Solar Orbiter: SPICE

Data Product Description Document

[SPICEFITS]

Issue **2.2**

	Name	Signature / Date
Prepared by	**Terje Fredvik and Stein Haugan**	
Reviewed by		
Approved by		
Approved by		

Institute of Theoretical Astrophysics
University of Oslo
P.O. Box 1029 Blindern
0315 Oslo
Norway

DPDDFD – Data Product Description Document for Dummies

The DPDD is a highly technical and very detailed document. Below we therefore give a brief and comprehensive summary of some of the most important aspects that users of SPICE files should be aware of.

1. **Write papers based on the newest official files**
   - For reproducibility, always use SPICE files that are publicly available from the Solar Orbiter Archive (SOAR) when preparing scientific papers [SOARURL].
   - Always use the latest version of the file: at the time of writing (April 2026), this means files with `'DR6'` in the filename (i.e. files belonging to Data Release 6 [IASDRURL]).
   - SPICE consortium members may take a sneak-peek of unreleased files using the Oslo SPICE archive [UIOARURL], but publications should be based on the publicly released SOAR files.
2. **Real scientists use L2**
   - Use L2 files for scientific analysis [4.4.2]. The reason: L2 files are calibrated [4.4.2.2]. Note that the calibration does not include corrections of effects caused by the point spread function [3.3.3].
   - Use L1 only if you need uncalibrated data or detailed telemetry information [4.4.1].
   - Note that in L2 files, adjacent spectral windows are merged into a single HDU, therefore the number of windows in L2 files may not be the same is in the corresponding L1 file or in the planning-tool observation definition [4.4.2].
   - Treat L3 files cautiously: L3 FITS files are useful for quickly inspecting automatically fitted line intensities, velocities, and widths [4.4.3]. L3 QL (JPEG/MP4) is for visual browsing only [3.3.5]. Note that any velocity trends have been removed from L3 QL files.
3. **The filenames tell a lot**
   - Basic information about the observation can be extracted from the filename [4.1].
   - `solo_L2_spice-n-ras_...` → calibrated (i.e. **L2**), **n**arrow-slit **ras**ter.
   - `...w-sit_...` → **w**ide-slit **sit**-and-stare.
   - `...-exp_...` → single **exp**osure.
   - `...V05_369099127-268-DR6.fits` → Version **05** of the file, raster repetition number **268** for SPIOBSID **369099127**, the file belongs to Data Release **6.**
4. **The FITS headers tell the rest**
   - Information from the telemetry, the planning tools, and other sources, can be found as FITS keywords or in special FITS file extensions [4.4.1.2, 4.4.2].
   - Example: where was SPICE pointed? Each HDU has a complete set of WCS keywords [4.3]. In addition, the metadata includes easily understandable keywords like e.g. `TARGET`, that may have a value like `'limb, South'`.
   - Example: is the S/C pointing unstable during the observation? No problem, the pointing variation is described in the FITS files [4.4.1.3.6].
   - Example: what was the exact time of observation for each exposure in a raster? Variable keywords store e.g. one value per exposure for this and other keywords [4.4.4].

- Example: what was done to the data during the L1 to L2 calibration? `PRSTEPn` and the other `PRxxxxn` keywords give a detailed description of all processing steps that were applied [4.4.2.2].

5. **Don't fear the `NaN`s!**
   - Some of the `NaN`s in L2 files mark data spikes and saturated pixels [4.4.2.1].
   - These `NaN`s are good for you: for line fitting, no data is better than incorrect data
   - Original and estimated values for pixels set to `NaN` due to spikes or saturation are stored in special extensions [4.4.2.1.2, 4.4.2.1.4]
   - The IDL `spice_data` object may be used to replace `NaN`s with original or estimated values where scientifically justified (or to make pretty pictures without doing line fits) [4.4.2.1.1, 4.4.2.1.3].

EXTERNAL DISTRIBUTION

Name	Organisation
spice-calibration.ias@services.cnrs.fr	*e-mail list*
https://spice-wiki.ias.u-psud.fr/	*SPICE team internal wiki*
https://idoc-projets.ias.u-psud.fr/redmine/	*SPICE operations redmine issue tracking tool*

CHANGE LOG

Date	Issue	Revision	Pages	Reason for change
28.06.2018	0	1	All	Document created based on SOL-SGS-OTH-0004-DPDDtemplate and [SPICELLFITS]
01.11.2018	0	9	All	Document rewritten and extended.
07.11.2018	0	91	All	Added comments based on IAS/UiO telecon.
07.11.2018	0	92		Fixed date, added READMODE in Section 4.7.1, removed an accidentally pasted random number
04.03.2019	0	93	All	– Updated sections that are copied from LLDPDD (from v1.7). Noticeable changes include a new paragraph on "Scanned Time Series" in Section 4.1, a new header example in Section 4.7.1 with some new LL01+ keywords, clarifications on the relationship between `OBS_TYPE` and the SPICE on-board Observation ID in Section 4.7.1.1.1, a new section 4.7.1.2.6 on keywords describing on-board processing steps, and a new table in Appendix B on the keywords that are derived from the value of `STUDYFLG`. See change log in LLDPDD for further details. – 3.3.3, 3.3.4.3, 4.6: the L2 data cubes will be corrected for geometric distortions, only S/C roll will be described by the `PCi_j` matrix. – 3.3: some restructuring of the text, changed temporary placeholder names of calibration routines to better match input from MPS. – 4.7.1: added L1+ keywords `UCD`, `OBS_ID`, `TIMEDER`, `TIMESYER`, `DATE-AVG`, `SOOPNAME`, `TELAPSE`, `LONPOLE`, `SPECSYS`, `VELOSYS`, `INFO_URL`, and Solar ephemeris data keywords. Renamed `SOOP_ID` to `SOOPTYPE`. – 4.7.1.1.1: added explanation of Solar Orbiter-wide keyword `OBS_ID`

				– 4.7.2: brief mention of L2+ keywords PRxxxxn, CRDERi, CSYERi, solar ephemeris data keywords. – 4.7.2.1: New section on keywords describing the calibration processing steps (L1 to L2) – 4.7.4: New section on keywords describing the processing steps when creating derived data products (L2 to L3) – Appendix C rewritten – Multiple minor changes, reformulations and fixing of typos
21.03.2019	0	94	All	– 4.7.1 and 4.7.3.1: corrected value of BUNIT – 3.2, 3.3.4.1, 3.3.4.3, 4.1, 4.2, 4.7.1.5, 4.7.3.2, and 4.7.4: Modified descriptions of intensity-windows to make it clear that such windows are not necessarily observed in line/background pairs. – 4.7.2.1: Added keyword PRPVERn giving the version of the processing function given by PRPROCn
03.05.2019	1	00	All	– Multiple clarifications, restructuring and moving of text, fixings of typos and minor modifications. – Updated sections that are copied from LLDPDD (v1.8). Noticeable changes include new section 4.7.1.1.7 on FILE_RAW, and rewording on intensity-windows in 4.7.1.5 – Keywords TIMEXOBT and TIMEXUTC renamed TIMAQOBT and TIMAQUTC.
02.12.2019	1	1	All. 3.3.1, 4.3.1, 4.4.1.2, 4.4.1.3.1, 4.4.2, 4.4.3.1, 4.4.4.2	– Updated sections that are copied from LLDPDD (v.1.10). Noticeable changes include new header example with new COMPRESS and COMP_RAT replacing COMPDESC. COMPPARA set to 0 for SHC and uncompressed data. New keyword SPIOBSID giving the SPICE Observation ID, no longer equal to OBS_MODE. Simplified sections by removing all examples of header differences between the example header and headers of HDUs with other window types and study types. Added new Section 4.4.4.2. – All: Removed references in figures, tables and text to the non-existing study/window type combination sit-and-stare intensity-windows.

				– Removed old Section 4.4.1.4.2 on description of processing steps in L1 files – 3.3.1: Added additional processing steps and new sub-section 3.3.3.1 describing the handling of telemetry loss. – 4.3.1: Added description of representation of space craft roll and geometrical distortions. – 4.4.1.2: Brand new L1 FITS header example. – 4.4.1.3.1: Rewritten. – 4.4.2: Added description of L1 to L2 change of window size. – 4.4.3.1: Use SI units All: Multiple minor corrections, additions, and deletions.
17.12.2019	1	2	1.3, 4, 4.4.1.2, 4.4.1.3.4, 4.4.1.4, 4.4.2	– 1.3: Added [SPICELOST] to Reference Documents. – Updated sections that are copied from LLDPDD (v.1.11). Noticeable changes include added description of HDU layout for two-exposure mode (4.1), clarification of text and fixing of typo regarding the representation of lost telemetry in binary table extensions, and new Section 4.4.1.4 explaining the value of the `BLANK` keyword and the data type of the HDU's data arrays – 4.4.1.2: New header example – 4.4.2: New default behaviour of the calibration routines is to crop the windows to preserve the number of pixels instead of padding the windows to preserve the data

01.09.2020	1	3	1.3, 4.3, 4.4.1.2, 4.4.2, 4.4.1.3.3, 4.4.1.4.1, 4.4.4.3, 4.4.5, Appendix B, Appendix C	– Multiple sections: Updated paragraphs that are copied from LLDPDD (v1.13), see changelog therein. Important changes are found in paragraphs describing the scan direction of the slit: 4.3, 4.4.1.3.3, and Appendix B.
				– 1.3: Added [DISPERSION] to Reference Documents
				– 4.4.2: reverted to the previous way of treating windows, i.e. with padding to preserve the data. Added information on changes in plate scale when going from L1 to L2
				– 4.4.1.2, 4.4.1.4.1, 4.4.4.3, 4.4.5: Renamed some of the FITS keywords used for lost telemetry bookkeeping to make the syntax more like the syntax of the SOLARNET keyword `NLOSTPIX`:
				`NCHKLOST -> NLOSTCHK`
				`NCHKFAIL -> NFAILCHK`
				`NPLNAPRX -> NAPRXPLN`
				`NPLNLOST -> NLOSTPLN`
				`PKTLOST  -> LOSTPKTS`
				`BINLOST  -> LOSTBINS`
				– 4.4.1.2: New FITS keyword `XSTART`. Removed `TIMSYER` and `TIMRDER` for the time being, may be reintroduced in the future
				– Appendix C: New paragraphs on IORs and Study Sets, fixed errors in Study Set table.
				– Multiple sections: future simple -> simple present…

31.05.2021	1	4	3.2, 3.3.3, 4, 4.1, 4.4.1.2, 4.4.1.3.7, 4.4.1.3.9, 4.4.1.4, 4.4.1.5, 4.4.1.6, 4.4.1.7, 4.4.2, 4.4.2.1, 4.4.3.1, 4.4.3.3, Appendix A, Appendix B, Appendix C	– 3.2: Rewritten text, updated all figures, added new figures 3-6 and 3-7. Updated all tables and added new table 3-4. – 3.3.3: Corrected names of calibration subroutines, added intensity units – 4: Updated Table 4.2. Added info on sit-and-stare studies and repeated full detector studies, highlighted these changes in the text – 4.1: `SPIOBSID-RASTERNO` is a part of the filename. – 4.4.1.2: Updated L1 header example with new keywords extracted from IORs and Study Sets, data statistics keywords, and updated keywords on data processing. – 4.4.1.3.7: New section on FITS keywords describing data processing – 4.4.1.3.9: New section FITS keywords describing linear fit to scan mirror position. – 4.4.1.4-7: Updated tables – 4.4.2: Added description of adjacent and near-adjacent windows. New value for `UCD`. – 4.4.2.1: Updated `PRxxxxn` keywords – Appendix A: Removed some descriptors that should not be there, e.g. [int] for sit-and-stare studies. – Appendix B: `RASTNUM` renamed to `RASTERNO`. – 4.4.3.1: New formatting of `BUNIT` values following recommendations given in the FITS standard – 4.4.3.3: Updated `PRxxxxn` keywords – Appendix C: Added new keywords extracted from the IORs: `STP` and `WINSHIFT`, and from the Study Sets: `MISOSTUD`, `XSTART`, and `MISOWIN`.

01.10.2021	1	5	4.1, 4.4.1.2, 4.4.2, 4.4.2.1, 4.4.5, Appendix C	– Multiple sections: Updated paragraphs that are copied from LLDPDD (v1.14), see changelog therein. Most changes are found in sections describing lost telemetry, e.g. 3.3.1, 3.3.1.1 and 4.4.1.3.5, – 4.1: Updated Table 4-2 – 4.4.1.2: Updated L1 header example – 4.4.2: Updated paragraphs on adjacent and near-adjacent windows in L2 files. Adjacent windows are now concatenated. – 4.4.2.1: Added example of `PRxxxxn` keywords describing window concatenation in L2 – 4.4.5: Rewritten – Appendix C: removed `DARKSPID` (the `SPIOBSID` of the dark that has been subtracted is given by `PRPARAn` (`n` being the processing step number of the `'DARK-SUBTRACTION'` processing step). Added `SETFILE` and `SETVER`. – A few minor changes in wording
13.10.2021	1	6	4.4.1.7, 4.4.2	– 4.4.1.7: `IWINBKG` is now the HDU *number* of the HDU storing its background window, not the HDU `EXTNAME`. `IWINLINE` gives a comma separated list of HDU numbers of all HDUs having this background window defined as background. – 4.4.2: Neither Intensity-windows nor dumbbells are merged.

03.01.2023	1	7	2.3.1, 2.3.2, 3.2, 3.3.2, 3.3.3, 3.3.4, 3.3.6, 3.4, 3.4.1, 3.4.2, 4, 4.1, 4.4.1.2, 4,4,1,3,3, 4.4.1.5, 4.4.1.3.5, 4.4.1.3.6, 4.4.1.3.9, 4.4.1.3.10, 4.4.2, 4.4.2.1, 4.4.3.2, 4.4.4.3, 4.4.5, Appendix A, Appendix C	–2.3.1, 2.3.2: Added information on calibration – 3.2 and throughout the text: emphasise that most user should use L2 files, not L1. – 3.2 and throughout the text: emphasise that dumbbells and intensity-windows are not used regularly – 3.3.2: Added information on SPICE-S/C coalignment and S/C pointing variations. Added information on pixel level offset subtraction, new variable keyword `RADCAL`, and the removal of lost telemetry binary table extensions in L2. – 3.3.3: specify correct physical units for wide-slit observations. Add merging of adjacent windows as L2 processing step – 3.3.4: added warning about outdated description of Level 3 FITS files – 3.3.6, 3.4, 3.4.1, 3.4.2: Replaced placeholder text with some very basic information – 4, Figure 4-1: new figure to illustrate the Y-shift of the spectra on the two detectors – 4, Table 4.2: Added information on actual usage of the different study type/window type combination. 4.4.1.2: New L1 header example – 4.4.1.3.3 and throughout the text: added information on the FITS keyword consequences of merging of adjacent windows in L2. Removed incorrect description of reversal of wide-slit images – 4.4.1.3.5: Added description of all keywords describing the telemetry – 4.4.1.3.6: New section coordinate distortions –4.4.1.3.9: New description of how to determine the study type – 4, 4.4.1, 4.4.1.5, etc: Added information on the lack of separate dumbbell HDUs due to the SW/LW detector y-offset 4.4.2.1: Updated processing steps – 4.4.3.2: `IWINBKG`/`IWINLINE` are HDU numbers, not the `EXTNAME` – 4.4.1.3.6: New section on coordinate distortions due to S/C pointing instability – 4.4.1.3.8: Added information on pixel level offset subtraction – 4.4.1.3.10: rewritten to explain that the study type is now determined from the

				IORs and study definition files instead of mirror positions. 4.4.2: Updated description on L2 FITS keywords 4.4.4.3: New section describing the variable keyword RADCAL – 4.4.5: Added information on telemetry completeness keywords – Appendix A: added information on actual data product type usage – Appendix C: Added FITS keyword SETVER. Modified description of EXTNAME in the case of merged windows
02.11.2023	1	8	1.3, 3, 3.2, 4, 4.4.1.2, 4.4.1.3.6, 4.4.2.2, 4.4.3, 4.4.3.1	– All: added information on the infrequent usage of intensity-windows and dumbbells. Removed some detailed descriptions of these types of observations. Simplified and shorted text a few places. Removed green colouring of text that indicated sections identical to sections in [SPICELLFITS] – 1.3: added references – 3: added information on IDL analysis tools – 3.2: removed detailed descriptions and illustrations of data flows of seldomly obtained observations. Updated Figure 3-1 – 4,: Updated Table 4-2 – 4.4.1.2: new L1 header example – 4.4.1.3.6: added information on new SolarSoft support of coordinate distortions. Updated Figure 4-2 including caption. – 4.4.2.2: Updated processing steps describing calibration, including correction for burn-in and time dependent detector response – 4.4.3: Rewritten from scratch, now describes the L3 R files being produced by the pipeline – 4.4.3.1: Rewritten from scratch

10.10.2024	2	0	3.2, 3.3.2, 3.3.3, 3.3.4, 3.3.5, 3.3.5.1, 3.3.5.2 3.3.5, 4, 4.2, 4.4.1.2, 4.4.1.3.7, 4.4.2, 4.4.2.2, 4.4.3	– All: Removed the concept of L3 C files. Removed all mentions of L2 files being used to create L3 QL files. Renamed L3 R files to L3 P to adhere to [S-META]. Removed all TBDs and “will be created” when it comes to L3 QL images and movies. – 3.2: New Figure 3-1. Removed table of file types created by the pipeline. Mentioned the future possibility of gathering repeated full detector observations in a single L2 file. – 3.3.2: Added info on new SPICE vs S/C pointing offset correction – 3.3.3: Shortened, added info on combined darks and cleaned darks, temperature dependent wavelength scale, and saturated pixels –3.3.4,: Updated description of the contents of L3 P files – 3.3.5: Simplified description of L3 QL files. – 3.3.5.1: new section – 3.3.5.2: new section – 4: Updated window/study usage in Table 4-2 with numbers up to September 2024 – 4.2: new definition of L3 QL filenames – 4.4.1.2: New L1 header example – 4.4.1.3.7: Updated list of PRxxxxn keywords – 4.4.1.2: Added reference to new section describing saturated pixels in L2 files. Updated list of L2 keywords that differ from L1 keywords – 4.4.2: Added info on and cross-references to sections describing saturation and binary table extensions storing information on lost telemetry – **Error! Reference source not found.**: New section, with two sub-sections, describing saturated pixels in L2 files – 4.4.2.2: Updated examples of PRxxxxn keywords in L2 files – 4.4.3: Updated description of L3 P FITS files

25.10.2024	2	1	4.2	– All: Corrected typos, minor adjustments of the text based on user input – 4.2: Replaced underscores with hyphens in the freefield part of the filename. Deleted [-db] as a possible part of the filename.
17.04.2026	2	2	1.3, 3.2, 3.3.3, 3.3.4, 3.3.5.1, 4, 4.1, 4.4.1.2, 4.4.2, 4.4.2.1, 4.4.2.1.1, 4.4.2.1.2, 4.4.2.1.3, 4.4.2.1.4, 4.4.2.1.5, 4.4.2.2	- 1.3: Links to FTOOLS utility package, Oslo sigma-clip routine, IAS Data Release URL, UIO SPICE archive URL, and SOAR URL. - 3.2, 3.3.3, 4, 4.4.2, 4.4.2.2: Description of the concatenation of repeated full detector observations into single L2 files - 3.3.3: Information on the lack of corrections of effects caused by the PSF - 3.3.4: Specified than only multi-exposure narrow-slit observations lead to L3 files, and that up to 25 lines may be fitted in a readout window - 3.3.5.1: Information on removal of velocity trends in QL images - 4.1: Added Data Release number to the file name - 4.4.1.2: New L1 example header - 3.3.3, 4.4.2: Information on sigma-clipping, including new FITS keywords and the increase in number of `NaN` pixels. - 4.4.2.1, 4.4.2.1.1, 4.4.2.1.2, 4.4.2.1.3, 4.4.2.1.4, and 4.4.2.1.5: New and rewritten sections describing pixel lists storing information on saturated and spike pixels

				- 4.4.2.2: updated list of keywords describing the L1 to L2 calibration, including the new 'DESPIKING' and 'FILE-CONCATENATION' processing steps - Added a DPDDFD at the beginning of the document

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose and Scope

This Data Product Definition Document describes the format and content of the SPICE Science data. It includes descriptions of the data products and associated metadata, including the data format, content, and generation pipeline. These products will be stored and distributed from the Solar Orbiter Archive (SOAR) of the SOC.

The specifications described in this document apply to all SPICE Science products submitted to ESA's Solar Orbiter SOC for further archival and exploitation. The specifications described in this document apply to all SPICE data products generated by the Science Data Pipeline running at the SPICE premises in Oslo. It does not address the Low Latency data (see [LLData]) delivered by the Low Latency Pipeline run at SOC, as these data products are described in [SPICELLFITS]. However, the LL01 data products produced by the Low Latency Pipeline are identical to the Level 0 data products created by the Science Data Pipeline, save for some minor FITS keyword differences.

1.2 Applicable Documents

APPL. DOC. SHORT	APPLICABLE DOCUMENT TITLE	DOCUMENT ID	ISSUE
[DPICD]	*Data Producer to archive ICD*	*SOL-SGS-ICD-0002?*	*Will be rewritten from scratch*

Table 1-1: Applicable Documents

1.3 Reference Documents

REF. DOC. SHORT	REFERENCE DOCUMENT TITLE	DOCUMENT ID	ISSUE
	Template for SoLO Data Product Description Document	SOL-SGS-OTH-0004-DPDDtemplate	
[ACRONYMS]	SPICE Acronym List	SPICE-RAL-LI-0001	1.0
[LLFITSICD]	Solar Orbiter Interface Control Document for Low Latency Data FITS Files	SOL-SGS-ICD-0005	1.4
[FITSpaper]	Definition of the Flexible Image Transport System (https://fits.gsfc.nasa.gov/standard40/fits_standard40aa-le.pdf)		4.0
[METADATA]	Metadata Definition for Solar Orbiter Science Data	SOL-SGS-TN-0009	2.6
[IOR ICD]	Solar Orbiter Instrument Operations Request ICD	SOL-SGS-ICD-0003	1.1
[SEGU]	SOC Engineering Guidelines for External Users	SOL-SGS-TN-0006	1.2
[SOAR]	Solar Orbiter Archive Plan	SOL-SGS-PL-0009	2.5
[MAN]	SPICE Instrument User Manual	SPICE-RAL-MAN-0001	13.0
[DATAICD]	SPICE Data Interface Control Document	SPICE-RAL-ICD-5003	10.0
[RAW]	SPICE FM Raw Science Data Handling Scheme	SPICE-RAL-TN-5202	1.0
[DECOMP]	SPICE Science Data Decompression Recipe	SwRI EM 17489-017	0
[SUNSPICE]	The SunSPICE Ephemeris Packeage for Solar Missions, William Thompson, 2018 (https://hesperia.gsfc.nasa.gov/ssw/packages/sunspice/doc/sunspice.pdf)		
[DISTORTIONS]	Simulated Image Distortions in SPICE Data, William Thompson, 2019 (https://spice-wiki.ias.u-psud.fr/lib/exe/fetch.php/documents:internal:simulate.pdf)		
[DISPERSION]	SPICE spectral dispersion function and detector geometry, Thompson and Young, 2020 (unpublished, under development)		
[TEMPWAVE]	SPICE pixel to wavelength relationship along the orbit, Giunta & Grundy, 2024, presented at 2024 SPICE Consortium meeting (https://spice-wiki.ias.u-psud.fr/lib/exe/fetch.php/meetings:consortium_202406:ag_spectral_shift_june2024.pdf)		
[CLEANDARK]	Cosmic Ray cleaned darks, Alfred Voyeux, 2024, unpublished, presented at 2024 SPICE Consortium meeting (https://spice-wiki.ias.u-psud.fr/lib/exe/fetch.php/meetings:consortium_202406:cosmic_removal.pdf)		
[MULTIDARK]	Dark correction using multi-dark observations , Sarah Leeks & Tim Grundy, 2024, unpublished, presented at 2024 SPICE Consortium meeting (https://spice-wiki.ias.u-psud.fr/lib/exe/fetch.php/meetings:consortium_202406:spice_darks_summary_jun_2024_v2.pdf)		
[LLData]	Solar Orbiter Low-Latency Data: Concept and Implementation	SOL-SGS-TN-0003	1.2
[SPICELLFITS]	SPICE Low Latency Data Product Description Document	SPICE-UIO-DPDD-0001	1.14

[SPICELOST]	Solar Orbiter SPICE Reconstructing Data with Lost Telemetry Packets (https://spice-wiki.ias.u-psud.fr/lib/exe/fetch.php/documents:internal:spice-uio-dpdd-0003-0.2-reconstructing_data_with_missing_telemetry_-_comresssed_images_hd.docx)	SPICE-UIO-DPDD-0003	0.2
[S-META]	SOLARNET Metadata Recommendations for Solar Observations (http://sdc.uio.no/open/solarnet/ , https://arxiv.org/abs/2011.12139,)		2.2-live
[FITSCOORD]	Representation of spectral coordinates in FITS, Greisen & al., 2006, A&A, 446, 747-771 (https://www.aanda.org/articles/aa/pdf/2006/05/aa3818-05.pdf)		
[WCSDISTORTIONS]	Representations of distortions in FITS world coordinate systems, Calabretta et al., 2004, draft (https://fits.gsfc.nasa.gov/wcs/dcs_20040422.pdf)		
[SSTRED]	SSTRED: Data- and metadata-processing pipeline for CHROMIS and CRISP, Löfdahl et al., 2021, A&A 653, A68 (https://www.aanda.org/articles/aa/pdf/2021/09/aa41326-21.pdf)		
[CFIT]	The Component Fitting System (CFIT) in IDL	CDS Software Note No. 47	2
[DATAMAN]	SPICE Data analysis user's manual (https://spice-wiki.ias.u-psud.fr/doku.php/data:data_analysis_manual)		
[IDLANA]	SPICE IDL Quicklook and Data Analysis Software (https://github.com/ITA-Solar/solo-spice-ql/wiki/)		
[FTOOLS]	Repository of high-performance C extensions for image processing and curve fitting: https://github.com/steinhh/ftools		
[SIGMACLIP]	Python function used to perform the sigma-clip calibration step: https://github.com/ITA-Solar/solo-spice-ql/blob/develop/utils/python/spice_sigma_clip2.py		
[IASDRURL]	Lists of files belonging to Data Releases: https://spice.ias.u-psud.fr/spice-data/		
[UIOARURL]	Oslo SPICE archive: http://astro-sdc-db.uio.no/vol/spice/fits/		
[SOARURL]	Solar Orbiter Archive: https://soar.esac.esa.int/soar/		

Table 1-2: Reference Documents

1.4 Abbreviations and Definitions

FITS	Flexible Image Transport System
JPEG	Joint Photographic Experts Group; a lossy image format used for L3 QL images (file extension .jpg)
MPEG-4	A lossy format for encoding and compressing video images, used for L3 QL movies (file extension .mp4)
LL01	Low Latency Level 1
L0	Level 0
L1	Level 1
L2	Level 2

L3	Level 3
L3 QL	Level 3 Quicklook – can be either L3 QL JPEG or L3 QL MP4
SHC	Spectral Hybrid Compression
SOAR	Solar Orbiter Archive
SOOP	Solar Orbiter Observing Program
WCS	World Coordinate System

A complete list of all acronyms used in the SPICE project can be found in [ACRONYMS].

