Data Manual

Data product description for Solar Wind Ion Spectrometer (SWIS-ASPEX)

SWIS-ASPEX

December, 2024

Doc Id: ASPEX_SWIS_DATA_MAN_V_1.2

Version control sheet

Version	Date	Change/Revisions
1.0	24/11/2024	1. Initial version. Includes details of data product from SWIS.
1.1	24/12/2024	Revisions after peer review suggestions mainly comprising of, 1. Description of file naming convention 2. Overview of data processing software 3. Sample C script to read SWIS CDF file
1.2	31/12/2024	Minor revisions mainly comprising of, 1. Note on bulk parameter generation and its availability

Contents

List of Tables

1 Introduction

Aditya Solar wind and Particle EXperiment(ASPEX) payload, on-board ADITYA-L1 satellite will facilitate in-situ, multi-directional measurements of the solar wind, supra-thermal and high energy particles in the energy range of 100 eV/n to 5 MeV/n. The ASPEX payload has two independent subsystems viz. Solar Wind Ion Spectrometer (SWIS) & Supra Thermal Energetic Particle Spectrometer (STEPS). SWIS is a low energy ion spectrometer measuring ion flux in the range of 100 eV to 20 keV, while STEPS measures the ion flux in the energy range of 20 keV/n to 5 MeV/n.

This document contains description of SWIS data products, finalized after the payload verification (PV) phase operations. The document is intended to give an overview of data products generated by SWIS data processing software which will be hosted at Indian Space Science Data Center (ISSDC) for public dissemination. File naming and other data attributes are governed by ISSDC and ASPEX payload operation centre (POC) interface control document (ICD). Some description of this can be found in this document.
It is expected from the users of SWIS data that they have some idea of the instrument and how it collects the data. Description on these aspects can be obtained from the SWIS instrument paper available on Research Square at `https://doi.org/10.21203/rs.3.rs-5180284/v1` . The description of level 0 data is avoided in the present document as it is expected that a user will use derived data products, made available by the POC. Although, payload team encourages users to directly use Level 2 (L2) data product as it is suitably processed, an informed user may use Level 1 (L1) data products and the software to generate refined, but non-standard versions of L2 data. It is strongly encouraged that a user must discuss any non-standard technique to generate L2 product with the payload team so as to avoid any misinterpretation of the data products. The description of the data products are provided in the sections 4 and 5.

2 Data access

ISSDC is the nodal agency for archival and dissemination of ASPEX data sets and a user is expected to download data from their server (`https://pradan.issdc.gov.in/al1/`). A user has to first register on their website to be able to download the data sets. The data sets are then accessible via a PRADAN interface. General information regarding data download, terms of use, etc are available on the ISSDC website.

3 File naming convention

A general file naming convention is followed for all data products related to SWIS. Common file name structure for SWIS L1 and L2 data products is:

```
AL1_ASW91_LN_TTT_YYYYMMDD_<obsId>_Vmn.cdf
```

File naming convention is explained is explained as follows:
AL1 : Spacecraft identifier (Aditya-L1)
ASW91 : Payload identifier (ASPEX-SWIS, 91 is the identifier as used by L0 team to identify ASPEX data products)
LN : Data product level, N = 1 for level 1 and N = 2 for level 2 data products
TTT: Description of file content. For level 1 SWIS files, 'TTT' can be either AUX for auxiliary file containing house keeping and health monitors of SWIS all packages, or TH1 for raw spectrum file from THA-1 or TH2 for raw spectrum file from THA-2. In level 2, 'TTT' can take values of TH1 for file containing flux information from THA-1, TH2 for file containing flux information from THA-2 and BLK for file containing the bulk parameters as obtained from SWIS measurements.
obsID: This is a 15 characters unique observation ID provided by ISSDC for proposal based observations. **Vmn:** This is the version of the data product in two digit format. It starts from '00' can go up to '99'.

4 Level 1 data product

Following table provides an overview of L1 data product description for SWIS.

#	Title	Description
1	L1 data products	SWIS - three CDF files per day per observation id. AUX: Auxiliary information file, contains health monitors, counter and other flags related to instrument operation TH1: THA-1 raw spectrum counts TH2: THA-2 raw spectrum counts *CDF files are compressed using CDF in-built Gzip compression of level 2.*
2	Associated files	1 meta-file for each L1 CDF file
3	Science ready	No
4	Data Dissemination Unit	Individual L1 CDF files

Table 1: Overview of the level-1 data product generated for SWIS.

