Royal Netherlands
Meteorological Institute
*Ministry of Infrastructure
and Water Management*

Input/output data specification for the TROPOMI L01b data processor

TROPOMI

document number : S5P-KNMI-L01B-0012-SD
CI identification : CI-6510-IODS
issue : 9.0.0
date : 2018-04-01
status : released

Document approval record

	digital signature
prepared:	
checked:	Nico C. Rozemeijer Digitally signed by Nico C. Rozemeijer DN: c=NL, st=Utrecht, o=Ministry of Infrastructure and the Environment, ou=Royal Netherlands Meteorological Institute, cn=Nico C. Rozemeijer, email=rozemeij@knmi.nl Date: 2018.03.29 14:42:05 +02'00'
approved PM:	
approved QA:	Nico C. Rozemeijer Digitally signed by Nico C. Rozemeijer DN: c=NL, st=Utrecht, o=Ministry of Infrastructure and the Environment, ou=Royal Netherlands Meteorological Institute, cn=Nico C. Rozemeijer, email=rozemeij@knmi.nl Date: 2018.03.29 14:42:33 +02'00'
approved CM:	
approved PI:	n/a
approved client:	n/a

Document change record

Issue	Date	Item	Comments
0.1.0	2012-12-03	All	Prepared for review.
0.2.0	2013-02-13	All	Processed informal review comments; see S5P-KNMI-L01B-1012-LI-RLF issue A.4.
1.0.0	2013-03-05	All	Modifications based on comments received on version 0.2.0. Editorial modifications.
1.0.1	2013-07-12	All	New sections on metadata (sections 7, 8 and 10.2). New section on input data (section 5). Redefinition of time variable. Redefinition of noise and error variables for radiance and irradiance. spectral_wavelength removed (radiance). spectral_wavelength replaced by calibrated_wavelength (irradiance). Reformatting of tables. Updated section 9.1 on L1b file structure.
1.0.2	2013-08-29	All	Modifications based on comments from internal review. Measurement class renamed to Processing class. Slicing for NRT products. Redefinition of quality_level. New TROPOMI logo
1.1.0	2013-09-11	All	Updated section on variable attributes Editorial modifications
1.2.0	2013-11-20	All	Processed formal review comments; see S5P-KNMI-L01B-1012-LI-RLF issue B.3.
2.0.0	2013-12-05	All	Modifications based on comments received on version 1.2.0. Editorial modifications.
2.1.0	2014-06-05	All	Document ported from Word to LaTeX. Definition of engineering product added (section 7.3.4). Product sizes table updated (Appendix A). Processing classes table updated (Appendix B). Type of variable time changed to int (section 8.4). Variables time and delta_time moved to ObservationsGroup (sections 7.3.1 and 7.3.2). Variables radiance, irradiance and small_pixel_radiance are normalized to 1AU. Variable descriptions added for instrument_settings, binning_table, housekeeping_data (sections 8.41 - 8.43). Moved sections on metadata models and profiles to document [RD9]).
2.2.0	2014-06-27	All	Modifications resulting from review comments. Ordering of dimensions in netCDF product changed.
2.3.0	2014-07-07	All	Some additional modifications resulting from review comments.
3.0.0	2014-07-07	All	Some additional modifications resulting from review comments.
3.0.1	2014-10-31	All	EO-FFS Fixed Header example added.
3.0.2	2014-11-18	All	Corrected EO-FFS example.
4.0.0	2014-12-09	All	Prepared for processor release 0.8.0
4.1.0	2015-09-21	All	Completed description of output products; prepared for processor release 0.9.0
5.0.0	2015-09-22	Title page, change record, references	Updated referenced documents; changed license.

Issue	Date	Item	Comments
6.0.0	2016-04-01	All	Updated data volume estimates; Prepared for processor release 0.10.0
7.0.0	2016-09-30	All	Added missing NetCDF4 global attributes. Regenerated group contents tables for L1b products. Regenerated cdl-descriptions of L1b product variables. Prepared for processor release 0.11.0
8.0.0	2017-06-01	All	Add out of band (oob) parameters to IODS. Prepare for processor release 0.12.0
8.1.0	2018-03-29	All	Preparations for L01b release 01.00.00
9.0.0	2018-04-01	All	Final version for L01b release 01.00.00

Contents

1 Introduction

1.1 Identification

This document, identified by S5P-KNMI-L01B-0012-SD, describes the data format of the TROPOMI Level-1b (L1b) data products. The logic for the L1b data format results from an analysis of different applicable standards and best practices in the Earth Observation (EO) data field. This document is identified in [AD1] as CI-6510-IODS.

1.2 Purpose and objective

The TROPOMI L01b processor developed by KNMI produces L1b data products from L0 input data and auxiliary data products. The TROPOMI L1b data products distinguish radiance, irradiance, calibration and engineering data. Although these products differ in their applicability, the objective is to define a common data format for all TROPOMI L1b products.

This document mainly addresses the output data of the L01b processor (i.e. the L1b data products), providing detailed specifications of the different L1b products. The input data (the Level 0 products and the various auxiliary products) are also identified and summarized in this document. One type of auxiliary data product is the calibration key data. Document [RD10] describes the types of calibration key data files. The L0 products are specified in [AD2].

1.3 Document overview

This document describes the official products that are the result from the Level 0 to Level 1b processing of the data collected by TROPOMI onboard the Sentinel-5 Precursor satellite. For all of the defined data products detailed technical information with respect to their contents and data formats is provided. This allows processing facilities and scientists to develop software for extracting information and in particular to produce higher level (i.e. Level 2) products.