2. SPICE INSTRUMENT DESCRIPTION

SPICE is a high-resolution imaging spectrometer operating at ultraviolet wavelengths. It will address the key science goals of the Solar Orbiter mission, by providing quantitative knowledge of the physical state and composition of the plasmas in the solar atmosphere, in particular investigating the source regions of outflows and ejection processes which link the solar surface and corona to the heliosphere. [MAN]

2.1 Science Objectives

See Table 2 in Section 2.1 of [MAN].

2.2 Operational Modes

Data from the SPICE instrument is read out in wavelength regions, see [MAN] Section 2.2.2.2. Depending on the operational mode (study definition), the data from each of these readout windows may result in 1, 2 or 3 data arrays. It is also possible to make a full-frame read-out, resulting in one data array per detector array.

2.3 Calibration

2.3.1 On-ground Calibration

The observations are calibrated on-ground by the Science Data Pipeline running in Oslo. Calibrated files are of Level 2, see Section 3.3.3. Most users should use Level 2 files in their data analysis.

2.3.2 In-flight Calibration

Dark current subtraction is the only form of calibration that may be applied in-flight. If not applied on-board, the dark current is subtracted on-ground by the Science Data Pipeline when creating Level 2 files.

3. DATA GENERATION PROCESS AND ANALYSIS PROCESS

The SPICE Science Data products are produced by the Science Data Pipeline running in Oslo. The data generation and analysis process are described in this section.

The science data products produced by the Science Data Pipeline are immediately available as to the SPICE consortium through the Oslo SPICE archive [UIOARURL]. The data are made available to the broad solar community through the Solar Orbiter archive [SOARURL] following the policies described in the Archiving Plan [SOAR].

For practical hints and tips regarding data analysis, please see the SPICE Data analysis user's manual [DATAMAN]. IDL users may use the tools developed at UiO for quicklook and analysis purposes, see the team's wiki pages for details [IDLANA].

3.1 Scientific Measurements

SPICE is a high-resolution imaging spectrometer operating at extreme ultraviolet (EUV) wavelengths.

3.2 Data Flow Overview

The SPICE team in Oslo retrieves the telemetry data from IAS, passing it as input to the Science Data Pipeline running in Oslo. The Science Data Pipeline produces science FITS files of Level 1, 2 and 3, and L3 Quicklook files.

Most users should use Level 2 files for their data analysis. Level 1 files are meant for advanced users. L3 files should only be used for quicklook purposes.

Level 1 and Level 2 files are in the FITS format (Sections 3.3.2 and 3.3.3).

Level 3 files come in three flavours: Level 3 FITS (L3 P, Section 3.3.4), Level 3 Quicklook images (L3 QL JPEG, Section 3.3.5.1), and Level 3 Quicklook movies (L3 QL MP4, Section 3.3.5.2).

The two SPICE detectors are read in wavelength regions, or windows. Each window is of a specific *window type*, see Table 4-1. A study is of a given *study type* (Table 4-2) and may consist of multiple window types. The study may be repeated. The dataflow differs for the multiple combinations of window types and study types, and for repeated vs non-repeated studies. See Section 4 for further information regarding window types, study types, and data products.

The most frequently obtained SPICE science observation is narrow-slit spectral rasters (see also Table 4-2). The dataflow inside the Science Data Pipeline for such an observation is illustrated in Figure 3-1.

Most SPICE observations are full detector single exposures. None of these are labelled as “Science”, although only a minority of these files are darks. Many of these files may therefore contain scientifically interesting data. Since most full detector observations are repeated, we concatenate all repetitions of such studies in single L2 FITS files starting in Data Release 6. For L1 the individual exposures are still stored in separate FITS files.

Wide-slit rasters are also obtained on a regular basis. Other observations are either mostly used for calibration purposes (full-detector single exposure observations), are not that frequently used (sit-and-stare observations), or have hardly been used at all (e.g. Intensity-windows and dumbbells). We therefore omit a thorough description of these types of data products in this document.

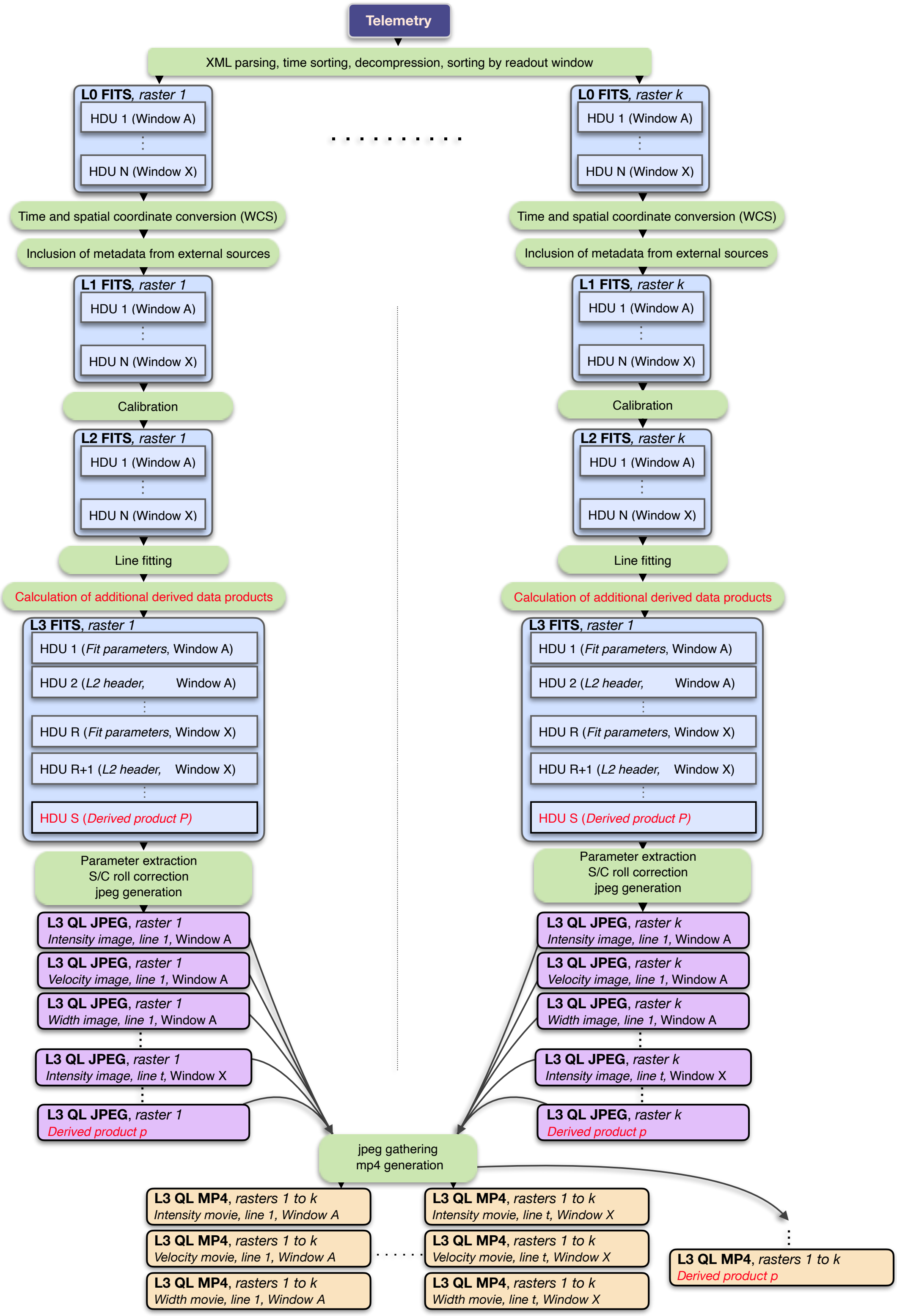

Figure 3-1 Pipeline dataflow: narrow-slit spectral-profile. Red colour signals planned processing steps and HDUs/file types that are not yet part of the pipeline. L3 FITS and L3 QL images are only made for rasters or sit-and-stare. L3 QL movies are only made for repeated rasters or sit-and-stare observations.

3.3 Data Generation

The Science Data Pipeline takes XML telemetry as input and produces files of all file levels described in Sections 3.3.1 to 3.3.4 sequentially. The pipeline is written in IDL, with the use of java objects and C executables. To speed up the processing the pipeline consists of tens of parallel IDL processes running simultaneously.

Most users should use Level 2 files for their data analysis. Level 1 files are meant for advanced users who either require full control of the calibration process or need access to the extended information on lost telemetry packets that these files contain. L3 files should only be used for quicklook purposes.

3.3.1 Level 0 – Raw Data (FITS)

Level 0 FITS files (L0) are used as temporary files within the Science Data Pipeline and will normally not be made available to the scientific community[1]. L0 files contain uncalibrated data expressed in engineering units (counts).

The processing steps performed by the Science Data Pipeline to create an L0 file:

1. The Science Data Pipeline regularly checks its input directory for processing requests.
2. If a request is found, the XML contents of the request is read, and the telemetry data is extracted.
3. The telemetry packets are sorted by time of packet creation.
4. A processing request may contain telemetry from multiple observations. The following steps are repeated for each observation found in the sorted telemetry stream:
5. Each telemetry packet is inspected, and the data and metadata of a packet is extracted according to Sections 4.2.6.1 to 4.2.6.3 of [DATAICD]. Data and metadata in telemetry packets are accumulated into collections called raster segments. The data array of a raster segment is a 1-dimensional 8-bit array.
6. When a raster segment is complete, or there are no packets left in the telemetry that belongs to the segment, it is decompressed according to [RAW] and [DECOMP]. Special care must be taken when telemetry packets are lost, see Section 3.3.1.1.
7. The decompressed raster segment data array is translated into a 3-dimensional 16-bit data cube with dimensions `(exposure,y,dispersion)`, according to [RAW]. If the telemetry was compressed using Spectral Hybrid Compression (SHC), this translation also involves an inverse Fourier transform as described in [DECOMP].
8. The decompressed raster segment data cubes are assembled into window data cubes. Window data cubes may consist of multiple raster segments, but each data cube only contains data stemming from a single readout window.
9. When all window data cubes of the observation are complete, or there are no more packets left in the telemetry, the data from each window data cube is reformatted into

[1] L0 files were made available to the SPICE consortium during the commissioning phase. The Science Data Pipeline up to (and including) the generation of L0 files is identical to the Low Latency Pipeline running at SOC. L0 files are indistinguishable from Low Latency Level 01 FITS files (LL01), except for the values of a few descriptive FITS keywords (e.g. `FILENAME`, `LEVEL`, `CREATOR`, etc.).

`(x,y,dispersion,time)` data cubes, with a singular `x` dimension for sit-and-stare observations and singular `time` dimension for raster scans.

10. A temporary L0 FITS file is created, with the data for each window stored in separate Header/Data Units (HDUs). The data from the first readout window are stored in the primary HDU, data from additional readout windows are stored in image extension HDUs.
11. If the telemetry contains dumbbell data that are downlinked in separate segments, i.e. not part of a spectral window, the data cubes are stored in the two last image extension HDUs (or in the last image extension HDU if only one dumbbell was recorded and/or received). Note that dumbbells are very seldomly downlinked separately, see Section 4 for additional information on dumbbell data.
12. Each L0 HDU that contains data from readout windows also contains all relevant metadata present in the telemetry packets, see Appendix B. Such HDUs also contain metadata that have been derived from the telemetry, in some cases using formulas and conversion tables stored in the virtual machine.
13. In the case of multi-exposure observations, the L0 file also contains a binary header extension storing the individual values of keywords that vary with exposure number, see Section 4.4.4.1.
14. In the case of multi-segment observations, the binary header extension mentioned in bullet 13 will also store the individual values of keywords that vary with segment number, see Section 4.4.4.2.
15. If telemetry packets are lost, the lost packet indices are stored in a binary table extension. If SHC compression was applied to the data, this binary table extension also stores the lost FFT Bin indices, see Section 4.4.4.4.
16. If telemetry packets are lost and JPEG or SHC compression was applied to the data, one or more image planes will either have approximated values or be completely lost (i.e. be set to the `BLANK` value). The coordinates of these image planes (or image plane ranges in the case of multi-segment observations) are stored in a binary table extension, see Section 4.4.5.[2]
17. GOTO 5

3.3.1.1 Reconstructing data with lost telemetry packets

During the first years of the mission, lost telemetry has not been a pronounced problem. We might not be that lucky in the future, and we have therefore implemented mechanisms for storing information on lost telemetry in the L1 FITS files. L2 files contain a simplified set of FITS keywords describing the completeness of the telemetry.

The implications of lost telemetry are discussed in [SPICELOST]. Missing telemetry packets affect the reconstructed data cube depending on the type of compression applied to the data:

- **No compression:** pixel columns in the dispersion-Y plane are lost, i.e., they have the `BLANK` value.
- **JPEG compression:** entire X-Y or dispersion-Y image planes of a single segment[3] are affected. *All* pixels in affected planes have *either*:

[2] For technical reasons the flagging of lost or approximated image plane ranges currently does not work if all packets of a segment are missing.
[3] Note that an observation may consist of multiple segments. Telemetry loss may in such cases affect only parts of the data cube image planes, see Section 4.4.5.

    - the BLANK value if the missing packets contained JPEG header information, or
    - approximated values if the missing packets did *not* contain JPEG header information
- **SHC compression:** *all* pixels stemming from the segment that the lost packets belonged to have approximated values. The degree of approximation depends on which FFT Bins are lost, see discussion in [SPICELOST].

In Level 0 and Level 1 files, detailed information concerning lost telemetry is stored in binary table extensions and in FITS header keywords of the observational HDUs, see Sections 4.4.4.4. and 4.4.5. These binary table extensions are not present in Level 2 files.

3.3.2 Level 1 – Engineering Data (FITS, uncalibrated)

Level 1 FITS files (L1) contain uncalibrated data expressed in engineering units (counts).

Note that most users should use Level 2 files for their data analysis. Level 1 files are meant for advanced users who either require full control of the calibration process or need access to the extended information on lost telemetry packets that these files contain.

The FITS headers include additional keywords giving times converted from on-board time (OBT) to UTC, and the spatial coordinates are converted from being given relative to the spacecraft boresight to coordinates relative to the Solar disc. L1 files also include additional metadata gathered from the Study Set files and the IORs. These additional metadata describe all available information about:

a) the study definition (e.g. name of the study, name of the readout windows, name of the author of the study, etc.), and
b) this instance of the study (e.g. the Solar Orbiter-wide ID of the observation, name of the SOOP, etc)

The L1 FITS files also include all metadata needed by the calibration routines to convert the file from L1 to L2, including Solar ephemeris data.

The processing steps performed by the Science Data Pipeline to create an L1 file:

1. An L0 file is given as input to the Level 1 FITS file generator.
2. The on-board time is converted to UTC using the SunSPICE IDL implementation of the SPICE[4] toolkit, see [SUNSPICE]. Additional FITS standard keywords DATE-BEG, DATE-END and DATEREF given in UTC are added to the L1 FITS headers.
3. Solar ephemeris data keywords are added using the SPICE toolkit.
4. IAS produces XML files with metadata that describe the specific studies and the specific instances of each study, i.e. the study definitions and the IORs. Oslo hosts a local database storing all this metadata. Based on this database and the FITS keywords in the input L0 file, additional metadata are added to the L1 file.
5. The keywords describing the pointing and FOV are converted from being relative to the spacecraft boresight to being relative to the solar disc (helioprojective) using SPICE toolkit routines and the XML metadata files obtained from IAS.
    - The spacecraft boresight pointing and roll are corrected for velocity aberration.

[4] NASA's Observation Geometry System for Space Science Missions.

- The offset between the SPICE and spacecraft boresight is corrected for, based on a coalignment analysis between SPICE and EUI (private communication, Auchère 2024)
- The `PCi_j` transformation matrix explained in Section 4.3 is updated to describe the resulting rotation of the field-of-view.
- The distortions of the Solar X and Solar Y coordinates due to any spacecraft pointing variations during the observation are stored in separate image extensions, see Section 4.4.1.3.6.

6. The L1 file inherits any binary table extensions present in the L0 file. An additional binary table column storing the acquisition time converted to UTC is added to the binary table extension. See Section 4.4.4 and 4.4.5 for details.
7. If a pixel level offset was added on-board, this offset is subtracted from the data array.

3.3.3 Level 2 – Science Data (FITS, calibrated)

Level 2 FITS files (L2) are calibrated L1 files[5] that are ready for scientific analysis.

The processing steps performed by the Science Data Pipeline to create an L2 file:

1. An L1 file is given as input to L2 FITS file generator. The file is sent to the calibration wrapper routine `spice_prep.pro` which performs multiple processing steps, including calling the calibration subroutines in bullet points 2 through 6:
2. `spice_prep_dark_offset_correction.pro:` subtract a dark frame from the data (only if a dark frame hasn't already been subtracted on-board). The dark image can be one the following:
    - A combined dark created from multiple darks ([MULTIDARK]),
    - A dark that has been cleaned for cosmic rays ([CLEANDARK]), or
    - A regular L1 dark file

   The combined darks and the cleaned darks will be made available in SolarSoft in the future.
3. `spice_prep_sigma_clip.pro`: identify data spikes using local sigma-clipping, set spike pixels to `NaN` (see Section 4.4.2.1)
4. `spice_prep_flat_field_correction.pro`: correct the data for flatfield.
5. `spice_prep_burnin_correction.pro`: correct the data for burn-in. Note that only sections of the detectors where strong lines have reduced the sensitivity significantly are corrected for burn-in (see Section 4.4.2.2)
6. `spice_prep_distortion_correction.pro`: correct the data for spatial and spectral distortions, e.g. slit tilts, spectral slant, detector misalignments and non-uniform dispersion. Apply the corrections to the data cubes by interpolation onto a linear grid. Update the WCS keywords describing the wavelength scale by correcting for the temperature dependency [TEMPWAVE].
7. `spice_prep_radiometric_calibration.pro`: calibrate the data to physical units: `W/m2/sr/nm` for narrow-slit observations and `W/m2/sr` for wide-slit observations. This step also includes correcting the data for the time-dependent response of the detectors (i.e. the reduction of detector sensitivity with time).
8. Set image planes to `NaN` if they contain approximated values in the L1 file, see Sections 4.4.4.4 and 4.4.5.
9. Set pixels to `NaN` if they are influenced by saturated L1 pixels, see Section 4.4.2.1.
10. Merge the data arrays of adjacent windows, see Section 4.4.2.

[5] Dark files or files originating from the special engineering setup study with `STUDY_ID=57` are not calibrated

11. The output from `spice_prep.pro` is a calibrated L2 FITS file. The data type of an L2 data array is floating point.
12. The L2 file inherits any binary table extension storing variable keywords from the L1 file, with an additional variable keyword `RADCAL`. Any binary table extensions storing information on lost telemetry are removed.

If an L2 file that is part of a series of repeated full detector observations have been created: concatenate all available L2 `-exp` files of this series. Replace all corresponding `-exp` files in the Oslo archive [UIOARURL] with a single concatenated L2 `-ras` or `-sit` file.

Note that the calibration does not include any corrections of effects caused by the point spread function. Users should be aware that the rotated PSF may cause both velocity artifacts and image displacements when scanning through the line profile. For further details on this issue and guidance on how this may affect your analysis of SPICE files, please confer with members of the SPICE consortium.

3.3.4 Level 3 – Higher Level Data (FITS, Gaussian fit parameters)

Level 3 FITS files contain the results of automatic Gaussian fitting of parameterized components to the line profiles, i.e. line peak intensity, line shift and line width. L3 FITS files are created from multi-exposure narrow-slit raster and sit-and-stare observations (i.e. windowed spectral-profile HDUs, see Table 4-1 and Table 4-2). Due to the contents of the files, we call the Level 3 FITS files “L3 P”, the letter P meaning “Parameterized”. See Appendix IX of [S-META] for a thorough description of such parameterized components files.

Note that line fitting is sensitive to the initial values of the Gaussian parameters and in some cases the derived data products produced by the pipeline may therefore be less trustworthy than if the line fitting had been performed with human supervision. Also, the algorithm that automatically detects emission lines may fail, and it may report false positives or miss some weaker emission lines. For scientific purposes, we therefore strongly encourage users to perform the line fitting manually, starting with L2 files.

The processing steps performed by the Science Data Pipeline to create an L3 P file:

1. An L2 file is given as input to the L3 FITS file generator, which calls the `::create_l3_file` method of the `spice_data` IDL object (see [IDLANA]).
2. Select HDUs with data stemming from multi-exposure narrow-slit spectral-profile windows and attempts to automatically detect and identify up to 25 emission lines in each readout window.
3. Give an initial guess of the amplitude, position, and width of each line as input to the line fitting routine `cfit.pro` (see [CFIT]), which returns the fitted gaussian line parameters peak intensity, line shift and line width.
4. Each observational HDU in the input L2 file results in two HDUs in the L3 P files. The first HDU contains the Gaussian fit parameters, the second contains the header of the parent L2 HDU. See Section 4.4.3. for details.
5. *To be implemented: based on the fitted gaussian line parameters, secondary derived parameters will be estimated. The format of future Level 3 FITS files containing secondary derived parameters is yet to be finalised.*

3.3.5 Level 3 QL – Higher Level Data Quicklook (JPEG/MP4, Gaussian fit parameters)

Level 3 Quicklook files are images and movies of the Gaussian line parameters found in L3 P FITS files. L3 Quicklook files are not suitable for scientific analysis, as they are not in an appropriate scientific file format, they are highly compressed, and they do not contain the necessary metadata required for a thorough analysis. They may, however, provide useful context information, and they can be used as a tool to find interesting observations that should be investigated further.