The data file description for level 1 files are provided below,

4.1 Auxiliary file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L1_AUX_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The aux file is generated by `swis_l1_gen` module of the data processing routine. It is obtained after separating the spectrum data from other header data present in the L0 bundle. Multiple L0 tar files are combined to generate daily data sets for a given observation ID.

File description: Inside L1 bundle, there exists an aux file which contains important HK parameters of the instrument. This file also contains set voltage values corresponding to each observation time which is later on used to derive energy values during L2 processing. Other parameters are integrated counters (used for flux calibration in L2 processing), temperature and voltage values of different subsystems. This file also contains spacecraft attitude and orbit information as derived from the SPICE kernels. UTC time stamping of intermediate observations within a frame is also done and stored in this file depending on the mode of operation.

Variables: 77, metadata is stored in CDF

Associated files: 1 meta-file. The parameters of meta-file are described in section 7.

4.2 THA-1 Raw spectrum file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L1_TH1_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The raw spectrum file is generated by `swis_l1_gen` module of the data processing routine. It is obtained after only selecting the spectrum data from L0 bundle. Multiple L0 tar files are combined to generate daily data sets for a given observation ID.

File description: The spectrum file contains the raw counts as recorded and stored by the instrument during on-board observation. The spectrum implies the position spectrum as observed by the resistive anode encoder (RAE). For THA-1, this file contains count vs track information from 470 tracks of four quadrants. This file is further used during L2 processing to generate integrated flux values after all corrections are is written to L2 flux spectrum file. UTC time stamping of intermediate observations within a frame is also done and stored in this file depending on the mode of operation.

Variables: 4, metadata is stored in CDF

Associated files: 1 meta-file. The parameters of meta-file are described in section 7.

4.3 THA-2 Raw spectrum file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L1_TH2_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The raw spectrum file is generated by `swis_l1_gen` module of the data processing routine. It is obtained after only selecting the spectrum data from L0 bundle. Multiple L0 tar files are combined to generate daily data sets for a given observation ID.

File description: The spectrum file contains the raw counts as recorded and stored by the instrument during on-board observation. The spectrum implies the position spectrum as observed by the RAE. For THA-2, this file contains count vs track information from 160 tracks of THA-2. This file is further used during L2 processing to generate integrated flux values after all corrections are is written to L2 flux spectrum file. UTC time stamping of intermediate observations within a frame is also done and stored in this file depending on the mode of operation.

Variables: 4, metadata is stored in CDF

Associated files: 1 meta-file. The parameters of meta-file are described in section 7.

5 Level 2 data product

Following table provides an overview of L2 data product description for SWIS.

#	Title	Description
1	L2 data products	SWIS: 3 CDF files per day per observation id TH1: Differential energy flux as obtained from THA-1 TH2: Differential energy flux as obtained from THA-2 BLK: Bulk parameters as derived from the flux data *CDF files are compressed using CDF in-built Gzip compression of level 2.*
2	Associated files	1 meta-file and 1 preview image for each L2 CDF file
3	Science ready	Yes
4	Data Dissemination Unit	Individual L2 CDF file

Table 2: Overview of the level-2 data product generated for SWIS.

The data file description for level 2 files are provided below,

5.1 THA-1 Flux spectrum file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L2_TH1_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The spectrum file is generated by `tha1_l2_gen` module of the data processing routine. It is obtained after converting the raw counts in calibrated energy flux values using the calibration database. It requires both AUX and TH1 spectrum file generated during L1 processing.

File description: The spectrum file contains the UTC time-tagged differential energy flux values integrated from sunward sectors in 50 energy channels of THA-1. Since energy scanning is done in time, central energy bins corresponding to each observation time is also written in the file. Spacecraft position in GSE coordinate system is also recorded in this file to facilitate comparison with other measurements. To derive FOV angles of the individual sectors, angle between the nominal sun-direction and the three axis of GSE coordinate system is derived from the SPICE kernels and written to this file. The estimated error in flux and energy are also written in the CDF file.

Variables: 11, metadata is stored in CDF

Associated files: 1 meta-file and 1 preview image. The parameters of meta-file are described in section 7.

