nd iteration LowPass2 = Boxcar(LowPass1)
Nth iteration LowPassN = Boxcar(LowPass(n-1))
```

- B. Subtract the resultant filter from the input signal:

```
LowPassNotchFilter = Input -- LowPassN.
```

- iii. Replace the glitch by random noise with noise characteristics of the filtered signal (for LOW-PASSNOTCH reconstruction).
 - iv. SIGNAL RECONSTRUCTION ITERATION Combine the glitch-removed LowPassNotch filter signal and the background signal and restore trending (for LOWPASSNOTCH reconstruction). This step then allows an iteration of steps 7 through 9 using the parameter reconstructionIterationCycles. The motivation for the iteration cycle is to allow smoothing of any residual glitch hump. So repeated iteration cycles will smooth out the residual hump using the low pass notch filter cycle. Note that this is not the same as the Large Loop Cycle (which also includes the glitch detection cycle (see below)). Since the glitch location has already been identified, there is no need to use the large loop cycle for smoothing.
5. LARGELOOPCYCLE This is a reprocessing of the data using the entire algorithm, but using the output from the previous cycle as input to the algorithm. The previous output is detrended, filtered, and processed using the sigma-kappa algorithm. Then the flags from all previous iterations are merged with the flags from this iteration and the glitch replacement for ALL flagged points is redone using the INITIAL input product.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
Long iterationNumber [Input, Optional, default=1]
Double kappa [Input, Optional, default=9.0]
Double gamma [Input, Optional, default=1.0]
Long randomSeed [Input, Optional, default=1984574303]
Long boxFilterWidth [Input, Optional, default=3]
Long medianFilterWidth [Input, Optional, default=3]
Integer largeGlitchDiscriminatorTimeConstant [Input, Optional, default=2]
Integer largeGlitchRemovalTimeConstant [Input, Optional, default=5]
String filterType [Input, Optional, default="BOXCAR"]
String correctionMode [Input, Optional, default="LOWPASSNOTCH"]
String largeGlitchMode [Input, Optional, default="OVERLAY"]
Long boxFilterCascade [Input, Optional, default=1]
Long lowPassNotchFilterWidth [Input, Optional, default=3]
Long lowPassNotchFilterIteration [Input, Optional, default=20]
Long reconstructionIterationCycles [Input, Optional, default=5]
String glitchinfo [Input, Optional, default=No default value]
String disableSigmaKappaDetection [Input, Optional, default=No default value]
String disableLargeGlitchDetection [Input, Optional, default=No default value]
String disableGlitchReconstruction [Input, Optional, default=No default value]

Properties
DetectorTimeline out [OUTPUT, Optional, default=No default value]

API details

Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT Input detector timeline
Long iterationNumber [Input, Optional, default=1]
Number of Large Loop iterations (total # of cycles, including first).
Double kappa [Input, Optional, default=9.0]
Optional glitch definition parameter ($\sigma \cdot \kappa - \text{mean}$) of filtered signal.
Double gamma [Input, Optional, default=1.0]
Optional parameter describing the glitch correction noise scale factor.
Long randomSeed [Input, Optional, default=1984574303]
Optional initialization kernel for glitch correction noise.
Long boxFilterWidth [Input, Optional, default=3]
Optional box filter window size (default is 3).
Long medianFilterWidth [Input, Optional, default=3]
Optional median filter window size (default is 3).
Integer largeGlitchDiscriminatorTimeConstant [Input, Optional, default=2]
Optional large glitch discriminator time constant (default is 2).
Integer largeGlitchRemovalTimeConstant [Input, Optional, default=5]
Optional large glitch removal time constant (default is 5).
String filterType [Input, Optional, default="BOXCAR"]
Optional filter type (default is BOXCAR (or optionally MEDIAN)).
String correctionMode [Input, Optional, default="LOWPASSNOTCH"]
Optional correction mode (default is LOWPASSNOTCH (or optionally DIRECT)).
String largeGlitchMode [Input, Optional, default="OVERLAY"]
Optional large glitch correction mode. Default parameter is OVERLAY (where the largeGlitchRemovalTimeConstant is overlaid on the largeGlitchDiscriminatorTimeConstant, i.e. if more than largeGlitchDiscriminatorTimeConstant consecutive glitch samples are detected, then the glitch is extended to a total of largeGlitchRemovalTimeConstant glitch samples).


String largeGlitchMode [Input, Optional, default="OVERLAY"]
Optional parameter is ADDITIVE (where the largeGlitchRemovalTimeConstant is added to the largeGlitchDiscriminatorTimeConstant to determine the total length of the glitch).
Long boxFilterCascade [Input, Optional, default=1]
Optional filter cascade sequence of high pass filters (default is single high pass filter).
Long lowPassNotchFilterWidth [Input, Optional, default=3]
Optional reconstruction notch filter window size in samples(default is 3).
Long lowPassNotchFilterIteration [Input, Optional, default=20]
Optional reconstruction notch filter iteration sequence of low pass filters (default is single low pass filter).
Long reconstructionIterationCycles [Input, Optional, default=5]
Optional iteration sequence, where the reconstruction is performed by removing the glitch from the low pass notched filtered signal, followed by recombining the deglitched notch filtered signal with the background signal. The notch filter is then reapplied to the resultant signal, and the glitch removal is again reapplied. This iterative procedure is applied N times as specified by the parameter.
String glitchinfo [Input, Optional, default=No default value]
Optional parameter will provide a listing of glitch information. Specify parameter as glitchinfo="glitchinfo"
String disableSigmaKappaDetection [Input, Optional, default=No default value]
Optional parameter will disable the Sigma Kappa glitch detection algorithm. Specify parameter as disableSigmaKappaDetection="disableSigmaKappaDetection"
String disableLargeGlitchDetection [Input, Optional, default=No default value]
Optional parameter will disable the large glitch detection algorithm. Specify parameter as disableLargeGlitchDetection="disableLargeGlitchDetection"
String disableGlitchReconstruction [Input, Optional, default=No default value]
Optional parameter will disable the glitch reconstruction algorithm. Specify parameter as disableGlitchReconstruction="disableGlitchReconstruction"
DetectorTimeline out [OUTPUT, Optional, default=No default value]
Output product corrected for bath temperature drift

History

- 2009-10-10 - AS: [SCR-1401] Completed basic SimpleDeGlitcher prototype conversion.
- 2009-10-26 - AS: Remove unused dependencies.
- 2009-11-23 - AS: Add new filter algorithm (filter cascade).

- 2010-1-28 - AS: Add new correction mode (LOWPASSNOTCH). Change name to Deglitcher-Reconstructor
- 2010-04-13 - AS: [SPR-2502] When iterationNumber>1 (so Large Loop cycle is selected) in LOWPASSNOTCH mode the incorrect reconstruction timeline is used (the direct correction timeline instead of the inverse timeline).
- 2010-05-10 - AS: [SPR-2517] Make detectortimeline unique in diagnostic table.

1.173. SignalJumpDetectorTask

Full Name:	herschel.spire.ia.pipeline.common.sigjump.SignalJumpDetectorTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.sigjump import SignalJumpDetectorTask
Category:	task

Description

This task detects jumps in temperature and dark pixel detectors and marks them in the mask.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
Double kappa [Input, Optional, default=2.0]
String filterType [Input, Optional, default="DISCRETEDERIVATIVE"]
String glitchinfo [Input, Optional, default=No default value]
DetectorTimeline out [OUTPUT, Optional, default=No default value]

API details


Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT Input detector timeline
Double kappa [Input, Optional, default=2.0]
Optional glitch definition parameter ($\sigma * \text{kappa} - \text{mean}$) of filtered signal.
String filterType [Input, Optional, default="DISCRETEDERIVATIVE"]
Optional filter type (default is DISCRETEDERIVATIVE).
String glitchinfo [Input, Optional, default=No default value]
Optional parameter will provide a listing of glitch information. Specify parameter as <code>glitchinfo="glitchinfo"</code>
DetectorTimeline out [OUTPUT, Optional, default=No default value]
Output product corrected for bath temperature drift

History

- 2010-4-23 - AS: Initial release of SignalJumpDetector {SCR-2370}
- 2010-4-27 - AS: Install JUMP_THERMISTORS_DARKS_SIGNAL mask bit.


1.174. SingleBlockContext

Full Name:	herschel.spire.ia.dataset.context.SingleBlockContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import SingleBlockContext

History

- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-10-12 - PPa: Refactoring
- 2008-02-25 - PPa: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-04-28 - PPa: Add modelName in the rule (SPR-0597)
- 2008-10-15 - PPa: Add getBbid and similar methods (SCR-0942).
- 2009-01-23 - PPa: Add bbTypeName to context [SCR-1091]
- 2009-01-28 - PPa: Add bbTypeName to context also in setProducts method [SCR-1091]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098].

1.175. SourceFluxTask

Full Name:	herschel.spire.ia.pipeline.phot.pointsource.SourceFluxTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.phot.pointsource import SourceFluxTask
Category:	herschel/spire/ia/pipeline/phot/pointsource

Description

This task takes a Jiggle Point Source Fit Product and produces a Jiggled Photometer Product. This task is part of the jiggle pipeline, between Level 0.5 and Level 1.

Example


Example 1: A Jython example

```
from herschel.spire.ia.modules.userproducts import *
task = SourceFlux()
outputProduct = task.perform(inputProduct)
```

Limitations

The class is just an empty container for the time being. It takes the input product, performs a few basic checks to ensure that all the needed datasets are present, then builds an empty output product.


1.176. SpecBandEdge

Full Name:	herschel.spire.ia.dataset.SpecBandEdge
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecBandEdge
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions.


1.177. SpecBeamParam

Full Name:	herschel.spire.ia.dataset.SpecBeamParam
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecBeamParam
Category:	Calibration Product

History

- 2010-05-24 - SG: First version


1.178. SpecBolPar

Full Name:	herschel.spire.ia.dataset.SpecBolPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecBolPar
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2007-11-09 - PP: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.179. SpecBsmOps

Full Name:	herschel.spire.ia.dataset.SpecBsmOps
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecBsmOps
Category:	Calibration product

History

- 2006-09-20 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-12-01 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.
- 2010-04-06 - PPa: Don't use SpecObsModes [SPIRE-2255]


1.180. SpecBsmPos

Full Name:	herschel.spire.ia.dataset.SpecBsmPos
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecBsmPos

History

- 2006-03-06 - RG: change degrees in seconds_arc
- 2007-03-06 - RG: standardization of Yangle and Zangle uppercase only for the first letter
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.181. SpecChanGain

Full Name:	herschel.spire.ia.dataset.SpecChanGain
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanGain
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.182. SpecChanMask

Full Name:	herschel.spire.ia.dataset.SpecChanMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanMask
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.183. SpecChanNoise

Full Name:	herschel.spire.ia.dataset.SpecChanNoise
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanNoise
Category:	Calibration product

History

- 2008-01-28 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.184. SpecChanNomRes

Full Name:	herschel.spire.ia.dataset.SpecChanNomRes
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanNomRes
Category:	Calibration product

History

- 2008-06-25 - PP: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.185. SpecChanNum

Full Name:	herschel.spire.ia.dataset.SpecChanNum
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanNum
Category:	Calibration product

History

- 2008-01-29 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-09-27 - PPa: Fix type and description [SPIRE-1995]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.186. SpecChanTimeConst

Full Name:	herschel.spire.ia.dataset.SpecChanTimeConst
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanTimeConst
Category:	Calibration product

History

- 2006-08-22 - PP: First version (PP)
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.187. SpecChanTimeOff

Full Name:	herschel.spire.ia.dataset.SpecChanTimeOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecChanTimeOff
Category:	Calibration product

History

- 2006-08-22 - PP: First version (PP)
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.188. SpecDetAngOff

Full Name:	herschel.spire.ia.dataset.SpecDetAngOff
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecDetAngOff
Category:	Calibration product

History

- 2006-08-18 - PP: First version (PP)
- 2008-05-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-05-25 - PP: Use new ArrayEnum
- 2008-05-20 - PP: Change name.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.189. SpecElecCross

Full Name:	herschel.spire.ia.dataset.SpecElecCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecElecCross
Category:	Calibration product

History

- 2006-02-04 - PP: First version (PP)
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-12-01 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.190. SpecExplorer

Full Name:	herschel.spire.ia.gui.SpecExplorer
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import SpecExplorer

Description

SpecExplorer:

SpecExplorer Graphical Interface, Listeners, and core methods that allow the creation of SpecExplorer and all its requirements.

1. The SpecExplorer allows the user to perform the following:
 - a. 1. Display all the Detectors in the honeycomb arrangement.
 - b. 2. Plotting and Over-Plotting of scans of the detectors. Moreover, it allows manipulating and customizing all the plots.
 - c. 3. Instantiates the Mosaic and performs the necessary calls to reflect user selections.
 - d. 4. Perform and Maintain Plotting objects such as PlotterManager and PlotterPlus.
2. A single SpecExplorer contains only one product that is set to be displayed. Multiple instances of this class are possible.

Example

Example 1: How to run specExplorer through command-line


```
>> from herschel.spire.ia.dataset.gui import SpecExplorer
>> sdi = new SpectrometerDetectorInterferogram()
>> SpecExplorer(sdi)
>>
>> sds = new SpectrometerDetectorSpectrum()
>> SpecExplorer(sds)
```

History

- 2008-12-01 - TRF: First Release in main branch
- 2008-12-02 - TRF: Deleted some unused, deprecated code.
- 2008-12-02 - TRF: Cleaned up some javadoc and jtag warnings -- [SPIRE SPR-0678], [HCSS SPR-3232] Deleted commented code
- 2008-12-10 - TRF: [SPIRE SPR-1070] Temporarily changing init() from private to protected in order to implement an automated test harness
- 2008-12-10 - TRF: [SPIRE SPR-1070] Temporarily adding accessor methods for private attributes clickedPixel and ArrayDisplay in order to implement an automated test harness
- 2008-12-10 - TRF: [SPIRE SPR-1070] Temporarily changing singleClick() from private to protected in order to implement an automated test harness
- 2008-12-11 - TRF: [SPIRE SPR-1073] Fixed the problem when input data has columns of different sizes.

- 2008-12-11 - TRF: Minor refactoring in constructor, added a sanity check in setObject
- 2009-04-20 - YFR: [SPIRE SCR-1343] Removed references to pixel, replaced with refs to channel
- 2009-04-20 - YFR: Modified imports statements to make the class extends from a no deprecated class
- 2009-04-22 - TRF: Removed SuppressWarnings(serial) and replaced with serialVersionUID = 1L
- 2009-07-19 - TRF: Added serialVersionUID = 1L
- 2009-07-19 - TRF: [SPIRE SPR-1680] %pixelName% -> %channelName%
- 2009-07-20 - YFR: [SPIRE SCR-0817] Removed deprecated methods, replaced with equivalent methods
- 2010-01-16 TRF/DS [SPIRE-1765] Added/improved functionality. Various refactoring.
- 2010-02-03 - TRF: [INFR-365] Migrated from herschel.spire.ia.dataset.gui


1.191. SpecFlatField

Full Name:	herschel.spire.ia.dataset.SpecFlatField
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecFlatField
Category:	Calibration product

History

- 2006-08-18 - PP: First version
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.192. SpecFluxConversionTask

Full Name:	herschel.spire.ia.pipeline.spec.fluxconv.SpecFluxConversionTask
Alias:	SpecFluxConversionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.fluxconv import SpecFluxConversionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Flux Conversion for the SPIRE Spectrometer

Apply flux conversion to a Spectrometer Detector Spectrum.

Examples

Example 1: Using a SPIRE calibration tree prior to spire_cal_3_1:

```
fluxConv = obs.calibration.spec.fluxConvList.getProduct(biasMode, startDate)
asds = specFluxConversion(sds=asds, fluxConv=fluxConv)
```

Example 2: Using a the spire_cal_3_1 SPIRE calibration tree:

```
fluxConv = obs.calibration.spec.fluxConvList.getProduct(cmdRes, apodApplied,
biasMode, startDate)
asds = specFluxConversion(sds=asds, fluxConv=fluxConv)
```

Example 3: Create a Spectrometer Point Source Spectrum (must use spire_cal_4_0 or later):

```
fluxConv = obs.calibration.spec.fluxConvList.getProduct("HR", apodApplied,
biasMode, startDate)
if apodApplied:
    beamApod=Boolean.TRUE
else:
    beamApod=Boolean.FALSE
beamParam = obs.calibration.spec.beamParamList.getProduct(beamParamApod,
biasMode, startDate)
spss = specFluxConversion(sds=sds, fluxConv=fluxConv, beamParam=beamParam,
APPLY_POINT_SOURCE=Boolean.TRUE)
```

API Summary

Properties
SpectrometerDetectorSpectrum sds [Input, Mandatory, default=No default value]
SpecFluxConv SCalSpecFluxConv [Input, Mandatory, default=No default value]
SpecBeamParam SCalSpecBeamParam [Input, Optional, default=No default value]
Boolean APPLY_POINT_SOURCE [INPUT, OPTIONAL, default=False]
SpectrometerDetectorSpectrum outsds [OUTPUT, Optional, default=No default value]

API details

Properties


SpectrometerDetectorSpectrum sds [Input, Mandatory, default=No default value]
SDS: Non flux-calibrated spectra from a SPIRE FTS building block
SpecFluxConv SCalSpecFluxConv [Input, Mandatory, default=No default value]
SPIRE Flux conversion calibration product: contains the wavenumber-dependent extended source flux conversion factors for all Spectrometer detectors.
SpecBeamParam SCalSpecBeamParam [Input, Optional, default=No default value]
SPIRE Beam Parameter conversion calibration product: contains the wavenumber-dependent point-source flux conversion factors for the central detectors of each Spectrometer array.
Boolean APPLY_POINT_SOURCE [INPUT, OPTIONAL, default=False]
Set this keyword to true to apply the point-source flux conversion.
SpectrometerDetectorSpectrum outsds [OUTPUT, Optional, default=No default value]
SDS: Flux-calibrated spectra from a SPIRE FTS building block

History

- 2008-10-24 - TRF: First Version
- 2008-10-28 - TRF: Added some functionality to the convertFlux() method. Implements actual unit multiplication.
- 2008-10-28 - TRF: Added checkWnGrids() that compares the WN grids of the Spectra and the cal Product to ensure that the FluxFactor vectors are parametric with the flux and fluxError columns.
- 2008-10-29 - TRF: [SPIRE SPR-0967, SPIRE SPR-0982] TaskParameter "SCalSpecFluxConv" is now Mandatory
- 2008-10-30 - PPa: [SPIRE SPR-0983] Added extrapolation.
- 2008-10-30 - TRF: Modified original bug fix to try interpolation first, catch an IllegalArgumentException, print a WARNING, then try extrapolation.
- 2008-11-06 - TRF: [SPIRE SPR-0967] Migrated from herschel.spire.ia.pipeline.spec.ft package
- 2008-11-10 - TRF: Modified the check made on the input cal product. Tests revealed that the method chosen to correct SPIRE SPR-0982 was not sufficient.
- 2008-11-12 - TRF: [SPIRE SPR-1025] Fixed a bug in the LR/MR computation
- 2008-12-03 - TRF: [HCSS SPR-3232] Fixed some jtag urm documentation
- 2008-12-03 - TRF: Fixed unchecked warnings
- 2008-12-15 - TRF: Cleaned-up some in-line documentation

- 2009-01-23 - TRF: [SPIRE SPR-1158] Explicitly cast getFlux() as a Double1d
- 2009-01-23 - TRF: [SPIRE SPR-1165] Added serialVersionUID = 1L, removed SuppressWarnings
- 2009-01-23 - TRF: [SPIRE SPR-1161] Added fluxConv TaskParameter, deprecated SpecFluxConv TaskParameter
- 2009-01-23 - TRF: [SPIRE SPR-1098] Added throws InterruptedException to javadoc tag to execute()
- 2009-01-29 - TRF: [SPIRE SPR-1161] Removed SpecFluxConv TaskParameter
- 2009-02-04 - TRF: [SPIRE SPR-1204] keySet().iterator() -> getScanNames()
- 2009-03-17 - TRF: [SPIRE SPR-1107] Set warning messages when non-nominal channels are not found in cal products to config level
- 2009-05-06 - TRF: [SPIRE SCR-1343] pixel->channel
- 2009-06-13 - TRF: [SPIRE SPR-1511] Correctly set creator metadata in output sds.
- 2009-06-13 - TRF: [SPIRE SPR-1107] R/DP/T channels are removed from output product
- 2009-06-13 - TRF: [SPIRE SPR-1107] Do not catch exception if expected channel not found in calibration product.
- 2009-06-14 - TRF: [SPIRE SCR-1270] Change INFO message to count scans in ascending order
- 2009-07-20 - YFR: [SPIRE-1689] Removed deprecated methods, replaced with equivalent methods
- 2010-01-26 - TRF: [SPIRE-2278] Fixed Javadoc
- 2010-01-26 - TRF: Removed usage of deprecated AbstractComposite() methods
- 2010-05-21 - TRF: [SPIRE-2278] Fixed URM/Javadoc
- 2010-05-22 - TRF: [SPIRE-2576] Cal product is regridded to match resolution.
- 2010-06-08 - TRF: [SPIRE-2577] Added a new beamParam Task Parameter, keyword: APPLY_POINT_SOURCE. Creates a Spire Point Source Spectrum when set.


1.193. SpecFluxConv

Full Name:	herschel.spire.ia.dataset.SpecFluxConv
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecFluxConv
Category:	Calibration Product

History

- 2008-10-24 - TRF: First version
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed constructor description
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods

1.194. SpecInterRef

Full Name:	herschel.spire.ia.dataset.SpecInterRef
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecInterRef
Category:	Product

Description


SpecInterRef.

A product used to store information regarding the SCalSpecScalBlankSky Calibration Product. The 2 constructors.

History

- 2008-07-23 - DB: first version
- 2008-10-15 - PP: fix SPR-0926 (PPa)
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-02-04 - DB: from TF [SPIRE SPR-1200, 1204] Changed sdi.keySet() to sdi.getScanNumbers()
- 2009-03-17 - DB: SPR1289 add version in SpecInterRef metadata , add getVersion and setVersion method
- 2009-06-11 - DB: SPR 1375 & SCR 1374
- 2009-06-13 - DB: correct bug checkMeta
- 2009-12-11 - DB: add implements SpireCalibration SPR-SPIRE 2128


1.195. SpecLpfPar

Full Name:	herschel.spire.ia.dataset.SpecLpfPar
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecLpfPar
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-09-18 - TRF: This product contains only the spec table
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.196. SpecNlp

Full Name:	herschel.spire.ia.dataset.SpecNlp
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecNlp
Category:	Calibration product

History

- 2007-08-21 - PP: First version
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 [HCSS SPR-3232] TRF Fixed some jtag urm documentation
- 2009-06-12 [SPIRE SCR-1341] TRF Deprecated SSCAL phases, added wavenumber error column, changed Telescope to Sky
- 2009-10-16 - TRF: [SPIRE-2092] Removed deprecated methods
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.197. SpecNonLinCorr

Full Name:	herschel.spire.ia.dataset.SpecNonLinCorr
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecNonLinCorr
Category:	Calibration product

History

- 2006-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2006-02-25 - PP: Use new ArrayEnum
- 2008-10-07 - AS: Change to descriptive comments.
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2008-12-01 - TRF: Fixed some jcategory urm documentation [HCSS SPR-3232]
- 2009-01-15 - PPa: Add bias amplitudes in metadata [SPIRE SPR-1044]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.198. SpecNonLinearityCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.nonlin.SpecNonLinearityCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.nonlin import SpecNonLinearityCorrectionTask

Description

The task corrects detector voltage non-linearities.

The non-linearity correction function is $S1 = K1*(V3-V0)+K2*\ln(V3-K3)-K2*\ln(V0-K3)$, after which the flux correction $J = S1 * F$ is applied.

API Summary

Properties
DetectorTimeline sdt [Input, Mandatory, default=No default value]
SpecNonLinCorr nonLinCorr [Input, Mandatory, default=No default value]
DetectorTimeline out [OUTPUT, Optional, default=No default value]

API details

Properties


DetectorTimeline sdt [Input, Mandatory, default=No default value]
SDT Input detector timeline
SpecNonLinCorr nonLinCorr [Input, Mandatory, default=No default value]
Input calibration product of correction coefficients
DetectorTimeline out [OUTPUT, Optional, default=No default value]
Output product corrected for bolometer responsivity function

History

- 24 Jan 2008 : R Gastaud & A.Schwartz (convert prototype)
- 15 Feb 2008 AS Incorporate calibration tables into calculations.
- 04 Apr 2008 AS Modify correction equation to be consistent with new version of pipeline requirements doc
- "The SPIRE Analogue Signal Chain and Photometer Detector Data Processing Pipeline", Issue 4
- dated: Feb 20 2008, by Matt Griffin.
- 23 Jul 2008 AS Add tests for out of range input data and

- ill conditioned input data (data that results in negative argument to log).
- 28 Oct 2008 AS Add OOL (out of limit) masking for data that is out of range of the fitted portion of the bolometer responsivity curve.
- 16 Mar 2008 AS [SPR-1295] Modify test of detector timeline metadata, as well as exception message.
- Purpose is to check for input that has already been corrected.
- 24 Mar 2009 AS [SCR-1201] Add capability to process fewer than the maximum number of detectors in an array (if the product has fewer).
- 24 Apr 2009 AS [SCR-1047] Handshaking between NonLinearityCorrection and TempDrfitCorr calibration products.
- 08 May 2009 AS [SPR-1144] Tooltips and URM cleanup.
- 12 Aug 2009 AS [SPR-1141] Tooltips and URM cleanup.
- 21 Aug 2009 AS [SCR-1280] Set mask bits for data below K3 voltage and data for which V0flag or Kflag is NOT set in the calibration product. Do not process these data.
- 04 Sep 2009 AS [SCR-1144] More tooltips issues.
- [SCR-1651] Rename calibration table input parameter.
- 18 Sep 2009 AS [SCR-1144] Tooltips issue.
- Dependencies:
- uses


1.199. SpecOeOpd

Full Name:	herschel.spire.ia.dataset.SpecOeOpd
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecOeOpd
Category:	Calibration Product

History

- 2007-03-01 - PP: First version (PP)
- 2007-08-20 - TRF: Migrate to the new Unit API
- 2007-10-23 - ZW: Corrected throw exception with invalid pixels
- 2007-11-08 - ZW: Added Copy Constructor.
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-06-26 - PP: Use isSeeing() instead of deprecated isVisible()
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed descriptions
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.200. SpecOffsetHist

Full Name:	herschel.spire.ia.dataset.SpecOffsetHist
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecOffsetHist
Category:	Calibration product

History

- 2006-11-18 - PP: First version (PP)
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.

1.201. SpecOptCrossCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.optcross.SpecOptCrossCorrectionTask
Alias:	SpecOptCrossCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.optcross import SpecOptCrossCorrectionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task to apply Optical crosstalk correction to a Spectrometer Detector Spectrum

Example

Example 1: from HIPE:

```
ssds = specOptCrossCorrection(sds=ssds, optCross=obs.calibration.spec.optCross)
```

API Summary

Properties
SpectrometerDetectorSpectrum sds [Input, MANDATORY, default=No default value]
SCalSpecOptCross optCross [Input, OPTIONAL, default=No default value]
SpectrometerDetectorSpectrum outsds [OUTPUT, OPTIONAL, default=No default value]

API details

Properties


SpectrometerDetectorSpectrum sds [Input, MANDATORY, default=No default value]
SDS: Non corrected spectra from a SPIRE FTS building block
SCalSpecOptCross optCross [Input, OPTIONAL, default=No default value]
SPIRE Optical crosstalk matrix calibration product: contains the optical crosstalk correction factors for all Spectrometer detectors.
SpectrometerDetectorSpectrum outsds [OUTPUT, OPTIONAL, default=No default value]
SDS: Corrected spectra from a SPIRE FTS building block

History

- 2008-10-24 - TRF: First Version
- 2008-10-07 - TRF: [SPIRE SCR-0886] Added actual functionality

- 2008-11-13 - TRF: [SPIRE SPR-1033] Suppressed warning messages (set logging level to config)
- 2008-12-02 - TRF: [HCSS SPR-3232] Fixed some jtag urm documentation
- 2008-12-02 - TRF: Fixed unchecked warnings
- 2009-01-23 - TRF: [SPIRE SPR-1157] Explicitly cast getFlux() as a DoubleId
- 2009-01-23 - TRF: [SPIRE SPR-1165] Added serialVersionUID = 1L, removed SuppressWarnings
- 2009-01-23 - TRF: [SPIRE SPR-1160] Added optCross TaskParameter, deprecated SCalSpecOptCross TaskParameter
- 2009-01-23 - TRF: [SPIRE SPR-1098] Added throws InterruptedException to javadoc tag to execute()
- 2009-01-23 - TRF: [SPIRE SPR-1159] Renamed to SpecOptCrossCorrectionTask
- 2009-01-29 - TRF: [SPIRE SPR-1160] Removed SCalSpecOptCross TaskParameter
- 2009-02-04 - TRF: [SPIRE SPR-1204] keySet().iterator() -> getScanNames()
- 2009-03-24 - YFR: [SPIRE SPR-1286] Fixed the issue that OpticalCrossTalkCorrection performs incorrect matrix manipulation
- 2009-04-14 - YFR: [SPIRE SCR-1343] Changed deprecated methods calls
- 2009-04-17 - YFR: [SPIRE SPR-1345] Fixed the bug
- 2009-06-13 - TRF: [SPIRE SPR-1511] Correctly set creator metadata in output sds.
- 2009-06-13 - TRF: [SPIRE SPR-1107] Do not catch exception if expected channel not found in calibration product.
- 2009-06-14 - TRF: [SPIRE SPR-1181] INFO message now has scan numbers increasing
- 2009-06-14 - TRF: [SPIRE SCR-1270] Change INFO message to count scans in ascending order
- 2009-07-20 - YFR: [SPIRE-1692] Removed deprecated methods, replaced with equivalent methods
- 2009-08-28 - YFR: [SPIRE-1880] Ignored bits where master masks are set while correcting optical crosstalk
- 2009-09-02 - TRF: [SPIRE-1906] level metadata in output sds is now set to "10"
- 2010-01-26 - TRF: Removed usage of deprecated AbstractComposite() methods
- 2010-01-26 - TRF: [SPIRE-2282] Fixed URM entries
- 2010-05-18 - TRF: [SPIRE-2282] Fixed jcategory URM entry


1.202. SpecOptCross

Full Name:	herschel.spire.ia.dataset.SpecOptCross
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecOptCross
Category:	Calibration product

History

- 2008-05-20 - PP: First version (PP)
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.203. SpecPcal

Full Name:	herschel.spire.ia.dataset.SpecPcal
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecPcal
Category:	Calibration product

History

- 2008-02-08 - PP: Fix description
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2010-02-05 - YFR: [SPIRE-2255] Added associated tasks' links.


1.204. SpecPhaseCorrLim

Full Name:	herschel.spire.ia.dataset.SpecPhaseCorrLim
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecPhaseCorrLim
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2008-02-25 - PP: Use new makeTable method
- 2009-06-03 - PPa: Include termistors, resistors and dark [SPIRE-SPR-1107]
- 2009-06-04 - PPa: Renamed from BandEdge [SCR-1378]
- 2009-06-19 - PPa: Throw IllegalArgumentException when an unknown channel is used. [SPIRE-SPR-1107]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods.


1.205. SpecScalEm

Full Name:	herschel.spire.ia.dataset.SpecScalEm
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecScalEm
Category:	Calibration product

History

- 2007-08-21 - CO: First version
- 2007-09-06 - TRF: Fixed javadoc
- 2007-09-18 - PP: Change name, add serialVersionUID
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed descriptions
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.206. SpecScalRsrf

Full Name:	herschel.spire.ia.dataset.SpecScalRsrf
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecScalRsrf
Category:	Calibration product

History

- 2007-08-21 - CO: First version
- 2007-09-06 - TRF: Fixed some Javadoc
- 2007-09-18 - PP: change name
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.207. SpecSmecStepFactor

Full Name:	herschel.spire.ia.dataset.SpecSmecStepFactor
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecSmecStepFactor
Category:	Calibration product

History

- 2007-09-07 - TRF: First version
- 2007-11-07 - ZW: Added copy constructor
- 2007-11-09 - ZW: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2009-06-03 - PPa: Remove N channels [SPR-1376].
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.208. SpecSmecZpd

Full Name:	herschel.spire.ia.dataset.SpecSmecZpd
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecSmecZpd
Category:	Calibration Product

History

- 2006-06-27 - TRF: Added some javadoc
- 2006-08-24 - TRF: Added private static final long serialVersionUID = 11;
- 2006-10-05 - PP: Rewritten
- 2006-10-17 - PP: changed in SpecSmecZpd, merging SCalSpecOeZpd and SCalSpecLvdtZpd
- 2007-04-02 - TRF: Added get/set methods for units. Also added default LVDT value and default UNITS for optEnc and LVDT columns
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-08-24 - AH: Remove deprecated methods using nT.quantity
- 2007-09-06 - TRF: Fixed some Javadoc
- 2007-09-07 - TRF: Removed deprecated tag on some methods
- 2007-10-22 - TRF: Changed setLvdtErrorUnits() to setLvdtErrorUnit()
- 2007-10-24 - TRF: Added copy constructor.
- 2007-11-09 - TRF: Added copy() method
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-02-25 - PP: Use new ArrayEnum
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2009-06-03 - PPa: Include termistors, resistors and dark [SPIRE-SPR-1107]
- 2009-06-04 - Remove: N channels
- 2010-02-05 - YFR: [SPIRE-2255] Fixed description
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods


1.209. SpecTempDriftCorr

Full Name:	herschel.spire.ia.dataset.SpecTempDriftCorr
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpecTempDriftCorr
Category:	Product

History

- 2006-02-28 - AS: Initial Release
- 2008-10-07 - AS: Descriptive comments.
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]

1.210. SpectrometerDetectorInterferogram

Full Name:	herschel.spire.ia.dataset.SpectrometerDetectorInterferogram
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpectrometerDetectorInterferogram
Category:	Product

Description

SpectrometerDetectorInterferogram.

A product used to store information regarding the Spectrometer Detector Interferogram. The 5 constructors.

Example

Example 1: from HIPE:

```
# initialize settings
channel = -"SLWC3"
scan = 1
size = 10
initSig = 2
#create sdi with one scan, one channel -- -"SLWC3", and set its signal data
to [2,2,2,2,2,2,2,2,2,2]
sdi = SpectrometerDetectorInterferogram()
sibd = SpireInterferogramCompositeDataset(scan)
sild = SpireInterferogramId(channel)
sig = DoubleId(size,initSig)
sild.setSignalData(sig)
sibd.setChannel(sild)
sdi.setScan(sibd)
#example of add integer
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi + 5
print output.getScan(scan).getChannel(channel).getSignalData()
#[7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0]
print -"
#example of subtract integer
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi -- 5
print output.getScan(scan).getChannel(channel).getSignalData()
#[-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0]
print -"
#example of multiply integer
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi * 5
print output.getScan(scan).getChannel(channel).getSignalData()
#[10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0]
print -"
#example of divide integer
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi -/ 5
print output.getScan(scan).getChannel(channel).getSignalData()
#[0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4]
print -"
#example of add double
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi + 5.0
print output.getScan(scan).getChannel(channel).getSignalData()
#[7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0,7.0]
```

Example 1: from HIPE:

```

print ""
#example of subtract double
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi -- 5.0
print output.getScan(scan).getChannel(channel).getSignalData()
#[-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0,-3.0]
print ""
#example of multiply double
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi * 5.0
print output.getScan(scan).getChannel(channel).getSignalData()
#[10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0]
print ""
#example of divide double
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi / 5.0
print output.getScan(scan).getChannel(channel).getSignalData()
#[0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4,0.4]
print ""
#make a copy
copy = sdi.copy()
#example of add sdi
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi + copy
print output.getScan(scan).getChannel(channel).getSignalData()
#[4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0]
print ""
#example of subtract sdi
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi -- copy
print output.getScan(scan).getChannel(channel).getSignalData()
#[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]
print ""
#example of multiply sdi
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi * copy
print output.getScan(scan).getChannel(channel).getSignalData()
#[4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0]
print ""
#example of divide sdi
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi / copy
print output.getScan(scan).getChannel(channel).getSignalData()
#[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]
print ""
#make interferogram
interferogram = SpireInterferogram1d(channel)
signal = Double1d.range(size)
signal = signal.add(1)
interferogram.setSignalData(signal)
#example of add interferogram
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi + interferogram
print output.getScan(scan).getChannel(channel).getSignalData()
#[3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0]
print ""
#example of subtract interferogram
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi -- interferogram
print output.getScan(scan).getChannel(channel).getSignalData()
#[1.0,0.0,-1.0,-2.0,-3.0,-4.0,-5.0,-6.0,-7.0,-8.0]

```

Example 1: from HIPE:

```

print -""
#example of multiply interferogram
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi * interferogram
print output.getScan(scan).getChannel(channel).getSignalData()
#[2.0,4.0,6.0,8.0,10.0,12.0,14.0,16.0,18.0,20.0]
print -""
#example of divide interferogram
print sdi.getScan(scan).getChannel(channel).getSignalData()
#[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
output = sdi / interferogram
print output.getScan(scan).getChannel(channel).getSignalData()
#[2.0,1.0,0.6666666666666666,0.5,0.4,0.3333333333333333,0.2857142857142857,0.25,0.2222222222222222]
print -""

```


History

- 2006-03-14 - TRF: Replaced the deprecated PixelMapper() with BolArray() and Bolometer()
- 2006-05-25 - PP: Cleanup un-needed code.
- 2006-06-15 - PP: small clean-up
- 2006-06-19 - PP: Fix nextScan calculation and more other miscellaneous corrections
- 2006-06-27 - TRF: Added some javadoc
- 2006-08-25 - TRF: Added copyright notice
- 2007-06-04 - AH: Added javadoc. Fixed javadoc conflicts
- 2007-06-07 - TRF: Deprecated obsolete methods. Fixed initMeta() so that numScans metaData is set.
- 2007-06-13 - TRF: Replaced portion of getScan() method to satisfy ft test harnesses.
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-10-12 - ZW: Delete all the deprecated methods.
- 2007-10-22 - TRF: Fixed behaviour of removeScan() and getScan()
- 2007-11-07 - TRF: Fixed behaviour of __getitem__(), __setitem__()
- 2007-11-08 - ZW: Added method copy()
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-05-08 - PK: Changed getNumScans so that it does not get its value from the actual number of scans in the Sdi contains but rather from the numScans MetaData.
- 2008-05-08 - PK: Changed setScan(SSCD) and removeScan(string) so that they update the numScans MetaData to the number of scans in the Sdi whenever they are called.
- 2008-05-08 - ES: Added getCommandedResolution and setCommandedResolution methods, modified initMeta to create a commandedResolution parameter.
- 2008-05-08 - ES: Fixed deleteScan methods to alter numScans MetaData appropriately
- 2008-05-13 - ES: Added propagateMeta() method

- 2008-05-15 - ES: Fixed propagateMeta() method to stop removing parameters
- 2008-05-16 - ES: Added missing MetaData to propagateMeta() method
- 2008-07-22 - ES: Added getUniqueDetectors(), to return a list of the unique detector names contained in the product, along with getScanNames(String) and getScanNumbers(String), which provide a list of the scan numbers that contain a given named detector, all as per SCR-0784.
- 2008-10-15 - TRF: Added get/setQcPhaseWrap(), accessor methods for the QcPhaseWrap MetaData keyword. [SPIRE SPR-0871]
- 2008-11-07 - TRF: Removed reference to deprecated DetectorTimeline.getCommandedResolution(). [SPIRE SPR-0871]
- 2008-11-07 - TRF: Modified propagateMeta() so that it propagates all SDT metaData but a select few, rather than the other way around
- 2008-11-21 - TRF: Removed redundant cast in getScanNames() [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNumbers() method to ignore 'History' as key
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScan(String) method to throw exception when get(scanNo) returns a History object
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNumbers(String) method to filter out 'History' scan number
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNames(String) method to ignore 'History' as key
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNames() method to ignore 'History' as key
- 2009-03-17 - TRF: Removed SuppressWarnings()
- 2009-04-16 - TRF: [SPIRE SCR-1343] Removed references to pixel, replaced with refs to channel
- 2009-06-05 - TRF: [SPIRE SCR-1343] Removed references to pixel, replaced with refs to channel
- 2009-06-09 - YFR: [SPIRE SCR-1184] Added arithmetic overloading methods
- 2009-09-08 - TRF: [SPIRE-1656] Added new metadata "apodType" and "phaseCorrApplied" as well as accessor methods
- 2009-10-17 - DMT: [SPIRE-1362] Added new metadata "missedFringes", "scanExtrema", "smecTemperature", "stageSpeedAverage" and "stageSpeedDeviation"
- 2009-10-21 - DMT: [SPIRE-1364] Added new metadata "phaseWrapSSW" and "phaseWrapSLW", removed "qcPhaseWrap"
- 2010-02-01 - TRF: Removed deprecated AbstractComposite() methods.
- 2010-02-01 - TRF: [SPIRE-2255] Fixed some URM and Javadoc
- 2010-02-01 - TRF: [SPIRE-2406] apodName metadata and accessor methods added. Fixed typo in apodType metadata

- 2010-02-22 - WT: [SPIRE-2376] Added new metadatum "smecOpenLoopCount"
- 2010-05-21 - TRF: [SPIRE-2468] Deprecated QC metadata accessor methods. Initial values of QC metadata no longer NaN.

1.211. SpectrometerDetectorSpectrum

Full Name:	herschel.spire.ia.dataset.SpectrometerDetectorSpectrum
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpectrometerDetectorSpectrum
Category:	Product

Description

Herschel-SPIRE Spectrometer Detector Spectrum Product

A class for generating empty SDS products.


History

- 2006-05-08 - KA: Class now extends DetectorTimeline instead of BuildingBlockProduct
- 2006-05-08 - KA: Added a temporary constructor that requires no arguments
- 2006-05-08 - KA: TODO: Update constructor with no arguments to provide more functionality
- 2006-05-09 - KA: Added 3 methods: setScan(int, SpectrumDataset[]), setScan(String, SpectrumDataset[]), and setScan(SpectrumDataset[]). These methods allow the setting of scan data
- 2006-05-09 - KA: Made the default constructor (one without arguments) to create one default scan (using getSpectrumSequence())
- 2006-05-10 - KA: Added __getitem__ and __setitem__ commands for jython usage
- 2006-05-10 - KA: Changed setScans from using SpectrumDataset arrays to CompositeDatasets
- 2006-05-10 - KA: Removed a lot of direct Metadata setting of data. Using provided setName methods (ie setObsid, setBbid, etc.)
- 2006-05-10 - KA: Added accessor methods for any metadata items that aren't already covered by super classes.
- 2006-05-18 - TRF: Added resolution metaData and get/setResoution() methods.
- 2006-05-25 - PP: Miscellaneous corrections
- 2006-06-19 - PP: Fix nextScan calculation and more other miscellaneous corrections
- 2006-06-27 - TRF: Fixed some javadoc
- 2006-08-24 - TRF: Added copyright notice
- 2007-03-01 - TRF: Changed jtag jcategory Task to jcategory Product to resolve HCSS SPR-2457
- 2007-03-27 - TRF: Added a setDescription() call in the constructor.
- 2007-06-07 - TRF: Deprecated obsolete methods.
- 2007-07-30 - AH: Migrate to new Unit API
- 2007-09-06 - TRF: Fixed some Javadoc
- 2007-10-12 - ZW: Delete all the deprecated methods.

- 2007-10-15 - TRF: Removed checkMeta() as it was redundant. Cleaned up some javadoc.
- 2007-10-22 - TRF: Fixed behaviour of getScan()
- 2007-11-05 - ZW: Fixed behaviour of removeScan(String)
- 2007-11-09 - ZW: Added constructor(description)
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-05-08 - PK: Changed getNumScans so that it does not get its value from the actual number of scans in the Sds contains but rather from the numScans MetaData.
- 2008-05-08 - PK: Changed setScan(SSCD) and removeScan(string) so that they update the numScans MetaData to the number of scans in the Sds whenever they are called.
- 2008-05-08 - PK: Added getActualResolution and setActualResolution methods, modified initMeta to create an actualResolution parameter.
- 2008-05-08 - ES: Added getCommandedResolution and setCommandedResolution methods, modified initMeta to create a commandedResolution parameter. Fixed deleteScan methods to alter numScans appropriately
- 2008-05-13 - ES: Added propagateMeta() method
- 2008-05-15 - ES: Fixed propagateMeta() method to stop removing parameters
- 2008-05-16 - ES: Added missing MetaData to propagateMeta() method
- 2008-07-22 - ES: Added getUniqueDetectors(), to return a list of the unique detector names contained in the product, along with getScanNames(String) and getScanNumbers(String), which provide a list of the scan numbers that contain a given named detector, all as per SCR-0784
- 2008-10-15 - TRF: Added get/setQcPhaseWrap(), accessor methods for the QcPhaseWrap MetaData keyword. [SPIRE SPR-0871]
- 2008-11-21 - TRF: Removed redundant cast in getScanNames() and removeScan(int scan) [SPIRE SPR-XXXX]
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNumbers(String) method to filter out input string 'History'
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScan(String) method to throw exception when get(scanNo) returns a History object
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNumbers() method to filter out 'History' key
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNames(String) method to filter out 'History' input
- 2009-02-03 - YFR: [SPIRE SPR-1200] Modified getScanNames() method to filter out 'History' key
- 2009-03-17 - TRF: [SPIRE SPR-1245] Modified propagateMeta() to copy start/endDate correctly.
- 2009-03-17 - TRF: [SPIRE SPR-1245] Overloaded propagateMeta() to correctly propagate from one sds to another.

- 2009-04-16 - TRF: [SPIRE SCR-1343] Deprecated references to pixel, replaced with refs to channel
- 2009-06-05 - TRF: [SPIRE SCR-1343] Removed references to pixel, replaced with refs to channel
- 2009-06-10 - TRF: [SPIRE SCR-1454] Removed propagateMeta()
- 2009-09-08 - TRF: [SPIRE-1656] Added new metadata "apodType" and "phaseCorrApplied" as well as accessor methods
- 2009-09-22 - DMT: [SPIRE-1366] Added new metadata "ratioSpectralOutliers" and "ratioSpectralDifference" and accessor methods
- 2010-02-01 - TRF: Removed deprecated AbstractComposite() methods.
- 2010-02-01 - TRF: [SPIRE-2255] Fixed some URM and Javadoc
- 2010-05-21 - TRF: [SPIRE-2468] Deprecated QC metadata accessor methods. Initial values of QC metadata no longer NaN.
- 2010-05-21 - TRF: [SPIRE-2595] Made initMeta() and setNumScans() visible for SpectrometerPointSourceSpectrum.

1.212. SpectrometerPointSourceSpectrum

Full Name:	herschel.spire.ia.dataset.SpectrometerPointSourceSpectrum
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpectrometerPointSourceSpectrum
Category:	Product


Description

Herschel-SPIRE Spectrometer Point Source Spectrum Product.

History

- 2010-05-21 - TRF: [SPIRE-2595] First version.

1.213. SpireCal

Full Name:	herschel.spire.ia.cal.SpireCal
Type:	Java Class - 
Import:	from herschel.spire.ia.cal import SpireCal

Description

Defines the SPIRE calibration tree and provides means of loading them.

The structure of the calibration tree looks like this. (This is slightly simplified as it does not show the specific editions available where dependencies exist).

cal.		
	phot.	
		chanNoise
		rsrf
	spec.	
		bandEdge
		interRef
		nlp
		nonLinCorr
		phaseCorrLim
		smecStepFactor
		smecZpd
	[both].	
		beamProf
		bolPar
		bsmOps
		bsmPos
		chanGain
		chanMask
		chanNomRes
		chanNum
		chanTimeConst
		chanTimeOff
		detAngOff
		fluxConv
		elecCross
		instModeMask
		LpfPar
		offsetHist
		optCross
		tempDrift

For documentation of the full API see the developer reference Javadoc.

Examples

Example 1: Accessing calibration data from an observation context