3.3.5.1 Level 3 Quicklook images (L3 QL JPEG)

The processing steps performed by the Science Data Pipeline to create L3 QL JPEG image files:

1. An L3 P file is given as input to the L3 quicklook image generator.
2. Extract a data array for each identified emission line, for each of the three Gaussian fit parameters.
3. For the line-shift Gaussian parameter (velocity): remove any vertical and horizontal velocity trends in the image. This also ensures that the median velocity of the image is 0 km/s.
4. Display each image with axes and a selection of metadata using the IDL Graphics functions. The images are adjusted for any space craft roll, using the IDL Graphics `rotate` method.
5. Create a JPEG file for each image, using the IDL Graphics `save` method.

Note that the removal of velocity trends in the images (bullet point 3) obscures any possibly real and physical velocity trends, as well as artificial trends created by inaccurate geometric distortion correction, overall line shifts caused by inaccurate wavelength calibration, some effects caused by the shape of the PSF, and possibly other instrumental effects. The reason for hiding all these effects in the QL velocity images is to make structures with higher velocity than the surroundings more visible. This makes the QL images more fitting for their intended usage, i.e. enabling the user to easily spot observations with interesting features.

3.3.5.2 Level 3 Quicklook movies (L3 QL MP4)

The processing steps performed by the Science Data Pipeline to create L3 QL MP4 movie files:

1. When all L3 P files for a given day has been processed to L3 QL JPEG as described above, pass the resulting list of unique `SPIOBSIDs` to the L3 Quicklook movie generator.
2. For each `SPIOBSID`, collect all JPEG images stemming from this observation.
3. Ignore `SPIOBSIDs/`images obtained from observations that are not repeated.
4. Loop through the JPEG file list: for each `SPIOBSID`, for each identified emission line, for each of the three Gaussian fit parameters, create an MP4 movie file using the `IDLffVideoWrite` object. The framerate is 5 fps.

3.3.6 CAL – Calibration data

The calibration routines and the necessary calibration data will be made available in SolarSoft (SSW) and in the SolarSoft Data Base (SSWDB).

3.3.7 HK – Housekeeping data

The Oslo pipeline does not process housekeeping telemetry, only science telemetry (both Low Latency Data and Science Data).

3.4 Validation

The FITS files produced by the Oslo pipeline are continuously validated by the Oslo team. Multiple validation procedures are run automatically every time new FITS files are created. The results of these tests are automatically sent to the Oslo team by e-mail. The team is also notified by e-mail and/or SMS if the pipeline encounters errors or crashes.

In earlier stages of the pipeline development the data products were compared to the output from RAL's *cube builder*.

3.4.1 Instrument Team Validation

The FITS files produced by the Oslo pipeline are automatically made available to the SPICE consortium for validation and analysis. Consortium members at IAS, MPS, RAL, SwRI, and GSFC are making invaluable contributions to the pipeline development and data product validation by reporting bugs, providing support, and participating in discussions.

3.4.2 SOC Validation

SOC validates the SPICE FITS files upon ingestion into the Solar Orbiter Archive.

4. DATA PRODUCT DESCRIPTIONS

SPICE data products stored in FITS files are formatted in accordance with the rules outlined in [METADATA]. This section provides details on the formats, metadata, and filenames for each of the products included in the SPICE data.

A SPICE FITS file contains a primary Header/Data Unit (HDU), and it may contain one or more image extensions. All primary and image extension HDUs containing observational data have a full header as described in [METADATA] – i.e. there is no distinction between primary HDUs and image extensions other than those required by [FITSpaper]. Each HDU is regarded as a data product.

In addition, Level 1 and Level 2 files may contain binary table extensions storing auxiliary data that have individual values for each exposure of the observation. Level 1 files may also contain binary table extensions describing lost telemetry, see Sections 4.4.4 and 4.4.5.

Level 1 and Level 2 files storing multi exposure observations always have two additional image extensions describing coordinate distortions due to spacecraft pointing instabilities, see Section 4.4.1.3.6.

Finally, L2 files have binary table extensions storing information about spike pixels, and they may have binary table extensions storing information about saturated pixels.

The SPICE detector is read in wavelength regions, or windows, of a specified wavelength range. The data array of an observational HDU stem from one of the following ***6 window types***, described in the `WIN_TYPE` FITS keyword:

	Window type description	WIN_TYPE
1	Narrow-slit spectral-profile (2”, 4”, and 6” slits)	`'Narrow-slit Spectral'`
2	Dumbbell stack	`'Dumbbell (lower)'` or `'Dumbbell (upper)'`
3	Wide-slit (30” slit)	`'Wide-slit'`
4	Intensity-window (2”, 4”, and 6” slits)	`'Intensity-window'`
5	Full detector read-out (2”, 4”, and 6” slits)	`'Full Detector Narrow-slit'`
6	Full detector read-out (30” slit)	`'Full Detector Wide-slit'`

Table 4-1: All available window types and corresponding values of `WIN_TYPE`. Note that most SPICE observations are of window type 1 and 5. Window type 2 and 4 have hardly ever been used, see Table 4-2.

Note that the spectra on the two detectors have a significant relative shift in the detector Y direction. Due to hardware limitations all type 1 windows within a study must have the same detector Y coordinates. Therefore, the readout windows must be significantly taller than the height of the slit to ensure that all windows on both detectors cover the full slit height, see Figure 4-1. This has two major consequences:

1) The upper and/or lower part of most windows of type 1 will contain dumbbell data. These pixels must be masked before any line fitting can be done to the spectral data of the window. The WCS keywords do not apply to the dumbbell pixels.
2) A multi-exposure Narrow-slit observation will normally only contain such extended windows of type 1, and no separate dumbbell windows.

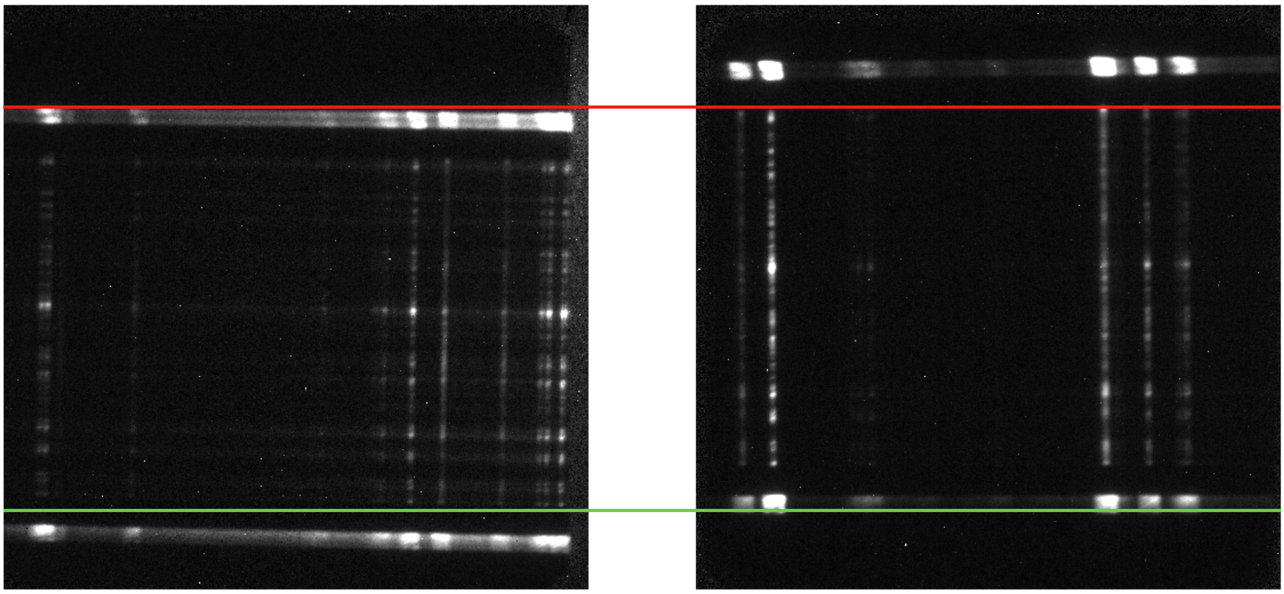

Figure 4-1: The spectra projected on the Short Wavelength detector (SW, left) and the Long Wavelength detector (LW, right) are shifted in the detector Y direction. The lower edge of the slit on the SW detector (green) is at a lower Y pixel value than the dumbbell region on the LW detector. The upper edge of the slit on the LW detector (red) is above the SW dumbbell region. To catch the full slit height on both detectors the readout windows must be covering the Y-range between the green and the red line, i.e., a spectral window will also contain dumbbell information. The figure shows SPIOBSID 150995892 corrected for flat field and dark current.

The window types may be of one the following ***3 study types***, described in the `STUDYTYP` FITS keyword:

	Study type	Possible window types	Actual usage, all L1 files	Actual usage, Science only	STUDYTYP	X	t
1	Sit-and-stare	1 2 3	1.6% – 0.1%	9.2% - 0.2%	'Sit-and-stare'	1	>1
2	Raster	1 2 3 4	13.9% <0.1% 2.1% <0.1%	**80%** - 10.6% -	'Raster '	>1	1
3	Single exposure	1 3 5 6	2.3% <0.1% **41%** 4.3%	- - - -	'Single Exposure'	1	1

Table 4-2: All available combinations of study types and window types, and their actual usage in percentage of the number of L1 files obtained after commissioning up to April 2026. Note that for science studies most observations are narrow-slit spectral rasters. However, even if a file is not labelled as *Science,* it may contain scientifically interesting data. As an example, even if none of the single exposure full detector observations are labelled as *Science*, only 2% of these files are darks. A series of repeated single-exposure full-detector observations are therefore concatenated into single L2 -ras or -sit files.

The very seldomly used window type 4 (Intensity-window) may not be of study type 1 (Sit-and-stare). Window types 5 and 6 (full-detector read-out) may only be of study type 3 (single exposure). However, it is possible to repeat full detector observations, with or without moving the scan mirror. For L1 the individual exposures are stored as separate FITS files. For L2 a series of repeated full detector observations are gathered in a single L2 FITS file.

Note that a full-frame read-out results in a FITS file with two HDUs, one for each detector array[6].

Considering all possible combinations of window types and study types, SPICE FITS files may contain `9` distinct types of science data products or HDUs. The data product description is stored in the keyword `DATAPROD`, which is simply the concatenation of `WIN_TYPE` and `STUDYTYP`, e.g. '`Narrow-slit Spectral Raster`'. See also Table 4-11.

Note that in other SPICE documents and applications, e.g. [MAN] and in the SPICE Study Generator tool, the nomenclature is a bit different from what has been outlined above. As an example, instead of differentiating between study types and window types, the Study Generator only uses the concept of study type, which may have one of the following values: "*Full Spectrum*", "*Spatial Scan*", "*Time Series*" and "*Scanned Time Series*". In the pipeline a "*Scanned Time Series*" is treated identical to a series of repeated "*Spatial Scans*": in both cases the series of spatial scans are stored in multiple L1, L2 and L3 P FITS files – each file contains a single spatial scan per readout window

4.1 Filename, L1, L2, and L3 FITS files

Following the specifications in [METADATA], the SPICE FITS file names have the following format:

```
solo_[level]_spice-[slit]-[type][-db][-int]_[time]_V[version]_[SPIOBSID]-[RASTERNO]-[DR].fits
```

- `[level]` is `L1`, `L2`, or `L3`.
- `[slit]` is either `w` or `n`, for "wide" (30") or "narrow" (2"/4"/6") respectively. We currently make L3 files only from narrow slit observations, i.e. for L3 this descriptor is always `n`.
- `[type]` is either -`ras` (for "raster"), -`sit` (for "sit-and-stare") or -`exp` (for "single exposure", applies to L1 and L2 only).
- `[-db]` is `-db` for files that include separate dumbbell extensions, otherwise empty. Note that due to technical reasons HDUs storing regular narrow-slit spectral-profile windows may contain dumbbell data, see page 30 of Section 4. The `[-db]` descriptor of such files is empty.
- `[-int]` is `-int` for files that include intensity-windows, otherwise empty.
- `[time]` is the UTC time at the beginning of the study.
- `[version]` is an incremental file number padded with '`0`' to two characters.
- The combination of `SPIOBSID` (SPICE Observation ID) and `RASTERNO` uniquely identifies a single observation. If a study is repeated all files in the series will have the same `SPIOBSID` but different `RASTERNO`.
- [`DR`] signals which Data Release the file belongs to, e.g. '`DR6`' (see [IASDRURL]).
- The combination of `level`, `version`, `SPIOBSID`, `RASTERNO` and `DR` uniquely identifies a single FITS file.

[6] SPICE can transmit full-frame data in both compressed and uncompressed format. In this case the L0+ files contain *four* HDUs, *two* for each detector array, one with decompressed data and one with the uncompressed data.

– The first time a file of a given `level`, `SPIOBSID`, and `RASTERNO` is created, `[version]` is set to ‘`V01`’. The version number is always incremented by 1 every time a file is reprocessed, regardless of the value of `[DR]`.

4.2 Filename, L3 QL

An L3 QL JPEG file is an image extracted from an L3 P FITS file. An L3 QL MP4 file is a movie made from multiple L3 QL JPEG files. The L3 QL file names have the following format:

```
solo_L3-spice-ql-[line]-[product]-[slit]-[type]_[time]_V[version]_[SPIOBSID]-
[first RASTERNO][-final RASTERNO]-[DR]-[WINNO]-[LINENO]-[product].[ext]
```

- `[line]` contains a string identifying the emission line.
- `[product]` contains the data product name: `int`, `vel`, or `wid`.
- `[slit]` is `n` for “narrow” (2"/4"/6"). In the future we may also create L3 QL files from wide slit observations, in that case this descriptor will have the value `w`..
- `[type]` is either `ras` (for “raster”), `sit` (for "sit-and-stare").
- `[time]` is the UTC time at the beginning of the study.
- `[version]` is an incremental file version number padded with ‘0’ to two characters. The first time a L3 file is created, `[version]` is ‘`V01`’.
- `[first RASTERNO]` equals the value of `RASTERNO` for images. For movies it is the value of `RASTERNO` of the first file in the movie.
- `[-final RASTERNO]` is empty for images, for movies it is the value of `RASTERNO` of the final file in the movie
- `[DR]` signals which Data Release the file belongs to, e.g. ‘`DR6`’ (see [IASDRURL]).
- `[WINNO]` is the value of `WINNO` of the parent L3 HDU
- `[LINENO]` is the line number identified in this window, starting at 0 for the strongest line in the window. Note that which line is the strongest in each window may depend on the observation target.
- `[ext]` is either `jpg` or `mp4`.

4.3 FITS WCS coordinates and the `PCi_j` transformation matrix

In general, the WCS coordinate c_i of a pixel with pixel indices $p_j=(p_1,p_2,p_3,\ldots,p_N)$ is expressed by:

$$c_i(p_1, p_2, p_3, \ldots, p_N) = \mathrm{CRVAL}i + \mathrm{CDELT}i \sum_{j=1}^{N} \mathrm{PCi\_j}(p_j - \mathrm{CRPIX}j) \quad (1)$$

Thus, for a four-dimensional data cube, the WCS coordinate c_i of a pixel with indices $p_j=(p_1,p_2,p_3,p_4)$ is expressed by:

$$\begin{aligned} c_i(p_1,p_2,p_3,p_4) = \mathrm{CRVAL}i + \mathrm{CDELT}i*(&\mathrm{PC}i\_1*(p_1-\mathrm{CRPIX1}) + \\ &\mathrm{PC}i\_2*(p_2-\mathrm{CRPIX2}) + \\ &\mathrm{PC}i\_3*(p_3-\mathrm{CRPIX3}) + \\ &\mathrm{PC}i\_4*(p_4-\mathrm{CRPIX4})) \end{aligned}$$

For HDUs where each coordinate is coupled only to "its own" dimension (i.e. coordinate $i=1$ is coupled to data cube dimension $j=1$ only, coordinate $i=2$ is coupled to dimension $j=2$ only, etc), the `PCi_j` matrix has only diagonal entries and all off-diagonal entries have the default value of `0`. This represents cases with no shear, rotation, mirroring, or transposition.

In L1+ (L1, L2 and L3) FITS files the rotation of the field-of-view is described by four `PCi_j` matrix elements: for a non-zero roll angle `PC1_1` and `PC2_2` do not have the default value of `1`, and `PC1_2` and `PC2_1` do not have the default value of `0`. The spacecraft counter-clockwise roll angle relative to Solar north is given in the `CROTA` keyword. In L2 FITS files all geometrical distortions other than the rotation of the FOV are corrected for by linear interpolation of the data, and these distortions are therefore not described by FITS keywords. However, the L1 to L2 calibration routine (`spice_prep.pro`) will be made publicly available and the users will be able to turn off the data interpolation if desirable. In such cases the geometrical distortions will be described by FITS keywords, see [DISTORTIONS] and [DISPERSION].

For both L1+ FITS files, the `j`th dimension of the data cube may contribute to the `i`th coordinate for raster scans and wide-slit (or separately downlinked dumbbell) observations. In these cases, the off-diagonal entries of the `PCi_j` matrix are not `0`:

- For raster observations, time increases for each new slit position. Since the scan direction of the scan mirror is from Solar West to Solar East this means that the time coordinate `t` *decreases* in the `X` dimension (`j=1`) of the data cube. Therefore, the `PC4_1` element of the transformation matrix is less than zero, coupling the `t` coordinate to the `X` dimension of the data cube.
- For all wide-slit (and dumbbell) HDUs, the `x` coordinate increases in the dispersion dimension (`j=3`), and the headers have a non-zero `PC1_3` matrix element. The value of `PC1_3` is given by the spatial pixel size in the dispersion dimension divided by `CDELT1`.

SPICE FITS files do not contain multiple repetitions of raster scans; therefore, the time dimension of raster scan data cubes is degenerate and there is no natural value for `CDELT4`. On the other hand, `CDELT4` cannot be zero, since the product `CDELT4*PC4_1` must be equal to the time between two consecutive exposures within a raster scan. We therefore let `CDELT4` have the default value of `1`, and `PC4_1` is thereby equal to the time between exposures.

For sit-and-stare observations the `t` coordinate is only dependent on the fourth dimension (`j=4`), and `PC4_1` in the equation above is `0`. The time between two consecutive exposures is `CDELT4` for sit-and-stare observations.

Note that the time coordinate `t` is the centre time of an exposure relative to the start time of the observation as a whole, i.e. the relative start time of the exposure + `XPOSURE/2`. Thus, the `t` coordinate of the first exposure of an observation is `XPOSURE/2`, the `t` coordinate of the exposure that corresponds to the reference pixel is `CRVAL4`, and `t` of the last exposure is the end time of the observation –`XPOSURE/2`. This is true for both rasters and sit-and-stare observations, even if the reference pixel for rasters corresponds to exposure number `NAXIS1/2`, while the reference pixel for sit-and-stare observations corresponds to exposure `1`.

4.4 Data arrays and FITS headers

4.4.1 Level 1 Data arrays and FITS headers

Although we encourage users to use L2 files for their scientific analysis, we start by describing L1 files. These files contain additional metadata and binary table extensions that are not present in L2 files.

4.4.1.1 General format of L1 data arrays

The dimensions of the L1 data arrays are `(X,Y,D,t)` for all 6 window types in Table 4-1.

The `X` dimension of a data array always denotes the number of slit positions during the observation, and `Y` is the height along the slit, in pixels[7].

The third dimension, `D`, always represents a position along the dispersion direction of the detector. For type 1, this dimension unambiguously corresponds to wavelength (lambda). For wide-slit (and dumbbell) HDUs, however, this dimension corresponds to both spatial `X` and wavelength. Thus, for any given `X` and `t`, the `(Y,D)` plane shows a transposed spatial image. However, the `X` position (relative to the centre of the solar disc) of every pixel in such an image is still given by the first WCS coordinate (`CTYPE1='HPLN-TAN'`). This means that there is an off-diagonal value in the `PCi_j` matrix to couple the `X` and `D` dimensions of the data cube, see Section 4.3

Dimension four, `t`, always represents the number of exposures per slit position. I.e. for sit-and-stare observations, `t` is the number of exposures and `X` is 1. For raster observations, where the slit is moved between each exposure, `t` is 1. For such observations, there is an off-diagonal element in the `PCi_j` matrix to couple time and the `X` position of the slit, see Section 4.3.

Intensity-window observations, window type 4, have only been tested briefly and are not included in regular SPICE operations. These observations are binned in the dispersion dimension with a binning factor equal to the width of the window. Intensity-windows can only be defined for raster studies. The size of the resulting data cube is `(X,Y,1,1)`, i.e. it has no spectral or temporal resolution. The wavelength coordinate is such that the value reflects the central wavelength (i.e. the midpoint between the central two pixels) of the readout window. Intensity-windows are normally observed in pairs: one window covers an emission line (line-window), and a nearby window covers a region with no strong emission lines (background-window). The data from the background-window may be used as an estimate of the background level of the window covering the emission line. Such a background intensity-window may be used as a background estimate for multiple line-windows. The link between an HDU storing the emission line data and the corresponding HDU storing the background data is established with FITS keywords, see Section 4.4.1.7.

For all study types, the maximum number of exposures is 480, i.e. `X` and `t` are always less than 480.

4.4.1.2 L1 FITS Header Example

Below is an example header of an L1 observational HDU including all keywords required by [METADATA].