The document is based on the results of discussions with user communities and of studies on data interoperability standards and on the lessons learned from previous missions (i.e. OMI).

After a short introduction of the TROPOMI system, its mission, the geophysical phenomena studied and the parameters measured by the detectors, the L1b products are described. Product specifications are presented in terms of file naming, file format and file structure. Comprehensive descriptions and specifications of all variables contained in the products are presented.

2 Applicable and reference documents

2.1 Applicable documents

[AD1] Software development plan for TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0002-PL; **issue:** 2.0.0; **date:** 2012-11-14.

[AD2] Sentinel-5 precursor PDGS L0 product format specification.
source: DLR; **ref:** S5P-PDGS-DLR-ISP-3011; **issue:** 1.3; **date:** 2015-11-30.

[AD3] Software product assurance plan for TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0003-PL; **issue:** 2.0.0; **date:** 2012-11-14.

[AD4] Software system specification for TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0005-RS; **issue:** 3.0.0; **date:** 2012-11-21.

[AD5] Tailoring of the Earth Observation File Format Standard for the Sentinel 5-Precursor Ground Segment.
source: ESA; **ref:** S5P-TN-ESA-GS-106; **issue:** 2.2; **date:** 2015-02-20.

[AD6] Earth Observation Ground Segment File Format Standard.
source: ESA; **ref:** PE-TN-ESA-GS-0001; **issue:** 2.0; **date:** 2012-05-03.

2.2 Standard documents

[SD7] Space Engineering – Software.
source: ESA/ECSS; **ref:** ECSS-E-ST-40C; **date:** 2009-03-06.

[SD8] Space Product Assurance – Software Product Assurance.
source: ESA/ECSS; **ref:** ECSS-Q-ST-80C; **date:** 2009-03-06.

2.3 Reference documents

[RD9] Metadata specification for the TROPOMI L1b products.
source: KNMI; **ref:** S5P-KNMI-L01B-0014-SD; **issue:** 5.0.0; **date:** 2018-04-01.

[RD10] Calibration key data specification for the TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0028-SD; **issue:** 6.0.0; **date:** 2017-06-01.

[RD11] Terms, definitions and abbreviations for TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0004-LI; **issue:** 3.0.0; **date:** 2013-11-08.

[RD12] Algorithm theoretical basis document for the TROPOMI L01b data processor.
source: KNMI; **ref:** S5P-KNMI-L01B-0009-SD; **issue:** 8.0.0; **date:** 2017-06-01.

[RD13] NetCDF Climate and Forecast (CF) Metadata Conventions.
source: CFConventions; **ref:** n/a; **issue:** 1.6; **date:** 2011-12-05.

[RD14] INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119.
source: EC JRC; **ref:** MD_IR_and_ISO_v1_2_20100616; **issue:** 1.2; **date:** 2010-06-16.

[RD15] Earth Observation Metadata profile of Observations Measurements.
source: OGC; **ref:** OGC 10-157r4; **issue:** 1.0.3-DRAFT; **date:** 2014-01-10.

[RD16] Command and Telemetry Handbook.
source: Dutch Space; **ref:** TROP-DS-0000-RP-0579; **issue:** 4.0; **date:** 2016-02-09.

2.4 Electronic references

[ER17] http://www.iers.org.

[ER18] http://www.unidata.ucar.edu/software/netcdf/docs/.

[ER19] http://en.wikipedia.org/wiki/University_Corporation_for_Atmospheric_Research.

[ER20] http://wiki.esipfed.org/index.php/Category:Attribute_Conventions_Dataset_Discovery.

[ER21] http://www.unidata.ucar.edu/software/thredds/current/tds/.

[ER22] http://wiki.esipfed.org/index.php/NetCDF,_HDF,_and_ISO_Metadata.

3 Terms, definitions and abbreviated terms

Terms, definitions and abbreviated terms that are used in the development program for the TROPOMI L01b data processor are described in [RD11]. Terms, definitions and abbreviated terms that are specific for this document can be found below.

3.1 Terms and definitions

There are no terms and definitions specific to this document.

3.2 Acronyms and Abbreviations

ACDD	Attribute Convention for Dataset Discovery
APID	Application Process Identifier
ADN	ADEPT/DLESE/NASA
AQA	Automated Quality Assurance
AU	Astronomical Unit
CCSDS	Consultative Committee for Space Data Systems
CF	Climate and Forecast
CKDS	Calibration Key Data Set
DEM	Detector Electronics Module
DIF	Data Interchange Format
EC	European Commission
EO-FFS	Earth Observation Ground Station File Format Standard
EOP	Earth Observation Product
ESA	European Space Agency
ESIP	Federation of Earth Science Information Partners
EU	European Union
FGDC	Federal Geographic Data Committee
GEMET	GEneral Multilingual Environmental Thesaurus
GMES	Global Monitoring for Environment and Security
HDF	Hierarchical Data Format
HMA	Heterogeneous Mission Accessibility
IcID	Instrument Configuration ID
ID	Identifier
IERS	International Earth Rotation and Reference Systems Service
INSPIRE	Infrastructure for Spatial Information in the European Community
IODS	Input/Output Data Specification
ISM	Instrument Specific Module
JRC	Joint Research Centre
LED	Light-Emitting Diode
LTAN	Local Solar Time at Ascending Node
NcML	NetCDF Markup Language
NetCDF	Network Common Data Form
NRT	Near Real Time
NUG	NetCDF User Guide
OGC	Open Geospatial Consortium
QI	Quality Indicator
SAA	South Atlantic Anomaly