```
cal = obs.calibration
print cal.phot.bsmPos
print cal.spec.smecZpd
```

Example 2: Loading a stored calibration tree

```
# Loads from preconfigured properties spire.cal.urn and/or spire.cal.pool
cal = SpireCal.getInstance()
#
# Loads from a local pool -"myCal"
cal = SpireCal.getInstance (ProductStorage ("myCal"))
#
# Loads a known version from the HSA interface
cal = SpireCal.getInstance (ProductStorage(HsaReadPool()), IdLookup
("spire_cal_2_0"))
```

Example 3: Configuring defaults

```
# Set default applicability to current time
from herschel.share.fltdyn.time import FineTime
from java.util import Date
now = FineTime (Date())
#
# Two ways of doing it
print cal.date
cal.date = now
print cal.date
cal.setDefault ("time", now)
print cal.date
```


Example 4: Updating an existing calibration tree in an observation to a new one

```
obs.calibration.update (newCalTree)
```

See also

- [SpireCal](#)
- http://www.spire.rl.ac.uk/icc/product_definitions/index.html

1.214. spireCal

Full Name:	herschel.spire.ia.cal.SpireCalTask
Alias:	spireCal
Type:	Java Task - 
Import:	from herschel.spire.ia.cal import SpireCalTask
Category:	SPIRE task

Description

SPIRE Calibration Task.

This task is used to get a SPIRE calibration tree from the archive or from a jar file.

Examples

Example 1: Get version 2_1 of the SPIRE calibration tree from the archive.

```
cal = spireCal (calTree="spire_cal_2_1")
```

Example 2: Read a calibration tree from a jar file.

```
cal = spireCal (jarFile="C:/spire/scripts/cal/spire_cal_2_1.jar")
```

API Summary

Jython Syntax

```
cal = spireCal ([id=cal-tree-id], [jar=jar-file])
```

Properties

[String calTree](#) [INPUT, OPTIONAL, default=default derived from spire.cal.jar property]

[String jarFile](#) [INPUT, OPTIONAL, default=No default value]

[SpireCal cal](#) [OUTPUT, MANDATORY, default=No default value]

API details

Properties

[String calTree](#) [INPUT, OPTIONAL, default=default derived from spire.cal.jar property]

Input id of calibration tree.

[String jarFile](#) [INPUT, OPTIONAL, default=No default value]

Name of a jar file containing the calibration tree. Overrides the id parameter if both are present.

[SpireCal cal](#) [OUTPUT, MANDATORY, default=No default value]

SPIRE calibration tree.


See also

- [SpireCal](#)

History

- 2010-02-25 - SG: First draft version

1.215. SpireCalibrationProduct

Full Name:	herschel.spire.ia.dataset.SpireCalibrationProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireCalibrationProduct


Example

Example 1: creation:
<pre>\$ x=SpireCalibrationProduct(description="This is my SPIRE calibration product",modelName="Flight") \$ x.creator="Me, myself and I" \$ x.version="0.1" \$ x["table"]=aTable</pre>

History

- 2006-01-10 - PP: First version (PP)
- 2006-08-10 - PP: Add serialVersionUID
- 2007-09-18 - PP: Fix copy constructor
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some javadoc [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-03-13 - SG: SCR-1059: dependency
- 2009-11-23 - SG: SCR-2128: implements HistoryIdentifiable
- 2009-12-10 - SG: SCR-2128: implements SpireCalibration
- 2010-02-05 - YFR: [SPIRE-2255] Fixed typo 'addes'

1.216. SpireDataProduct

Full Name:	herschel.spire.ia.dataset.SpireDataProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireDataProduct


Example

Example 1: creation:
<pre>\$ x=SpireDataProduct(description="This is my SPIRE data product",modelName="Flight") \$ x.creator="Me, myself and I" \$ x.instMode="POF1" \$ x.obsid=0x3000C251 \$ x["Spectrum"]=aSpectrum</pre>

History

- 2006-01-14 - PP: First version.
- 2006-02-06 - PP: Add more constructors.
- 2006-02-10 - PP: Now extends SpireProduct instead of Product.
- 2006-02-20 - PP: Change AOT into Observing mode.
- 2006-03-10 - PP: Add getObsIdHex() method.
- 2006-03-13 - PP: Improve Javadoc.
- 2006-03-31 - PP: Fix obsId description. Improve Javadoc.
- 2006-04-19 - PP: Start changing obsId into obsid, as agreed with the HDPG.
- 2006-06-29 - PP: Remove the deprecated method getBbId().
- 2006-07-04 - PP: Remove the deprecated method setObsId().
- 2006-08-10 - PP: Add serialVersionUID
- 2007-09-18 - PP: Fix copy constructor
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-05-22 - PP: Use ObsParameter Enum to add all compulsory metadata keywords. Introduce instMode.
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2009-06-02 - PPa: Add propagateMeta(Product) method [SPIRE SCR-1454]. Don't propagate fileName anymore.
- 2009-08-04 - PPa: Change propagateMeta method to exclude "level" [SPIRE-1596]
- 2010-02-05 - YFR: [SPIRE-2255] Fixed typo 'compilsory'


1.217. SpireInterferogram1d

Full Name:	herschel.spire.ia.dataset.SpireInterferogram1d
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireInterferogram1d
Category:	Dataset

History

- 2006-06-27 - TRF: Changed some javadoc
- 2006-12-18 - TRF: Added Float1d options to setOpd(), setSignal(), setErrorSig(), and setErrorOpd()
- 2007-03-01 - TRF: Changed units of opd and errorOpd columns from METRES to CENTIMETRES.
- 2007-03-26 - TRF: Added new methods to access data and units.
- 2007-03-26 - TRF: setPixelName() now a public method.
- 2007-03-27 - TRF: accessor methods now feature generic ArrayData
- 2007-06-07 - TRF: Deprecated some obsolete methods.
- 2007-07-30 - AH: Migrate to new Unit API, add new methods for set/get of Unit, deprecated old methods for set/get of Units
- 2007-08-24 - AH: Remove deprecated methods using nT.quantity
- 2007-10-12 - ZW: Delete all the deprecated methods.
- 2007-10-22 - TRF: Deleted unused set() methods
- 2008-02-25 - PP: Remove serialVersionUID: TableDataset doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-03-11 - YFR: [SPIRE SCR-1290] Added Ra and Dec attributes
- 2009-03-17 - YFR: [SPIRE SCR-1290] Added initMeta method to initialize Ra, Dec, and pixel name
- 2009-03-17 - YFR: [SPIRE SCR-1290] Added description for pixel name
- 2009-03-20 - YFR: [SPIRE SPR-1319] Initialized ra and dec values to be -1 to fix saving product problem
- 2009-04-14 - YFR: Modified RA_DESCRIPTION and DEC_DESCRIPTION
- 2009-04-14 - YFR: Removed redundant castings
- 2009-04-16 - YFR: [SPIRE SCR-1343] Deprecated methods with name 'pixel', added new methods with name 'channel'
- 2009-06-05 - TRF: [SPIRE SCR-1343] Removed deprecated methods with name 'pixel'

1.218. SpireInterferogramCompositeDataset

Full Name:	herschel.spire.ia.dataset.SpireInterferogramCompositeDataset
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireInterferogramCompositeDataset
Category:	Dataset


Description

Herschel-SPIRE SpireInterferogramCompositeDataset Dataset

History

- 2006-06-27 - TRF: Changed some javadoc
- 2006-10-20 - TRF: added getScanDir/setScanDir methods
- 2007-06-04 - AH: added some javadoc. resolved javadoc conflicts
- 2007-06-07 - TRF: Deprecated obsolete methods. Deprecated get/setCount as that metaData parameter is being phased out in favour of scanNumber.
- 2007-10-12 - ZW: Delete all the deprecated methods.
- 2007-10-22 - TRF: Fixed behaviour of constructors where String scanNo cannot be parsed to a long
- 2007-10-23 - ZW: Fixed constructors where String scanNo cannot be parsed to a long.
- 2007-10-23 - ZW: Fixed behaviour of removePixel() method
- 2007-11-08 - ZW: Added method copy()
- 2008-02-25 - PP: Remove serialVersionUID: Dataset doesn't implement Serializable anymore
- 2008-11-21 - TRF: Removed redundant cast in hasPixel() and getPixelNames() [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-04-16 - YFR: [SPIRE SCR-1343] Deprecated methods with name 'pixel', added new methods with name 'channel'
- 2009-04-16 - YFR: [SPIRE SCR-1343] Deprecated methods with name 'pixel', added new methods with name 'channel'
- 2010-02-01 - TRF: Removed deprecated AbstractComposite() methods.
- 2010-02-01 - TRF: [SPIRE-2255] Fixed some URM and Javadoc
- 2010-02-01 - TRF: Removed deprecated AbstractComposite() methods.
- 2010-02-01 - TRF: [SPIRE-2255] Fixed some URM and Javadoc


1.219. SpireListContext

Full Name:	herschel.spire.ia.dataset.context.SpireListContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import SpireListContext

History

- 2007-10-12 - PPa: First version
- 2008-02-25 - PPa: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2009-02-23 - PPa: Add addRef(ProductRef) method [SPIRE SCR-1244]
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]. Remove setting of creator.


1.220. SpireMapContext

Full Name:	herschel.spire.ia.dataset.context.SpireMapContext
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import SpireMapContext

History

- 2007-04-23 - PPa: Fix serialization exception (E2E tests)
- 2007-20-12 - PPa: Change name and refactoring
- 2008-02-25 - PPa: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2009-04-28 - PPa: Remove unneeded exception declarations, improve javadocs [SPR-1098]. Remove setting of creator.
- 2009-06-11 - PPa: Remove setting of fileName [SPR-1390].


1.221. SpireParameter

Full Name:	herschel.spire.ia.dataset.SpireParameter
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireParameter

History

- 2008-06-26 - PP: First version
- 2008-07-01 - PP: Set bias freq unit.
- 2008-07-09 - PP: Add bias amplitude keywords.
- 2008-07-16 - PP: Add copy() in getParameter().
- 2008-11-04 - PP: Add PTC bias amplitude.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-02-18 - PPa: Fix scanLineNum description [SPIRE SPR-1238]
- 2009-03-03 - SG: Add cal dependency [SPIRE SCR-1059]

1.222. SpirePhotLargeMapPipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpirePhotLargeMapPipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpirePhotLargeMapPipelineTask
Category:	task

Description

Task to execute the SPIRE Photometer Large Map Pipeline (POF5)

This task executes the SPIRE Photometer Large Map Pipeline (POF5)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String mapping [INPUT, OPTIONAL, default="naive"]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

ObservationContext obs [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
String mapping [INPUT, OPTIONAL, default="naive"]
Mapping algorithm
String level [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-29 - PPa: First version
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]

1.223. SpirePhotParallelPipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpirePhotParallelPipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpirePhotParallelPipelineTask
Category:	task

Description

Task to execute the SPIRE Photometer Parallel Pipeline (PARALLEL)

This task executes the SPIRE Photometer Parallel Pipeline (PARALLEL)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String mapping [INPUT, OPTIONAL, default="naive"]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

ObservationContext obs [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
String mapping [INPUT, OPTIONAL, default="naive"]
Mapping algorithm
String level [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-29 - PPa: First version
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]. Use PARALLEL script [SPR-1230]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]

1.224. SpirePhotPointSourcePipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpirePhotPointSourcePipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpirePhotPointSourcePipelineTask
Category:	task

Description

Task to execute the SPIRE Photometer Point Source Pipeline (POF2)

This task executes the SPIRE Photometer Point Source Pipeline (POF2)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

ObservationContext obs [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
String level [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-29 - PPa: First version
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]

1.225. SpirePhotSmallMapPipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpirePhotSmallMapPipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpirePhotSmallMapPipelineTask
Category:	task

Description

Task to execute the SPIRE Photometer Small Map Pipeline (either POF3 or POF10)

This task executes the SPIRE Photometer Small Map Pipeline (either POF3 or POF10)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

<code>ObservationContext obs</code> [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
<code>String level</code> [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-29 - PPa: First version
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]
- 2010-01-28 - PPa: Recognise new Small Map implementation.

1.226. SpirePointingProduct

Full Name:	herschel.spire.ia.dataset.SpirePointingProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpirePointingProduct

Description

This class implements the Spire Pointing Product.

The Spire Pointing Product is a context containing the Herschel Pointing Product, the SIAM product, the DetAngOff product and the Bsm Angle Timeline. This class allows the user to obtain the position of one or more SPIRE detectors on the sky at a given time.

Examples

Example 1: Creation:

```
$ spp=SpirePointingProduct(hpp=obs.auxiliary.pointing,siam=obs.auxiliary.siam)
$ spp.detAngOff=obs.calibration.phot.detAngOff
$ spp.bat=bat
```

Example 2: Obtain the Gyro propagated sky position of the PSWE8 and PSWE4 at a given time