A selection of the FITS keywords is further described in Section 4.4.1.3. The correspondence between the metadata parameter names used in [DATAICD] and the FITS keywords of the L1 HDUs is explained in Appendix B. The correspondence between the Study Generator fields and FITS keywords is outlined in Appendix C.

```
XTENSION= 'IMAGE   '           / Written by IDL:  Tue Mar 24 10:09:51 2026
BITPIX  =                   16 / Integer*2 (short integer)
```

[7] When referring to pixels along detector Y (height along the slit) or pixels along the dispersion direction (lambda/X'), these might represent *binned* pixels, depending on the study.

```
NAXIS   =                    4 / Number of dimensions
NAXIS1  =                  224 / Number of slit positions (x)
NAXIS2  =                  768 / Number of pixels along slit (y)
NAXIS3  =                   16 / Number of pixels in dispersion dimension
NAXIS4  =                    1 / Number of exposures per slit position (time)
PCOUNT  =                    0 /  number of random group parameters
GCOUNT  =                    1 /  number of random groups
DATE    = '2026-03-24T10:09:51' / Date and time of FITS file creation

EXTNAME = 'Ne VIII 770 / Mg VIII 772 - SH' / Extension name
FILENAME= 'solo_L1_spice-n-ras_20260225T014443_V03_369099136-001-DR6.fits' / Fil


           --------------------------------------------------------
           | Study parameters valid for all Obs-HDUs in this file |
           --------------------------------------------------------
STUDYTYP= 'Raster  '           / Sit-and-stare, Raster or Single Exposure
STUDYDES= 'Composition for High Latitude Polar Survey' / Study description
STUDY   = 'SCI_POLAR-SURVEY-COMP_SC_SL04_59.7S_FF' / SPICE Study name
OBS_MODE= 'SCI_POLAR-SURVEY-COMP_SC_SL04_59.7S_FF' / = STUDY
OBS_TYPE= 'sFrx    '           / Unique code for OBS_MODE
AUTHOR  = 'Alessandra Giunta'  / Author of study
OBS_ID  = 'SSPI_220C_RS9_112_sFrx_11F' / SOC Observation ID
SPIOBSID=            369099136 / SPICE Observation ID (hex 16000180)
OBS_DESC= 'High latitude polar survey. Second mosaic position - composition 2 &'
CONTINUE  'repeats.' / Observation description
PURPOSE = 'Science '           / Purpose of study (Science/Calibration/Checkout)
READMODE= 'Destructive'        / Destructive or non-destructive
TRIGGERD= 'none    '           / Event that triggered observation
OBJECT  = 'Sun     '           / Type of object observed
TARGET  = 'limb, South'        / Coarse human interpretable pointing info
SOOPNAME= 'R_SMALL_HRES_MCAD_Polar-Observations' / SOOP Campaign name(s)
SOOPTYPE= 'RS9     '           / SOOP Campaign name code(s)
LTP     =                   22 / SoLO Long-Term Plan number
STP     =                  399 / SoLO Short-Term Plan number
SETFILE = 'flt26-study-set.xml' / Study Set (study definitions) filename
SETVER  =                   35 / Study Set version
APID    =                 1404 / Application Process ID
NRASTERS=              2.00000 / Number of planned rasters for this SPIOBSID
RASTERNO=                    1 / Raster number (starting at 0)
STUDY_ID=                   18 / On-board Study ID slot (0-63)
MISOSTUD=                 2371 / Ground study ID used in MISO planning tool
XSTART  =                  416 / [arcsec] Slit instr. x rel. to instr. boresight
XPOSURE =              59.7000 / [s] Total effective exposure time
FOCUSPOS=                12237 / Focus position
NSEGMENT=                    7 / Number of segments per window

POINT_ID= '00006609'           / SVO pointing ID
MOSAICID= '        '           / Blank when study is not a mosaic
SVO_SEP1= 'INSTRUME,PURPOSE,SOOPNAME,STUDY,SPIOBSID' / SVO very fine split level
SVO_SEP2= 'INSTRUME,PURPOSE,SOOPNAME,STUDY' / SVO fine split level
SVO_SEP3= 'INSTRUME,PURPOSE,SOOPNAME' / SVO medium split level
SVO_SEP4= 'INSTRUME'           / SVO coarse split level
SVO_GRP = '369099136'          / SVO file group ID = string(SPIOBSID)

NWIN    =                   13 / Total number of windows in this file
```

```
NWIN_PRF=                   13 / Number of windows not Dumbbell or Intensity
NWIN_DUM=                    0 / Number of Dumbbell windows
NWIN_INT=                    0 / Number of Intensity-windows

PXCOV3  = '42-73, 74-105, 205-236, 742-773, 774-805, 852-883, 918-949, 950-965&'
CONTINUE  ', 1133-1164, 1715-1746, 1791-1822, 1841-1872, 1873-1904&' / [pixel]
CONTINUE  '' / All PXBEG3-PXEND3


           ------------------------------------------------------
           | Other keywords valid for all Obs-HDUs in this file |
           ------------------------------------------------------
TIMESYS = 'UTC     '           / TIMESYS included for readability
DATEREF = '2026-02-25T01:44:43.781' / [UTC] Zero point of time coordinate
DATE-BEG= '2026-02-25T01:44:43.781' / [UTC] Beginning of data acquisition
DATE-OBS= '2026-02-25T01:44:43.781' / [UTC] Equals DATE-BEG
DATE-AVG= '2026-02-25T03:36:43.792' / [UTC] Data acquisition midpoint
DATE-END= '2026-02-25T05:28:43.506' / [UTC] End of data acquisition
SEQ_BEG = '2026-02-24T22:00:32' / [UTC] Approximated start of observation series
TELAPSE =        13439.7249999 / [s] Elapsed time between beg. and end of acqu.
OBT_BEG =        825298747.851 / Start acquisition time in OBT
LEVEL   = 'L1      '           / Data processing level
CREATOR = 'SDP-SPICE'          / Name of pipeline
ORIGIN  = 'University of Oslo' / Name of institution
VERS_SW = '6093    '           / UiO SVN revision number of L1 pipeline
VERSION = '03      '           / Incremental file version number
SOAR_DR =                    6 / SOAR Data Release number
OBSRVTRY= 'Solar Orbiter'      / Observatory Name
INSTRUME= 'SPICE   '           / Instrument name
CROTA   =       -2.77496467802 / [deg] S/C counter-clockwise roll rel to Solar N
ROT_COMP=                    1 / Feature tracking probably off (0) or on (1)

COMPLETE= 'C       '           / Complete data set, all windows combined
PCT_CMPL=              100.000 / Completeness of data set, all windows combined


           -------------------------------------
           | STUDYFLG and its derived keywords |
           -------------------------------------
STUDYFLG=                    0 / Study flags
NOSPECTR=                    0 / Applies only to dumbbells
CALMODE =                    0 / Applies only to full frame readouts
DBLEXP  =                    0 / If set, double exposure is enabled
DBLEXPNO=                    0 / Applies only when DBLEXP=1
DARKMAP =                    0 / If set, a dark map was subtracted on-board
BLACKLEV=                    0 / If set, a bias frame was subtracted on-board


           ----------------------------------------------
           | Keywords valid for this HDU (WINDOW3_76.92) |
           ----------------------------------------------
WIN_TYPE= 'Narrow-slit Spectral' / Description of window type
DATAPROD= 'Narrow-slit Spectral Raster' / WIN_TYPE+STUDYTYP
TELESCOP= 'SOLO/SPICE/SW'      / Telescope/Sensor name/Detector array name
DETECTOR= 'SW      '           / Detector array name
WINNO   =                    3 / Window number (starting at 0) within this study
```

```
WINTABID=                  136 / Index in on-board window data table (0-255)
MISOWIN =                 3222 / Ground window ID used in MISO planning tool
WINSHIFT=                   -9 / [pixel] Win redshift rel to win 3222 base pos.
SLIT_ID =                    2 / Slit ID (0-3)
SLIT_WID=                    4 / [arcsec] Slit width
DUMBBELL=                    0 / 0/1/2: not a dumbbell/lower dumbbel/upper dumbb

WAVEUNIT=                   -9 / Power of 10 by which the metre is multiplied
WAVEREF = 'vacuum  '           / Wavelengths are given in vacuum
WAVEMIN =        76.8332217698 / [nm] Left edge of first read detector pixel
WAVEMAX =        77.1409745510 / [nm] Right edge of last read detector pixel
WINWIDTH=             0.611633 / [nm] Window width

BTYPE   = 'Intensity'          / Type of data
UCD     = 'phot.count;em.line' / Unified Content Descriptors v1.23
BUNIT   = 'adu     '           / Units of uncalibrated data

BSCALE  =              1.00000 / Data value = BZERO + BSCALE*FITS array value
BZERO   =                    0 / Default value for unsigned integers
BLANK   =                65535 / Value of undefined and lost pixels
NTOTPIX =              2752512 / Number of potentially usable pixels =all for L1
NDATAPIX=              2752512 / Number of usable pixels excl. saturated & BLANK
NSATPIX =                    0 / Number of fully saturated pixels
NLOSTPIX=                    0 / Number of lost pixels
NAPRXPIX=                    0 / Number of approximated pix. b/c telemetry loss

PCT_DATA=              100.000 / NDATAPIX/NTOTPIX*100
PCT_SATP=              0.00000 / NSATPIX/ NTOTPIX*100
PCT_LOST=              0.00000 / NLOSTPIX/NTOTPIX*100
PCT_APRX=              0.00000 / NAPRXPIX/NTOTPIX*100

DATAMIN =              1223.00 / [adu] Minimum data value
DATAMAX =              11454.0 / [adu] Maximum data value
DATAMEAN=              1190.84 / [adu] Mean    data value
DATAMEDN=              1072.00 / [adu] Median  data value
DATAP01 =              897.000 / [adu] 1st  percentile of data values
DATAP10 =              938.000 / [adu] 10th percentile of data values
DATAP25 =              981.000 / [adu] 25th percentile of data values
DATAP75 =              1270.00 / [adu] 75th percentile of data values
DATAP90 =              1560.00 / [adu] 90th percentile of data values
DATAP95 =              1814.00 / [adu] 95th percentile of data values
DATAP98 =              2197.00 / [adu] 98th percentile of data values
DATAP99 =              2521.00 / [adu] 99th percentile of data values
DATARMS =        392.052354221 / [adu] RMS dev: sqrt(sum((data-DATAMEAN)^2)/N)
DATANRMS=       0.329222650855 / Normalised RMS dev: DATARMS/DATAMEAN
DATAMAD =        230.660892206 / [adu] Mean abs dev: sum(abs(data-DATAMEAN))/N
DATASKEW=        6.92552364823 / Data skewness
DATAKURT=        101.555169678 / Data kurtosis

PXBEG1  =                  224 / [pixel] First read-out pixel in X dimension
PXEND1  =                    1 / [pixel] Last  read-out pixel in X dimension
PXBEG2  =                  101 / [pixel] First read-out pixel in Y dimension
PXEND2  =                  868 / [pixel] Last  read-out pixel in Y dimension
PXBEG3  =                  742 / [pixel] First read-out pixel in dispersion dim.
PXEND3  =                  773 / [pixel] Last  read-out pixel in dispersion dim.
PXBEG4  =                    1 / [pixel] First read-out pixel in time dimension
```

```
PXEND4  =                    1 / [pixel] Last  read-out pixel in time dimension

NBIN1   =                    1 / Binning factor in X dimension
NBIN2   =                    1 / Binning factor in Y dimension
NBIN3   =                    2 / Binning factor in dispersion dimension
NBIN4   =                    1 / Binning factor in time dimension
NBIN    =                    2 / Total binning factor

COMPRESS= 'Spatial Lossy'      / JPEG Compression description
COMP_RAT=              4.00000 / Compression ratio decompressed/compressed
COMPQUAL=             0.250000 / Compression ratio compressed/decompressed
COMPTYPE=                    6 / Compression type (0-7)
COMPPARA=                   64 / Compression amount parameter (0-255)
SHCFFTID=                    0 / Applies only to SHC-compressed data
COMP_ALG= 'Lossy/Spatial Lossy/4/6/64' / Lossy/COMP(RESS/_RAT/TYPE/PARA)


           ---------------------------------------------------------
           | Keywords describing telemetry of non-missing segments |
           ---------------------------------------------------------
NPACKETS=                  336 / Number of packets with observational data

LOSTPKTS=                    0 / Number of lost packets w/data, variable keyword
LOSTBINS=                    0 / Applies only to SHC-compressed data

NLOSTCHK=                    0 / Number of lost checksum packets
NFAILCHK=                    0 / Number of checksums failed
NAPRXPLN=                    0 / Number of approximated X-Y plane sections
NLOSTPLN=                    0 / Number of lost X-Y plane sections



           ------------------------------------------
           | World Coordinate System (WCS) keywords |
           ------------------------------------------
WCSNAME = 'Helioprojective-cartesian' / Name of coordinate system
LONPOLE =              180.000 / [deg] Native longitude of celestial pole
SPECSYS = 'TOPOCENT'           / Spectral coord. not corrected for S/C velocity
VELOSYS =              0.00000 / [m/s] Default for SPECSYS='TOPOCENT'

CTYPE1  = 'HPLN-TAN'           / Type of 1st coordinate
CNAME1  = 'Helioprojective longitude (Solar X), increases towards Solar West' /
CUNIT1  = 'arcsec  '           / Units for 1st coordinate (for CRVAL1, CDELT1)
CRVAL1  =       -401.140509661 / [arcsec] 1st coordinate of reference point
CDELT1  =              4.00000 / [arcsec] Increment of 1st coord at ref point
CRPIX1  =              112.500 / [pixel] 1st pixel index of reference point
PC1_1   =       0.998827386841 / Non-default value due to CROTA degrees S/C roll
PC1_2   =      0.0132929712494 / Contribution of dim 2 to coord 1 due to roll
CRDER1  =       0.167201113070 / [arcsec] Mean stddev of Solar X
CWERR1  =       0.362123136213 / [arcsec] Max absolute distortion, Solar X
CWDIS1  = 'Lookup  '           / Type of WCS distortion correction
DW1     = 'EXTVER: 1'          / Extension version
DW1     = 'NAXES: 1'           / Axes in the distorion array
DW1     = 'AXIS.1: 1'          / Correction of Solar X
DW1     = 'ASSOCIATE: 1'       / Association stage is pixel coordinates
DW1     = 'APPLY: 6'           / Application stage is world coordinates
```

```
CTYPE2  = 'HPLT-TAN'           / Type of 2nd coordinate
CNAME2  = 'Helioprojective latitude (Solar Y), increases towards Solar North' /
CUNIT2  = 'arcsec  '           / Units for 2nd coordinate (for CRVAL2, CDELT2)
CRVAL2  =      -2835.80738823 / [arcsec] 2nd coordinate of reference point
CDELT2  =             1.09829 / [arcsec] Increment of 2nd coord at ref point
CRPIX2  =             384.500 / [pixel] 2nd pixel index of reference point
PC2_1   =     -0.176322603339 / Contribution of dim 1 to coord 2 due to roll
PC2_2   =      0.998827386841 / Non-default value due to CROTA degrees S/C roll
CRDER2  =      0.178798752753 / [arcsec] Mean stddev of Solar Y
CWERR2  =       7.68412855102 / [arcsec] Max absolute distortion, Solar Y
CWDIS2  = 'Lookup  '           / Type of WCS distortion correction
DW2     = 'EXTVER: 2'          / Extension version
DW2     = 'NAXES: 1'           / Axes in the distorion array
DW2     = 'AXIS.1: 1'          / Correction of Solar Y
DW2     = 'ASSOCIATE: 1'       / Association stage is pixel coordinates
DW2     = 'APPLY: 6'           / Application stage is world coordinates

CTYPE3  = 'WAVE    '           / Type of 3rd coordinate
CNAME3  = 'Wavelength'         / Description of 3rd coordinate
CUNIT3  = 'nm      '           / Units for 3rd coordinate (for CRVAL3, CDELT3)
CRVAL3  =       76.9101362811 / [nm] 3rd coordinate of reference point
CDELT3  =    0.00961542169011 / [nm] Increment of 3rd coord at ref point
CRPIX3  =             8.50000 / [pixel] 3rd pixel index of reference point
PC3_3   =             1.00000 / Default value, no rotation

CTYPE4  = 'UTC     '           / Type of 4th coordinate
CNAME4  = 'Time (Degenerate Dimension)' / Description of 4th coordinate
CUNIT4  = 's       '           / Units for 4th coordinate (for CRVAL4, CDELT4)
CRVAL4  =       6719.85000000 / [s] 4th coord of ref point, relative to DATEREF
CDELT4  =       1.00000000000 / [s] Degenerate dimension; default value
CRPIX4  =             1.00000 / [pixel] 4th pixel index of reference point
PC4_4   =             1.00000 / Default value, no rotation
PC4_1   =      -60.0000000000 / Contribution of dimension 1 to coordinate 4


           ----------------------------------------------------------------
           | Auxiliary data and reference to bintab with variable keywords |
           ----------------------------------------------------------------
VAR_KEYS= 'VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_GRAT,TN_SW,TN_LW,T_F&'
CONTINUE  'OCUS,T_GRAT,T_SW,T_LW,VN_MCPSW,VN_MCPLW,VN_GAPSW,VN_GAPLW,V_MCPSW,V&'
CONTINUE  '_MCPLW,V_GAPSW,V_GAPLW,TIMAQUTC,CRDER1,CRDER2                    &'
CONTINUE  '' / Variable keywords
TIMAQOBT=       825305437.851 / [OBT] Average Start time of data acquisition
MIRRPOS =             40998.7 / [adu] Average Scan mirror position
TN_FOCUS=             1468.69 / [adu]     Average SFM focus   temperature
TN_GRAT =             1379.95 / [adu]     Average SFM grating temperature
TN_SW   =             2778.70 / [adu]     Average HAS SW      temperature
TN_LW   =             2778.98 / [adu]     Average HAS LW      temperature
T_FOCUS =             28.7482 / [Celsius] Average SFM focus   temperature
T_GRAT  =             31.9560 / [Celsius] Average SFM grating temperature
T_SW    =            -19.9990 / [Celsius] Average HAS SW      temperature
T_LW    =            -20.0064 / [Celsius] Average HAS LW      temperature
VN_MCPSW=                2707 / [adu] Average MCP SW voltage
VN_MCPLW=                2581 / [adu] Average MCP LW voltage
VN_GAPSW=                2927 / [adu] Average GAP SW voltage
```

```
VN_GAPLW=                2935 / [adu] Average GAP LW voltage
V_MCPSW =                 898 / [V]   Average MCP SW voltage
V_MCPLW =                 846 / [V]   Average MCP LW voltage
V_GAPSW =                2784 / [V]   Average GAP SW voltage
V_GAPLW =                2793 / [V]   Average GAP LW voltage
TIMAQUTC= '2026-02-25T03:36:13.793' / [UTC] Average Start t. of data acquisition


           ----------------------------------------
           | Keywords describing processing steps |
           ----------------------------------------
PRSTEP1 = 'DISPERSION-BINNING' / Type of processing, step 1
PRPROC1 = 'Dispersion Binning (On-board)' / Name of procedure, step 1

PRSTEP2 = 'COMPRESSION'        / Type of processing, step 2
PRPROC2 = 'JPEG Compression (On-board)' / Name of procedure, step 2

PRSTEP3 = 'TELEMETRY-PARSING'  / XML decoding, decompression if applicable, etc
PRPROC3 = 'spice_process_telemetry.pro' / Name of procedure, step 3
PRPVER3 = '03.07.00'           / Version of procedure, step 3
PRLIB3A = 'uio-spice-pipeline/telemetry_parsing' / Library containing PRPROC3
PRVER3A = '6076    '           / UiO SVN revision number of PRLIB3A (2026-03-19)


PRSTEP4 = 'SPATIAL-COORDINATE-CALCULATION' / Type of processing
PRPROC4 = 'get_sunspice_hpc_point.pro' / Proc. returning s/c helioproj. coords
PRPVER4 =                60067 / Mod. julday of PRPROC4 (2023-05-03)
PRPARA4 = 'correction = "lt+s"' / Correct for velocity aberration
PREF4   = 'spk="solo_ANC_soc-orbit-stp_20200210-20301120_403_V1_00528_V01.bsp"&'
CONTINUE  ',spk_date = "2026-03-10"                                            &'
CONTINUE  ',ck="solo_ANC_soc-flown-att_20260224T224357-20260225T092301_V01.bc"&'
CONTINUE  ',ck_date  = "2026-02-28"                                            &'
CONTINUE  '' / Name and date of loaded ephemeris (spk) and attitude (ck) files
COMMENT
COMMENT    As-flown attitude file applied in coordinate calculation
COMMENT
PRLIB4A = '$SSW/packages/sunspice/idl/' / Software library containing PRPROC4
PRVER4A =                60921 / Mod. julday of PRLIB4A (2025-09-03)

PRSTEP5 = 'SPATIAL-COORDINATE-CORRECTION' / Type of processing, step 5
PRPROC5 = 'correct_spice_offset_relative_to_spacecraft.pro' / Name of procedure,
PRPVER5 = '3.0     '           / Version of procedure, step 5
PRREF5  = 'delta_instrument_x = -79,                                           &'
CONTINUE  'delta_instrument_y = -45                                            &'
CONTINUE  '' / [arcsec] constant SPICE vs spacecraft offset
PRLIB5A = 'uio-spice-pipeline' / Software library containing PRPROC5
PRVER5A = '6093    '           / UiO SVN revision number of PRLIB5A (2026-03-23)


           -------------------------------------------------
           | Hardware and software processing environment  |
           -------------------------------------------------
LONGSTRN= 'OGIP 1.0'           / The OGIP long string convention may be used.
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords.  This convention uses the  '&'
COMMENT character at the end of a string which is then continued
```

```
COMMENT on subsequent keywords whose name = 'CONTINUE'.
PRENV3  = '  Kernel: Linux                                                    &'
CONTINUE  '  Kernel release number: 5.14.0-570.49.1.el9_6.x86_64              &'
CONTINUE  '  Architecture: x86_64                                             &'
CONTINUE  '  Host name: astro-sdc-fs2.uio.no                                  &'
CONTINUE  '  OS: Red Hat Enterprise Linux 9.7 (Plow)                          &'
CONTINUE  '  CPU: Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz                 &'
CONTINUE  '  IDL 9.2.0 (Jul 21 2025 (481746)), memory bits: 64, file offset bi&'
CONTINUE  'ts: 64      ' / Hardware and software


           ----------------------------------------------
           | SOLARNET keywords, and additional keywords |
           ----------------------------------------------
SOLARNET=              1.00000 / Fully/Partially/No SOLARNET compliant (1/0.5/-1
OBS_HDU =                    1 / HDU contains observational data (1) or not (0)
PARENT  = 'solo_L0_spice-n-ras_0825298748_V202603240957C_369099136-001.fits' / L
FILE_RAW= 'sc_2026_02_24.xml;sc_2026_02_25.xml' / Telemetry file
INFO_URL= 'https://spice.ias.u-psud.fr/' / URL to additional information


           ----------------------------
           | Solar Ephemeris Keywords |
           ----------------------------
DSUN_OBS=        49004572359.2 / [m]  S/C distance from Sun
DSUN_AU =       0.327575348136 / [AU] S/C distance from Sun
RSUN_ARC=        2928.06913591 / [arcsec] Apparent photospheric Solar radius
RSUN_REF=          6.95700E+08 / [m]      Assumed  photospheric Solar radius
SOLAR_B0=       -16.3220476790 / [deg] Tilt angle of Solar North toward S/C
SOLAR_P0=       -25.5654536810 / [deg] S/C Celestial North to Solar North angle
SOLAR_EP=       -6.32785431438 / [deg] S/C Ecliptic  North to Solar North angle
CAR_ROT =                 2307 / Carrington rotation number
HGLT_OBS=       -16.3220476790 / [deg] S/C Stonyhurst latitude (B0 angle)
HGLN_OBS=        68.1652116652 / [deg] S/C Stonyhurst longitude
CRLT_OBS=       -16.3220476790 / [deg] S/C Carrington latitude (B0 angle)
CRLN_OBS=        1.00842492752 / [deg] S/C Carrington longitude (L0 angle)
HEEX_OBS=        19070148293.0 / [m]   S/C Heliocentric Earth Ecliptic X
HEEY_OBS=        43888703977.0 / [m]   S/C Heliocentric Earth Ecliptic Y
HEEZ_OBS=       -10562159789.1 / [m]   S/C Heliocentric Earth Ecliptic Z
HCIX_OBS=       -40099364209.0 / [m]   S/C Heliocentric Inertial X
HCIY_OBS=        24572743721.0 / [m]   S/C Heliocentric Inertial Y
HCIZ_OBS=       -13772050252.8 / [m]   S/C Heliocentric Inertial Z
HCIX_VOB=       -19301.8893913 / [m/s] S/C Heliocentric Inertial X Velocity
HCIY_VOB=       -58987.0129068 / [m/s] S/C Heliocentric Inertial Y Velocity
HCIZ_VOB=        8857.03463655 / [m/s] S/C Heliocentric Inertial Z Velocity
HAEX_OBS=        53056.3311365 / [m]   S/C Heliocentric Aries Ecliptic X
HAEY_OBS=       -33372.2481151 / [m]   S/C Heliocentric Aries Ecliptic Y
HAEZ_OBS=        1340.32472648 / [m]   S/C Heliocentric Aries Ecliptic Z
HEQX_OBS=        17491772537.1 / [m]   S/C Heliocentric Earth Equatorial X
HEQY_OBS=        43655659855.9 / [m]   S/C Heliocentric Earth Equatorial Y
HEQZ_OBS=       -13772050252.8 / [m]   S/C Heliocentric Earth Equatorial Z
GSEX_OBS=        129006385398. / [m]   S/C Geocentric   Solar Ecliptic X
GSEY_OBS=       -43888703977.0 / [m]   S/C Geocentric   Solar Ecliptic Y
GSEZ_OBS=       -10562159789.1 / [m]   S/C Geocentric   Solar Ecliptic Z
OBS_VR  =       -16273.1485594 / [m/s] Radial velocity of S/C away from the Sun
EAR_TDEL=        330.468491411 / [s] Time(Sun to Earth)-Time(Sun to S/C)
```

```
SUN_TIME=        163.461658396 / [s] Time(Sun centre to S/C)
DATE_EAR= '2026-02-25T01:50:14.249' / [UTC] DATE-BEG + EAR_TDEL
DATE_SUN= '2026-02-25T01:42:00.319' / [UTC] DATE-BEG - SUN_TIME


           -------------------------
           | HISTORY and checksums |
           -------------------------
HISTORY process_request.pro
HISTORY spice_process_telemetry.pro
HISTORY spice_l0_to_l1.pro
DATASUM = '1562167775'         / data unit checksum created 2026-03-24T09:10:24
CHECKSUM= 'QCVDTAS9QASCQAS9'   / HDU checksum created 2026-03-24T09:10:24
O_BLANK =                32767 / Original BLANK value
O_BZERO =              32768.0 / Original BZERO Value
END
```

4.4.1.3 Brief description of some selected FITS keywords

All keywords that are described in this section apply for both L1 and L2 files, except for the L1 only keywords mentioned in Sections 4.4.1.3.4 and 4.4.1.3.5.3.

4.4.1.3.1 Keywords identifying the study

The keyword `OBS_TYPE` is a 3-character alphanumeric string that uniquely identifies the `OBS_MODE`, which contains the name of the SPICE study. The value of the Solar Obiter-wide `OBS_MODE` is repeated in the SPICE-specific keyword `STUDY`.

The keyword `OBS_ID` contains the SOC Observation ID. The `OBS_ID` is built up of parameters that together uniquely identify the observation in a Solar Orbiter-wide context. The format of the SOC Observation ID is:

`AAAA_PPvV_SSS_III_OOOO_JJJ`

The components of this string are:

`AAAA`: 4-character instrument ID, is always “`SSPI`” for SPICE
`PPvV`: 4-character string, Long-Term Planning plan ID
`SSS`: 3-digit alphanumeric SOOP Type, equals FITS keyword `SOOPTYPE`
III: 3-digit (base-58) SOOP Instance
OOOO: 4-character (base-58) Observation Type, equals FITS keyword `OBS_TYPE`
`JJJ`: 3-digit (base-58) Observation Instance

An observation may belong to multiple SOOPs. In such cases, the SOC Observation ID contains multiple strings as described above, each sub-string being separated by a semicolon. See [IOR-ICD] for details.