5.2 THA-2 Flux spectrum file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L2_TH2_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The spectrum file is generated by `tha2_l2_gen` module of the data processing routine. It is obtained after converting the raw counts in calibrated energy flux values using the calibration database. It requires both AUX and TH2 spectrum file generated during L1 processing.

File description: The spectrum file contains the UTC time-tagged differential energy flux values integrated from sunward sectors in 50 energy channels of THA-2. Since energy scanning is done in time, central energy bins corresponding to each observation time is also written in the file. Spacecraft position in GSE coordinate system is also recorded in this file to facilitate comparison with other measurements. To derive FOV angles of the individual sectors, angle between the nominal sun-direction and the three axis of GSE coordinate system is derived from the SPICE kernels and written to this file. The estimated error in flux and energy are also written in the CDF file.

Variables: 11, metadata is stored in CDF

Associated files: 1 meta-file and 1 preview image. The parameters of meta-file are described in section 7.

5.3 Bulk parameter file

File name: The file naming convention follows the description mentioned in section 3. It is as mentioned below:
`AL1_ASW91_L2_BLK_YYYYMMDD_<obsId>_Vmn.cdf`

Frequency of generation: 1 file per day, per observation ID

File generation: The bulk parameter file is generated by `bulk_l2_gen` module of the data processing routine. It is obtained after calculating moments from the flux data of THA-1 and THA-2 and requires L2 flux spectrum files generated by L2 processing. The bulk parameters are obtained from non-linear least square fitting of the flux data. The values provided in the data sets are result of fitting the core component of the solar wind particle flux as observed by the instrument.

File description: The bulk parameter file contains number density, temperature and solar wind speed for protons as derived from non-linear fitting of the energy histogram generated from flux observations. Since multiple cycles are merged to generate this file, time cadence of the file is different from the previous two L2 files and depends on the number of cycles used for averaging. Spacecraft position in GSE coordinate system is also recorded in this file to facilitate comparison with other measurements. The estimated error in bulk parameters are also written in the CDF file.

Variables: 10, metadata is stored in CDF

Associated files: 1 meta-file and 1 preview image. The parameters of meta-file are described in section 7.

6 Overview of the data pipeline software

Figure 1 shows the overall layout of various elements of the SWIS data processing software (DPS). The level 0 to level 1 data processing chain contains one module, *swis_l1_gen* which generates two raw spectrum files (corresponding to THA-1 and THA-2) and one AUX file containing HK and other instrument parameters. The level 1 to level 2 data processing chain is separate for THA-1 and THA-2 providing separate science files in CDF as output.