```
$ pdt=obs.level0_5.getProduct(5).pdt
$ ra,dec=spp.getSkyPosition(["PSWE8","PSWE4"],pdt.sampleTime[0],"gyro")
```


History

- 2006-02-01 - EH: First version
- 2006-02-01 - EH: add method setMeta (this simplify the code an makes it compatible java 1.4) and set the description to "Spire Pointing Product", add a new table for offsets TableOffsetMake
- 2006-02-01 - EH: add getPixelDec getPixelRa
- 2006-02-20 - PP: standardize TableOffsetMake and TableMake methods. Moved in the herschel.spire.ia.dataset package (PP)
- 2006-03-08 - PP: some code clean up. Remove autoboxing usage
- 2006-03-10 - EH: product redefinition; two separate TableDatasets (HPP and BSM) store HPP and BSM timelines (EH)
- 2006-03-21 - PP: change makeHPPTable() -> makeHppTable() and makeBSMTable() -> makeBsmTable().
- 2006-03-21 - PP: Fix getDec(), getOnBoardTime().
- 2006-04-06 - RG: standardization of Yangle and Zangle uppercase only for the first letter (RG)
- 2006-04-14 - PP: all spacecraft and bsm TableDatasets setters removed; HPP and BAT copied in SPP
- 2006-05-23 - PP: Code cleanup

- 2006-06-16 - EH: metadata keyword "detector" added, with default value "PHOT"; getYangle and getZangle return 0.0 if bsm TableDataset not present
- 2006-06-19 - PP: Allow extrapolation for LinearInterpolator. Fix getYangle and getZangle. Add new getPixelDec/Ra methods.
- 2006-06-21 - PP: Use the Pointing Product API
- 2006-06-26 - PP: Add new getPixelDec/Ra methods.
- 2006-11-13 - PP: Use PixAngOffsetTable. Implement set method.
- 2007-08-01 - NC: Add a MapContext
- 2007-09-09 - PP: Improvement of the getSkyPosition method to get the position of pixels
- 2007-09-21 - NC: Code cleanup of several classes, add serialVersionUID to PixAngOff classes.
- 2007-10-24 - NC: Add new SiamProduct to be used in the the getSkyPosition method. This product contains the information of the relative offset between the spacecraft pointing and the spire apertures.
- 2007-11-23 - PP: Make it work also in the case that the SiamProduct was not set
- 2007-11-23 - PP: Fix directions creation when pointing is out of range: z/yoffset are in seconds
- 2008-02-11 - PP: Enforce rules. Use PSWE8 pixel as SIAM reference.
- 2008-02-25 - PP: Remove serialVersionUID: Context doesn't implement Serializable anymore
- 2008-05-19 - PP: Fix getSkyPosition
- 2008-05-27 - PP: Use DetAngOff instead of PixAngOff
- 2008-06-10 - PP: Fix SIAM aperture. Add SIAM aperture for the spectrometer.
- 2008-09-12 - PP: Fix SPR-0862
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-01-23 - PPa: Fix SPR-1085
- 2009-03-04 - NC: Invert the Siam Matrix (SPR-1237)
- 2009-05-25 - PPa: Implemente aberration correction (SCR-1383)
- 2009-07-12 - SG: Wrong implementation of correctAberration (SPR-1617)
- 2009-08-10 - PPa: Add new convenience methods [SPIRE-1800]
- 2009-08-24 - PPa: Add new methods to get gyro propagated and commanded pointing [SPIRE-1801] Use aperture from the detAngOff [SPIRE-1857]
- 2009-08-27 - PPa: Add code examples [SPIRE-1852]
- 2009-12-07 - AP: [SPR - 2237] Added new public method getSkyPositions that takes as input a Double1d array of times and returns a table of Directions
- 2010-04-01 - AP: [SPIRE-2471] Refactored getSkyPosition to use Aberration.correct() and remove deprecated PoiningUtils.correcAberration() . The refactoring is inefficient because Aberration.correct() requires PointingArray as input.

- 2010-04-01 - AP: [SPIRE-2471] Refactored getSkyPositions to use Aberration.correct() and remove PoiningUtils.correcAberration().
- 2010-04-06 - PPa: Fix getSkyPosition.

1.227. SpirePreprocessCubeTask

Full Name:	herschel.spire.ia.pipeline.spec.regrid.SpirePreprocessCubeTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.regrid import SpirePreprocessCubeTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

SpirePreprocessCubeTask

Creates a SpirePreprocessedCube from flux and error data copied from SpectrometerDetectorSpectrum input.

SpirePreprocessCubeTask operates on one or several SpectrometerDetectorSpectrum products. The resulting SpirePreprocessedCube is an unordered set of spectra which can then be used by a NearestNeighbourProjectionTask to create a regular spectral data cube.

There are some input preconditions that must be true of ALL channels within the input SpectrometerDetectorSpectrum products:

- All wave column lengths must be equal.
- All wave column values must be equal within 1e-8.
- All units must be consistent; that is, consistent for flux, error on flux, RA, DEC, and wave.
- The unit of flux and error on flux must be equal.

Example

Example 1: Usage:

```
#Instantiate the task
task = SpirePreprocessCubeTask()
#Get Rebinned Cube from a single SDS
cube1 = task(sds=mySDS)
#Get Rebinned Cube from a list of SDS's
cube2 = task(sdsList=mySDSArray)
```

API Summary

Properties
SpectrometerDetectorSpectrum sds [INPUT, OPTIONAL, default=no default value]
SpectrometerDetectorSpectrum[] sdsList [INPUT, OPTIONAL, default=no default value]
Boolean UNVIGNETTED [INPUT, OPTIONAL, default=false]
StringId detectors [INPUT, OPTIONAL, default=null]
String arrayType [INPUT, OPTIONAL, default=null]
SpirePreprocessedCube preprocessedCube [OUTPUT, no default value, default=no default value]

API details

Properties

SpectrometerDetectorSpectrum sds [INPUT, OPTIONAL, default=no default value]

A single SpectrometerDetectorSpectrum product to be preprocessed.

SpectrometerDetectorSpectrum[] sdsList [INPUT, OPTIONAL, default=no default value]

A list of SpectrometerDetectorSpectrum products to be preprocessed.

[Boolean](#) UNVIGNETTED [INPUT, OPTIONAL, default=false]

If true, only unvignetted detectors are included in the output of this task.

StringId detectors [INPUT, OPTIONAL, default=null]

Specifies the detectors which may be included in the output of this task.

[String](#) arrayType [INPUT, OPTIONAL, default=null]

Specifies which arrays ("SLW" or "SSW") to include in the output of this task.

SpirePreprocessedCube preprocessedCube [OUTPUT, no default value, default=no default value]

The resulting preprocessed cube, effectively a list of SpireSpectrumId datasets.

See also


- [???](#)
- [SpirePreprocessedCube](#)
- [SpectrometerDetectorSpectrum](#)

History

- 2009-03-02 - CM: [SPIRE SCR-1262] First version in order to meet the requirements of the SCR
- 2009-04-16 - TRF: [HCSS SPR-6051], [SPIRE SPR-1135]: Changed jcategory tag from Task to task
- 2009-05-05 - CM: [SPIRE SCR-1262] Added new features like setError and setFlag of SpirePreprocessedCube to complete the requirements
- 2009-06-02 - CM: [SPIRE SPR-1471] Changed the input parameters: 2 different types, sds to preprocess a single SDS and sdsList to preprocess a list of SDS's. Changed the definition of 'sds' to fix the issue of the SPR
- 2009-08-12 - PK: [SPIRE SPR-1724] Added logger, interruptibility to makeCubeForSds. Removed obsolete comments
- 2009-08-13 - PK: [SPIRE SPR-1724] Improved speed of setting Flag objects by copying all at once from a Bool3d. Removed all references to deprecated setFlagValue. Removed unnecessary member variable _bflag.

- 2009-08-14 - PK: [SPIRE SPR-1724] Reorganized makeCubeForSds and makeCube. Added method getWavenumberGrid.
- 2009-08-25 - PK: [SPIRE SPR-1724] Modified checkSds and makeCubeForSds so that bad data is not included in the output cube. Added method isBadSpectrum.
- 2009-09-13 - TRF: [SPIRE-1959] Reduced wavenumber grid sensitivity threshold to 1e-8
- 2009-09-17 - DS: [SPIRE SPR-1996] Added unit consistency checking and unit propagation.
- 2009-09-18 - DS: Replaced repeated scan name look-ups in checkSDSs(...) with variables.
- 2009-09-24 - DS: [SPIRE SPR-1996] Added metadata propagation.
- 2009-12-02 - DS: [SPIRE-2226] Fixed misuse of SpireSpectrumCompositeDataset.getScan(int scanNo).
- 2010-01-19 - DS: [SPIRE-2284] Documentation updated, deprecated the VERSION public field.
- 2010-01-19 - DS: [SPIRE-2345] Refactored consistencyCheck method and delegate methods.
- 2010-01-19 - DS: [SPIRE-2334] Added OBSID, BBID, SDS and scan information to consistency error messages.
- 2010-01-26 - DS: Removed usage of deprecated AbstractComposite() methods.
- 2010-02-04 - DS: Consistency checks now throw BadDataException, not InvalidParameterException.
- 2010-05-06 - PK: [SPIRE-2446] Added UNVIGNETTED and detectors parameters for specifying eligible detectors. Added includeSpectrum method for determining whether a spectrum should be included in output.
- 2010-05-12 - PK: [SPIRE-2566] Added arrayType parameter for specifying filtering of SSW or SLW detectors.
- 2010-05-22 - PK: [SPIRE-2487] Propagated raster position and jiggle position from Sds input to SpirePreprocessedCube output

1.228. SpirePreprocessedCube

Full Name:	herschel.spire.ia.dataset.SpirePreprocessedCube
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpirePreprocessedCube
Category:	Product

Description

Herschel-SPIRE Preprocessed Cube


A convenient container for storing spectral data. This class is an intermediate product to facilitate the transition from `SpectrometerDetectorSpectrum` to `SpectralSimpleCube`. It is recommended to use `SpirePreprocessCubeTask` to build a `SpirePreprocessedCube` (see the task for an example).

`herschel.ia.toolbox.spectrum.projection.NearestNeighbourProjectionTask` uses `SpirePreprocessedCube` output to produce a `SpectralSimpleCube`.

History

- 2009-03-03 - CM: [SPIRE SCR-1262] First initial version in order to meet the requirements listed in SCR
- 2009-05-05 - CM: [SPIRE SCR-1262] Implemented new methods like `get/setError` and `get/setFlag` to complete the requirements
- 2009-08-12 - PK: Deprecated `getFlagValues()`, `setFlagValues()`. Removed use of `_bflag` from `getFlag(String)`
- 2009-08-13 - PK: Added `serialVersionUID`
- 2009-09-16 - DS: [SPIRE SPR-1996] Added `get/set` for units `flux`, `fluxError`, `dec`, `ra`, and `wave`
- 2009-09-24 - DS: [SPIRE SPR-1996] Added various error checks for bolometer (detector) handling
- 2010-02-05 - DS: [SPIRE-2255] Updated Javadoc and URM. Updated email address (CM's is defunct).
- 2010-05-21 - PK: [SPIRE-2335] Changed `appendDetectorsToMap` to enable empty constructor to be set by this method.
- 2010-05-21 - PK: [SPIRE-2487] Added `getDetectors(String type)`. Added `jiggle` and `raster` methods and attributes.

1.229. SpirePreprocessThread

Full Name:	herschel.spire.ia.pipeline.pg.SpirePreprocessThread
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import SpirePreprocessThread

Description

SPIRE preprocessing as a thread.

This class implements the preprocessing stage of data processing. The logical steps are the following:

1. Create a `ObservationContext` for the requested observation.
2. Run the `CalibrationPlugin` to extract the DPU reset history and detector offset history calibration products extractin the data from a telemetry database (via network or locally) and to attach the calibration context to the `ObservationContext`
3. Create a dummy `PointingProduct` and attach it to the `ObservationContext`
4. Attach a SIAM product (version 0122_0001) to the `ObservationContext`
5. Download all the telemetry packets for the requested observation, convert them into Level 0 products and attach them to the `ObservationContext` running the `Level0Plugin`
6. If required, convert Level 0 product into engineering data (level 0.5) and attach them to the `ObservationContext`
7. Finally save the `ObservationContext` into a specified pool.

This class can be used in two different ways:

- via a GUI implemented by the class `SpirePreprocessTool`
- via the command line. The syntax for the command line call is:

```
java herschel.spire.ia.pipeline.pg.SpirePreprocessThread obsid [edpFlag [poolName]]
```

where `obsid` is the observation ID, `edpFlag` is "true" or "false" to specify if the engineering conversion should performed or not, and `poolname` is the identifier of the pool where the `ObservationContext` shall be saved. The "edpFlag" and "poolName" are optional.

Nota bene:


- The Herschel Pointing Product attached to the `ObservationContext` by this tool is **dummy**. This tool build the `ObservationContext` using **only** the SPIRE telemetry. The information needed to create the Herschel Pointing Product it is **not** contained in the SPIRE telemetry. The Herschel Pointing Product can be generated **only** by the HSC, so the user will need to attach it to the `ObservationContext` by himself/herself.
- The SIAM product attached to the `ObservationContext` is the one shipped with this software version. There is no guarantee that it is the most updated/convenient one. It is left to the user to attach a proper vesion to the `ObservationContext` by himself/herself.

History

- 2007-10-18 - PPa: Complete CLI implementation.

- 2007-10-19 - PPa: Better handling of exceptions.
- 2007-11-03 - PPa: Register the operational day pool in the environment.
- 2007-12-04 - PPa: Add usage of chanMask product.
- 2007-12-13 - PPa: Use new EngConversionTask
- 2008-01-09 - PPa: Add SIAM usage
- 2008-01-29 - PPa: Use new ObservationContext API.
- 2008-03-27 - PPa: Use simplified syntax of EngConversion.
- 2008-05-23 - PPa: Make possible to use SpirePreprocessThread from jide
- 2008-07-10 - PPa: Register exceptions in log and in the gui. [SPR-0765]
- 2008-07-10 - PPa: Use ProductSink for EngConversion.
- 2008-08-06 - PPa: Implements SPR-0810: don't use AhfProcessor
- 2008-09-22 - PPa: Implements SPR-0875
- 2008-10-06 - PPa: Remove deprecated code. Better handling of calibration properties.
- 2008-10-20 - PPa: Improve message when the observation is not found.
- 2008-11-03 - PPa: Improve message in case of problems.
- 2008-12-10 - PPa: Get the SIAM product from FITS file instead of from SIAM file (SPR-1054)
- 2009-01-13 - PPa: Save the SIAM as serial archive instead of FITS [SPR-1117]. Fix jhistory format. Add some javadocs.
- 2009-01-28 - PPa: Add computation of the cusMode and instMode [SPR-1177]. Add some info in javadoc.
- 2009-08-10 - PPa: Change "useSink" into "tempStorage" [SPIRE-1646]
- 2010-01-25 - PPa: Improve javadocs [SPIRE-2266]
- 2010-02-16 - PPa: Use new level 0 format [SPIRE-2139], improve URM [SPIRE-2266]

1.230. SpirePreprocessTool

Full Name:	herschel.spire.ia.pipeline.pg.SpirePreprocessTool
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.pg import SpirePreprocessTool

Description


Spire Preprocess Tool.

Tool to download data of observations from the ICC database to convert them in an ObservationContext and to save them in a storage, using a GUI. This tool is not intended for use by the general user who doesn't have access to telemetry databases, but only for ICC members.

History

- 2007-10-16 - PPa: rewritten. Processing moved into SpirePreprocessThread.java
- 2007-10-19 - PPa: adjust layout and reintroduce tooltips.
- 2008-05-23 - PPa: make possible to use SpirePreprocessThread from jide
- 2008-09-22 - PPa: Implements SPR-0875
- 2008-10-20 - PPa: Implements SCR-0951 (check calibration properties)
- 2008-11-03 - PPa: Implements SCR-0979 (check pal version property)
- 2008-11-10 - PPa: Use TwoColDialogLayout
- 2009-07-23 - PPa: Remove empty catch statements [SPIRE-SPR-1707]
- 2010-01-25 - PPa: Use JInputLongField from herschel.spire.ia.util.gui package [SPIRE-2266]
- 2010-05-11 - PPa: don't use ProductPool.exists(String) deprecated method


1.231. SpireProduct

Full Name:	herschel.spire.ia.dataset.SpireProduct
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireProduct

History

- 2006-02-10 - PP: First version (PP)
- 2006-08-10 - PP: Add serialVersionUID
- 2008-02-25 - PP: Remove serialVersionUID: Product doesn't implement Serializable anymore
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]


1.232. SpireProductRule

Full Name:	herschel.spire.ia.dataset.context.SpireProductRule
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import SpireProductRule


History

- 2007 Apr 23: Fix serialization exception (E2E tests)
- 2007 Oct 11: Complete rewriting (PP)

1.233. SpireProductTypeRule

Full Name:	herschel.spire.ia.dataset.context.SpireProductTypeRule
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset.context import SpireProductTypeRule

1.234. SpireSpecFullMapPipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpireSpecFullMapPipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpireSpecFullMapPipelineTask
Category:	task

Description

Execute the SPIRE Spectrometer Fully Sampled Map Pipeline (SOF2)

This task executes the SPIRE Spectrometer Fully Sampled Map Pipeline (SOF2)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

<code>ObservationContext obs</code> [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
<code>String level</code> [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-27 - TRF: First version
- 2009-01-29 - PPa: Remove creation of output. Uncomment parameter validation.
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]

1.235. SpireSpecPointSourcePipelineTask

Full Name:	herschel.spire.ia.pipeline.hipe.SpireSpecPointSourcePipelineTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.hipe import SpireSpecPointSourcePipelineTask
Category:	task

Description

Execute the SPIRE Spectrometer Point Source Pipeline (SOF1)

This task executes the SPIRE Spectrometer Point Source Pipeline (SOF1)

API Summary

Properties
ObservationContext obs [INPUT, MANDATORY, default=no default value]
String level [INPUT, OPTIONAL, default="level0"]

API details


Properties

ObservationContext obs [INPUT, MANDATORY, default=no default value]
ObservationContext to be processed
String level [INPUT, OPTIONAL, default="level0"]
Starting level of pipeline processing

History

- 2009-01-27 - TRF: First version
- 2009-01-29 - PPa: Remove creation of output. Uncomment parameter validation.
- 2009-05-15 - PPa: Add propagation of progress from pipeline scripts [SCR-1260]
- 2010-01-28 - PPa: Use new method for Modifiers [SPIRE-2348], update javadoc [SPIRE-2312]

1.236. SpireSpectrum1d

Full Name:	herschel.spire.ia.dataset.SpireSpectrum1d
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireSpectrum1d
Category:	Dataset

Description

Herschel-SPIRE SpireSpectrum1d Dataset

Extended Spectrum1d dataset that allows convenience methods for get/set Flux (Complex1d) and get/set Wavenumber (Double1d).


History

- 2006-06-27 - TRF: Changed some javadoc
- 2006-10-10 - TRF: Removed the resizing of the data columns. getComplexFlux() now does not throw an error even if flux is of type Double1d. The returned vector has all of its imaginary componenets set to zero.
- 2007-03-26 - TRF: Added new methods to access data and units.
- 2007-03-27 - TRF: setPixelName() now a public method.
- 2007-06-04 - AH: added some javadoc. resolved javadoc conflicts.
- 2007-06-07 - TRF: Deprecated some obsolete methods
- 2007-07-30 - AH: Migrate to new Unit API, add new methods for set/get of Unit, deprecated old methods for set/get of Units
- 2007-08-24 - AH: Remove deprecated methods using nT.quantity
- 2007-10-12 - ZW: Delete all the deprecated methods.
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-03-11 - YFR: [SPIRE SCR-1290] Added Ra and Dec attributes
- 2009-03-17 - YFR: [SPIRE SCR-1290] Added initMeta method to initialize Ra, Dec, and pixel name
- 2009-03-17 - TRF: [SPIRE SPR-1062] New constructor to propagate metadata from SpireInterferogram1d
- 2009-03-17 - YFR: [SPIRE SCR-1290] Added description for pixel name
- 2009-03-20 - YFR: [SPIRE SPR-1319] Initialized ra and dec values to be -1 to fix saving product problem
- 2009-04-14 - YFR: Modified RA_DESCRIPTION and DEC_DESCRIPTION
- 2009-04-14 - YFR: Removed redundant castings
- 2009-04-14 - YFR: [SPIRE SCR-0817] Deprecated methods: getWavenumberData, getWavenumberUnit, setWavenumber, setWavenumberData, setWavenumberUnit, getComplexFlux-

Data, getComplexFluxUnit, setComplexFlux, setComplexFluxData, setComplexFluxUnit, setComplexFluxError, setComplexFluxErrorData, setComplexFluxErrorUnit, getComplexFluxErrorData, getComplexFluxErrorUnit

- 2009-04-14 - YFR: [SPIRE SCR-0817] Added methods: setFluxError, setFluxErrorData, setFluxErrorUnit, getFluxErrorData, getFluxErrorUnit
- 2009-04-16 - YFR: [SPIRE SCR-1343] Deprecated methods with name 'pixel', added new methods with name 'channel'
- 2009-06-05 - YFR: [SPIRE SCR-1343] Removed deprecated methods with name 'pixel'
- 2009-08-05 - TRF: [SPIRE-1784] (see SPIRE-817) Removed deprecated methods: getWavenumberData, getWavenumberUnit, setWavenumber, setWavenumberData, setWavenumberUnit, getComplexFluxData, getComplexFluxUnit, setComplexFlux, setComplexFluxData, setComplexFluxUnit, setComplexFluxError, setComplexFluxErrorData, setComplexFluxErrorUnit, getComplexFluxErrorData, getComplexFluxErrorUnit
- 2009-11-12 - YFR: [SPIRE-1936] Fixed setter methods which circumventing column row-length equality check

1.237. SpireSpectrumCompositeDataset

Full Name:	herschel.spire.ia.dataset.SpireSpectrumCompositeDataset
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SpireSpectrumCompositeDataset
Category:	Dataset


Description

Herschel-SPIRE SpireSpectrumCompositeDataset Dataset

History

- 2006-06-27 - TRF: Changed some javadoc
- 2006-10-20 - TRF: added getScanDir/setScanDir methods
- 2007-06-04 - AH: added javadoc. resolved javadoc conflicts
- 2007-06-07 - TRF: Deprecated obsolete methods. Deprecated get/setCount as that metaData parameter is being phased out in favour of scanNumber.
- 2007-10-12 - ZW: Delete all the deprecated methods.
- 2007-10-22 - TRF: Fixed behaviour of constructors where String scanNo cannot be parsed to a long
- 2007-10-23 - ZW: Fixed constructors where String scanNo cannot be parsed to a long.
- 2007-10-23 - ZW: Fixed behaviour of removePixel() method
- 2007-11-08 - ZW: Added Copy Constructor and method copy()
- 2008-02-25 - PP: Remove serialVersionUID: Dataset doesn't implement Serializable anymore
- 2008-05-27 - ES: Added "scanNumber" parameter to initMeta(), resolving a bug that caused NullPointerExceptions when getScanNumber() was called in some cases.
- 2008-11-21 - TRF: Removed redundant cast in hasPixel() and getPixelNames() [SPIRE SPR-0684]
- 2008-11-27 - TRF: Fixed some jtag urm documentation [HCSS SPR-3232]
- 2009-04-16 - YFR: [SPIRE SCR-1343] Deprecated methods with name 'pixel', added new methods with name 'channel'
- 2009-06-05 - TRF: [SPIRE SCR-1343] Removed deprecated methods with name 'pixel'
- 2010-02-01 - TRF: Removed deprecated AbstractComposite() methods.
- 2010-02-01 - TRF: [SPIRE-2255] Fixed some URM and Javadoc

1.238. SpireTmPacketConverter

Full Name:	herschel.spire.ia.pipeline.common.rawdata.SpireTmPacketConverter
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.rawdata import SpireTmPacketConverter

Description

SPIRE Telemetry packet converter


This class is used by the RawProductCreatorTask to convert telemetry packets into SPIRE data frames. For each TmPacket recived on the input queue it produces an array of SpireDataFrames.

Note: this class is not intended to be used by normal users since only ICC members have access to the telemetry database.

History

- 2009-08-10 - PPa: New class copied from spire.ccm [SPIRE-1369]
- 2009-08-23 - PPa: Fix setting of data in dataframes [SPIRE-1849]
- 2009-09-16 - PPa: Fix computation of data frame time [SPIRE-1962]
- 2010-04-27 - PPa: Add jhelp [SPIRE-2262]

1.239. splitRawData

Full Name:	herschel.spire.ia.pipeline.common.engdata.SplitRawDataTask
Alias:	splitRawData
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import SplitRawDataTask
Category:	SPIRE/Pipeline/Common Task

Description

Split Raw Data Task.

This task is used by the EngConversionTask to split raw (Level-0) data products.

API Summary

Properties
BuildingBlockProduct rst [INPUT, MANDATORY, default=No default value.]
ArrayList array [OUTPUT, OPTIONAL, default=No default value.]

API details


Properties

BuildingBlockProduct rst [INPUT, MANDATORY, default=No default value.]
Input raw SPIRE Timeline Data Product to be split.
ArrayList array [OUTPUT, OPTIONAL, default=No default value.]
Array of Raw SPIRE Data Products.

History

- 2010-02-14 - PPa: First version.
- 2010-02-18 - PPa: propagate the tmDrops metadata.
- 2010-02-19 - PPa: Remove unneeded log message.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].


1.240. SubtractIfgmTask

Full Name:	herschel.spire.ia.dataset.SubtractIfgmTask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import SubtractIfgmTask

History

- 2009-04-21 - YFR: First Version
- 2010-02-05 - YFR: [SPIRE-2255] Added missing descriptions and fixed vague descriptions


1.241. TelemMask

Full Name:	herschel.spire.ia.dataset.TelemMask
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import TelemMask
Category:	Calibration product

History

- 2009-06-08 - First: version
- 2010-02-05 - YFR: [SPIRE-2255] Added description for constructor and method 'isUnreliable'
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods

1.242. TelescopeScalSubtractionTask

Full Name:	herschel.spire.ia.pipeline.spec.scal.TelescopeScalSubtractionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.scal import TelescopeScalSubtractionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

The TelescopeScalSubtractionTask Workpackage :

API Summary

Properties
SpectrometerDetectorInterferogram sdi [Input, Mandatory, default=No default value the]
NhkTimeline nhkt [Input, Mandatory, default=No default value]
SpectrometerDetectorInterferogram interRef [Input, Mandatory, default=No default value]
type scaloutput [Output, MANDATORY, default=no default value]

API details

Properties


SpectrometerDetectorInterferogram sdi [Input, Mandatory, default=No default value the]
sdi product
NhkTimeline nhkt [Input, Mandatory, default=No default value]
nhkt the HK product
SpectrometerDetectorInterferogram interRef [Input, Mandatory, default=No default value]
interRef the SDI product
type scaloutput [Output, MANDATORY, default=no default value]
herschel.spire.ia.dataset.SpectrometerDetectorInterferogram, Optional, No default value the SCAL SDI product scaloutput

History

- 2008-05-23 - DB: first version
- 2008-10-07 - DB: add description to task Parameters
- 2008-10-16 - DB: update Double1d to Float1d output signal
- 2008-11-20 - DB: ScalTasks becomes interruptible

- 2008-11-21 - DB: implement progress bar 2008-12-02 DB remove LOG.setLevel SPR-1048
- 2009-01-23 - DB: corrected jhistory sdi input product first +@SuppressWarnings("unchecked")
- 2009-01-28 - DB: change the compatibility between enter and SpecInterRef calibration same OPD but may have defferent length
- 2009-02-04 - DB: from TF [SPIRE SPR-1200, 1204] Changed sdi.keySet() to sdi.getScanNumbers()
- 2009-02-10 - DB: change test on opd grid : add checkOpdStep() and modified isCompatible() remove //test same opd grid
- 2009-02-19 - DB: erase comment and same length by detector (no update)
- 2009-03-23 - DB: update output signal data from float1d to double1d
- 2009-04-24 - DB: SCR 1363 ScalTask to include QC metadata
- 2009-05-19 - DB: correct deprecated methods
- 2009-06-01 - DB: modified setCreator, remove "filename in metadata
- 2009-06-22 - DB: modification name metadata "deltaSCalTemp"
- 2009-07-21 - DB: to change the task name from "ScalTask" to "TelescopeScalSubtraction-Task" "calSpecReferenceIfgm" to accept the calibration product of SPIRE SPR 1655
- 2009-07-28 - DB: SPIRE-1655 change the name in constructor
- 2009-07-31 - DB: to change "ReferenceIfgm" to "interRef" SPIRE SPR 1655
- 2009-08-06 - DB: corrected back to interRef for jparameter tag
- 2009-09-02 - DB: SPIRE-1815 TelescopeScalCorrection does not apply the calibration product if number detectors not equal in interRef and input SDI
- 2009-10-09 - DB: add description parameter metadata
- 2009-11-08 - DB: SPIRE-2135 change warning to error throw exception
- 2009-11-08 - DB: change mandatory nhkt
- 2009-01-29 - DB: SPIRE-2285 add java doc method, task description, change exception
- 2009-02-01 - DB: correct usage of deprecated method
- 2010-04-26 - DB: SPIRE-2476 TelescopeScalSubtractionTask is too verbose , remove info message add fine message for temperature
- 2010-05-11 - DB: use throw new BadDataException
- 2010-05-11 - DB: removed isReverseForward , increasing test coverage
- 2010-05-11 - DB: add ParameterValidator
- 2010-05-11 - DB: re-open SPIRE-2476 TelescopeScalSubtractionTask is too verbose update info message misunderstanding on message


1.243. TempDriftCorr

Full Name:	herschel.spire.ia.dataset.TempDriftCorr
Type:	Java Class - 
Import:	from herschel.spire.ia.dataset import TempDriftCorr
Category:	TempDriftCorrCalibTable Calibration product

History

- 2007-11-11 - AS: Add methods to get/set data by column; rename pixel get/set methods.
- 2007-11-14 - AS: Modify get/set by column to get/set by ArrayData type
- 2008-02-25 - AS: Extend ChannelCalibTable
- 2008-11-27 - TRF: Fix some javadoc warnings [SPIRE SPR-0684]
- 2010-02-05 - YFR: Removed usage of deprecated AbstractComposite methods

1.244. TemperatureCondition

Full Name:	herschel.spire.ia.pipeline.spec.scal.TemperatureCondition
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.scal import TemperatureCondition
Category:	SPIRE/Pipeline/Spectrometer Task

Description

This task checks the sources temperature fluctuations. Its result is a boolean :

API Summary

Properties
DoubleId temp [Input, Mandatory, default=No default]
DoubleId temp2 [Input, Mandatory, default=No default]
DoubleId temp4 [Input, Mandatory, default=No default]
ShortId scanct [Input, Mandatory, default=No default]
Boolean result [Output, Optional, default=no default value]

API details

Properties


DoubleId temp [Input, Mandatory, default=No default]
temp value scal bath temp
DoubleId temp2 [Input, Mandatory, default=No default]
temp2 value scal2 temp
DoubleId temp4 [Input, Mandatory, default=No default]
temp4 value scal4 temp
ShortId scanct [Input, Mandatory, default=No default]
scanct value scan count list form HK["SMECSCANCNT"]
Boolean result [Output, Optional, default=no default value]
result check status

History

- 2006-09-12 - CO: change deprecated 'setMode' method. define delta temperature variations domain for scal2 scal4 and scal bath (checkmethod)
- 2006-10-06 - CO: scan count parameter added compute delta temperature variations for each scan (checkAllScans) volt2 and volt4 parameters removed class renamed -> TemperatureCondition
- 2006-12-12 - CO: preamble and postamble codes moved into the execute method

- 2007-03-07 - CO: set _scanct to ShortId type. Format numbers in LOG message.
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.scal->herschel.spire.ia.pipeline.spec.scal
- 2009-01-23 - DB: corrected jhistory
- 2009-04-24 - DB: SCR 1363 ScalTask to include QC metadata
- 2009-04-24 - DB: SCR 1363 ScalTask calculated value not delta accepted
- 2009-10-01 - DB: update method getDeltaScalTemp... SPIRE 1363
- 2009-01-29 - DB: SPIRE-2285 add java doc method,
- 2009-02-01 - DB: correct usage of deprecated method
- 2010-04-26 - DB: SPIRE-2476 TelescopeScalSubtractionTask is too verbose remove info message
- 2009-05-11 - DB: SPIRE-2285 corrected naming getDeltaScalTemp4/2

1.245. TemperatureDriftCorrectionTask

Full Name:	herschel.spire.ia.pipeline.common.tempdrift.TemperatureDriftCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.tempdrift import TemperatureDriftCorrectionTask

Description

This task corrects the detectors for bath temperature drift.

The algorithm uses a spline function to smooth the thermistor timeline, which is then used to correct the temperature drift.

API Summary

Properties
DetectorTimeline data [Input, Mandatory, default=No default value]
PhotTempDriftCorr tempDriftCorr [Input, Mandatory, default=No default value]
Double timeSpan [Input, Optional, default=5.0]
Optional pswThermistorSelect [java.lang.String, "T1", default=no default value]
Optional plwThermistorSelect [java.lang.String, "T1", default=no default value]
Optional pmwThermistorSelect [java.lang.String, "T1", default=no default value]
String glitchinfo [Input, Optional, default=No default value]
DetectorTimeline out [OUTPUT, Optional, default=No default value]

API details

Properties

DetectorTimeline data [Input, Mandatory, default=No default value]
PDT or SDT Input detector timeline
PhotTempDriftCorr tempDriftCorr [Input, Mandatory, default=No default value]
Input calibration product of correction coefficients
Double timeSpan [Input, Optional, default=5.0]
Optional parameter describing spline binning interval in seconds (5.0 def.)
Optional pswThermistorSelect [java.lang.String, "T1", default=no default value]
Optional thermistor selection (default is use value specified in input calibration product).

Optional plwThermistorSelect [<code>java.lang.String</code> , "T1", default=no default value]

Optional thermistor selection (default is use value specified in input calibration product).
--

Optional pmwThermistorSelect [<code>java.lang.String</code> , "T1", default=no default value]

Optional thermistor selection (default is use value specified in input calibration product).
--

String glitchinfo [<code>Input</code> , <code>Optional</code> , default=No default value]

Optional parameter will provide a listing of glitch information. Specify parameter as glitchinfo="glitchinfo"

DetectorTimeline out [<code>OUTPUT</code> , <code>Optional</code> , default=No default value]

Output product corrected for bath temperature drift


History

- 24 Jan 2008 : R Gastaud & A.Schwartz (convert prototype)
- 15 Feb 2008 AS Incorporate calibration tables into calculations.
- 05 Mar 2009: AS Modify code to use same algorithm as IDL model.
- Use average time of interval for spline point, rather than
- using time of median point. Change spline fit to use time values
- biased to the first time in timeline for greater accuracy.
- Also clean up some small errors:
- intervals are shifted by one point.
- 24 Mar 2009: [SCR-1201] Add capability to process fewer than the
- maximum number of detectors in an array, if the
- input product has fewer.
- 08 May 2009: [SCR-1140] Revise tooltips and URM doc.
- 08 May 2009: [SCR-1047] Add checks that nonlinearity/fluxconversion
- calibration files are consistent with temperature drift
- calibration product by checking product and metadata control
- stamps.
- 10 Jul 2009: [SPR-1618] Correct error message about missing control stamp
- in input product, and not requirement that input data must
- have been processed by NonLinearityCorrectionTask first.
- 16 Jul 2009: 1. [SCR-1279] Added progress indicator.
- 2. [SCR-1598] Changed timeSpan default to 5.0 .

- 3. [SCR-1583] Changed timeSpan type from long to double.
- 4. [SPR-1534] Added package.html.
- 07 Aug 2009: [SCR-1548] Add RobustMean function to isolate glitches, and
- omit glitch-masked data (in thermistors) from smoothing calculations.
- 12 Aug 2009: [SPR-1078] Modify code to remove fileName metadata (i.e. do not propagate it).
- 13 Aug 2009: [SCR-1547,1755,1462,1359] Analyze thermistor voltage data,
- and flag (in mask) points that fall out of accepted range.
- Also generate metadata reporting the frequency of occurrence of
- out of range data for each array.
- [SPR-1140] Modified tooltip and URM parameter descriptions
- to be consistent.
- 02 Sep 2009: [SPR-1893] Corrected bright source mode to extract bright
- source reference voltage (i.e. DP refe voltage), and to
- set dark pixel masks as appropriate.
- [SPR-1140] Change hipe tooltips again.
- Also change output parameter to "out".
- [SPR-1652,1826] Change table input name to "tempDriftCorr".
- [SCR-1462,SPR-1359] Rename metadata to be consistent with quality
- control names.
- 23 Oct 2009: [SCR-1547] Fix error in masking when both thermistors are out of range.
- 10 Nov 2009: {SCR-2040} Modify code to exclude various thermistor samples
- from the spline binning-smoothing processing. The
- mask categories to be excluded are: master, ADC-latch
- time shift error, no resp data error, dead channels,
- as well as samples identified as glitches. If less than
- a specified number of samples remains in a bin, exclude
- that bin from the spline fit. Also if
- more than 5 percent of thermistor data is marked as
- ADC truncated switch to use of other thermistor, or
- if only one is available terminate with exception.
- 12 Jan 2010: [SCR-1755] Correct error in mask implementation for out of range
- thermistor value for thermistor T2.

- 11 Feb 2010: [SCR-2202] Add new parameters for selecting thermistors for each bolometer array. These values will override the specifications given in the calibration product.
- 22 Feb 2010: [SCR-2202] Fix error in ingest of new thermistor select parameters.
- 25 Feb 2010: [SPR-2036] Modify TemperatureDriftCorrectionDone metadata (change first character to lower case).
- 29 Apr 2010: [SCR-2454] Add processing to check for signal jump mask bit, and modify use of thermistors accordingly.
- 10 May 2010: URL, javadoc mods; change exception types.
- 21 May 2010: [SCR-2513] Change processing to set NO_THERMISTOR_AVAILABLE mask bit if either jumps or out of range conditions occur.
- Remove data exceptions and instead process all bolometer arrays with valid thermistor data.

1.246. timeConvReord

Full Name:	herschel.spire.ia.pipeline.common.engdata.TimeConvReordTask
Alias:	timeConvReord
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import TimeConvReordTask
Category:	SPIRE/Pipeline/Common Task

Description

Time Conversion and Reordering Task.

This task is part of the Engineering Conversion. Its purpose is to recompute the sample time of timelines and to reorder the data in the case of unsorted telemetry.

API Summary

Properties
EdpProduct rawData [INPUT, MANDATORY, default=No default value.]
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value.]
ResetHist resetHist [INPUT, OPTIONAL, default=No default value.]
Double offset [INPUT, OPTIONAL, default=0.079.]
Double drift [INPUT, OPTIONAL, default=-0.39.]
EdpProduct converted [OUTPUT, OPTIONAL, default=No default value.]

API details

Properties


EdpProduct rawData [INPUT, MANDATORY, default=No default value.]
Input SPIRE Data Product to be converted.
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value.]
Nominal House Keeping timeline.
ResetHist resetHist [INPUT, OPTIONAL, default=No default value.]
DPU Reset History.
Double offset [INPUT, OPTIONAL, default=0.079.]
Time constant to correct the offset between detectors and pointing timing, in seconds.
Double drift [INPUT, OPTIONAL, default=-0.39.]
Time constant to correct the drift between detectors and pointing timing, in milliseconds per minute.
EdpProduct converted [OUTPUT, OPTIONAL, default=No default value.]
Converted SPIRE Data Product.

History

- 2005-02-00 - PPa: First version copied from DataConversionProcess.java.
- 2005-08-00 - PPa: moved to the new doc system.
- 2006-02-21 - PPa: use BuildingBlockProducts, use new EDP product format.
- 2006-05-23 - PPa: Remove usage of deprecated methods of BuildingBlockProduct.
- 2006-05-31 - PPa: If it gets wrong inputs, now it throws a SignatureException rather than returning.
- 2006-08-11 - PPa: Some refactoring.
- 2006-09-14 - PPa: Compatible with optimized products.
- 2006-11-10 - PPa: Supports SOT and POT products. Refactoring.
- 2007-01-09 - PPa: Implementation of the case of frametime counter rolling over (SPR-0569).
- 2007-03-09 - PPa: skip "quality" table.
- 2007-03-13 - PPa: Use new product structure.
- 2007-05-15 - PPa: Adapt to the new Product API.
- 2007-05-29 - PPa: Adapt to the new DetectorTimeline structure.
- 2007-07-27 - PPa: Migrate to new Unit API.
- 2008-07-04 - PPa: Renamed and merged with the DataSorterTask. Fix logic and fix SPR-0702.
- 2008-07-11 - PPa: Change sampleTime unit from Duration.SECONDS to TimeInstant.TAI (SCR-0762). Refactoring.
- 2008-07-17 - PPa: Change logic again.
- 2008-07-18 - PPa: Remove time table usage.
- 2008-08-27 - PPa: Add mask setting for invalid time.
- 2008-11-14 - PPa: Remove seqCount column which is now propagated from FormatConversionTask. The seqCount column is still not used.
- 2009-01-28 - PPa: Add parameter descriptions (SPR-0985). Add jparameter tags. Fix jhistory format. Add serialVersionUID.
- 2009-02-18 - PPa: Fill the missing description of resetHist [SPR-0985].
- 2009-03-30 - PPa: Remove deprecated setResetOnExecute(true).
- 2009-05-01 - PPa: Remove seqCount in housekeeping timelines [SPR-1452].
- 2009-06-09 - PPa: Simplify the algorithm. Remove duplicated code. Adapt to new requirements.
- 2009-06-12 - PPa: Set as invalid times where packetTime is smaller of reset.
- 2009-07-22 - PPa: Assume that MCUENG packets have frames [SPR-1699].
- 2009-07-23 - PPa: Use $2 * (\text{Integer.MAX_VALUE} + 1L)$ instead of $2 * \text{Integer.MAX_VALUE}$ [SPIRE-SPR-1664].

- 2009-07-29 - PPa: Fix algorithm run-out.
- 2009-09-05 - PPa: Generate output with correct class.
- 2009-09-16 - PPa: Recompute start and end time of products with frames. [SPIRE-1962].
- 2009-12-16 - PPa: Add correction for the drift and the offset between DCU signal timeline and pointing timeline [SPIRE-2293 and 2292].
- 2010-02-24 - PPa: Update offset value [SPIRE-2412].
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.247. TimeCorrelationTask

Full Name:	herschel.spire.ia.pipeline.common.timecorr.TimeCorrelationTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.timecorr import TimeCorrelationTask
Category:	herschel/spire/ia/common/timecorr

Description

This module applies a time correction from on-board-time (OBT) to TAI. These times are also known as

uncorrelated and correlated times respectively. The conversion from TAI to UTC is well-known, well-defined, and out of the scope of this module.

API Summary

Properties
SpireDataProduct inProduct [Input, Mandatory, default=No Default value]
TimeCorrProduct timeCorrProduct [Input, Mandatory, default=No default value]
SpireDataProduct outProduct [Output, Mandatory, default=No Default value]

Limitations

None

API details

Properties


<code>SpireDataProduct inProduct</code> [Input, Mandatory, default=No Default value]
SpireDataProduct: Product timestamps to be converted from OBT to TAI
<code>TimeCorrProduct timeCorrProduct</code> [Input, Mandatory, default=No default value]
Time Correlation data product
<code>SpireDataProduct outProduct</code> [Output, Mandatory, default=No Default value]
SpireDataProduct: Product timestamps converted from OBT to TAI

History

- 02 Sep 2009 (AP) First upload
- 13 Feb 2010 (AP) SPIRE-2264 FIXED: No description for the class or the constructor.

- 10 May 2010 (AP) SPIRE-2264 FIXED: added (at)see links for input and output products
- 10 May 2010 (AP) Changed limitations tag to None
- 10 May 2010 (AP) Added Initials to all history tags.
- 10 May 2010 (AP) Added added jcategory tag

1.248. TimeDomainPhaseCorrectionTask

Full Name:	herschel.spire.ia.pipeline.spec.phase.TimeDomainPhaseCorrectionTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.spec.phase import TimeDomainPhaseCorrectionTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Task used to correct SPIRE FTS science timelines for the delay induced by the readout electronics and the thermal time constant of the bolometers.

(from the SPIRE Spectrometer Pipeline Description Document, SPIRE-BSS-DOC-002966):

Example

Example 1: The following jython script shows an example of the TimeDomainPhaseCorrectionTask.