Note that the SPICE Observation ID is given in the keyword `SPIOBSID`. This keyword uniquely identifies an observation series in a SPICE context (the combination of `SPIOBSID` and `RASTERNO` uniquely identifies a single observation). The `SPIOBSID` is a 32-bit on-board monotonically increasing observation counter.

4.4.1.3.2 Keywords derived from the value of `STUDYFLG`

The 8-bit value of the STUDYFLG keyword stems from the Science Header Packets, and the 6 keywords following STUDYFLG in the header example are set based on this value, in accordance with Section 4.2.6.1. of [DATAICD].

4.4.1.3.3 Keywords describing the size of the readout windows

Note that in L2 files any adjacent windows are merged, see Section 4.4.2. In such cases the keywords described below apply to the merged window.

The window start column on the detector and the window width are both given in pixel coordinates in the telemetry, see Table 4-13. The values of WAVEMIN and WINWIDTH are derived from these pixel values, using the conversion from pixel indices to nm given in [DISPERSION] and [MAN]. Note that for L2 files all wavelength related keywords are recalculated to correct for the temperature dependent wavelength scale [TEMPWAVE]. WAVEMIN is the wavelength of the leftmost edge of the first detector pixel column of the readout window, ignoring any binning, and WINWIDTH is the edge-to-edge window width. WAVEMAX is the wavelength corresponding to the rightmost edge of the window's last pixel column. For narrow-slit observations the pixel value of the window start column (starting at 1 for the leftmost pixel column on the SW detector, 1025 for the leftmost column on the LW detector) can be found in the Solar Orbiter mandatory keyword PXBEG3, and the pixel value of the last column is stored in PXEND3. The edge-to-edge pixel width of the window is therefore PXEND3 – PXBEG3 + 1. PXCOV3 contains a comma separated list of the detector coverage in the dispersion dimension of all HDUs in the file, i.e. a list of all PXBEG3-PXEND3 ranges.

The window start *row*, starting at 1 for the lowermost[8] pixels on the detector, is also given in the telemetry and is stored in the FITS files as PXBEG2. The window end row is stored in PXEND2.

The scan direction of the scan mirror is from Solar West to Solar East. This is indicated by always having PXEND1 = 1. For raster observations PXBEG1 > 1, for other study types PXBEG1 = PXEND1 = 1.

PXBEG4 = 1 for all observation types. For sit-and-stare observations PXEND4 > 1, for other study types PXEND4 = PXBEG4 = 1.

4.4.1.3.4 L1 only: The BLANK keyword

The data type of an L1 HDU's data array is normally 16-bit *unsigned* integer. However, when reconstructing SHC-compressed data some pixel values may end up being negative, and the data type for SHC-compressed data is therefore always 16-bit *signed* integer. Due to this difference the value of undefined pixels (BLANK) is set to $2^{16}-1$ for unsigned integer array HDUs and $2^{15}-1$ for signed integer array HDUs.

Note that in L2 files the data arrays are of type floating point and the BLANK keyword is not present. Undefined pixels in L2 files are set to NaN.

4.4.1.3.5 Keywords describing telemetry and telemetry loss and consequences thereof

L1 and L2 files contain a collection of FITS keywords that briefly describe the completeness of the telemetry of the file as a whole, and of each of the observational HDUs.

4.4.1.3.5.1 Keywords having the same values for all observational HDUs in the file

[8] i.e. the southernmost detector pixel row in the case of no spacecraft roll

`COMPLETE`: Completeness of all windows in the file combined. '`C`' if complete, '`I`' if incomplete.
`PCT_CMPL`: Completeness of all windows in the file combined, in percent.

4.4.1.3.5.2 Keywords describing the telemetry of an individual HDU

`NTOTPIX`: The number of potentially usable pixels
`NDATAPIX`: The number of usable pixels excluding saturated & `BLANK` (L1) or `NaN` (L2)
`NSATPIX`: The number of fully saturated pixels
`NLOSTPIX`: The number of lost pixels due to telemetry loss
`NAPRXPIX`: The number of pixels with approximated values due to loss of compressed telemetry
`PCT_DATA: NDATAPIX/NTOTPIX*100`
`PCT_SATP: NSATPIX/ NTOTPIX*100`
`PCT_LOST: NLOSTPIX/NTOTPIX*100`
`PCT_APRX: NAPRXPIX/NTOTPIX*100`

4.4.1.3.5.3 L1 only: keywords describing the telemetry of an individual HDU in detail

Note that neither the keywords mentioned in this section nor the binary table extensions they are referring to are present in L2 files.

`NPACKETS`: the total number of packets *with observational data* that were prepared on-board for downlink. This number includes the total number of Data Packets of all the window's *n* segments, plus *n* if the *n* Final Packets contained observational data.
`LOSTPKTS`: the number of lost packets with observational data
`LOSTBINS`: the number of lost FFT bins (in the case of SHC data)
`NLOSTCHK`: the number of lost Final Packets (with or without observational data)
`NFAILCHK`: the number of lost data checksum test that have failed
`NAPRXPLN`: the number of approximated image plane ranges due to lost compressed telemetry
`NLOSTPIX`: the number of lost pixels due to lost compressed or uncompressed telemetry
`NLOSTPLN`: the number of lost image plane ranges due to lost compressed telemetry (or lost FFT coefficient planes for SHC data)

Sections 4.4.4.4 and 4.4.5 describe how the indices of lost telemetry packets, lost FFT coefficient planes, approximated image plane ranges, lost image plane ranges, and lost FFT bins are stored in binary table extensions in L1 files. Lost pixels (and lost image planes) can easily be identified by selecting pixels having the `BLANK` value and therefore the indices of such pixels and image planes are not stored in binary table extensions.

4.4.1.3.6 Keywords describing coordinate distortions

After a significant re-pointing or a wheel off-loading it may take tens of minutes, even hours, until the pointing of Solar Orbiter is stable. If SPICE is observing during a period of unstable spacecraft pointing it may be necessary to take the pointing instability into account when calculating the Solar X and Solar Y coordinates of the observation.

For single exposure observations `CRDER1` and `CRDER2` give the standard deviation of the S/C pointing in arc seconds during the exposure. For multi-exposure observations these keywords give the mean of the standard deviations of the S/C pointing. The individual standard deviations are stored in a binary table extension, see Section 4.4.4. for details on *variable keywords*.

The `CWERR1` and `CWERR2` keywords give the maximum absolute deviation in arc seconds from the Solar X and Solar Y coordinates calculated by Equation (1). Following [WCSDISTORTIONS], these deviations, or coordinate distortions, are stored in two separate image extension both having `EXTNAME` = ‘`WCSDVARR`’. The distortions of Solar X are stored in the `WCSDVARR` extension having `EXTVER=1`, the Solar Y distortions are stored in the `WCSDVARR` extension having `EXTVER=2`.

The distortion mechanisms described in [WCSDISTORTIONS] support distortions of *pixel coordinates* that are used to calculate the world coordinates. In SPICE files, however, we use the SOLARNET mechanisms mentioned in [SSTRED] to store the distortions of the calculated Solar X and Solar Y world coordinates.

We represent the distortions according to the `Lookup` mechanism outlined in Section 3.4 of [WCSDISTORTIONS]. The spacecraft roll is assumed to be constant during the observation, making each of the two coordinate distortion arrays 1-dimensional, with one coordinate distortion value per exposure. See Figure 4-2 for an example of Solar X and Solar Y coordinate distortion arrays. Figure 4-3 shows the corresponding CIII raster image without and with corrected coordinates.

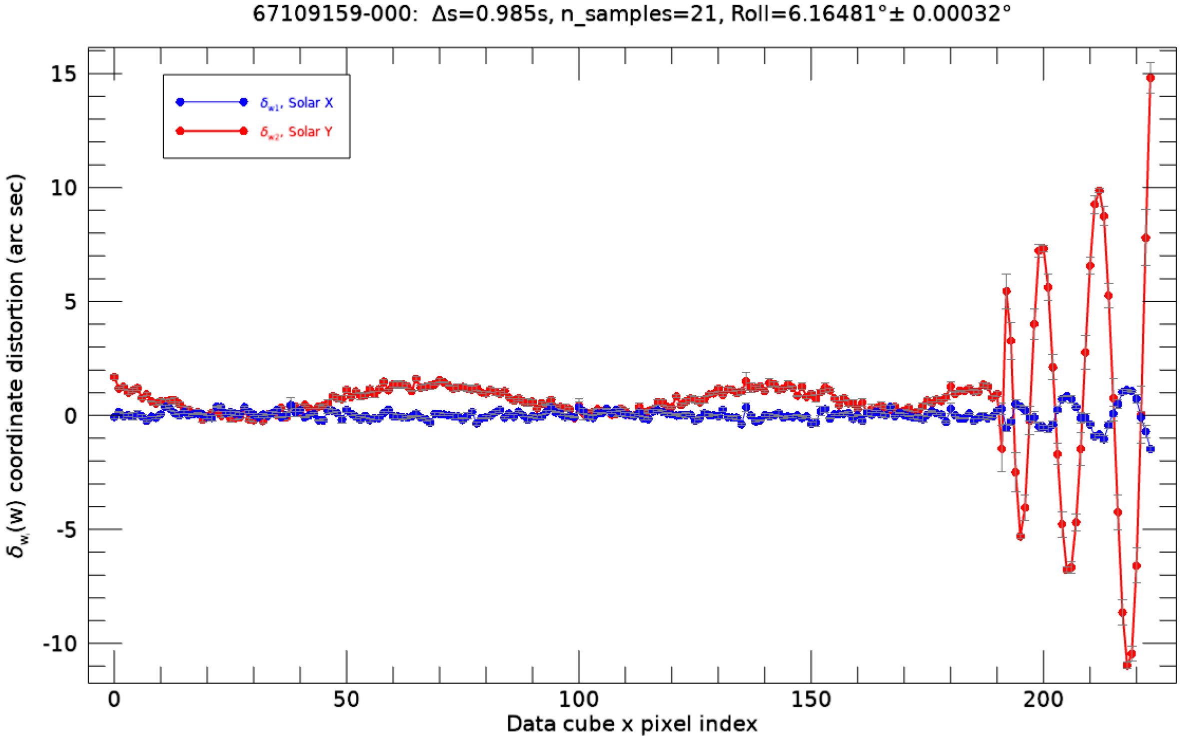

Figure 4-2: 2021-09-14, `SPIOBSID` 67109159: An example of Solar X and Solar Y coordinate distortions due to unusually pronounced spacecraft pointing instability. The error bars correspond to the standard deviation of the coordinate distortions during a 20 sec exposure. These standard deviations are given in the variable keywords `CRDERi`. The maximum amplitudes of the distortions are given by `CWERRi`. The duration of the raster is 1 hour 15 minutes, 224 exposures were taken. Note that the distortion correction values for Solar Y are not fluctuating around 0, but are shifted towards positive distortion. The reason for this shift is that the distortion correction at the reference pixel `CRPIX1=112.5` is by definition 0.

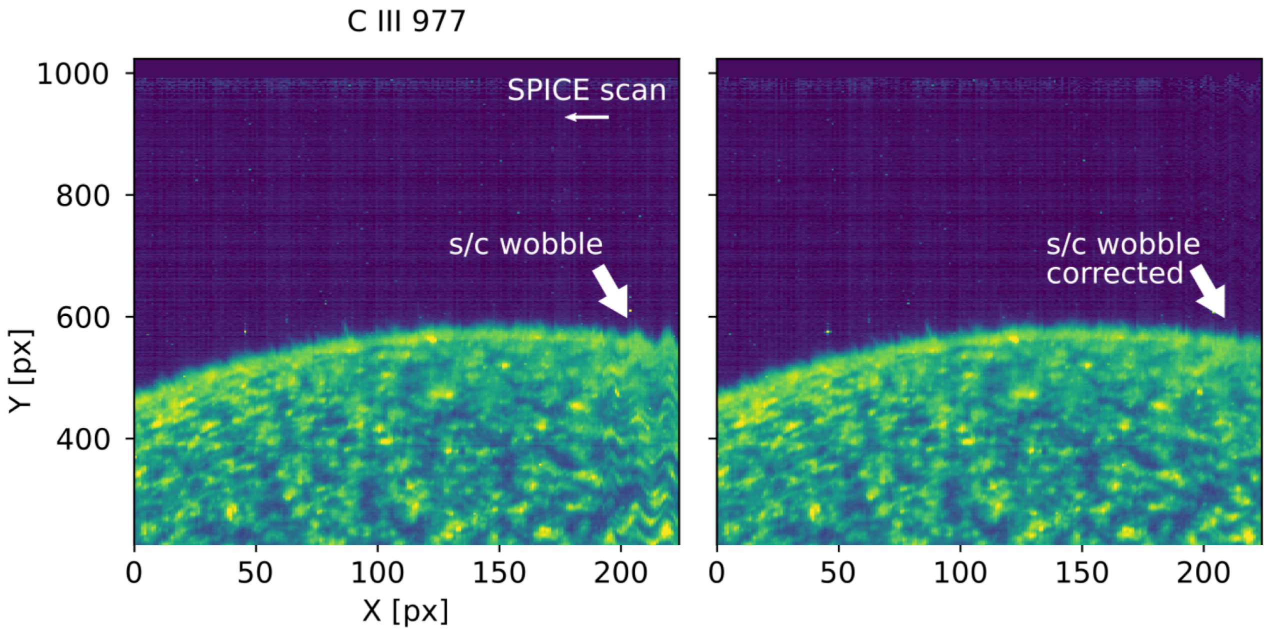

Figure 4-3: Example of a SPICE CIII raster before and after coordinate distortion correction. (G. Pelouze)

The IDL SolarSoft function `wcs_get_coord.pro` supports the SOLARNET interpretation of the `Lookup` mechanism that is used in SPICE FITS files[9]. When calculating coordinates, the distortions are automatically taken into account:

```
IDL> data = readfits(file, header, ext=0)
IDL> wcs = fitshead2wcs(header, filename=file)
IDL> corrected_coordinates = wcs_get_coord(wcs)
```

The SPICE IDL quicklook and analysis tools [IDLANA] use `wcs_get_coord.pro` to calculate corrected coordinates.

As of January 2023, the existing WCS tools in astropy/python do not support the `Lookup` method for calculating coordinate distortions, nor the SOLARNET implementation of the distortion mechanisms. Python users will therefore have to calculate the Solar X and Solar Y coordinates using the existing tools, before manually adding the distortions to obtain the corrected coordinates. I.e., for slit position `n`:

- calculate the Solar X coordinates of all pixels along the slit and add element `n` of the distortion array stored in the `WCSDVARR` extension having `EXTVER=1`
- calculate the Solar Y coordinates of all pixels along the slit and add element `n` of the distortion array stored in the `WCSDVARR` extension having `EXTVER=2`

4.4.1.3.7 Keywords describing the processing steps

A set of `PRxxxxn` keywords describes each processing step that have been applied to the data:

`PRSTEPn:` a description of the processing step
`PRPROCn:` the name of the routine performing the processing step

[9] `wcs_get_coord.pro` version 11 (11th May 2023) and later

`PRPVERn`: version of the processing routine
`PRPARAn`: input parameters to the processing routine
`PRPREFn`: other relevant parameters used by or calculated by the processing routine
`PRLIBnA`: name of the library that contains the processing routine
`PRVERnA`: version of the library

Processing performed on-ground is described with the complete list of `PRxxxxn` keywords. On-board processing is described by the first two keywords.

Finally, the keyword `PRENVn` describes the hardware and software environment of the on-ground processing. `PRENVn` is valid for all processing steps starting at step `n`, until a new `PRENVn` keyword is defined. However, for SPICE FITS files created by the pipeline, the processing environment is normally the same for all processing steps, and therefore only a single `PRENVn` keyword that is valid for all processing steps performed on ground will normally be defined.

4.4.1.3.8 Keyword giving the name(s) of the telemetry file(s)

The name of the source telemetry file is given in the keyword `FILE_RAW`. If also a telemetry file from the preceding day is given as input to the pipeline, the value of `FILE_RAW` is the names of the telemetry files of the preceding day and the current day, separated by a semicolon.

4.4.1.3.9 Keywords used to determine the study type: sit-and-stare or raster

The pipeline uses the IORs and study definition files to determine whether a multi-exposure observation is a raster or a sit-and-stare observation. These files are only available for cruise phase and nominal mission phase observations. To determine the study type for commissioning phase observations, we must therefore instead use the scan mirror positions reported in the telemetry. Due to readout noise the scan mirror position values are neither constant for sit-and-stare observations nor monotonically increasing for raster observations. Therefore, the study type is determined from a linear fit of the scan mirror positions. For pipeline debugging purposes the parameters of the fit are included in the headers of commissioning observation files: `MIRRDELT` gives the slope of fit and `SMIRRDEL` gives the 1-sigma uncertainty estimate of the fit.

4.4.1.4 L1 HDUs with narrow-slit spectral-profile data (window type 1)

Data arrays:
SPICE narrow-slit (2”, 4” or 6”) spectral-profile data arrays have dimensions according to Table 4-3 below. Readout windows have a width in the dispersion direction of `D = 4, 8, 16` or `32` pixels.

Study type	Dimensions
Sit-and-stare	`(1,Y,D,t)`
Raster	`(X,Y,D,1)`
Single exposure	`(1,Y,D,1)`

Table 4-3: Dimensions of narrow-slit spectral-profile data arrays

4.4.1.5 L1 HDUs with dumbbell stack data (window type 2)

Note that due to technical reasons stand-alone dumbbell observations are rarely recorded, see page 30 of Section 4.

At each end of the slit in the `Y` direction there is an area of nominal size 30" x 30" used for making small context images, so called dumbbell or alignment windows. It is possible to downlink one or both dumbbells, but only for a single window per study. The dumbbells may be downloaded in addition to, or instead of, the spectral-profile data cube, and are included in the same FITS file as the spectral-profile data, in separate HDUs.

Data arrays:
SPICE dumbbell stack data arrays have dimensions according to Table 4-4 below.

Study type	Dimensions
Sit-and-stare	`(1,32,64,t)`
Raster	`(X,32,64,1)`
Single exposure	*Not applicable*

Table 4-4: Dimensions of dumbbell stack data arrays

4.4.1.6 L1 HDUs with wide-slit data (window type 3)

Data array:
SPICE wide-slit (30”) data arrays have dimensions according to Table 4-5 below. Readout windows have in most cases a width in the dispersion direction of `D = 32` pixels.

Study type	Dimensions
Sit-and-stare	`(1,Y,D,t)`
Raster	`(X,Y,D,1)`
Single exposure	`(1,Y,D,1)`

Table 4-5: Dimensions of wide-slit data arrays

4.4.1.7 L1 HDUs with intensity-window data (window type 4)

Note that intensity-window observations are rarely recorded, see page 30 of Section 4.

An intensity-window is binned in the dispersion dimension with a binning factor equal to the width of the window. Intensity-windows are normally observed in pairs, with one window covering a spectral line, and a nearby window covering a part of the spectrum without any strong emission lines. The intensity-window data cubes are saved in separate HDUs. The value of the wavelength coordinate is the central wavelength (i.e. the midpoint between the central two pixels before binning) of the readout window.

Data array:
SPICE intensity-window data arrays have dimensions according to Table 4-6 below.

Study type	Dimensions
Sit-and-stare	*Not applicable*
Raster	`(X,Y,1,1)`
Single exposure	*Not applicable*

Table 4-6: Dimensions of intensity-window data arrays

If the HDU stems from an intensity-window that was defined as either a line window or a background window in the MISO planning tool, the HDU has two of the following three keywords that are not present in the HDUs of any other window type: `IWINTYPE`, and either `IWINBKG` or

IWINLINE. HDUs storing stand-alone intensity-windows do not include any of these three keywords.

If the data of the HDU stems from a window covering an emission line, the value of IWINTYPE is 'LINE'. In that case an additional keyword IWINBKG gives the HDU number of the HDU storing the data that is to be regarded as the background level.

If the data stems from a background window the value of IWINTYPE is 'BACKGROUND'. A background window may be defined as background for multiple 'LINE' windows. IWINLINE gives a comma separated list of the HDU numbers of all 'LINE' window HDUs that has this window defined as the background.

As an example, the following keywords may be present in an HDU stemming from a window covering the emission line Ne VIII 770 A:

```
EXTNAME  = 'Ne VIII 770 A'      / Extension name
…
WINNO    =                    1 / Window number (starting at 0) within this study
…
IWINTYPE = 'LINE'               / This intensity-window covers an emission line
IWINBKG  =                    4 / HDU number (primary=0) storing background
```

The following keywords may be present in the corresponding HDU with data to be used for background-subtraction from the Ne VIII 770 A line:

```
EXTNAME  = 'Red wing of Ne VIII 770 A' / Extension name
…
WINNO    =
…
IWINTYPE = 'BACKGROUND'        / This intensity-window considered as background
IWINLINE = '0,1,2   '          / HDU numbers (primary=0) storing emission line
```

4.4.1.8 L1 HDUs with full-detector read-out data (window types 5 and 6)

It is possible to make a full-frame read-out where the full areas of both detector arrays are recorded. A full-frame read-out always consists of a single exposure[10]. Data from each detector array are stored in separate HDUs[11]. Both the three narrow slits and the wide slit may be used, in the latter case the PC1_3 transformation matrix element has a non-zero value.

[10] However, it is possible to repeat full detector observations, with or without moving the scan mirror. Rasters and sit-and-stare observations may therefore be constructed also for full detector observations by combining the individual exposures that are stored in separate FITS files.
[11] SPICE can transmit full-frame data in both compressed and uncompressed format. In this case the L1 file contains *four* HDUs, *two* for each detector array, one with decompressed data and one with the uncompressed data.

Data array:
SPICE full-detector read-out data arrays have dimensions according to Table 4-7 below.

Study type	Dimensions
Sit-and-stare	*Not applicable*
Raster	*Not applicable*
Single exposure	(1,1024,1024,1)

Table 4-7: Dimensions of full-detector read-out data arrays

4.4.2 Level 2 Data arrays and FITS headers

All keywords described in Section 4.4.1.3 are present in L2 files as well, with the exception of the keywords mentioned in Sections 4.4.1.3.4 and 4.4.1.3.5.3.

Compared to earlier data releases, L2 files in Data Release 6 contain a significantly larger fraction of pixels set to NaN. This is intentional: as part of the L1 to L2 calibration all pixels that are identified as data spikes are set to NaN. As in earlier data releases, L2 pixels influenced by saturated L1 pixels are also set to NaN, although saturation affects only a few pixels in a minority of files. However, for pixels that are set to NaN due to spikes or saturation, estimated and/or original values are stored in the FITS files and may be restored by the users; see Section 4.4.2.1.

To include data from all pixels present in the L1 data array, the geometrically corrected L2 data arrays are by default a few pixels wider in the dispersion dimension compared to the L1 data arrays described in Section 4.4.1. The values of padded pixels are set to NaN.

To ensure that an emission line is completely covered, or to catch multiple closely spaced lines, two or more adjacent readout windows may be defined. In L2 files the data arrays of such adjacent windows are merged (concatenated) if the windows have the same binning and compression. Neither Intensity-windows nor dumbbells are merged (but neither of these window types have been used other than for a very few test runs).

If adjacent windows have been merged the L2 file will contain fewer image HDUs than the L1 file.

The metadata that describe the data array are updated to reflect the wider dispersion dimension of the merged window, and the reduced number of HDUs in the file. This window concatenation is described by a set of PRxxxxn keywords, see the 'WINDOW-CONCATENATION' processing step in Section 4.4.2.2.

Two windows may be very close without being adjacent. Due to the geometrical correction of L2 files, pixels close to the adjoining edge of two closely spaced L2 windows may originally stem from the nearby window. A future update of the calibration routines will ensure that pixels are only present in their original window.

In L2 files the spatial plate scale in the direction of the slit is the same for the two detector arrays, and a given pixel along the slit on the SW or LW detector correspond the same location on the Sun. The plate scale in the dispersion dimension is adjusted to get the same spatial plate scale in the dispersion and y dimensions for wide-slit (and dumbbell) observations.

Note that if telemetry is lost for compressed observations, entire image planes or even full data cubes may be missing (i.e. are set to NaN) in L2 files, see Sections 4.4.4.4 and 4.4.5. L1 files contain binary table extensions storing information on lost telemetry and keywords describing the contents of these extensions, see Section 4.4.1.3.5.3. Neither these extensions nor the corresponding keywords are present in L2 files.

A series of repeated full-detector observations is concatenated into a single L2 FITS file. The individual exposures remain as separate L1 files.

Keywords that are present in L1 files are updated to account for e.g. calibrated data array values, merged windows, and merged files.