SZA	Solar Zenith Angle
THREDDS	Thematic Realtime Environmental Distributed Data Services
TOA	Top Of Atmosphere
UCAR	University Corporation for Atmospheric Research
UML	Unified Modeling Language
UTC	Coordinated Universal Time
WGS	World Geodetic System
WLS	White Light Source
XML	Extensible Markup Language

4 TROPOMI system overview

4.1 Mission

The Sentinel-5 Precursor (S5p) mission will be implemented as part of the Global Monitoring for Environment and Security (GMES) programme, which is a joint initiative of the European Commission (EC) and of the European Space Agency (ESA). The S5p mission is a single-payload satellite in a low Earth orbit that provides daily global information on concentrations of trace gases and aerosols important for air quality, climate forcing, and the ozone layer. The payload of the mission is the TROPOspheric Monitoring Instrument (TROPOMI), which is jointly developed by The Netherlands and ESA. TROPOMI is a spectrometer with spectral bands in the ultraviolet (UV), the visible (VIS), the near-infrared (NIR) and the shortwave infrared (SWIR). The selected wavelength range for TROPOMI allows observation of key atmospheric constituents, including ozone (O_3), nitrogen dioxide (NO_2), carbon monoxide (CO), sulfur dioxide (SO_2), methane (CH_4), formaldehyde (CH_2O), aerosols and clouds.

4.2 Instrument description

The TROPOMI instrument (TROPOMI) is a space-borne nadir-viewing hyperspectral imager with four separate spectrometers covering non-overlapping and non-contiguous wavelength bands between the ultraviolet and the shortwave infrared. The instrument is the payload on the ESA/GMES Sentinel 5 Precursor mission.

The purpose of TROPOMI is the measurement of atmospheric properties and constituents. The instrument uses passive remote sensing techniques to attain its objective by measuring at the top of the atmosphere the solar radiation reflected by and radiated from the Earth. The instrument operates in a push-broom configuration with a wide swath. Light from the entire swath is recorded simultaneously and dispersed onto two-dimensional imaging detectors: the position along the swath is projected onto one direction of the detectors, and the spectral information for each position is projected on the other direction.

The instrument images a strip of the Earth on a two dimensional detector for a period of approximately 1 second during which the satellite moves by about 7 km. This strip has dimensions of approximately 2600 km in the direction across the track of the satellite and 7 km in the along-track direction. After the 1 second measurement a new measurement is started thus the instrument scans the Earth as the satellite moves. The two dimensions of the detector are used to detect the different ground pixels in the across track direction and for the different wavelengths. The measurement principle of TROPOMI is shown in Figure 1.

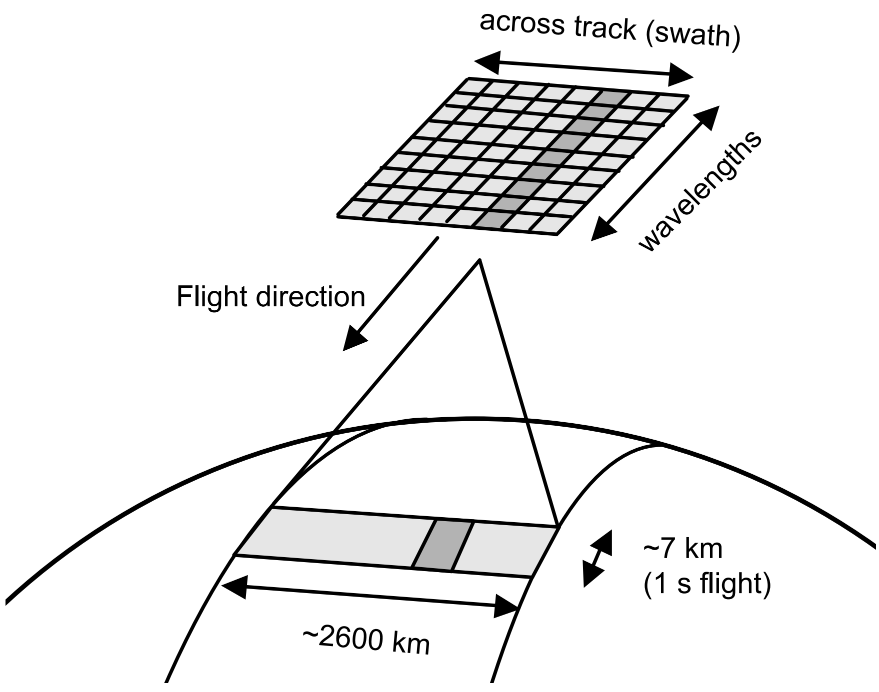

Figure 1: TROPOMI measurement principle

TROPOMI utilizes a single telescope to form an image of the target area onto a rectangular slit that acts as the entrance slit of the spectrometer system. There are four different spectrometers, each with its own optics and detector: medium wave ultraviolet (UV), long wave ultraviolet combined with visual (UVIS), near infrared (NIR), and short wave infrared (SWIR). The spectrometers for UV, UVIS and NIR are jointly referred to as UVN. Table 1 lists the spectral characteristics of the four TROPOMI spectrometers and the definition of the TROPOMI spectral bands with identifiers 1–8.