```
#import all necessary packages
from herschel.ia.dataset import *
# ... assume that a SPIRE DetectorTimeline Product has been created
#create and instantiate the Time Domain Phase Correction Task
timeDomainPhaseCorrection=TimeDomainPhaseCorrectionTask()
#execute the task
sdt = timeDomainPhaseCorrection(sdt=sdt,
                               nhkt = nhkt,
                               lpfPar = obs.calibration.spec.lpfPar,
                               chanTimeConst= obs.calibration.spec.chanTimeConst,
                               phaseCorrLim = obs.calibration.spec.phaseCorrLim)
```

API Summary

Properties
DetectorTimeline sdt [INPUT, MANDATORY, default=No default value]
NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value]
SpecLpfPar lpfPar [INPUT, MANDATORY, default=No default value]
SpecChanTimeConst chanTimeConst [INPUT, OPTIONAL, default=No default value]
DetectorTimeline outsdt [OUTPUT, MANDATORY, default=No default value]

API details

Properties

DetectorTimeline sdt [INPUT, MANDATORY, default=No default value]

Input SPIRE DetectorTimeline product. Uncorrected detector timelines.

NhkTimeline nhkt [INPUT, OPTIONAL, default=No default value]

Input NhkTimeline product. Housekeeping product corresponding to DetectorTimeline product.


SpecLpfPar lpfPar [INPUT, MANDATORY, default=No default value]
Input electric filter parameters calibration product. This product contains the resistor and capacitor values for the electrical low-pass filter in the Spectrometer Detector read-out chain.
SpecChanTimeConst chanTimeConst [INPUT, OPTIONAL, default=No default value]
Input time constant calibration product. It contains the thermal time constant for each bolometer detector
DetectorTimeline outsdt [OUTPUT, MANDATORY, default=No default value]
Output SPIRE DetectorTimeline product. This product will contain the shifted detector timelines.

History

- 2007-10-21 - AH: First implementation.
- 2007-06-28 - AH: Added readout electronic and time constant calibration products.
- 2007-09-27 - TRF: Slight refactoring.
- 2007-10-02 - ZW: Added super() in constructor
- 2008-04-18 - TRF: Changed FFT references to herschel.ia.numeric.toolbox.xform.util
- 2008-04-28 - TRF: Changed deprecated get/setChannelSignal() to get/setSignal()
- 2008-07-17 - TRF: Changed dt.getSignal() so that it now expects type ArrayData. Also modified the public timeDomainPhaseCorrection()
- 2008-07-17 - TRF: Modified the public timeDomainPhaseCorrection() method so that the signal argument is of type ArrayData. timeDomainPhaseCorrection() now checks if the values returned by dt.getSignal() is of type Float1d or Double1d before taking further actions. These changes are in response to SPIRE SPR-0775.
- 2008-07-29 - TRF: [SPIRE SPR-0775] Added an information logging message to show progress.
- 2008-08-21 - TRF: [SPIRE SPR-0825] New class named TimeDomainPhaseCorrectionTask to replace the misnamed TimeDomainPhaseCorrection class
- 2008-10-07 - TRF: [SPIRE SPR-0916] Added descriptions to TaskParameters. Made jparameter descriptions more descriptive. Added a reference to the SPIRE Pipeline Description Document.
- 2008-10-07 - TRF: [SPIRE SCR-0912] Changed state of generateFrequencyResponse() from public to public
- 2008-10-10 - TRF: [SPIRE SPR-0937] Fixed bug in the manner in which setProgress() was being updated
- 2008-10-17 - TRF: [SPIRE SPR-0952] Sampling rate now determined by the median of all dts
- 2008-11-12 - TRF: [SPIRE SPR-0949] Truncate the mask vectors
- 2008-11-13 - TRF: [SPIRE SPR-1028] Added checkInterrupted() to execute()
- 2008-11-19 - TRF: [SPIRE SPR-1037] Changed the way columns are truncated in the tables of the output SDT

- 2008-11-21 - TRF: Fixed unchecked, serial, and javadoc warnings
- 2008-12-09 - TRF: [HCSS SPR-3232] Fixed some jtag urm documentation
- 2009-01-23 - TRF: [SPIRE SPR-1165] Added serialVersionUID = 1L, removed SuppressWarnings
- 2009-01-23 - TRF: [SPIRE SPR-1164] Added new TaskParameters to agree with SPIRE Pipeline policy, deprecated old TaskParameters
- 2009-01-23 - TRF: [SPIRE SPR-1098] Added throws javadoc tag to execute()
- 2009-01-29 - TRF: [SPIRE SPR-1164] Removed deprecated TaskParameters
- 2009-02-05 - TRF: [SPIRE SPR-1204, 1210] Changed the loop through the SDT tables to exclude non TableDatasets
- 2009-02-19 - TRF: [SPIRE SPR-XXXX] Fixed the example in the Javadoc
- 2009-02-19 - TRF: [SPIRE SPR-XXXX] Fixed the package to which the logger is assigned
- 2009-02-19 - TRF: [SPIRE SPR-XXXX] Fixed the way the range() method is accessed
- 2009-02-19 - TRF: [SPIRE SPR-XXXX] Added a warning if the SDT signal type is other than FLOAT or DOUBLE
- 2009-06-12 - TRF: [SPIRE-1511] Task now sets Creator data in accordance with SPIRE Pipeline policy
- 2009-06-24 - PK: [SPIRE-1269] Added methods to shift masks to the same degree as index shifted by convolution
- 2009-11-23 - TRF: [SPIRE-2196] Smec speed is now only calculated when NHKT.SCANS > 0
- 2010-01-26 - TRF: Removed usage of deprecated AbstractComposite() methods
- 2010-05-18 - TRF: [SPIRE-2283] Fixed Javadoc/URM entries

1.249. TimelineSimplePlot

Full Name:	herschel.spire.ia.gui.TimelineSimplePlot
Type:	Java Class - 
Import:	from herschel.spire.ia.gui import TimelineSimplePlot
Category:	SPIRE/IA GUI

Description

Plotter for DetectorTimeline products

This plotter allows to plot the SPIRE DetectorTimeline products from HIPE command line.

Example

Example 1: How to use from HIPE command line.