Some keywords are modified as required by [FITSpaper]. As an example, the `BLANK` keyword only applies to integer data HDUs, as is the case for L1. The floating-point data HDUs in L2 files do not have a `BLANK` keyword, and undefined pixels are set to `NaN`.

L2 files contain additional keywords that are not present in L1 files:

- Keywords that describe the calibration. This includes:
  – `VERS_CAL`, a Solar Orbiter-wide keyword giving the version of the calibration software
  – Additional sets of `PRxxxxn` keywords describing the L1 to L2 calibration steps, see Section 4.4.2.2.
  – `RADCAL`, the radiometric calibration factor as a function of dispersion pixel, see Section 4.4.4.3
- `WAVECOV`, a comma separated list of the wavelength coverage (i.e. '`WAVEMIN-WAVEMAX`') of all image HDUs in the file.
- Keywords that describe random and systematic errors (`CRDERi` and `CSYERi` respectively, see Table 3-8 of [METADATA])
- The keyword `UCD` giving the Unified Content Descriptors, e.g.:
  ```
  UCD = 'phot.radiance; em.line' ; for narrow-slit observations
  UCD = 'phot.radiance; em.UV'   ; for wide-slit/dumbbell observations
  ```
  The intensity unit of SPICE L2 files, given by `BUNIT`, is `W/m2/sr/nm` (narrow-slit) or `W/m2/sr` (wide-slit/dumbbell). The intensity unit is described by `BTYPE`, with the value '`Spectral Radiance`' or '`Radiance`' respectively.
- `NWIN_ORIG`, the total number of windows defined for the study in the MISO planning tool. The value will be equal to `NWIN` if no adjacent windows have been merged.
- `NSPIKPIX`, the number of spike pixels (i.e. the number of pixels set to `NaN` due to influence from L1 spike pixels)
- `PCT_SPIK`, the percentage of spike pixels

For files with two or more merged windows the following keywords will have a different value in L2 than in L1:

- `EXTNAME` is based on the `EXTNAMEs` of the individual windows, ending with the string "`(Merged)`".
- `NWIN` is the total number of observational HDUs in the L2 file. Note that `NWIN` is smaller than `NWIN_ORIG`
- `WINNO` is the window number in the L2 file (starting at 0)
- `MISOWIN` and `WINTAB` will have the value of the first window in the group of merged windows.
- Keywords describing the size of the windows, the pixel contents of the windows, etc.

4.4.2.1 Pixel lists describing saturated and spike pixels

A pixel may be identified as saturated, i.e. the number of detected counts reaches the 14-bit limit of the instrument (taking binning, pixel level offset and compression into account). As an approximation we regard the detector as linear up to this 14-bit threshold, i.e. we treat all L1 pixels below the threshold as not being saturated.

A pixel may also be identified as a spike pixel, i.e. it has an improbably high or low intensity compared to its neighbouring pixels. Spike pixel detection is performed using local sigma-clipping; see Section 4.4.2.1.5 for a detailed description of the method.

An L2 file contains pixel lists storing information on saturated and spike pixels: one binary-table pixel list for each observational HDU that contains saturated pixels, and one binary-table pixel list for each observational HDU that contains spike pixels.

The data arrays are corrected for geometric distortions when the FITS file is converted from L1 to L2. As a result, the signal from a single L1 pixel is distributed over several L2 pixels. Likewise, a given L2 pixel receives contributions from more than one L1 pixel. In L2 files, all pixels that contain a non-zero contribution from at least one saturated or spike L1 pixel are set to `NaN`.

The contribution that an L1 pixel makes to a given L2 pixel can vary greatly, depending on where on the detectors the L1 pixel is located. Instead of ignoring all L2 pixels that are influenced by one or more saturated or spike L1 pixels, users may want to fill in values for those L2 pixels that are not too heavily affected by data spikes or saturation. Therefore, both types of pixel lists store:

- the pixel coordinates of all pixels in the observational HDU that are influenced by saturated or spike L1 pixels.
- the fractional contribution of saturated or spike L1 pixels to each influenced L2 pixel.

The two types of pixel lists also store original and/or estimated values for the influenced pixels:

- Pixel lists for saturated pixels:
  - an estimate of the intensity of the influenced pixels if all saturated L1 pixels are ignored (set to 0) during calibration.
- Pixel lists for spike pixels:
  - the intensity that the influenced pixels would have had if sigma-clipping had not been applied during calibration ("original" values).
  - an estimate of the intensity of the influenced pixels as calculated by the sigma-clipping code.

Section 4.4.2.1.1 shows how to use the IDL `spice_data` object to fill in saturated pixels. Section 4.4.2.1.3 shows how the same object can be used to fill in spike pixels. Sections 4.4.2.1.2 and 4.4.2.1.4 describe how the information on saturated and spike pixels is stored in the FITS file.

4.4.2.1.1 Using the IDL `spice_data` object to fill partially saturated pixels with estimated values

If an L2 pixel receives all its contribution from saturated L1 pixels we refer to the pixel as "fully saturated". If an L2 pixel receives contributions from both saturated and un-saturated L1 pixels we refer to the pixel as "partially saturated".

The IDL `spice_data` object (see [IDLANA]) makes it easy to work with saturated pixels. We create an object:

```
IDL> o = spice_data(file)
```

When we extract the data for a window that has saturated pixels, we get a message like this:

```
IDL> d_default = o->get_window_data(winno)
```

```
|-----------------------------------------------------------------------------------------|
| EXTNAME = 'S IV (Merged)'                                                                |
| 217 pixels are set to NaN due to contribution from saturated pixels.                     |
|                                                                                          |
| * Set keyword FILL_SATURATED to fill in selected saturated pixels with estimated values. |
| By default all saturation-affected pixels are selected for the fill operation. Fully     |
| saturated pixels are set to max(data).                                                   |
|                                                                                          |
| * Set keyword MAX_SATURATION_FRACTION to a value between 0 and 1 to only select          |
| pixels having a fractional contribution from saturated pixels below the keyword          |
| value E.g. to fill only saturated pixels with a saturation contribution less than        |
| 50%, set MAX_SATURATION_FRACTION to 0.5.                                                 |
|                                                                                          |
| The value of a filled-in pixel is corrected for the "filling factor" by                  |
| upscaling the pixel value to value=value/(1-fractional_contribution_from_saturated_pixels) |
|                                                                                          |
-------------------------------------------------------------------------------------------|
```

A zoom-in of a line profile with saturated pixels is displayed in the left panel of Figure 4-4.

To fill all partially saturated pixels with estimated values, and set all fully saturated pixels to the maximum value of the data array, set the keyword `FILL_SATURATED` (the resulting line profile is displayed in the right panel of Figure 4-4):

```
IDL> d_all_sat_filled = o->get_window_data(winno, /fill_saturated)
 ------------------------------------------
|                                          |
| 135 partially saturated pixels filled in |
|                                          |
 ------------------------------------------
 --------------------------------------------
|                                            |
| 82 fully saturated pixels set to max(data) |
|                                            |
 --------------------------------------------
```

To fill in only partially saturated pixels that have a contribution from saturated L1 pixels less than e.g. 30%, set `MAX_SATURATION_FRACTION` to `0.3`, in addition to setting `FILL_SATURATED` (the line profile is displayed in the middle panel of Figure 4-4):

```
IDL> d_sat_LT30pct_filled = $
       o->get_window_data(winno, /fill_saturated, max_saturation_fraction=0.3)
 -----------------------------------------
|                                         |
| 54 partially saturated pixels filled in |
|                                         |
 -----------------------------------------
 --------------------------------------------
|                                            |
| 82 fully saturated pixels set to max(data) |
|                                            |
 --------------------------------------------
```

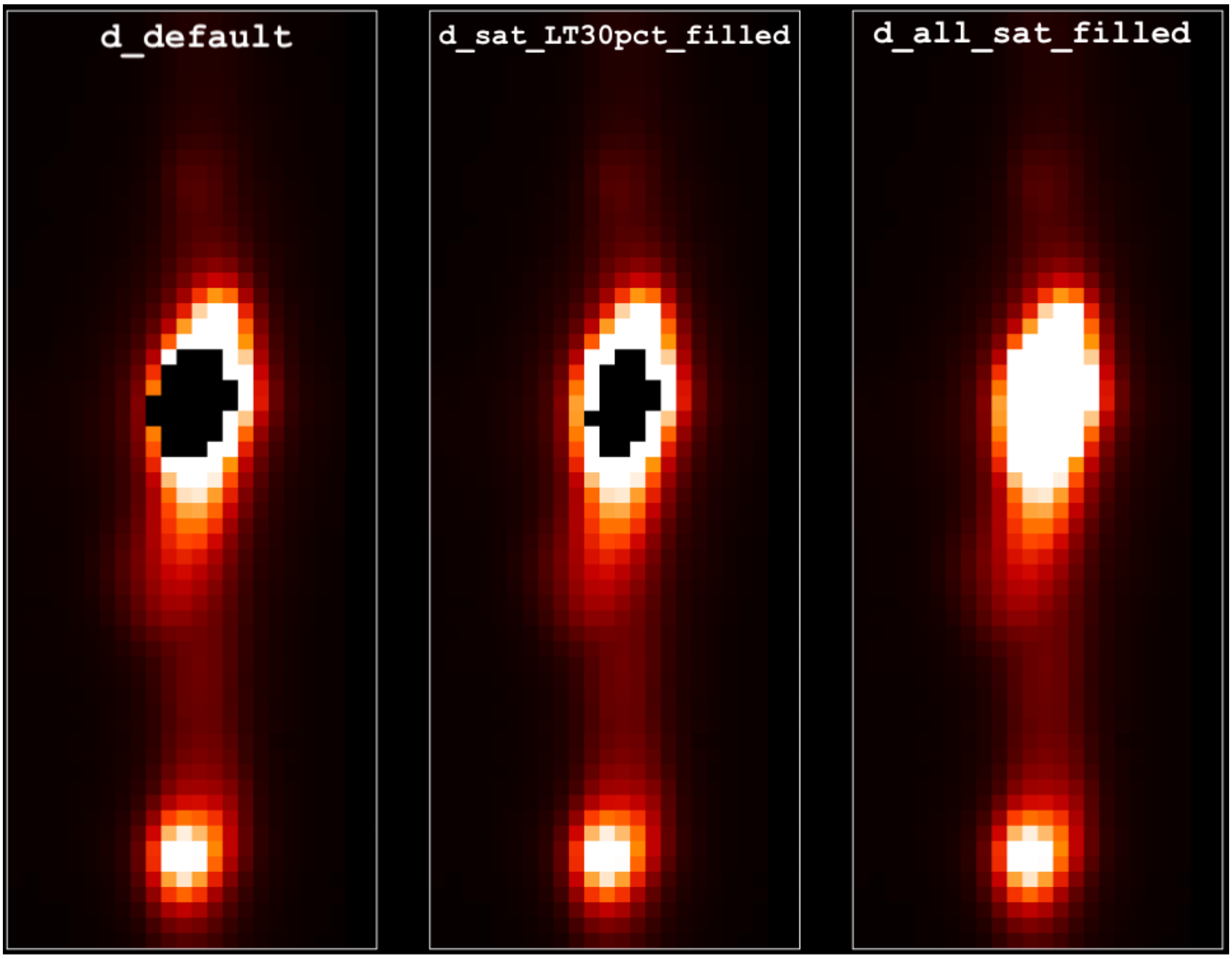

Figure 4-4: A crop of one of the exposures of the data cubes obtained with the previous IDL commands. Left: all L2 pixels with a contribution from saturated L1 pixels are set to `NaN`. Middle: L2 pixels with less than 30% contribution from saturated L1 pixels are filled in with estimated values. Right: all L2 pixels that are not fully saturated are filled in with estimated values, fully saturated L2 pixels are set to the maximum value of the L2 data array.

4.4.2.1.2 Extracting information on saturated pixels from the pixel lists

SPICE L2 store information on individual pixels that are influenced by saturation in binary table pixel lists. We use the SOLARNET implementation of the pixel list FITS standard [S-META].

If an HDU contains saturated pixels, the `EXTNAME` of the binary table extension storing the information on saturated pixels for this HDU can be extracted from the `PIXLISTS` keyword. As an example, if `PIXLISTS` has the value:

```
PIXLISTS= 'SATPIXLIST[Ly-g-CIII];ESTIMATED,SATPIX_CONTRIBUTION'
```

then the `EXTNAME` of the relevant binary table extension is `SATPIXLIST[Ly-g-CIII]`.

The SPICE `SATPIXLIST` binary table extensions always contain 6 columns, each column having one row for each flagged L2 pixel, see Table 4-8.

Column #	1	2	3	4	5	6
TTYPE	DIMENSION1	DIMENSION2	DIMENSION3	DIMENSION4	ESTIMATED	SATPIX_CONTRIBUTION

Table 4-8: TTYPE **values of the columns of a** SATPIXLIST **binary table pixel list**

The first 4 columns, with TTYPEn='DIMENSIONn' (n = 1,2,3 4), store the n pixel indices of the flagged pixels. The column 'ESTIMATED' contains the estimated values of L2 pixels when saturated L1 pixels were set to 0 during the calibration. The column 'SATPIX_CONTRIBUTION' contains the fraction of contribution of saturated L1 pixels to the flagged L2 pixel.

4.4.2.1.3 Using the IDL spice_data object to fill spike pixels with original or estimated values

We extract window data using the IDL spice_data object (see [IDLANA]):

```
IDL> o = spice_data(file)

IDL> d_default = o->get_window_data(winno)
```

If the window contains spike pixels, a message like this is displayed:

```
 ------------------------------------------------------------------------------------------
|                                                                                          |
| EXTNAME = 'Ne VIII 780 / Mg VIII 782 (Merged)'                                           |
| 393353 pixels are set to NaN due to contribution from spike pixels.                      |
|                                                                                          |
| * Set keyword FILL_SPIKES to fill in selected spike pixels with estimated values.        |
| * Set keyword RESTORE_SPIKES to restore selected spike pixels to their original values.  |
| By default all spike-affected pixels are selected for the fill/restore operations.       |
|                                                                                          |
| * Set keyword MAX_SPIKE_FRACTION to a value between 0 and 1 to only select               |
| pixels having a fractional contribution from spike pixels below the keyword              |
| value E.g. to fill only spike pixels with a spike contribution less than                 |
| 50%, set MAX_SPIKE_FRACTION to 0.5.                                                      |
|                                                                                          |
 ------------------------------------------------------------------------------------------
```

We apply the keywords listed in the message to replace the pixels that are set to NaN due to spikes with original or estimated values. The data arrays resulting from the commands below are displayed in Figure 4-5:

```
IDL> d_spikes_restored    = o->get_window_data(winno,/restore_spikes)

IDL> d_all_spikes_filled = o->get_window_data(winno,/fill_spikes)

IDL> d_spikes_LT30pct_filled = $
                o->get_window_data(winno,/fill_spikes,/max_spike_fraction=0.3)
```

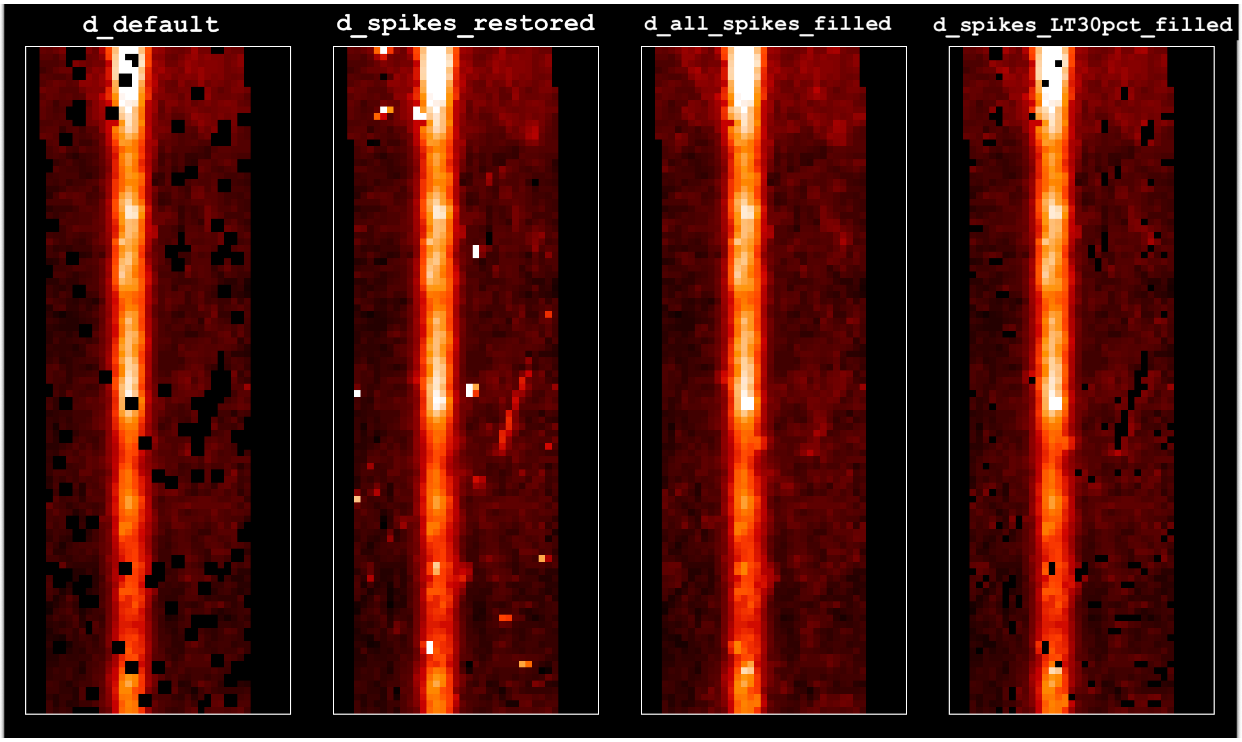

Figure 4-5: A crop of one of the exposures of the data cubes obtained with the `o->get_window_data()` commands above. From left to right: `d_default` – all L2 pixels with a contribution from spike L1 pixels are set to `NaN` and displayed in black (default); `d_spikes_restored` – no spike detection; `d_all_spikes_filled` – all detected spike pixels are replaced with values estimated by the sigma-clipping code; `d_spikes_LT30pct_filled` – all pixels with less than 30% contribution from L1 spike pixels are replaced with estimated values, pixels with more contribution from spike pixels than this threshold remain `NaN`

Note that the keywords that control the filling in of saturated and spike pixels may be combined, e.g.:

```
IDL> d = o->get_window_data(winno,/fill_spikes,/max_spike_fraction=0.25, $
                              /fill_saturated, /max_saturation_fraction=0.6)
```

4.4.2.1.4 Extracting information on spike pixels from pixel lists

SPICE L2 store information on individual pixels that are influenced by data spikes in binary table pixel lists. We use the SOLARNET implementation of the pixel list FITS standard [S-META].

If an HDU contains spike pixels, the `EXTNAME` of the binary table extension storing the information on spike pixels for this HDU can be extracted from the `PIXLISTS` keyword. As an example, if `PIXLISTS` has the value:

```
PIXLISTS= 'SPIKPIXLIST[S IV (Merged)];ORIGINAL,SPIKPIX_CONTRIBUTION,ESTIMATED'
```

then the `EXTNAME` of the relevant binary table extension is `SPIKPIXLIST[S IV (Merged)]`

The SPICE `SPIKPIXLIST` binary table extensions always contain 7 columns, each column having one row for each flagged L2 pixel, see Table 4-9.

Column #	1	2	3	4	5	6	7
TTYPE	DIMENSION1	DIMENSION2	DIMENSION3	DIMENSION4	ORIGINAL	SPIKPIX_CONTRIBUTION	ESTIMATED

Table 4-9: TTYPE values of the columns of a SPIKPIXLIST binary table pixel list. Note the column number that the ESTIMATED column has in this pixel list compared to its number it has in the SATPIXLIST pixel list. This somewhat confusing difference is due to technicalities in the pipeline code.

The first 4 columns with TTYPEn='DIMENSIONn' (n = 1,2,3 4), store the n pixel indices of the flagged pixels. The binary table column 'ORIGINAL' contains the original values of the flagged pixels, i.e. the values the flagged pixels would have had if no spike detection had been performed. The 'SPIKPIX_CONTRIBUTION' contains the fractional contribution of L1 spike pixels to the flagged L2 pixel. Finally, 'ESTIMATED' contains the values estimated by the sigma-clipping code.

4.4.2.1.5 Details on the spike detection

The pipeline uses local sigma-clipping for detecting data spikes and calculating estimated values for the spikes. The sigma-clipping code is written in Python and calls code written in C for speed (see [SIGMACLIP] and [FTOOLS]). The Python code is based on the sigma-clipping code provided by F. Auchère (IAS).

The sigma-clipping is done as the second L1 to L2 calibration step, right after the dark subtraction. At this stage of the processing, both raster and sit-and-stare data cubes have been temporarily reformed to 3 dimensions, (dispersion,Y,exposure_no).

There are two distinct types of data spikes that the sigma-clipping detects:

1) a fixed pattern of single bright/dark spike pixels that can be found in *all* exposures of an observation series (raster or sit-and-stare). This fixed pattern is caused by subtracting a dark with a warm/hot pixel configuration that differs from that of the scientific observation.
2) cosmic rays (blobs and streaks) that are found in individual exposures

For multi-exposure observations, the two types of data spikes are detected in two separate steps.

The fixed pattern is detected by sigma-clipping a *minimum image* (this step is omitted for full-detector observations that are not repeated):

- A detector *minimum image* is created by setting each (dispersion,Y) pixel to the minimum value that this pixel has in all exposures of the 3D data cube. Such a minimum image enhances the fixed pattern while obscuring any cosmic rays occurring in single exposures.
- The minimum image is sigma-clipped.
- All spike pixels of the minimum image identified by the sigma-clipping are set to NaN.

The cosmic rays are detected by:

- Using the sigma-clipped minimum image from the previous step, all detected (dispersion,Y) spike pixels are set to NaN for *all* exposures of the 3D data cube. The fixed pattern is now removed from the 3D data cube. (This step is omitted for full-detector observations that are not repeated)
- The 3D data cube (now having the fixed pattern pixels replaced by NaNs) is sigma-clipped to detect spikes occurring in individual exposures. Note that the full 3D data cube is sent as input to the sigma-clipping, i.e. we do *not* sigma-clip one exposure at a time.

Finally, all spike pixels identified by any of the two sigma-clipping processes are set to NaN, and the rest of the calibration steps are performed

The input parameters to the `spice_sigma_clip2.py` function can be found in the `PRxxxxn` keywords of the FITS file (see Section 4.4.2.2) describing the '`DESPIKING`' processing step. Table 4-10 lists the input parameters used for Data Release 6.

	kernel	sigma_lower	sigma_upper	exclude_center
2D Minimum Image	3	8	8	1
3D Data Cube	3	2.5	3	1

Table 4-10: input parameters to the sigma-clipping function.

To populate the pixel lists with information on the spike pixels, the pipeline then runs several data cubes through some or all steps of the L1 to L2 calibration: a spike pixel mask, a data cube that has not been sigma-clipped, and a data cube where all detected spike pixels have been replaced by values estimated by the sigma-clipping routine.

4.4.2.2 Keywords describing the individual L1 to L2 calibration steps

As mentioned in Section 4.4.1.3.7 the individual processing steps are described by a set of `PRxxxxn` keywords:

`PRSTEPn:` a description of the processing step
`PRPROCn:` the name of the routine performing the processing step
`PRPVERn:` version of the processing routine
`PRPARAn:` input parameters to the processing routine
`PRPREFn:` other relevant parameters used by or calculated by the processing routine
`PRLIBnA`: name of the library that contains the processing routine
`PRVERnA`: version of the library
`PRENVn`: hardware and software environment of the on-ground processing, valid for processing step `n` and for all remaining steps

Below is an example of a collection of `PRxxxxn` keywords used to describe the processing of a file. For completeness the list includes processing steps performed to create the parent L1 file (in this example, steps 1 through 6), and the steps performed to calibrate the L1 file to L2 (steps 7 through 12). In Section 4.4.2.2.1 we give some additional comments on the values that two of these keywords may take.