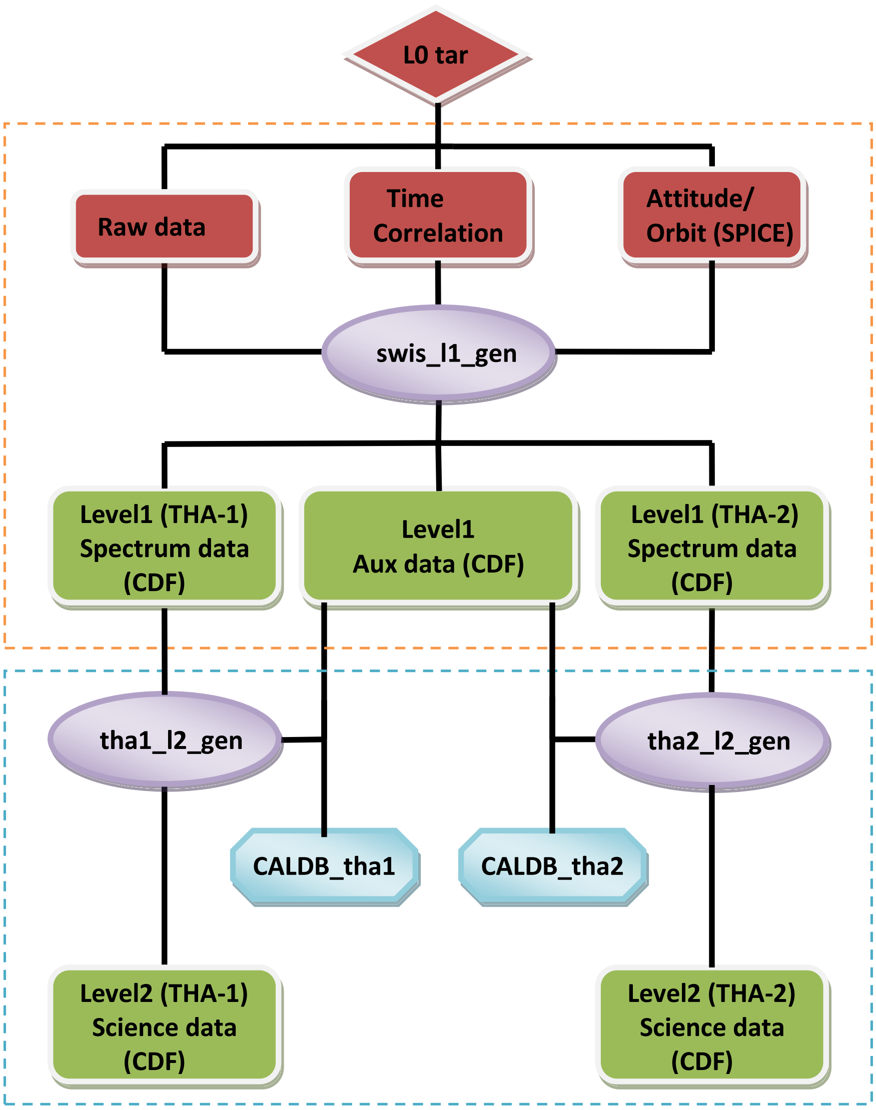

Figure 1: Overall schematic of SWIS data pipeline software

The details of modules are presented in the subsequent sections.

6.1 Module 1: *swis_l1_gen*

6.1.1 Purpose:

This module mainly re-organizes the level 0 data as received from ISSDC into two separate spectrum files corresponding to THA-1 and THA-2. It also generates an auxiliary file containing housekeeping and other header data as obtained in the level 0 dump. The schematic is shown in figure 2.

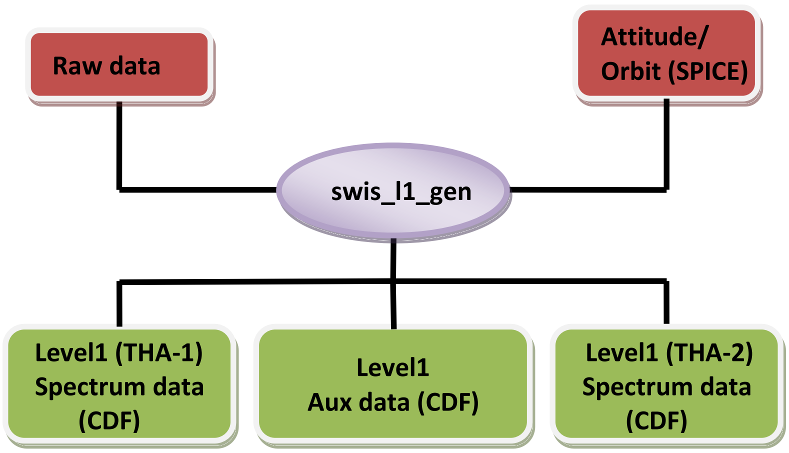

Figure 2: Schematic of module: *swis_l1_gen*

6.1.2 Input output requirements:

- **Input files:** Raw Payload file, SPICE kernels
- **Output files:** 3 files, THA-1 spectrum data, THA-2 spectrum data, Aux data(all HK, headers, SPICE)

6.1.3 Processes:

1. Read multiple dumps/single payload data file of ASPEX-SWIS.
2. Remove repetitive and bad data frames from the payload data file
3. Find UTC information from the ground level header of the payload data file.
4. Sort valid data frames with respect to UTC time.
5. Segregate valid sorted payload data files into different variables using predefined data packet structure of SWIS.
6. If SPICE kernels are specified, calculate the position and velocity of spacecraft with respect to Geocentric Solar Ecliptic (GSE) coordinate systems.
7. Calculate angle between nominal FOV axis of respective sectors with the Sun-spacecraft line and write in aux file.
8. Write Level-1 spectrum files (TH1 and TH2), AUX file according to ISTP/IACG guidelines.
9. In addition to the above, a sanity check with respect to expected value of all instrument parameters is performed.

6.2 Module 2: *tha1_l2_gen*

6.2.1 Purpose:

This module generates level 2 science data product from level 1 files for THA-1. Level 2 data processing has been made separate for THA-1 and THA-2. This module has an interface with THA-1 specific calibration database (CALDB) and uses both L1 science and AUX file as input. The schematic is shown in figure 3.