Instrument module			**UVN**				**SWIR**	
Detector	**UV**		**UVIS**		**NIR**		**SWIR**	
Band ID	1	2	3	4	5	6	7	8
Spectral range [nm]	270–300	300–320	320–405	405–500	675–725	725–775	2305–2345	2345–2385
Spectral resolution [nm]	0.5	0.5	0.5	0.5	0.5	0.5	0.23	0.23
Spectral sampling [nm/pixel]	0.065	0.065	0.20	0.20	0.124	0.124	0.084	0.097

Table 1: Main spectral characteristics of the four TROPOMI spectrometers and the definition of the spectral bands with identifiers 1–8. Remark: the figures mentioned for range, resolution and sampling are taken from [RD12]. These figures might change in future releases of [RD12]; this table will be updated accordingly in next issues.

4.3 Instrument operations

For TROPOMI instrument operations, an orbital scheduling approach is used. An orbit is defined from spacecraft midnight to spacecraft midnight. Earth radiance measurements will be performed on the day side of the orbit. At the north side of the orbit, near the day-night terminator, the Sun is visible in the instrument's solar port. Approximately once a day, a solar irradiance measurement is performed. The night side of the orbit is used for calibration and background measurements. The following constraints apply to the calibration measurements:

1. Background and calibration measurements can only be performed when the spacecraft is in eclipse.
2. No measurements can be taken around spacecraft midnight, in order to facilitate data processing.
3. All orbits must contain background measurements in the eclipse part of the orbit that match the radiance measurements on the day side for in-orbit calibration by the L01b Processor.
4. Calibration measurements must be performed outside the South Atlantic Anomaly (SAA) area, in order to minimize interference of proton radiation.
5. Calibration measurements must have a regular, fixed repetition interval.

In Figure 2 the Sentinel-5p orbit overview is presented. Each S5p orbit has a day (lit) side and a night (dark) side, as illustrated in the figure. On the day side the spacecraft flies from south to north; on the night side it flies from north to south. Spacecraft midnight is defined as the time halfway the nadir day-night terminator and the nadir night-day terminator; spacecraft noon is the time halfway the nadir night-day terminator and the nadir day-night terminator. Both the instrument operations as well as data processing will use the spacecraft midnight as the start and end points of an orbit. Due to seasonal variation, the position of the equator with respect to the spacecraft midnight will change. As a result, spacecraft midnight is not at a fixed latitude.

The S5p reference orbit is a near-polar frozen sun-synchronous orbit, adopted for mission optimization with a mean Local Solar Time at Ascending Node (LTAN) of 13:30 h and a repeat cycle of 17 days or 24155 orbits. More important than this repeat cycle is the operational repeat cycle. To accommodate regular, fixed repetition intervals for the calibration measurements, a scheme of 360 orbits is used. As 360 is divisible by many numbers, it is possible to accommodate many different repetition intervals. For sake of simplicity, the 360 orbits are divided in 24 blocks of 15 orbits, each block corresponding to approximately 25 hours, or roughly to a day. A 'week' is defined to be 6 of these 15-orbit blocks and a 'month' as 4 of these weeks. This allows for easy definition of calibration measurements that have (roughly) daily, weekly, biweekly or monthly repetition cycles.