Note that not all files include the same processing steps, i.e. the user can't assume that e.g. step 9 will always be burn-in correction.

```
PRSTEP1 = 'DISPERSION-BINNING' / Type of processing, step 1
PRPROC1 = 'Dispersion Binning (On-board)' / Name of procedure, step 1

PRSTEP2 = 'COMPRESSION'         / Type of processing, step 2
PRPROC2 = 'JPEG Compression (On-board)' / Name of procedure, step 2

PRSTEP3 = 'TELEMETRY-PARSING'   / XML decoding, decompression if applicable, etc
PRPROC3 = 'spice_process_telemetry.pro' / Name of procedure, step 3
PRPVER3 = '03.07.00'            / Version of procedure, step 3
PRLIB3A = 'uio-spice-pipeline/telemetry_parsing' / Library containing PRPROC3
PRVER3A = '6076    '            / UiO SVN revision number of PRLIB3A (2026-03-19)
```

```
PRSTEP4 = 'SPATIAL-COORDINATE-CALCULATION' / Type of processing
PRPROC4 = 'get_sunspice_hpc_point.pro' / Proc. returning s/c helioproj. coords
PRPVER4 =                60067 / Mod. julday of PRPROC4 (2023-05-03)
PRPARA4 = 'correction = "lt+s"' / Correct for velocity aberration
PRREF4  = 'spk="solo_ANC_soc-orbit-stp_20200210-20301120_403_V1_00528_V01.bsp"&'
CONTINUE  ',spk_date = "2026-03-10"                                           &'
CONTINUE  ',ck="solo_ANC_soc-flown-att_20260224T224357-20260225T092301_V01.bc"&'
CONTINUE  ',ck_date  = "2026-02-28"                                           &'
CONTINUE  '' / Name and date of loaded ephemeris (spk) and attitude (ck) files
COMMENT
COMMENT    As-flown attitude file applied in coordinate calculation
COMMENT
PRLIB4A = '$SSW/packages/sunspice/idl/' / Software library containing PRPROC4
PRVER4A =                60921 / Mod. julday of PRLIB4A (2025-09-03)

PRSTEP5 = 'SPATIAL-COORDINATE-CORRECTION' / Type of processing, step 5
PRPROC5 = 'correct_spice_offset_relative_to_spacecraft.pro' / Name of procedure,
PRPVER5 = '3.0     '           / Version of procedure, step 5
PRREF5  = 'delta_instrument_x = -79,                                          &'
CONTINUE  'delta_instrument_y = -45                                           &'
CONTINUE  '' / [arcsec] constant SPICE vs spacecraft offset
PRLIB5A = 'uio-spice-pipeline' / Software library containing PRPROC5
PRVER5A = '6093    '           / UiO SVN revision number of PRLIB5A (2026-03-23)

PRSTEP6 = 'DARK-SUBTRACTION'   / Type of processing, step 6
PRPROC6 = 'spice_prep_dark_offset_correction.pro' / Name of procedure, step 6
PRPVER6 = '3.4     '           / Version of procedure, step 6
PRREF6  = 'dark = "combined_dark_60s_ID1_V01",                                &'
CONTINUE  'constituent_darks = "218103847-000,                                &'
CONTINUE  '218103847-001,                                                     &'
CONTINUE  '218103847-002,                                                     &'
CONTINUE  '218103847-003,                                                     &'
CONTINUE  '218103847-004,                                                     &'
CONTINUE  '218103847-005,                                                     &'
CONTINUE  '218103847-006",                                                    &'
CONTINUE  'constituent_darks_date = "2023-09-29",                             &'
CONTINUE  'combined_dark_id_table_version="v04"                               &'
CONTINUE  '' / Additional info, step 6
PRLIB6A = 'uio-spice-pipeline/calibration' / Software library containing PRPROC6
PRVER6A = '6135    '           / Version of software library (2026-03-30)

PRSTEP7 = 'DESPIKING'          / Type of processing, step 7
PRPROC7 = 'spice_sigma_clip2.py' / Name of procedure, step 7
PRPVER7 = '2025-11-10T20:35'   / Version of procedure, step 7
PRREF7  = 'n_sigma_clip_targets: 2,                                           &'
CONTINUE  'sigma_clip_target_1: "Lambda-Y min intensity projection (MinIP)",  &'
CONTINUE  'sigma_clip_target_2: "Lamda-Y-exp data cube"                       &'
CONTINUE  '' / Additional info, step 7
PRPARA7 = '[minIP] size=3,                                                    &'
CONTINUE  '[minIP] sigma_upper=8,                                             &'
CONTINUE  '[minIP] sigma_lower=8,                                             &'
CONTINUE  '[minIP] centerfunc="median",                                       &'
CONTINUE  '[minIP] exclude_center=1,                                          &'
CONTINUE  'size=3,                                                            &'
CONTINUE  'sigma_upper=3,                                                     &'
```

```
CONTINUE  'sigma_lower=2.5,                                                   &'
CONTINUE  'centerfunc="median",                                               &'
CONTINUE  'exclude_center=1                                                   &'
CONTINUE  '' / Parameters for PRPROC7
PRLIB7A = 'uio-spice-pipeline/python' / Software library containing PRPROC7
PRVER7A = '2026-01-16T09:08'   / Version of software library

PRSTEP8 = 'FLATFIELDING'       / Type of processing, step 8
PRPROC8 = 'spice_prep_flat_field_correction.pro' / Name of procedure, step 8
PRPVER8 = '1.6     '           / Version of procedure, step 8
PRREF8  = 'flat = "ground-calibration flat field"' / Additional info, step 8
PRLIB8A = 'uio-spice-pipeline/calibration' / Software library containing PRPROC8
PRVER8A = '6135    '           / Version of software library (2026-03-30)

PRSTEP9 = 'BURN-IN-CORRECTION' / Type of processing, step 9
PRPROC9 = 'spice_prep_burnin_correction.pro' / Name of procedure, step 9
PRPVER9 = '1.4     '           / Version of procedure, step 9
PRREF9  = 'burn_in_correction_data_version = "2026-03-23",                    &'
CONTINUE  'burn_in_correction_defined_for_this_hdu = 1' / Additional info, step
PRLIB9A = 'uio-spice-pipeline/calibration' / Software library containing PRPROC9
PRVER9A = '6135    '           / Version of software library (2026-03-30)

PRSTEP10= 'SPATIAL-SPECTRAL-DISTORTION-CORRECTION' / Type of processing, step 10
PRPROC10= 'spice_prep_distortion_correction.pro' / Name of procedure, step 10
PRPVER10= '4.1     '           / Version of procedure, step 10
PRREF10 = 'distortion_correction_matrix_version = "2024-07-04",               &'
CONTINUE  'lambda_vs_temperature_correction_version = "2024-04-18"            &'
CONTINUE  '' / Additional info, step 10
PRLIB10A= 'uio-spice-pipeline/calibration' / Software library containing PRPROC1
PRVER10A= '6135    '           / Version of software library (2026-03-30)

PRSTEP11= 'RADIOMETRIC-CALIBRATION' / Type of processing, step 11
PRPROC11= 'spice_prep_radiometric_calibration.pro' / Name of procedure, step 11
PRPVER11= '3.0     '           / Version of procedure, step 11
PRREF11 = 'method = "based on comparison to QS SUMER spectrum",               &'
CONTINUE  'response_file_version = "2024-08-28",                              &'
CONTINUE  'interpolated_sw_response = 0.39174                                 &'
CONTINUE  '' / Additional info, step 11
PRLIB11A= 'uio-spice-pipeline/calibration' / Software library containing PRPROC1
PRVER11A= '6135    '           / Version of software library (2026-03-30)

PRSTEP12= 'WINDOW-CONCATENATION' / Type of processing, step 12
PRPROC12= 'spice_prep.pro'     / Name of procedure, step 12
PRPVER12= '6093    '           / Version of procedure, step 12
PRREF12 = '   WINNOs_of_concatenated_windows = [3,4],                         &'
CONTINUE  'WINTABIDs_of_concatenated_windows = [136,148],                     &'
CONTINUE  ' MISOWINs_of_concatenated_windows = [3222,3223],                   &'
CONTINUE  ' EXTNAMEs_of_concatenated_windows = "Ne VIII 770 / Mg VIII 772 - SH&'
CONTINUE  ',Ne VIII 770 / Mg VIII 772 - LH"                                   &'
CONTINUE  '' / Additional info, step 12
PRLIB12A= 'uio-spice-pipeline/calibration' / Software library containing PRPROC1
PRVER12A= '6135    '           / Version of software library (2026-03-30)

PRENV2  = '  Kernel: Linux                                                    &'
CONTINUE  '  Kernel release number: 5.14.0-503.40.1.el9_5.x86_64              &'
CONTINUE  '  Architecture: x86_64                                             &'
```

```
CONTINUE  '  Host name: pleiades13.uio.no                                    &'
CONTINUE  '  OS: Red Hat Enterprise Linux 9.5 (Plow)                         &'
CONTINUE  '  CPU: AMD EPYC 7543 32-Core Processor                            &'
CONTINUE  '  IDL 9.2.0 (Jul 21 2025 (481746)), memory bits: 64, file offset bi&'
CONTINUE  'ts: 64      ' / Hardware and software
```

A series of repeated full-detector observations are concatenated into a single L2 file. This is described by the ‘`FILE-CONCATENATION`’ processing step, e.g.:

```
PRSTEP12= 'FILE-CONCATENATION' / Type of processing, step 12
PRPROC12= 'spice_concatenate_repeated_l2_exp_files.pro' / Name of procedure, ste
PRPVER12= '5452    '            / Version of procedure, step 12 (2025-10-24)
PRREF12 = 'constituent_L2_exp_files =                                     &'
CONTINUE  '"solo_L2_spice-n-exp_20260224T134617_V01_369099132-*-DR6.fits",    &'
CONTINUE  'n_constituent_L2_exp_files: 30,                                 &'
CONTINUE  'n_constituent_L2_exp_files_missing: 0                           &'
CONTINUE  '' / Additional info, step 12
PRLIB12A= 'uio-spice-pipeline' / SW library containing PRPROC12
PRVER12A= '5895    '            / Version of software library (2026-02-25)
```

4.4.2.2.1 `PRPARAn` and `PRREFn` keywords

The `PRPARAn` keywords describe the input parameters to the processing routine performing the processing step.

The `PRREFn` keywords give other relevant information relating to the processing step, e.g. dates of input files, the file name of the dark map, etc. The format of a `PRREFn` keyword is a comma separated list of pairs of parameter names and parameter values. The parameter names will vary depending on what kind of processing step they are describing. As an example, if a combined dark ([MULTIDARK]) was subtracted, the corresponding `PRREFn` keyword contains a comma separated list with the parameter ‘`dark`’ followed by the name of the dark, the parameter ‘`constituent_darks`’ followed by the `SPIOBSID-RASTERNO`s of the darks that were used to create the combined dark, and the parameter ‘`constituent_darks_date`’ followed by the observation date of these darks:

```
PRREF6   = 'dark = "combined_dark_20s_ID1_V01",                          &'
CONTINUE  'constituent_darks = "218103845-000,                           &'
CONTINUE  '218103845-001,                                                &'
CONTINUE  '218103845-002,                                                &'
CONTINUE  '218103845-003,                                                &'
CONTINUE  '218103845-004,                                                &'
CONTINUE  '218103845-005,                                                &'
CONTINUE  '218103845-006",                                               &'
CONTINUE  'constituent_darks_date = "2023-09-29"                         &'
CONTINUE  '' / Additional info, step 6
```

Note that all observational HDUs of L2 files will include a description of the burn-in correction processing step. However, the burn-in data file contains burn-in corrections only for sections of the detectors where strong lines have significantly reduced the detector’s sensitivity. Whether the burn-in data file contains information that allowed for a correction of the HDU in question is given by the `PRREFn` parameter `burn_in_correction_defined_for_this_line`. In the header example above the value of this parameter is `1`, i.e. this line has been corrected for burn-in. Finally, if one or more windows that make up a merged window are corrected for burn-in, the merged window is also marked as corrected for burn-in.

The sensitivity of the SPICE detectors decreases with time. The radiometric calibration includes a correction for this time-dependent responsivity. From a table of estimated response values, we interpolate a single response correction value for each detector for the `DATE-AVG` of the observation. This factor is given by the `PRREFn` parameter `interpolated_lw_response`.

4.4.3 Level 3 Data arrays and FITS headers

The pipeline is as of April 2026 creating simplified L3 P FITS files containing Gaussian fit parameters (see [IDLANA]). L3 P FITS files are created from narrow-slit raster and sit-and-stare observations (i.e. windowed spectral-profile HDUs, see Table 4-1 and Table 4-2). Each observational HDU in the input L2 file leads to two L3 P HDUs. The first HDU contains the Gaussian fit parameters as described in Appendix IX of [S-META]. The second HDU contains the header of the parent L2 HDU.

The L3 P file contains a reference to the parent L2 file. The user should store the L2 and L3 P files in the file structure recommended by the UiO team, e.g. by downloading SPICE data using the IDL routine `spice_wget_files`. Analysis tools will then be able to restore the data array that was used for the line fitting. This enables the user to manually modify the automatically determined initial values of the fit and re-run the line fitting with the code of the user's choice. The line fitting routine must use the parameterisation used by `cfit.pro` (see [CFIT]) but may of course be written in other programming languages than IDL. However, for IDL users manipulating L3 files is especially convenient:

- use `spice_xfiles` to select the file you want to analyse
- click the "copy window to user file" button in the "LEVEL 3 – official" column
- click the "View/Edit window" button in the "LEVEL 3 – user" column to start `xcfit_block`

To be implemented: based on the fitted Gaussian line parameters, secondary derived parameters can be estimated:

- Abundances
- FIP bias
- Density
- Temperature

The format of L3 files including secondary derived parameters is not yet finalised.

4.4.3.1 Keywords describing individual L2 to L3 processing steps

As outlined in Section 4.4.2.2 the individual processing steps are described using the `PRxxxxn` keywords. When processing L2 files to L3 additional processing steps are described: finding the peak(s) in the spectrum to which Gaussian fits were made, and the actual line fitting:

```
PRSTEP11= 'PEAK-FINDING'        /  Processing step type
PRPROC11= 'spice_gt_peaks'      /  Name of procedure performing PRSTEP11
PRPVER11=                     1 /  Version of procedure PRPROC11
PRLIB11A= 'solarsoft/so/spice/idl/quicklook' /  Software library containing PRPR

PRSTEP12= 'LINE-FITTING'        /  Processing step type
PRPROC12= 'spice_data::create_l3_file' /  Name of procedure performing PRSTEP12
PRPVER12=                     5 /  Version of procedure PRPROC12
PRPARA12= 'LINE_LIST = 0,                                                     &'
CONTINUE  'MASKING   = 1,                                                     &'
CONTINUE  'FITTING   = 1,                                                     &'
```

```
CONTINUE  'POSITION  = 0,                                                      &'
CONTINUE  'VELOCITY  = -999,                                                   &'
CONTINUE  'POSSIBLE_MANUAL_EDITING = 0' /  Parameters for PRPROC12
PRLIB12A= 'solarsoft/so/spice/idl/quicklook' /  Software library containing PRPR
```

In future L3 files, the calculation of secondary derived data products will be described by e.g.:

```
PRSTEP9 = 'ABUNDANCE-ESTIMATION'   / Processing step type 9
PRPVER9 =                     0.32 / Version of procedure PRPROC9
PRPROC9 = 'spice_abundances.pro'   / Name of procedure performing PRSTEP9

PRSTEP10= 'FIP-BIAS-ESTIMATION'    / Processing step type 10
PRPVER10=                      1.0 / Version of procedure PRPROC10
PRPROC10= 'spice_fip_bias.pro'     / Name of procedure performing PRSTEP10

PRSTEP11= 'DENSITY-ESTIMATION'     / Processing step type 11
PRPVER11=                        2 / Version of procedure PRPROC11
PRPROC11= 'spice_densities.pro'    / Name of procedure performing PRSTEP11

PRSTEP12= 'TEMPERATURE-ESTIMATION'/ Processing step type 12
PRPVER12=                      0.1 / Version of procedure PRPROC12
PRPROC12= 'spice_temperatures.pro'/ Name of procedure performing PRSTEP12
```

4.4.4 Storing variable keyword values in binary table extensions

Some SPICE FITS keywords may have multiple values. Such keywords have either:

- one value per exposure: acquisition times, mirror positions, and temperatures, see Section 4.4.4.1, or
- one value per segment: voltages, see Section 4.4.4.2, or
- one value per dispersion pixel: the radiometric calibration factor that converts the intensities in Level 2 files to counts, see Section 4.4.4.3, or
- L1 only: values that are not directly linked to individual exposures or other dimensions of the data cube: lost telemetry packet indices and lost FFT Bin indices, see Section 4.4.4.4.

In all cases we use the *variable-keyword mechanism* outlined in Appendix I of [S-META] (see http://sdc.uio.no/open/solarnet/ for the latest version) to store the individual values of variable keywords in binary table extensions.

4.4.4.1 Variable keywords with one value per exposure: times, mirror positions, and temperatures (L1 and L2)

The acquisition time (OBT), scan mirror position and 4 instrument temperatures are recorded for each exposure of a SPICE observation. For single exposure observations, the values of these measurements are stored in the FITS keywords `TIMAQOBT`, `MIRRPOS`, `TN_FOCUS`, `TN_MIRR`, `TN_SW`, and `TN_LW`, and the temperatures converted from data numbers to degrees Celsius in `T_FOCUS`, `T_MIRR`, `T_SW`, and `T_LW`. In L1+ FITS files, `TIMAQUTC` stores the acquisition time converted to UTC. For multi exposure observations, these keywords hold the *average* values, and the *individual values* of each keyword, i.e. one value per exposure, are stored in a binary table extension using the variable-keyword mechanism.

In the header of a SPICE observational HDU that uses the variable-keyword mechanism, the `VAR_KEYS` keyword always has the following value:

```
VAR_KEYS= 'VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_MIRR,TN_SW,TN_LW',&'
```

```
CONTINUE  'T_FOCUS, T_MIRR, T_SW, T_LW, TIMAQUTC'/ Variable keywords
```

This means that in the binary table extension `VARIABLE_KEYWORDS`, the individual values of the keywords `TIMAQOBT`, `MIRRPOS`, `TN_xxxxx`, `T_xxxxx`, and `TIMAQUTC` are stored in columns with `TTYPEn` equal to the keyword names. Following Appendix I-d of [S-META], the binary table columns and the referring HDU are to be associated pixel-by-pixel, since the `WCSNn` keywords are set to '`PIXEL-TO-PIXEL`'. The dimensions of the value columns, given by `TDIMn`, are `(1,1,1,NAXIS4)`, the singular dimensions signalling that there is one value per exposure that is valid for all `(x,y,d)` pixels for that exposure.

Below is an excerpt of such a binary table extension header, including column specific keywords that define the columns storing the individual `TIMAQOBT` and `TN_FOCUS` values.

```
XTENSION= 'BINTABLE'           / Written by IDL:  Mon Sep 25 12:03:41 2017
:
EXTNAME = 'VARIABLE_KEYWORDS'  / Extension name
:
WCSN1   = 'PIXEL-TO-PIXEL'     / Value column/referring HDU association type
TFORM1  = '64D     '           / Real*8 (double precision)
TTYPE1  = 'TIMAQOBT'           / [OBT] Start time of data acquisition
TDIM1   = '(1,1,1,64)'         / Array dimensions for column 1
TUNIT1  = '        '           / Units of column 1
TDMIN1  =        481295089.350 / Minimum value in column 1
TDMAX1  =        481295146.051 / Maximum value in column 1
TDESC1  = 'Variable values for TIMAQOBT' / Axis labels for column 1
:
WCSN3   = 'PIXEL-TO-PIXEL'     / Value column/referring HDU association type
TFORM3  = '64I     '           / Unsigned Integer*2 (short integer)
TTYPE3  = 'TN_FOCUS'           / [adu] SFM focus adu temperature
TDIM3   = '(1,1,1,64)'         / Array dimensions for column 3
TUNIT1  = '        '           / Units of column 3
TSCAL3  =                    1 / Scale parameter for column 3
TZERO3  =                32768 / Zero offset for column 3
TDMIN3  =                  846 / Minimum value in column 3
TDMAX3  =                  871 / Maximum value in column 3
TDESC3  = 'Variable values for TN_FOCUS' / Axis labels for column 3
```

4.4.4.2 Variable keywords with one value per segment: voltages (L1 and L2)

4 instrument voltages per segment are downlinked in the science telemetry. For single-segment observations the raw values of these measurements are stored in the FITS keywords `VN_MCPSW`, `VN_MCPLW`, `VN_GAPSW`, and `VN_GAPLW`, and the voltages converted from data numbers to Volt in `V_MCPSW`, `V_MCPLW`, `V_GAPSW`, and `V_GAPLW`. For multi-segment observations these keywords hold the *average* values, and the *individual* values for each keyword, i.e. one value per segment, are stored in the same binary table extension described in the previous Section.

In the header of a SPICE observational HDU stemming from a multi-segment observation, the `VAR_KEYS` keyword has the following value:

```
VAR_KEYS= 'VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_GRAT,TN_SW,TN_LW,T_F&'
CONTINUE  'OCUS,T_GRAT,T_SW,T_LW,VN_MCPSW,VN_MCPLW,VN_GAPSW,VN_GAPLW,V_MCPSW,V&'
CONTINUE  '_MCPLW,V_GAPSW,V_GAPLW,TIMAQUTC' / Variable keywords
```

Below is an excerpt of the header of the 'VARIABLE_KEYWORDS' binary table extension, including column specific keywords that define the column storing the individual VN_MCPSW values, one value for each segment.

```
XTENSION= 'BINTABLE'           / Written by IDL:  Fri Oct 25 13:47:08 2019
:
EXTNAME = 'VARIABLE_KEYWORDS'  / Extension name
:
           -------------------------------
           | Column 12 specific keywords |
           -------------------------------
WCSN12  = 'PIXEL-TO-PIXEL'     / Value column/referring HDU association type
TFORM12 = '4I       '          / Unsigned Integer*2 (short integer)
TTYPE12 = 'VN_MCPLW'           / [adu] MCP LW voltage
TDIM12  = '(1,1,1,4)'          / Array dimensions for column 12
TUNIT12 = '        '           / Units of column 12
TSCAL12 =                    1 / Scale parameter for column 12
TZERO12 =                32768 / Zero offset for column 12
TDMIN12 =                  553 / Minimum value in column 12
TDMAX12 =                  553 / Maximum value in column 12
TDESC12 = 'Variable values for VN_MCPLW' / Axis labels for column 12
```

4.4.4.3 Variable keyword with one value per lambda pixel: radiometric calibration factor (L2)

In the Level 1 to Level 2 calibration the photon counts are converted to physical intensity units. To get back the counts from the Level 2 intensity the variable keyword RADCAL should be applied to the Level 2 data.

In the header of a Level 2 observational HDU the RADCAL keyword gives the mean radiometric conversion factor of the window. The individual values of the conversion factor are stored in the binary table extension VARIABLE_KEYWORDS. To retrieve the counts the Level 2 data array is multiplied with the conversion factor array values:

```
FOR lam=0,n_lam-1 DO counts[*,*,lam,*] = L2_int[*,*,lam,*]*radcal_array[lam]
```

The binary table column storing the individual RADCAL values for a given observational HDU is named RADCAL followed by a tag enclosed by square brackets. This tag equals the EXTNAME of the observational HDU, possibly shortened to make the length of the string RADCAL[tag] shorter than 68 characters.

As an example, the value of the VAR_KEYS keyword of an observational HDU may be:

```
VAR_KEYS='VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_GRAT,TN_SW,TN_LW,T_F&'
CONTINUE'OCUS,T_GRAT,T_SW,T_LW,VN_MCPSW,VN_MCPLW,VN_GAPSW,VN_GAPLW,V_MCPSW,V&'
CONTINUE  '_MCPLW,V_GAPSW,V_GAPLW,TIMAQUTC,CRDER1,CRDER2,RADCAL[O III 703 / Mg&'
CONTINUE  ' IX 706 (Merged)]
```

The relevant part of the header of the binary table extension 'VARIABLE_KEYWORDS' may then be:

```
WCSN22  = 'PIXEL-TO-PIXEL'     / Value column/referring HDU association type
TFORM22 = '80D     '           / Real*8 (double precision)
TTYPE22 = 'RADCAL[O III 703 / Mg IX 706 (Merged)]' / [DN/(W/m2/sr/nm)] Calibrati
TDIM22  = '(1,1,80,1)'         / Array dimensions for column 22
TUNIT22 = 'DN/(W/m2/sr/nm)]'   / Units of column 22
TDMIN22 =        335.757886613 / Minimum value in column 22
```

```
TDMAX22 =        367.519855635 / Maximum value in column 22
TDESC22 = 'Variable values for RADCAL' / Axis labels for column 2
```

4.4.4.4 Variable keywords describing lost telemetry packets and lost FFT Bins (L1)

In L1 files, if telemetry packets containing observational data are lost the packet indices (starting at 0 for the first data packet) are stored in the binary table extension with name `'LOST_TELEMETRY'`. Note that this extension is not present in L2 and L3 FITS files. Instead, the user should refer to the keywords described in Sections 4.4.1.3.5.1 and 4.4.1.3.5.2 for an overview of the telemetry completeness.