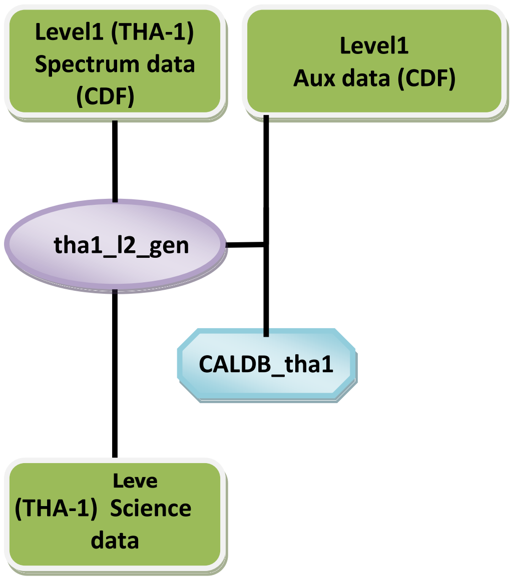

Figure 3: Schematic of module: *tha1_l2_gen*

6.2.2 Input output requirements:

- **Input files:** Level1 THA-1 spectrum data, level1 aux data , CALDB for THA1
- **Output files:** 1 file, Calibrated THA-1 level2 science data

6.2.3 Processes:

1. Read multiple /single payload level 1 data file of ASPEX-SWIS.
2. Split data frames into separate variables based on the presence of calibration being on or off.
3. For calibration 'OFF' data frames, process data files as mentioned below, for calibration 'ON' frames, move the relevant frames to a separate folder for updating CALDB

4. Covert voltage to energy based on relevant CALDB files
5. Unfold the raw spectrum data from THA-1, provide sector demarcations, use CALDB files to separate proton and other mass bins based on ion energy
6. Apply count rate based correction, cross-talk corrections
7. Apply mass/energy based corrections for GF to counts, obtain flux for proton
8. Compute error in flux based on CALDB file and FEE settings as provided in level1 AUX file
9. Write Level-2 Science file according to ISTP/IACG guidelines.

6.3 Module 3: *tha2_l2_gen*

6.3.1 Purpose:

This module generates level 2 science data product from level 1 files for THA-2. The level 2 data processing for THA-2 is identical to THA-1. THA-2 doesn't have mass separation capability and will provide integrated flux as output. This module has an interface with THA-2 specific CALDB and uses both L1 science and AUX file as input. The schematic is shown in figure 4.

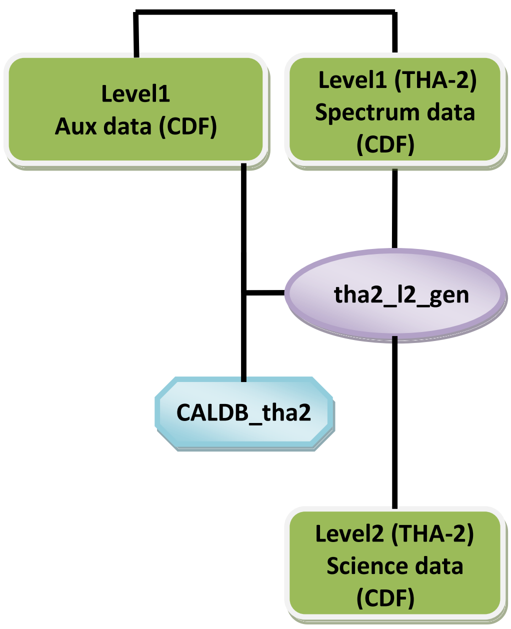

Figure 4: Schematic of module: *tha2_l2_gen*

6.3.2 Input output requirements:

- **Input files:** Level1 THA-2 spectrum data, level1 aux data , CALDB for THA2
- **Output files:** 1 file, Calibrated THA-2 level2 science data

6.3.3 Processes:

1. Read multiple/single payload level 1 data file of ASPEX-SWIS.
2. Split data frames into separate variables based on the presence of calibration being on or off.
3. For calibration 'OFF' data frames, process data files as mentioned below, for calibration 'ON' frames, move the relevant frames to a separate folder for updating CALDB
4. Covert voltage to energy based on relevant CALDB files
5. Unfold the raw spectrum data from THA-2, provide sector demarcations.
6. Apply count rate based corrections
7. Apply energy based corrections for GF to counts, obtain integrated particle flux
8. Compute error in flux based on CALDB file and FEE settings as provided in level1 AUX file
9. Write Level-2 Science file according to ISTP/IACG guidelines.