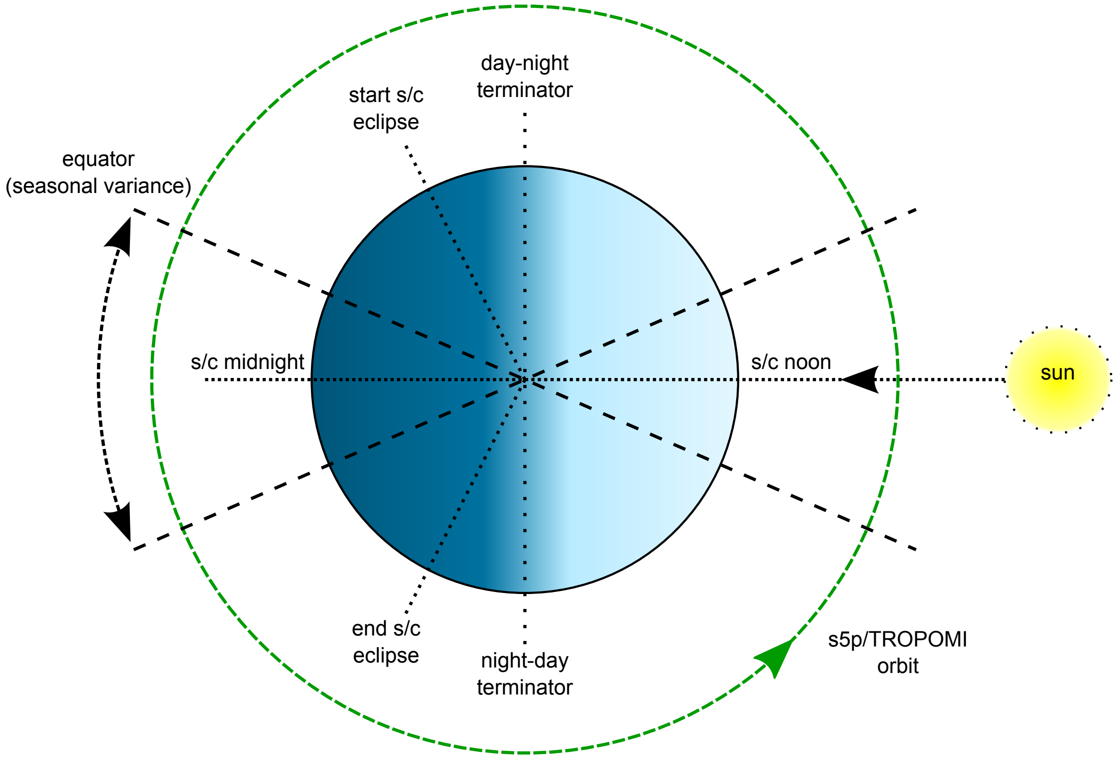

Figure 2: Sentinel-5p orbit overview

4.3.1 Co-addition and small pixels

The signals detected by the spectrometers are digitized in the detector electronics modules (DEMs). The data is saved and co-added in the instrument specific modules (ISMs) in the instrument control unit (ICU). The number of those pixels to be co-added for each detector half (or band) is individually programmable between 0 and 512. It is possible to co-add up to 256 consecutive images. The two halves of one detector can use different co-addition factors.

Information concerning the individual signals of a pixel that contribute (i.e. add up to) to a co-addition is lost, with one exception. One configurable detector pixel, in every row, for both detector output chains, i.e., two columns per detector, is also stored separately for every exposure/co-addition of an image. The data for these ‘small-pixel columns’ are included in the science data and provide information on a higher spatial resolution than the data for other columns, which may be useful for certain studies.

Clearly, co-addition increases the signal to noise ratio. Pixels in the small pixel columns are excluded from this operation. These pixels provide the only way to get some information about changes in a temporal sense during the co-addition time.

4.3.2 Earth radiance measurements

The Earth radiance measurements form the bulk of the measurements. Apart from the optical properties of the instrument, there is some flexibility in the electronics that determines the Earth radiance ground pixel size. The co-addition period determines the ground pixel size in the along-track direction. Row binning (which is possible for UVN detector modules only) determines the ground pixel size cross-track.

For the Earth radiance measurements, the co-addition period will be set to 1080 ms. This effectively results in a ground pixel size of approximately 7 km along-track. For the SWIR-DEM, which contains a CMOS detector, row binning is not supported. This means that, effectively, the binning factor is 1 for the SWIR bands, resulting in a ground pixel size across-track between 7 km at the center and 34 km at the edges of the across-track field of view.

The binning factors and across-track ground pixel size are summarized in Table 2 (taken from [RD12]).

Band	DEM	Binning factor	Across-track ground pixel size
1	UV	8…16	28 … 68 km
2	UV	1…2	3.5 … 8.5 km
3	UVIS	1…2	3.5 … 8.5 km
4	UVIS	1…2	3.5 … 8.5 km
5	NIR	1…2	3.5 … 8.5 km
6	NIR	1…2	3.5 … 8.5 km
7	SWIR	n/a	7 … 34 km
8	SWIR	n/a	7 … 34 km

Table 2: Binning factors and across-track ground pixel sizes for Earth radiance measurements

4.3.3 Solar irradiance measurements

The Sun is visible in TROPOMI's solar irradiance port every orbit for a period of approximately 1.5 minutes around orbit phases 0.75[1]. Every 15 orbits - approximately once every calendar day - TROPOMI will be commanded to perform a solar irradiance measurement. As the main purpose of the solar irradiance measurement is to calculate top-of-atmosphere reflectance, the solar irradiance measurement follows the same binning scheme as the Earth radiance measurements. The remaining parameters will be optimized for the best signal-to-noise ratio. The signal-to-noise ratio is improved even further by averaging the solar irradiance measurements within the L01b Processor.

4.3.4 Background measurements

The background signal for measurements will be calibrated in-orbit. For this to work, every measurement should have accompanying background measurements in the same orbit. These background measurements are performed using the exact same settings as the measurement they accompany. A different IcID for the background measurement ensures that on-ground it is being processed as a background measurement. The background measurements are performed on the eclipse side of the orbit.

4.3.5 Calibration measurements

Calibration measurements will be performed on the night side of the orbit, outside the SAA. The binning scheme that is used for a calibration measurement depends on the objective of that measurement. Calibration measurements that have a strong relation with Earth radiance measurements will use the same binning scheme as Earth radiance measurements. Most calibration measurements however will use a so-called unbinned scheme, that reads out all the pixels of the detector. For these measurements, the co-addition period may be slightly longer than for Earth radiance measurements, to avoid data rate bottlenecks within the instrument or the platform.

Since for instrument operations, the orbits are defined without any seasonal dependency, only a small part of the orbit is guaranteed to be unaffected by the SAA throughout the seasons. This part of the orbit will be used for calibration measurements, while the remainder of the orbit where the spacecraft is in eclipse will be used for background measurements. This is shown in Figure 3. These background measurements are susceptible for proton radiation too, but the L01b Processor will use a filter to avoid background measurements taken in the SAA being used for in-orbit calibration

[1] The orbit phase is defined as $1/(2\pi)$ times the angle in radians traversed by the spacecraft since spacecraft midnight as seen from the center of the Earth. Spacecraft midnight is the point on the night side of the Earth where the spacecraft crosses the orbital plane of the Earth about the Sun. This makes the orbit phase a quantity that runs from 0 to 1, while the spacecraft moves between each spacecraft midnight.