```
from herschel.spire.ia.gui import TimelineSimplePlot
sp=TimelineSimplePlot(my_product, -"my_dataset", -"my_channel")
# or
sp=TimelineSimplePlot()
sp.setObject(DetectorTimeline my_product, String -"my_dataset",
String -"my_channel")
```

API Summary

Constructors
TimelineSimplePlot(DetectorTimeline dt, String tableName, String chanName)
TimelineSimplePlot()
Methods
setObject(DetectorTimeline dt, String tableName, String chanName)
clearObject()
changeObject(DetectorTimeline dt)
changeDataset(String data)
changeChannel(String chann)
setScale(int abs)
setMask()
clearMask()
setColorMask(Color col)

API details

Constructors

<code>TimelineSimplePlot(DetectorTimeline dt, String tableName, String chanName)</code>
Arguments

TimelineSimplePlot(DetectorTimeline dt, [String](#) tableName, [String](#) chanName)

```
DetectorTimeline dt [INPUT, MANDATORY]
String tableName [INPUT, MANDATORY]
String chanName [INPUT, MANDATORY]
```

Example

TimelineSimplePlot(my_product, "my_dataset", "my_channel").

```
from herchel.spire.ia.gui import TimelineSimplePlot
sp=SimplePlot(DetectorTimeline my_product, String -"my_dataset",
String -"my_channel")
# E.G. my_dataset=voltage, signal, mask...
# E.G. my_channel=PLWC5, SLWC5, -...
```

TimelineSimplePlot()

Example

TimelineSimplePlot().

```
from herchel.spire.ia.gui import TimelineSimplePlot
sp=TimelineSimplePlot()
sp.setObjects(DetectorTimeline my_product, String -"my_dataset",
String -"my_channel")
# E.G. my_dataset=voltage, signal, mask, -...
# E.G. my_channel=PLWC5, SLWC5, -...
```

Methods

setObject(DetectorTimeline dt, [String](#) tableName, [String](#) chanName)

Arguments

```
DetectorTimeline dt [INPUT, MANDATORY]
String tableName [INPUT, MANDATORY]
String chanName [INPUT, MANDATORY]
```

Example

setObject(my_product, "my_dataset", "my_channel").

```
from herchel.spire.ia.gui import TimelineSimplePlot
sp=SimplePlot()
sp.setObjects(DetectorTimeline my_product, String -"my_dataset",
String -"my_channel")
# E.G. my_dataset=voltage, signal, mask...
# E.G. my_channel=PLWC5, SLWC5, -...
```

clearObject()

Example

clearObject()

```
from herchel.spire.ia.gui import TimelineSimplePlot
sp=SimplePlot(DetectorTimeline my_product, String -"my_dataset",
String -"my_channel")
sp.clearObject()
```

changeObject(DetectorTimeline dt)

Argument

changeObject(DetectorTimeline dt)

DetectorTimeline **dt** [INPUT, MANDATORY]

Example

changeObject(my_product): this method allow to CHANGE the product to plot.

```
from herschel.spire.ia.gui import TimelineSimplePlot
sp=SimplePlot()
sp.changeObjects(DetectorTimeline my_product)
```

changeDataset(String data)**Argument**

String **data** [INPUT, MANDATORY]

Example

changeDataset(String data): this method allow to CHANGE the dataset to plot.

```
from herschel.spire.ia.gui import TimelineSimplePlot
sp=SimplePlot()
sp.setDataset(String data)
# E.G. data="voltage", -"signal", -...
```

changeChannel(String chann)**Argument**

String **chann** [INPUT, MANDATORY]

Example

changeChannel(String data): this method allow to CHANGE the channel to plot.

```
sp.setChannel(String chan)
# E.G. chan="PLWC5", -"SLWA5", -...
```

setScale(int abs)**Argument**

int **abs** [INPUT, MANDATORY]

Example

setScale(int n): method to set the x axis scale.

```
sp.setScale(n)
# n=0 ----> Relative time scale
# n=1 ----> Absolute time scale
# n=2 ----> Index scale
# default n=0
```

setMask()**Example**

setMask(): method to allow the mask overplotting.

```
sp.setMask()
```

clearMask()**Example**

clearMask(): method to remove the mask overplotting.

clearMask()

```
sp.clearMask()
```

setColorMask([Color](#) col)**Argument**

[Color](#) col [INPUT, MANDATORY]

Example

setColorMask(java.awt.Color color): method to set the color to use to display the mask.

```
sp.setColorMask(java.awt.Color.yellow)
sp.setColorMask(java.awt.Color.BLUE)
sp.setColorMask(java.awt.Color.GREEN)
sp.setColorMask(java.awt.Color.black)
```


See also

- [DetectorTimelineExplorer](#)
- [DetectorTimelineExplorerComponent](#)
- [User's Manual](#)

History

- 2010-01-11 - GM: First Version
- 2010-02-03 - TRF: [INFR-365] Migrated from herschel.spire.ia.dataset.gui

1.250. TimePos

Full Name:	herschel.spire.ia.pipeline.spec.ifgm.TimePos
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.spec.ifgm import TimePos
Category:	SPIRE/Pipeline/Spectrometer Task

Description

Helper class to store Time/Position doublets for the CreateIfgmTask.


See also

- [CreateIfgmTask](#)

History

- 2007-09-07 - TRF: First version
- 2009-01-19 - TRF: [SPIRE SPR-1133] Fixed URM warnings.
- 2009-04-16 - TRF: [HCSS SPR-6051], [SPIRE SPR-1135]: Changed jcategory tag from Task to task
- 2010-05-18 - TRF: [SPIRE-2280]: Fixed URM issues.

1.251. translateMcuet

Full Name:	herschel.spire.ia.pipeline.common.engdata.TranslateMcuetTask
Alias:	translateMcuet
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.engdata import TranslateMcuetTask
Category:	SPIRE/Pipeline/Common Task

Description

Translate MCUET Task.

Task to translate column names of MCUET products.

API Summary

Properties
EdpProduct mcuet [INPUT, MANDATORY, default=No default value.]
NhkTimeline nhkt [INPUT, MANDATORY, default=No default value.]
EdpProduct mcuet_out [OUTPUT, OPTIONAL, default=No default value.]

API details


Properties

EdpProduct mcuet [INPUT, MANDATORY, default=No default value.]
Input MCUET.
NhkTimeline nhkt [INPUT, MANDATORY, default=No default value.]
Nominal Housekeeping Timeline.
EdpProduct mcuet_out [OUTPUT, OPTIONAL, default=No default value.]
Output MCUET.

History

- 2009-09-05 - PPa: First version.
- 2009-12-16 - PPa: Use new AbstractComposite methods instead of deprecated ones.
- 2010-05-17 - PPa: Improve URM [SPIRE-2259].

1.252. WaveletDeglitcherTask

Full Name:	herschel.spire.ia.pipeline.common.deglitch.WaveletDeglitcherTask
Type:	Java Task - 
Import:	from herschel.spire.ia.pipeline.common.deglitch import WaveletDeglitcherTask
Category:	SPIRE/Pipeline/Spectrometer Task

Description

WaveletDeglitcherTask

API Summary

Properties
DetectorTimeline input [Input, Mandatory, default=no default value]
Boolean correctGlitches [Input, Optional, default=Default value true]
Double scaleMax [Input, Optional, default=Default value]
Double scaleMin [Input, Optional, default=Default value]
Integer scaleInterval [Input, Optional, default=Default value]
Double holderMin [Input, Optional, default=Default value]
Double holderMax [Input, Optional, default=Default value]
Double correlationThreshold [Input, Optional, default=Default value]
Integer reconstructionPointsBefore [Input, Optional, default=Default value]
Integer reconstructionPointsAfter [Input, Optional, default=Default value]
String optionReconstruction [Input, Optional, default=Default value]
BuildingBlockProduct result [Output, MANDATORY, default=no default value]
Product.class debug [Output, MANDATORY, default=no default value]

API details

Properties

DetectorTimeline input [Input, Mandatory, default=no default value]
Input DetectorTimeline product.
Boolean correctGlitches [Input, Optional, default=Default value true]
correctGlitches option parameter true reconstruction will be done.

Double scaleMax [Input, Optional, default=Default value]
scaleMax value for the analyses. suggested range values 5 to 10.
Double scaleMin [Input, Optional, default=Default value]
scaleMin value for the analyses. suggested range values 1 to 5.
Integer scaleInterval [Input, Optional, default=Default value]
scaleInterval number of voice thickness of scale exploration
Double holderMin [Input, Optional, default=Default value]
holderMin value for Holder coefficient the analyses. suggested around -1.4.
Double holderMax [Input, Optional, default=Default value]
holderMax for holder coefficient suggested around -0.6.
Double correlationThreshold [Input, Optional, default=Default value]
correlationThreshold for correlation coefficient of linear regression range values 0.75 to 1
Integer reconstructionPointsBefore [Input, Optional, default=Default value]
reconstructionPointsBefore number of samples affected by the reconstruction before detected sample, range values 0 to 30.
Integer reconstructionPointsAfter [Input, Optional, default=Default value]
reconstructionPointsAfter number of samples affected by the reconstruction after detected sample, range values 0 to 30.
String optionReconstruction [Input, Optional, default=Default value]
optionReconstruction choices the reconstruction method ex : linearInterpolation.
BuildingBlockProduct result [Output, MANDATORY, default=no default value]
corrected BuildingBlockProduct product
Product.class debug [Output, MANDATORY, default=no default value]
debug only for debug Product product.


History

- 2007-02-28 - CO: first java version
- 2007-05-02 - CO: add default values for task parameters
- 2007-07-10 - CO: delete Level and Level1DeglitchingTask. DeglitchingTask now deals only
- 2007-05-02 - CO: with SDT, PDT or SDI. A class in /scripts transform a RAW sdt prod into a DetectorTimeline (sdt)

- 2007-05-02 - CO: adapt with new definition of detectorTimeline prod
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching->herchel.spire.ia.pipeline.common.deglitch
- 2007-10-12 - TRF: Replaced deprecated SDI setSignal() and getSignal() methods.
- 2007-10-26 - use: a copy of the input argument to build the output data. use getChannelNames() method to retrieve pixel names.replace perform() method by execute()
- 2008-04-29 - DB: correction warning Trevor Fulton version build 814
- 2008-09-12 - DB: update add mask traitement
- 2008-10-07 - DB: add description to task Parameters
- 2008-11-14 - DB: update SpireMask GLITCH -> GLITCH_FIRST_LEVEL
- 2008-11-20 - DB: input becomes Detector TimeLine instead of BuildingBlockProduct
- 2008-11-20 - DB: DeglitchingTasks becomes interruptible]
- 2008-11-20 - DB: SpireMask.GLITCH_FIRST_LEVEL_UNCORR). SpireMask also used to update MetaData in output product. [SPIRE SPR-1030]
- 2008-11-21 - DB: implement progress bar]
- 2008-12-15 - DB: rename 'reconstruction' input parameter as 'correctGlitches'
- 2008-12-15 - DB: remove the fileName meta data in output product
- 2009-01-23 - DB: correction jhistory suppress @SuppressWarnings("unchecked")
- 2009-01-28 - DB: add description @jparameter tag
- 2009-03-10 - DB: SCR-1201 no fail if there is no signal in input product
- 2009-03-12 - DB: SCR-1201 no fail if there is no dataset signal in input product and product type set to SDT or PDT
- 2009-03-18 - DB: SCR-1060 Modification of signification of SPIRE 1st level deglitching masks
- 2009-04-02 - DB: update default values of parameters
- 2009-04-23 - DB: add Meta data for QC SCR 1360
- 2009-04-24 - DB: add Metadata for QC even if equal 0
- 2009-04-29 - DB: suppress @jparameter scaleMax, tag in excess (SPR 906)
- 2009-06-01 - DB: modified setCreator
- 2009-06-02 - DB: SCR-1460 Add quality control metrics to First Level Deglitching
- 2009-06-09 - DB: add modified setCreator
- 2009-07-09 - DB: change task name to "deglitchTimelineTask" SPIRE 1650 and modif input names
- 2009-08-06 - DB: changed script name to deglitchTimeline in constructor
- 2009-09-10 - DB: SPIRE-1935 new reconstruction by linear interpolation, new parameters

- 2009-09-18 - DB: corrected s in parameters names
- 2009-10-09 - DB: SPIRE-2026 add description parameter metadata, correct SprireMask metadata setting SPIRE-2041
- 2009-11-12 - DB: Reconstruction by Polynomiale fitting
- 2009-11-30 - DB: update deprecated method
- 2010-01-25 - DB: SPIRE-2347 DeglitchTimelineTask should set QC metadata based on usable channels, only use not MASTER bit
- 2010-01-25 - DB: SPIRE-2257 update javadoc (public/protected field ->private)
- 2010-02-04 - DB: SPIRE-2347 DeglitchTimelineTask should set QC metadata based on usable channels, also for the count of all samples
- 2010-04-27 - DB: SPIRE-2478 DeglitchTimelineTaskis too verbose , update info message
- 2010-04-29 - DB: SPIRE-2530 rename as WaveletDeglitcherTask
- 2010-05-11 - DB: add ParameterValidator
- 2010-05-11 - DB: re-open SPIRE-2478 DeglitchTimelineTaskis too verbose , update info message misunderstanding on message

1.253. Wavelet

Full Name:	herschel.spire.ia.pipeline.common.deglitch.wavelet.Wavelet
Type:	Java Class - 
Import:	from herschel.spire.ia.pipeline.common.deglitch.wavelet import Wavelet
Category:	class

History

- 2006-02-24 - CO: first version
- 2007-02-22 - CO: admissibility factor added in variables definition
- 2007-03-08 - CO: compute only c_psi between scales 0 and f_x_max
- 2007-09-04 - CO: move to new package herschel.spire.ia.modules.deglitching ->herschel.spire.ia.pipeline.common.deglitch
- 2009-01-23 - DB: correction jhistory
- 2010-01-25 - DB: SPIRE-2257 update javadoc method return and param tag