7 Meta-file parameter definitions

The meta-file format along with the description of parameters are provided in this section. The file format is same for all data products generated from SWIS. Definitions of the parameters are also same, unless otherwise stated. This meta-file format is standard for all Aditya-L1 isntruments and is as per the requirement by the dissemination software deployed at ISSDC. The structure and parameters of the meta-file is shown table 3.

Parameter		Description
Identification Details	Mission	Name of the mission (Aditya-L1)
	Payload Name	Name of payload (ASPEX SWIS)
	Payload description	Description of payload
	Proposal id	Proposal ID, SWIS supports proposal driven observations and this will be reflected by a change in the ID
	Observation id	Observation ID SWIS supports proposal driven observations and this will be reflected by a change in the ID
	Version	File version. See description at the end of this table.
Observation Details	Time Coordinates: start_date_time stop_date_time exposure_time	Start and Stop time of observations written in the CDF file. Exposure time in sec is also provided
	Energy scan: Coverage and description	Specifies whether energy scan is partial or full. Refer to text at the end of this table.
	Event Flag: Assessment and description	A flag to identify whether the file contains data for transient event. Refer to text at the end of this table.
File Details	Name, file size, md5 checksum, creation date and content type	Generic details related to the data file. Content describes the content of data product as defined earlier. For L1 it can be: AUX, TH1 and TH2 while for L2 It can be: TH1, TH2, BLK
	Input files (number), input file names	Input files used to generate the present data file
Processing details	Software version and level	Software version used to create the file. Level identifier specifies the data level of the product as per definition in this document
	Quality assessment and description	Quality of data level. For a full description please refer to text at the end of this table.
Remarks	File type: AUX/TH1/TH2/BLK	

Table 3: Parameters written in the XML file associated with each CDF file generated by SWIS data pipeline.

Some of the parameters in the meta-file description table 3 are described below.

7.1 Data product versions

Each data product generated by the SWIS pipeline software is assigned a version number, represented in the CDF filenames using two variables, m and n, where both can range from 0 to 9. The first version of a data product is assigned the version number 01, and subsequent versions increment accordingly (e.g., 02, 03, etc.). The version number may be updated under several conditions, including modifications to the data processing software, updates in calibration methodologies, changes to SPICE kernels, or other significant alterations.

7.2 Energy scan

Since there is a provision to change energy value and number of steps in case of SWIS, an additional variable in meta-file related to energy scan coverage is also written. This variable can take string value and is represented by either ‘FULL’ or ‘PARTIAL’. FULL energy scan implies that the energy coverage is sufficient to observe both proton and alpha flux and bulk parameters from such data can be derived. If the value of the variable is ‘PARTIAL’, it implies that data is obtained for limited energy channels and generation of bulk parameters may not be possible.

7.3 Data quality

Another important parameter in the meta-file is related to data quality levels. Standard data level definition for Aditya-L1 instruments requires data levels to be categorized in levels in the range of 1-10. However, number of levels prescribed by this standard is too large and for SWIS, following definition of data quality level is decided.

Data level	Quality level	Description
Level 1	1-6	Very poor quality, highly restricted energy scan , uncertainty in detector and ESA voltage >20%
	7	Poor quality, restricted energy scan, uncertainty in detector and ESA voltage > 15%
	8	Average quality, slightly restricted energy scan, uncertainty in detector and ESA voltage > 10%
	9	Good quality, uncertainty in detector and ESA voltage > 5%
	10	Very good quality, uncertainty in detector and ESA voltage < 5%
Level 2	1-6	Very poor data, huge flux errors (>50%), highly restricted energy scan, subsequent estimation of bulk parameters not possible, for BLK data file all values are NaN
	7	Poor data, huge flux errors (>40%), restricted energy scan, subsequent estimation of bulk parameters are restricted, BLK data file contains data for limited time, error in bulk parameters > 50%
	8	Average quality data, high flux errors (>30%), restricted energy scan, subsequent estimation of bulk parameters are somewhat compromised, BLK data file contains data for limited time, error in bulk parameters > 40%
	9	Good quality data, low flux errors (>20%), reasonable estimation of bulk parameters, BLK data file contains data for most of the time, error in bulk parameters > 30%
	10	Very good quality data, lowest flux errors (<20%), good estimation of bulk parameters, BLK data file contains data for most of the time, error in bulk parameters < 30%

Table 4: Description of the data quality levels for SWIS. Quality definitions are different for level-1 and level-2 files. These definitions are tentative and may be modified once year long data is reprocessed and compared with existing measurements at L1 point.