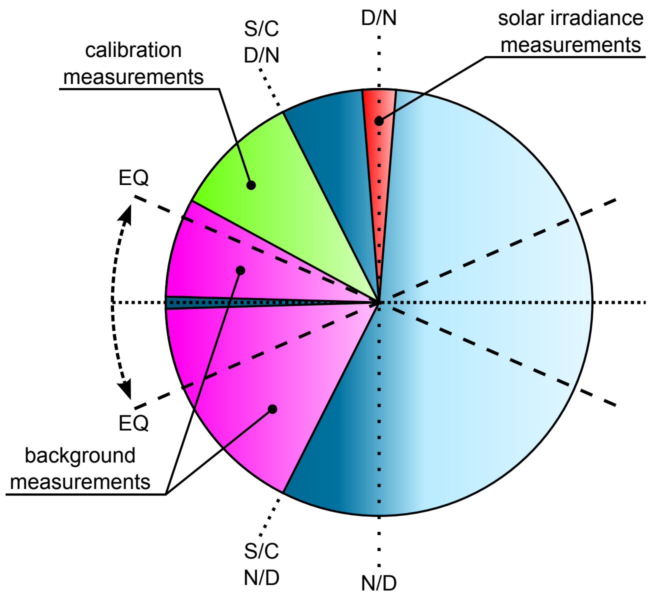

Figure 3: Position of irradiance, calibration and background measurements in the orbit. S/C = spacecraft, EQ = equator, D = day, N = night.

5 Input data products

The main inputs for the L01b are the L0 data products, as described in Table 3. Each of these L0 data products will contain L0 data of a different Application Process Identifier (APID), i.e. 1 APID per product, separate products for each of the APIDs. The L0 product format is specified in [AD2].

Input product	Description
L0__ENG_A	L0 Engineering data (X-Band telemetry)
L0__ODB_1 ... L0__ODB_8	L0 Instrument data for bands 1 through 8
L0__SAT_A	L0 Ancillary data (containing S/C ephemeris and attitude)

Table 3: L0 input products

Another important input for the L01b is the Calibration Key Data. It is foreseen that the Calibration Key Data is provided as a set of data products that has a specified validity range (i.e. the set of orbits to which these Calibration Key Data can be applied and as described in metadata). For the production rules, such a Calibration Key Data Set (CKDS) is treated as an auxiliary data product. The frequency at which the CKDS will be updated depends on the performance of the instrument; a first assumption is that daily updates will be made available. The CKDS is described in [RD10]. An overview of all the auxiliary data products that are currently foreseen is provided in Table 4.

Input product	Description
ICM_CKDUVN	Calibration Key Data Set containing dynamic CKD parameters generated by UVN in-flight calibration processor Generated each orbit.
ICM_CKDSIR	Calibration Key Data Set containing dynamic CKD parameters generated by SWIR in-flight calibration processor. Generated each orbit.
AUX_L1_CKD	Calibration Key Data Set containing semi-static CKD parameters delivered by IDAF system. Generated when nescesarry
IERSB	IERS Bulletin B, see [ER17]. The IERS Bulletin B files can be obtained using anonymous FTP from the IERS public FTP server ftp.iers.org in directory `ftp://ftp.iers.org/products/eop/bulletinb/format_2009/`. These products are generated once per month and are approximately 17 kB in size.
IERSC	IERS Bulletin C, see [ER17]. The IERS Bulletin B files can be obtained using anonymous FTP from the IERS public FTP server ftp.iers.org in directory `ftp://ftp.iers.org/products/eop/bulletinc/`. These products are generated approximately twice per year and are approximately 2 kB in size.

Table 4: L0 auxiliary input products

Finally, there are several static input files that determine the run-time configuration of the L01b. These will be delivered with the L01b and are considered part of the run-time environment of the L01b. These files are, for example, used to tailor the L01b for a specific processing mode. This means that for each of the different modes, there will / can be separate deliveries of the L01b. These deliveries could differ in terms of binaries or in term of these static input files or both.

6 TROPOMI L1b product overview

The Level-1b processor output consists of the following data products:

Level-1b radiance The Level-1b radiance products contain the Earth radiance measurements, including annotation data such as geolocation. For each data granule, typically of the size of one orbit, there is a data product for each of the eight bands. The radiance products are the main input for the Level-2 processors.

Level-1b irradiance The Level-1b irradiance products contain the averaged solar irradiance measurements, including annotation data. For each data granule, there is a data product for each of the two modules, UVN and SWIR. The Level-2 processors will use the irradiance products to calculate reflectance from the Earth radiance data. The irradiance data is used for calibration processing as well. Every 15 orbits - approximately once every calendar day - TROPOMI will be commanded to perform a solar irradiance measurement. If no solar measurements are available in the data granule being processed, no irradiance product will be generated.

Level-1b calibration The Level-1b calibration products contain the calibration and background measurements, including annotation data, as well as any calibration data that are derived from radiance and irradiance measurements. For each data granule, there is a data product for each of the two modules, UVN and SWIR. The calibration products are the main input for the calibration processors that will use these products for generating updates to the calibration key data and for generating trending and monitoring products.

Level-1b engineering The Level-1b engineering products contain the instrument's engineering data converted to physical units. For each data granule, there is a single data product. The engineering products are input for the calibration processors who will use these products for generating updates to the calibration key data and for generating trending and monitoring products. The L1b engineering product is only intended for calibration and monitoring purposes. All instrument information needed or relevant for L2 processing will be contained within the radiance and irradiance products. The operational perspective of the L01b processing chain is depicted in Figure 4.