The name of the binary table column storing the indices of the lost packets is `'LOSTPKTS'`, followed by a tag giving the `EXTNAME` of the observational HDU that uses the variable-keyword mechanism (the referring HDU). The lost packet indices are stored primarily for pipeline debugging purposes.

Below is an excerpt of the header of a `'LOST_TELEMETRY'` binary table extension of an L1 file, including keywords that define the column storing the individual lost packet indices:

```
XTENSION= 'BINTABLE'            / Written by IDL:  Mon Sep  9 11:08:01 2019
:
EXTNAME = 'LOST_TELEMETRY'      / Extension name
:
TFORM1  = '3J      '            / Integer*4 (long integer)
TTYPE1  = 'LOSTPKTS[WINDOW0_724.05A]' / Lost packets w/data, variable keyword
TUNIT1  = '        '            / Units of column 1
TDMIN1  =                     1 / Minimum value in column 1
TDMAX1  =                    28 / Maximum value in column 1
TDESC1  = 'Indices of lost packets containing observational data' / Axis labels
```

The value of the `VAR_KEYS` keyword of the referring HDU contains the binary table extension name and the column name:

```
VAR_KEYS= 'VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_GRAT,TN_SW,TN_LW,T_F&'
CONTINUE  'OCUS,T_GRAT,T_SW,T_LW,TIMAQUTC,                                     &'
CONTINUE  'LOST_TELEMETRY;LOSTPKTS[WINDOW0_724.05A]' / Variable keywords
```

As described in [SPICELOST] the implications of lost telemetry packets in the case of SHC data highly depends on how many and which FFT Bins are lost. This information is therefore stored in the FITS files to help the advanced user to determine the degree of degradation of the approximated data cube. An additional binary table column storing the lost FFT Bin indices is added to the `'LOST_TELEMETRY'` binary table extension. The name of this column is `'LOSTBINS'` plus a tag with the `EXTNAME` of the referring HDU.

L1 files may contain data cubes that are reconstructed from incomplete SHC telemetry, but in the L2 and L3 files produced by the Science Data Pipeline all pixels in such cubes are set to `NaN`. However, the user may choose to keep incomplete data cubes when running the L1 to L2 calibration routines manually and use the information in the `LOSTBINS` column when interpreting the data.

Below is an excerpt of the header of a `'LOST_TELEMETRY'` binary table extension, including keywords that define the column storing the individual lost FFT Bin indices:

```
XTENSION= 'BINTABLE'           / Written by IDL:  Mon Sep  9 11:08:01 2019
:
EXTNAME = 'LOST_TELEMETRY'     / Extension name
:
TFORM2  = '8I      '           / Integer*2 (short integer)
TTYPE2  = 'LOSTBINS[WINDOW0_724.05A]' / Lost FFT bins, variable keyword
TUNIT2  = '        '           / Units of column 2
TDMIN2  =                    0 / Minimum value in column 2
TDMAX2  =                   31 / Maximum value in column 2
TDESC2  = 'Indices of lost FFT Bins' / Axis labels for column 2
```

The `VAR_KEYS` keyword of the referring HDU contains the binary table extension name and the column name:

```
VAR_KEYS= 'VARIABLE_KEYWORDS;TIMAQOBT,MIRRPOS,TN_FOCUS,TN_GRAT,TN_SW,TN_LW,T_F&'
CONTINUE  'OCUS,T_GRAT,T_SW,T_LW,TIMAQUTC,                                   &'
CONTINUE  'LOST_TELEMETRY;LOSTPKTS[WINDOW0_724.05A],LOSTBINS[WINDOW0_724.05A]&'
/
CONTINUE  '' / Variable keywords
```

The referring HDU contains representative scalar values of variable keyword (see Appendix I of [S-META]). The representative scalar values of `LOSTPKTS` and `LOSTBINS` are the number of lost telemetry packets and the number of lost FFT Bins respectively:

```
LOSTPKTS =                   3 / Number of lost packets w/data, variable keyword
LOSTBINS =                   8 / Number of lost FFT bins, variable keyword
```

4.4.5 Flagging image planes with approximated or missing values due to lost compressed telemetry using binary table pixel list (L1)

As described in [SPICELOST] it is possible to reconstruct a data cube in the case of lost telemetry, even if the data is compressed. However, one or more X-Y or dispersion-Y image planes of the data cube may in such cases be lost or have approximated values, depending on the file level, which kind of compression was applied, and which telemetry packets were lost.

If a lost telemetry packet contained JPEG header information it is not possible to decompress the JPEG image, and the entire image plane is lost. Decompression is possible if a missing telemetry packet did *not* contain JPEG header information, but the resulting values will be approximated. In such cases L1 files contain the approximated values, but in L2 and L3 files all pixels affected by telemetry loss are set to `NaN`.

If compressed telemetry packets are missing from a multi-segment observation, then *ranges* of image planes may be set to `NaN` or have approximated values (L1 only). Each image plane range corresponds to an image plane of a single segment. Flagging of image plane ranges may also be applied for compressed full-detector readouts. For such observations each detector array is split into 16 Lambda-Y regions of 64x1024 pixels that are JPEG compressed on-board separately.

Note that for technical reasons the flagging of lost image plane ranges in L1 files described below currently does *not* work if *all* packets of a segment are missing. However, the L1+ FITS keywords mentioned in Sections 4.4.1.3.5.1 and 4.4.1.3.5.2 are correctly set even when entire segments are missing.

In L1 files, we use the pixel list mechanism described in Appendix II-a of [S-META] to flag image planes ranges that have approximated values due to missing compressed telemetry.

For every observational HDU that uses the pixel list mechanism there is a corresponding binary table extension containing a single pixel list where the vertices of approximated or lost image plane ranges are stored. The names of these binary table extensions are `'APRXPLNPIXLIST'` and `'LOSTPLNPIXLIST'` respectively, plus a tag with the `EXTNAME` of the referring HDU. In L2 and L3 files, pixels corresponding to L1 Image plane ranges that are defined in an `'APRXPLNPIXLIST'` extension are set to `NaN`. In the future the advanced user will be able to run the Level 1 to Level 2 calibration routines manually with an option to retain approximated pixel values in L2 files.

In the header of the L1 observational HDU the names of the any pixel list binary table extensions are given by `PIXLISTS`. `NLOSTPLN` and `NAPRXPLN` give the total number of lost or approximated image plane ranges in the L1 data cube.

As an example, we consider an actual full-detector JPEG compressed observation obtained in March 2021. A telemetry packet belonging to the third Lambda-Y JPEG of the Long Wavelength detector never made it to the ground, leading to approximated pixel values. The corresponding range of the full LW Lambda-Y image plane is flagged by the pixel list binary table by the following two rows:

	DIMENSION1	DIMENSION2	DIMENSION3	DIMENSION4	PIXTYPE
Row 1	1	1	65	1	1
Row 2	1	1024	128	1	2

An excerpt of the header of the L1 pixel list binary table extension looks like this:

```
XTENSION= 'BINTABLE'            / Written by IDL:  Wed Sep 29 13:20:51 2021
…
TFIELDS =                     5 / Number of columns
EXTNAME = 'APRXPLNPIXLIST[Full LW 4:1 Focal Lossy]' / Extension name


           ------------------------------
           | Column 1 specific keywords |
           ------------------------------
TFORM1   = '1I      '            / Integer*2 (short integer)
TTYPE1   = 'DIMENSION1'          / Pixel indices dimension 1
TCTYP1   = 'PIXEL   '            / Indicates that column 1 is a pixel index
TPC1_1   =                     1 / Indicates that column 1 is a pixel index
TUNIT1   = '        '            / Units of column 1
TDMIN1   =                     1 / Minimum value in column 1
TDMAX1   =                     1 / Maximum value in column 1
TDESC1   = 'Lower Left/Upper Right pixel indices of 1 approximated Lambda-Y ima&'
CONTINUE  'ge plane ranges due to loss of compressed telemetry packets&' / Axis
CONTINUE  '' / labels for column 1
```

```
           ------------------------------
           | Column 2 specific keywords |
           ------------------------------
TFORM2   = '1I      '           / Integer*2 (short integer)
TTYPE2   = 'DIMENSION2'         / Pixel indices dimension 2
TCTYP2   = 'PIXEL   '           / Indicates that column 2 is a pixel index
TPC2_2   =                    1 / Indicates that column 2 is a pixel index
TUNIT2   = '        '           / Units of column 2
TDMIN2   =                    1 / Minimum value in column 2
TDMAX2   =                 1024 / Maximum value in column 2
TDESC2   = 'Lower Left/Upper Right pixel indices of 1 approximated Lambda-Y ima&'
CONTINUE  'ge plane ranges due to loss of compressed telemetry packets&' / Axis
CONTINUE  '' / labels for column 2


           ------------------------------
           | Column 3 specific keywords |
           ------------------------------
TFORM3   = '1I      '           / Integer*2 (short integer)
TTYPE3   = 'DIMENSION3'         / Pixel indices dimension 3
TCTYP3   = 'PIXEL   '           / Indicates that column 3 is a pixel index
TPC3_3   =                    1 / Indicates that column 3 is a pixel index
TUNIT3   = '        '           / Units of column 3
TDMIN3   =                   65 / Minimum value in column 3
TDMAX3   =                  128 / Maximum value in column 3
TDESC3   = 'Lower Left/Upper Right pixel indices of 1 approximated Lambda-Y ima&'
CONTINUE  'ge plane ranges due to loss of compressed telemetry packets&' / Axis
CONTINUE  '' / labels for column 3


           ------------------------------
           | Column 4 specific keywords |
           ------------------------------
TFORM4   = '1I      '           / Integer*2 (short integer)
TTYPE4   = 'DIMENSION4'         / Pixel indices dimension 4
TCTYP4   = 'PIXEL   '           / Indicates that column 4 is a pixel index
TPC4_4   =                    1 / Indicates that column 4 is a pixel index
TUNIT4   = '        '           / Units of column 4
TDMIN4   =                    1 / Minimum value in column 4
TDMAX4   =                    1 / Maximum value in column 4
TDESC4   = 'Lower Left/Upper Right pixel indices of 1 approximated Lambda-Y ima&'
CONTINUE  'ge plane ranges due to loss of compressed telemetry packets&' / Axis
CONTINUE  '' / labels for column 4


           ------------------------------
           | Column 5 specific keywords |
           ------------------------------
TFORM5   = '1I      '           / Integer*2 (short integer)
TTYPE5   = 'PIXTYPE '           / Pixel type
TUNIT5   = '        '           / Units of column 5
TDMIN5   =                    1 / Minimum value in column 5
TDMAX5   =                    2 / Maximum value in column 5
TDESC5   = 'Pixel indices types: 1 = lower left corner indices, 2 = upper right&'
CONTINUE  ' corner indices' / Axis labels for column 5
```

```
           ------------------------------
           | File and study identifiers |
           ------------------------------
FILENAME= 'solo_L2_spice-n-exp_20210322T033925_V26_50331861-018-DR6.fits' /
Filename
…
```

Note that for SHC data *all* image planes stemming from a single segment are approximated if telemetry packets with observational data are lost. The amount of approximation may be estimated using the variable L1 keyword `LOSTBINS` described in Section 4.4.4.4.

APPENDIX A SPICE DATA PRODUCTS MATRIX

A FITS file may contain different types of HDUs/data products. Each HDU type may occur in files with several different combinations of descriptor elements.

L1 and L2:
Table 4-11 lists all possible L1 and L2 data products, and the file descriptors that a FITS file containing such a product may have. The overwhelming majority of Science observations recorded the first 3 years of the mission are `n-ras`, with a few `n-sit` and fewer still `w-sit`. For all observation types including calibration observations, `n-exp` is the most common descriptor. The usage of `-int` and `-db` is negligible. See Table 4-2 for actual usage of the different data products.

Data Product (HDU type)	Description	Descriptors	Avg cadence	Expected Daily Vol
Narrow-slit spectral-profile raster		n-ras[-db][-int]		
Narrow-slit intensity-window raster		n-ras[-db]-int		
Wide-slit raster		w-ras		
Dumbbell raster		n-ras-db[-int]		
Narrow-slit spectral-profile sit-and-stare		n-sit[-db]		
Wide-slit sit-and-stare		w-sit		
Dumbbell sit-and-stare		n-sit-db		
Narrow-slit full detector single exposure		n-exp		
Wide-slit full detector single exposure		w-exp		

Table 4-11: SPICE L1 and L2 Data Products Matrix

L3 P:
In Table 4-12 we have listed all possible L3 P data products, and the file descriptors that a FITS file containing such a product may have.

Data Product (HDU type)	Description	Descriptors	Avg cadence	Expected Daily Vol
Narrow-slit spectral-profile raster		n-ras[-db][-int]		
Narrow-slit spectral-profile sit-and-stare		n-sit[-db]		

Table 4-12: SPICE L3 P Data Products Matrix

APPENDIX B RELEATIONSHIP BETWEEN SCIENCE HEADER PACKET PARAMETERS AND FITS KEYWORDS

Section 4.2.6.1 of [DATAICD] lists the Science Header Packet parameters that describe the collection of Science Data Packets that builds up a raster segment. These parameters may apply to all windows of an observation, a single window, a single raster segment, or the parameters may have one value per exposure. The values of these parameters are stored as FITS keywords, see Table 4-13. Note that the "StudyFlags" parameter is an 8-bit integer with the value of each bit indicating different instrument settings, see Table 4-14. Both the primary HDU and all image extensions store all the FITS keywords described in the table. The shaded table rows indicate parameters/keywords with one value for each exposure. The mean values of these keywords are stored in the primary HDU and all image extensions, and the individual values for each exposure are stored in a binary table extension, see Section 4.4.4.

Science Header Packet Parameters	FITS keywords
Total number of CCSDS packets for this window	NPACKETS -1 or -2 (NPACKETS is the number of packets with observational data, excluding the Header Packet and the Final Packet if the latter only contains a checksum)
Observation ID	SPIOBSID
Focus position	FOCUSPOS
Slit position	SLIT_ID
Exposure time	XPOSURE * 10
Study ID	STUDY_ID
StudyFlags	STUDYFLG
Total Raster Segments per Window	NSEGMENT
ObsRasterNumber	RASTERNO
Raster Segment Number	-
Window Total Number	NWIN
Window Number	WINNO
Window Data Table ID	WINTABID
Window Start Column	PXBEG3
Window Width	PXEND3 – PXBEG3 + 1
Wavelength Binning Factor	NBIN3
Window Starting Row	PXBEG2
Window Height	PXEND2 – PXBEG2 + 1
Y Binning Factor	NBIN2
Compression Type	COMPTYPE
Compression Amount Parameter	COMPPARA
SHC FFT ID	SHCFFTID
Pixel Level Offset	PIXELOFF
Alignment Window Status	DUMBBELL
MCP SW Monitor Voltage	VN_MCPSW (and V_MCPSW)
MCP LW Monitor Voltage	VN_MCPLW (and V_MCPLW)
Gap SW Monitor Voltage	VN_GAPSW (and V_GAPSW)

Science Header Packet Parameters	FITS keywords
Gap LW Monitor Voltage	VN_GAPLW (and V_GAPLW)
Segment X Size	–
Acquisition Time when initiating black level reset after arriving at X	TIMAQOBT
Scan Mirror Position	MIRRPOS
Temperature 1	TN_FOCUS (and T_FOCUS)
Temperature 2	TN_GRAT (and T_GRAT)
Temperature 3	TN_SW (and T_SW)
Temperature 4	TN_LW (and T_LW)

Table 4-13: Telemetry Science Header Packet Parameters and their keyword equivalents. Note that the two parameters that describe a single raster segment do not have FITS keyword equivalents. Keywords in red are Solar Orbiter-wide FITS keywords. The values of the individual bits of STUDYFLG determine the values of the 6 derived keywords in Table 4-14. The voltages and temperatures are given in engineering units in the VN_xxxxx and TN_xxxxx keywords, and are converted to Volt and Celsius in the V_xxxxx and T_xxxxx keywords. Orange shading indicates parameters with one value per segment; blue shading indicates parameters with one value per exposure. In L1+ FITS files the onboard time TIMAQOBT is converted to UTC and given in TIMAQUTC (this keyword is not present L0 FITS files).

Bit	Study Flag Description	FITS Keywords
0 -1	*Spare*	–
2	AlignExcludeSpectral – 0 = Do not exclude spectral window data – 1 = Exclude spectral window data	NOSPECTR
3	Cal Mode Config – 0 = 1 Data Plane – 1 = 2 Data Planes	CALMODE
4	Double Exposure Number – 0 = First Exposure – 1 = Second Exposure	DBLEXPNO
5	Double Exposure Enabled – 0 = False – 1 = True	DBLEXP
6	Dark Map Subtraction Used? – 0 = False – 1 = True	DARKMAP
7	Black Level Subtraction Used – 0 = False – 1 = True	BLACKLEV

Table 4-14: FITS keywords derived from the value of STUDYFLG. All FITS keywords have values 0 or 1.

APPENDIX C RELEATIONSHIP BETWEEN STUDY SETS, IORS AND FITS KEYWORDS

IAS provide Study Set files containing the definitions of all on-board studies, and IORs[12] containing the commanded parameters of each observation (i.e. each instance of a study). These files are used by the Science Data Pipeline to populate L1+ FITS keywords.

For each observation the SPICE Observation ID found in the telemetry (SPIOBSID, see Appendix B) is used to find the IOR containing the information about that specific observation. The IOR contains information about which Study Set was used by the planning tool in creating the IOR (the Study Set **<version>**, FITS keyword SETVER). Having found the correct Study Set file (FITS keyword SETFILE), the STUDY_ID of the telemetry is then used to find the Study Set's definition of the study in question.

If a parameter found in the IOR or Study Set is also found in the telemetry, the telemetry value is the one used as a FITS keyword value. This is to ensure that the metadata describe the actual contents of file, and not what was commanded.

The following FITS keywords are set based on XML tags, parameters values and XML comments in the IORs:

STP – Solar Orbiter Short-Term Plan number
WINSHIFT – The number of pixels the window is shifted towards the red on the detector relative to the base position of windows with the current MISOID
OBS_ID – SOC Observation ID (not to be confused with the SPICE Observation ID, SPIOBSID)
OBS_TYPE – Unique code for OBS_MODE/STUDY, derived from OBSID
SOOPTYPE – Unique code for SOOPNAME, derived from OBSID
SOOPNAME – The name of the SOOP (given in XML comments in the IORs)
NRASTERS – The number of planned rasters for a given SPIOBSID

Table 4-15 below summarises the Study Set XML tags describing a study and their translation into FITS keywords.

XML tags	XML tag data type and values	FITS keywords	FITS keyword data type and values	In TLM
<studySet>				
<version>	0 ≤ int ≤ ?	SETVER	*Same as XML tag*	No
<studyInfo>				
<groundId>	0 ≤ int ≤ ?	MISOSTUD	*Same as XML tag*	No
<name>	String, free text	STUDY and OBS_MODE	*Same as XML tag*	No
<purpose>	String, {"Calibration", "Science", "Engineering"}	PURPOSE	*Same as XML tag*	No
<description>	String, free text	STUDYDES	*Same as XML tag*	No

[12] Instrument Operation Requests

XML tags	XML tag data type and values	FITS keywords	FITS keyword data type and values	In TLM
<author>	String, `free text`	`AUTHOR`	*Same as XML tag*	No
</studyInfo>				
<onboardInfo>				
<onboardID>	`0` ≤ int ≤ `63`	`STUDY_ID`	*Same as XML tag*	Yes
</onboardInfo>				
<type>	String, `{"Full Spectrum", "Spatial Scan", "Time Series", "Scanned Time Series"}`	A mix of `STUDYTYP` and `WIN_TYPE`. These two keywords are populated using information in the telemetry.	`STUDYTYP, {"Sit-and-stare", "Raster", "Single Exposure"}` `WIN_TYPE, {"Narrow-slit Spectral", "Dumbbell (lower)", "Dumbbell (upper)", "Wide-slit", "Intensity-window", "Full Detector Narrow-slit", "Full Detector Wide-Slit"}`	Yes, implicitly
<slit>	`1` ≤ int ≤ `4`	`SLIT_ID`	*Same as XML tag*	Yes
<exposures>	`1` ≤ int ≤ `480` [13]	`CDELT1` or `CDELT4` (when **<exposures>** eq actual number of exposures)	*Same as XML tag* (when **<exposures>** eq actual number of exposures)	Yes
<expTime>	`0.1` ≤ float ≤ `1023.5` [14]	`XPOSURE`	*Same as XML tag*	Yes
<totalExpTime>		`XPOSURE` for double exposure	*Same as XML tag*	No
<xStart>	float, default `0`	`XSTART`. Also used when calculating `CRVAL1`	*Same as XML tag*	No
<stepSize>	float, default ?	`CDELT1`	*Same as XML tag*	No
<delayTime>	float, default ?	*None*	–	No
<alignmentWindowConfig>				

[13] Any number from 1 to 64, multiples of 2 between 66 and 128, multiples of 4 between 132 and 256, multiples of 8 between 264 and 480

[14] Overheads for resetting detector pixels are added to these numbers. The overhead scales with the number of pixels in the dispersion dimension. Resolution of expTime is 0.1 s between 0.1 s and 204.7 s, 0.5 s between 205 s and 1023.5 s.

XML tags	XML tag data type and values	FITS keywords	FITS keyword data type and values	In TLM
<excludeSpectral>	boolean, default `false`	NOSPECTR	Int, {0,1}	Yes
<compressionUpper>				
<type>	String, {"Spatial Lossy", "Focal Lossy", "Spatial Uncompressed", "Focal Uncompressed", "SHC Low", "SHC High"}	COMPRESS	*Same as XML tag*	Yes, implicitly
<rawFactor>	1 ≤ Int ≤ 256?	COMP_RAT = raw data volume/compressed volume. Applies to all observations (not only JPEG compressed)	Int, {1,128}	Yes, implicitly
<shcCoef>	0 ≤ Int ≤ 7	SHCFFTID	*Same as XML tag*	Yes
</compressionUpper>				
<compressionLower>				
		See <compressionUpper>		
</compressionLower>				
</alignmentWindowConfig>				
<options>				
<readoutMode>	string, {"Destructive", "Non Destructive"}	READMODE	*Same as XML tag*	No
<calMode>	string, {"Uncompressed", "Compressed", "Both"}	CALMODE	Int, {0,1} (0 = one data plane (compressed or uncompressed), 1 = two data planes (compressed ***and*** uncompressed)	Yes
<focusCal>		*None*	–	No
<macroSteps>		*None*	–	No
<macroDelay>		*None*	–	No
<darkMap>	boolean	DARKMAP	Int, {0,1}	Yes
</options>				
<window>				
<windowInfo>				

XML tags	XML tag data type and values	FITS keywords	FITS keyword data type and values	In TLM
<description>	string	EXTNAME. For merged windows the EXTNAME is based on the EXTANMEs of the constituent windows. For dumbbells, either "DUMBBELL_UPPER_" or "DUMBBELL_LOWER_" is prepended to the description string.	*Same as XML tag*	No
<wavelength>		*None* (keywords describing wavelength, e.g. WAVEMIN, WAVEMAX) are calculated from telemetry parameters)	–	Yes. implicitly
<onboardID>	0 ≤ int ≤ 255	WINTABID	*Same as XML tag*	Yes
<groundID>	0 ≤ int ≤ ?	MISOWIN	*Same as XML tag*	No
<iwin_type>	String, {"", "Background", "Line"}	IWIN_TYPE	*Same as XML tag*	
<background_number>	0 ≤ int ≤ 255	Used to determine IWINBKG	EXTNAME of extension storing background intensity-window	
<line_number>	0 ≤ int ≤ 255	Used to determine IWINLINE	(Comma-separated list of) EXTNAME(s) of extension(s) storing line intensity-window	
</windowInfo>				
<axis axisName="lambda">				
<start>		PXBEG3		Yes
<size>		PXEND3 – PXBEG3 + 1		Yes
<binFactor>		NBIN3		Yes
</axis>				
<axis axisName="Y">				
<start>		PXBEG2		Yes
<size>		PXEND2 – PXBEG2 + 1		Yes
<binFactor>		NBIN2		Yes
</axis>				
<compression>				

XML tags	XML tag data type and values	FITS keywords	FITS keyword data type and values	In TLM
		See <compressionUpper>		
</compression>				
<alignmentWindow>		DUMBBEXT *Not implemented yet*	(Comma-separated list of) EXTNAME(s) of the extension(s) containing dumb-bell data for this window	No
</window>				

Table 4-15: Relationship between Study Generator XML tags and FITS keywords.