7.4 Event flag

It is planned to provide event flag denoting data of transient event from SWIS data sets. A value of '1' will be written if the data-set contains data from an event, while '0' denotes normal data. Although this variable is included in the meta-file, identification and labeling of data-sets for transient events is still not implemented and will be updated in future revision.

8 Data sets available and future versions

At present, data sets for August-September, 2024 are released and further data sets will be provided gradually. The bulk parameters, at present, are provided only for August, 2024. The pipeline is still undergoing changes to provide a robust estimate of bulk parameters and we expect to release complete data sets within a few months time. Directional energy flux will also be made available along with the bulk parameters in a future revision of the data-sets. Similarly, anisotropy analysis and resulting data sets using the combined measurement of direction differential flux from SWIS and magnetic field data from MAG instrument will be made available in due course once the pipeline is tested.

9 Other data products

In addition to the regular science observation data generated by ASPEX POC for SWIS, there are some more data sets which are generated and send to ISSDC. These are discussed in subsequent sections.

9.1 Outreach data product

An image file consisting of derived plots from level-2 data will be provided to ISSDC for public outreach. An overview of the data product is shown in table 5 and description is provided subsequently.

#	Title	Description
1	Outreach Product	1 png file Contains 4 sub-plots obtained from level-2 data
2	Associated file	1 meta-file
3	Purpose	Outreach, quick look of dataset

Table 5: Overview of the outreach data product planned for SWIS.

File name: The file naming convention for the out reach product is as mentioned below: `OR_AL1_ASW_YYYYMMDD.png`

Frequency of generation: 1 file per day

File generation: The outreach product image file is generated by `swis_outreach_gen` module of the data processing routine. The image file requires level-2 data products from a previous day's observation.

File description: As of now, it is planned to provide four subplots in the image file. These subplots are, energy spectrogram, proton number density, proton temperature and bulk velocity of proton as obtained from SWIS observations.

Associated files: 1 meta-file.

9.2 Calibration data

The data collected during the payload's calibration operations will be processed and transferred to ISSDC similar to level 1 and level 2 data files as described previously. Since it is expected that calibration operations will have a different proposal ID, it is expected that dissemination of these files will be restricted only to the PI institute. In this context, it may be noted that the data files obtained directly from calibration operations are termed as 'raw calibration data files' and will follow the above suggested plan for generation, archival and dissemination.

These files may be processed to further generate 'derived calibration data files' which may be used to update the calibration database. The derived calibration data files may also be generated by POC even without a calibration operation. These files along with the software will be hosted separately by ISSDC and they are not a part of the regular data stream. All calibration data files along with a calibration index file will be bundled together and sent to ISSDC as a single zip file with appropriate versioning. These data-sets will be hosted at 'Other Downloads' section of Aditya-L1 ISSDC webpage along with the software. At present, we expect the updation in calibration database 1-2 times in a year.

10 Software to read data files

All data products are generated as CDF v3.9. To read these files, a user must have CDF libraries v3.9 or above installed. Python v3.6 and above (with CDF libraries) can be used to read these files. Since all files are in CDF format, all metadata information is stored in the file itself. Moreover the files can be read from CDF tools available on NASA's (`https://cdf.gsfc.nasa.gov/`). A simple Python script to read attributes from the file and a C script to read the file content is attached below.