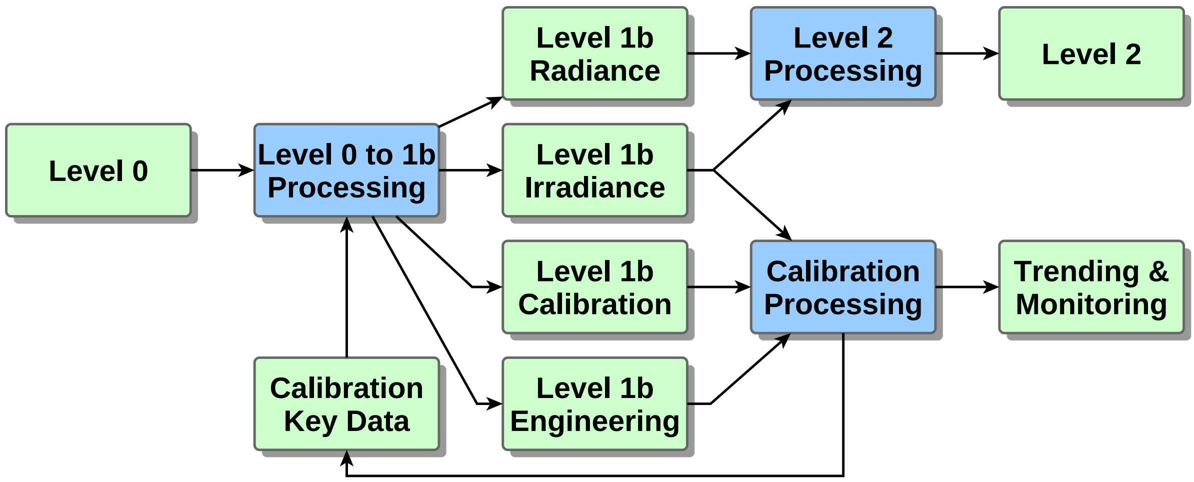

Figure 4: Operational perspective of the L01b processing chain, showing its data products and their position in the processing chain. The blue blocks denote processors; the green blocks denote data products.

The L01b Processor is operationally used in two different modes: **standard product processing** and **near-real-time (NRT) product processing**. The products from standard product processing have the highest quality but less stringent requirements for timeliness. This as opposed to the NRT products, which are required to be available within 2 hours 15 minutes after observation for L1b and 3 hours after observation for L2. To achieve this requirement, speed is favored over quality for the NRT products. The standard products can be distinguished from the NRT products by means of their product or file names and the metadata.

The operational perspective of the NRT processing chain differs from the standard L10b processing chain in that it not includes the generation of irradiance products nor that it involves calibration processing. This is show in Figure 5.

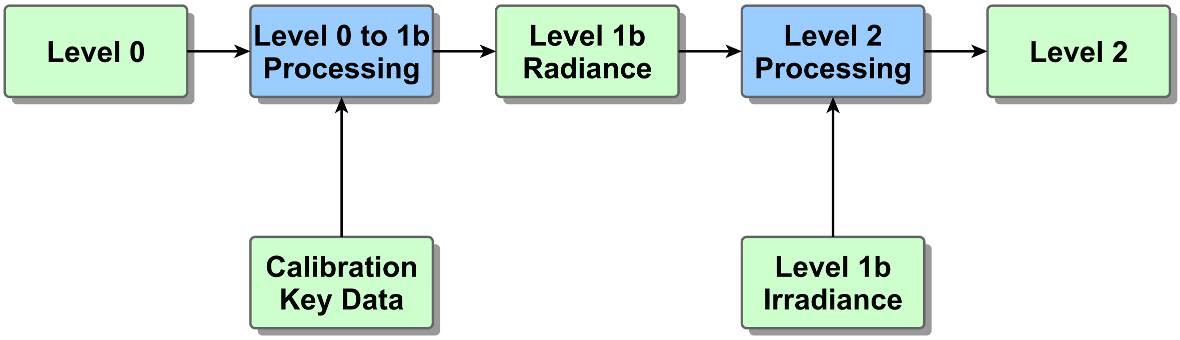

Figure 5: Operational perspective of the NRT processing chain. The blue blocks denote processors; the green blocks denote data products. The irradiance product shown is the result of the standard processing chain.

The data granule (defined as the data time span that is to be processed by the processor) is one orbit for standard product processing. For reasons of efficiency (i.e. data transmission), the volume of one data downlink will be sliced into smaller data volumes. These smaller data volumes form the base of the NRT products, leading approximately to 1 NRT product per data slice.

Table 5 presents an overview of the products: two radiance products will be made for each detector (one for each spectral band). Irradiance and calibration products are instrument module specific.

Instrument module			**UVN**				**SWIR**	
Detector	**UV**		**UVIS**		**NIR**		**SWIR**	
Spectral band	1	2	3	4	5	6	7	8
Radiance product (standard) (# of products/orbit)	1	1	1	1	1	1	1	1
Radiance product (NRT) (# of products/orbit)	10-20	10-20	10-20	10-20	10-20	10-20	10-20	10-20
Irradiance product (# of products/day)			1 (UVN)				1 (SWIR)	
Calibration product (# of products/orbit)			1 (UVN)				1 (SWIR)	
Engineering product (# of products/orbit)				1 (UVN + SWIR)				

Table 5: Overview of L1b products that are generated per day (irradiance product) and per orbit (all other products); The number of NRT products depends on the number of downlink slices and the data volume per slice. The numbers mentions here are indicative.