Python Script:

```
import cdflib

# Replace with the actual path to your CDF file
cdf_file = r'<CDF FILENAME>'

# Open the CDF file
cdf = cdflib.CDF(cdf_file)

# Get global attributes
global_attributes = cdf.globalattsget()

# Print global attributes and their descriptions
print("Global Attributes and Descriptions:\n")
for attr_name, attr_value in global_attributes.items():
    print(f"{attr_name}: {attr_value}")

# Get CDF info
cdf_info = cdf.cdf_info()

# Print variable attributes and their descriptions
print("\nVariable Attributes and Descriptions:\n")
for var_name in cdf_info.zVariables:  # Access the zVariables attribute directly
    var_attrs = cdf.varattsget(var_name)
    print(f"Attributes for {var_name}:")
    for attr_name, attr_value in var_attrs.items():
        print(f"  {attr_name}: {attr_value}")
    print()  # Add an empty line between variables
```

C-Script:

```
/*
DESCRIPTION:

This is sample C routine to read SWIS CDF file content.
Following are the pre-requisites:
```

```
1. GNU C Libraries
2. CDF Libraries


*/

/* Begin preamble */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <assert.h>
#include "cdf.h"


/* End preamble */

/*****************************************************************
* USER DEFINED prototypes
*****************************************************************/
void StatusHandler (CDFstatus status, int lno);
/****************************************************************
* MAIN.
*****************************************************************/

int main(int argc, char *argv[])
{
int i, j, k;
CDFid id;            /* CDF identifier. */
CDFstatus status;
char *file_name;
int buffer1 = 0;
long attr_num, var_num;
double *obs_time, *flux;
char temp_char[23];
long dim_indices[3], dim_counts[3];
```

```
long dim_interval[3], dim_sizes[3];
long dim_varys[3];

file_name = argv[1];


status = CDFopenCDF (file_name, &id);
if (status != CDF_OK) StatusHandler (status, __LINE__);

dim_varys[0] = VARY;
dim_varys[1] = VARY;
dim_varys[2] = VARY;
dim_interval[0] = 1;
dim_interval[1] = 1;
dim_interval[2] = 1;
dim_indices[0] = 0;
dim_indices[1] = 0;
dim_indices[2] = 0;

attr_num = CDFgetAttrNum (id, "Record_length"); //Number of observation records
status = CDFgetAttrgEntry (id, attr_num, 0, &buffer1);
if (status != CDF_OK) StatusHandler (status, __LINE__);

obs_time = malloc(buffer1 * sizeof(obs_time)); /* Allocate memory for
observation record */
flux = malloc(buffer1 * sizeof(flux)); /* Allocate memory for flux record*/

var_num = CDFgetVarNum (id, "epoch_for_cdf"); /* read variabe number of
CDF epoch*/
if (var_num < CDF_OK) StatusHandler (var_num, __LINE__);

dim_counts[0] = 1;

status = CDFhyperGetzVarData (id, var_num, 0L, buffer1, 1L, dim_indices,
dim_counts, dim_interval, obs_time); /*Get CDF epoch values and store
it in a variable */
if (status != CDF_OK) StatusHandler (status, __LINE__);
```

```
var_num = CDFgetVarNum (id, "integrated_flux"); /* read variabe number of
integrated_flux, to read other variable replace this */
if (var_num < CDF_OK) StatusHandler (var_num, __LINE__);

dim_counts[0] = 1;

status = CDFhyperGetzVarData (id, var_num, 0L, buffer1, 1L, dim_indices,
dim_counts, dim_interval, flux); /* Get flux values variable and store
it in a variable*/
       if (status != CDF_OK) StatusHandler (status, __LINE__);

/*
fout1 = fopen(acii_output.dat, "w");
for(i = 0; i<buffer1; i++)
{
encodeEPOCH4(obs_time[i], temp_char);
temp_char[23] = '\0';
fprintf(fout1, "%s\t%lf\n", temp_char, flux[i]);
}
       fclose(fout1);
*/

  status = CDFcloseCDF (id);
if (status != CDF_OK) StatusHandler (status, __LINE__);

free(obs_time);
free(flux);

}
void StatusHandler(CDFstatus status, int lno)
{
char message[CDF_ERRTEXT_LEN+1];

if (status < CDF_WARN)
```

```
{
printf ("LINE %d (%s): An error has occurred, halting...\n", lno, __FILE__);
CDFgetStatusText (status, message);
printf ("%s\n", message);
   exit (1);
}
else if (status < CDF_OK)
{
printf ("LINE %d (%s): Warning, function may not
have completed as expected...\n", lno, __FILE__);
CDFgetStatusText (status, message);
printf ("%s\n", message);
}
return;
}
```

To compile the routine, use following:

```
gcc -o <exe> -I <CDF HEADER PATH> <C-SCRIPT-FILE> <PATH OF CDF LIBRARY> -lm
```