7 TROPOMI L1b product description

7.1 L1b file structure

The “Earth Observation Ground Station File Format Standard (EO-FFS)” standard [AD04] is relevant to all data files exchanged between ground segment systems within the Earth Observation Missions and as such applicable for the TROPOMI L1b product. This standard provides guidance for data files structures and their encoding, naming and syntax. The EO-FFS standard is used in all recent and upcoming ESA Earth Observations missions, including Sentinel Missions 1, 2 and 3. For the Sentinel 5 Precursor ground segment a tailoring document [AD5] has been made available with a mission specific implementation of the EO-FFS standard.

Within this standard, Earth observation data files are defined as logical files composed of one header and one data block. The logical file can be structured as one physical file or as two physical files separate (i.e. a header file and data block file). The *physical header file* is defined as an XML file containing a fixed part and a variable part of header/metadata information. The header contains configuration control or organizational data. A physical header file has a file name extension “.HDR.”

The *data block* can be either an ASCII/XML file or a binary file. In case of a binary file, a self-describing format is preferred. Binary data blocks are always stored as a separate file with file name extension “.DBL”. For the TROPOMI L1b products the use of a binary data block is applicable containing one netCDF4 file (see section 7.2 for a discussion on netCDF4). In order to be in conformance with the CF-Metadata conventions [RD13], the tailoring permits to use the file name extension “.nc” for the physical filename instead.

For the TROPOMI L1b products this leads to the following convention with respect to the naming of the physical header and the data block files:

header file: `logical_file_name.HDR`

data block file: `logical_file_name.nc`

When these files are distributed the baseline for packaging is “zip”, but other formats (i.e. “gzip”, “tar” or “tar/gzip”) are allowed. However, because of the considerable processing overhead introduced in compressing and decompressing L1b products, it is recommended to use either “zip” without compression or “tar” (uncompressed by definition). In case of packed files the file name extension is “.zip” or “.tar”.

7.1.1 L1b logical file name convention

The files shall be named using a fixed set of elements, each of fixed size, separated by underscores “_”. The file names are composed of a *Mission ID* (<MMM>), a *File Class* (<CCCC>), a *File Type* (<TTTTTTTTTT>) and a *File Instance ID* (<instanceID>):

L1b logical file name: <MMM>_<CCCC>_<TTTTTTTTTT>_<instance ID>

The next subsections show how the L1b logical file name will be based on the S5p tailoring defined in [AD5].

7.1.1.1 Mission ID and File Class

The Mission ID and File Class elements for S5p TROPOMI products are listed in Table 6.

Name	Value	Definition
MMM	S5P	Mission identifier (*fixed value: “S5P”*)
CCCC	[TEST, OGCA, GSOV, OPER, NRTI, OFFL, RPRO]	The file class refers to the type of activity for which the file is used. TEST for internal testing OGCA for on-ground calibration GSOV for ground segment overall validation, system level testing OPER for operational processing NRTI for near-real time processing OFFL for offline processing RPRO for reprocessing

Table 6: Mission identifier and file class specification

7.1.1.2 File Type

The File Type element identifies the product and consists of 10 characters, either uppercase letters, digits or underscores "_". For S5p, the File Type can be subdivided into two sub-elements of respectively 4 and 6 characters, as follows:

File Type: <TTTTTTTTTT> = <FFFF><DDDDDD>

where:

File Category: <FFFF>

Product Semantic Descriptor: <DDDDDD>

File Category The File Category element consists of 4 characters (3 uppercase letters, digits or underscores "_" + 1 underscore "_").

For the S5p TROPOMI L1b products (science data products) the File Category FFFF = L1B_

Product Semantic Descriptor The Product Semantic Descriptor must be unique for a given File Type and be as descriptive as possible. It consists of 6 characters, either uppercase letters, digits or underscores "_".

For S5p L1b data product files (radiance and irradiance data, calibration and engineering products) identified with the File Type element set to "L1B_", the Product Semantic Descriptor is defined as shown in Table 7.

Product Semantic Descriptor	Comment
RA_BD1	Radiance product band 1 (UV detector)
RA_BD2	Radiance product band 2 (UV detector)
RA_BD3	Radiance product band 3 (UVIS detector)
RA_BD4	Radiance product band 4 (UVIS detector)
RA_BD5	Radiance product band 5 (NIR detector)
RA_BD6	Radiance product band 6 (NIR detector)
RA_BD7	Radiance product band 7 (SWIR detector)
RA_BD8	Radiance product band 8 (SWIR detector)
IR_UVN	Irradiance product UVN module
IR_SIR	Irradiance product SWIR module
CA_UVN	Calibration product UVN module
CA_SIR	Calibration product SWIR module
ENG_DB	Engineering product

Table 7: Product Semantic Descriptor for L1b products. See Table 1 for a definition of the bands, modules and detectors.

7.1.1.3 File Instance ID

For science data products (with the File Type “L1B_”), the File Instance ID consists of 63 characters, either uppercase letters, digits or underscores “_”, with the following shape:

File Instance ID: <yyyymmddThhmmss>_<YYYYMMDDTHHMMSS>_<ooooo>_<cc>_<pppppp>_<YYYYMMDDTHHMMSS>

where:

product validity start time: <yyyymmddThhmmss>

product validity stop time: <YYYYMMDDTHHMMSS>

absolute orbit number: <ooooo>

collection number: <cc>

processor version number: <pppppp>

production (start) time: <YYYYMMDDTHHMMSS>

Notes:

- For standard products the product validity start time is set to spacecraft midnight, which is the start time of the orbit. The product validity stop time is set to the end time of that orbit. For near real-time (NRT) products the validity start and stop times are equal to the start and stop time of the data slice.
- The absolute orbit number starts at 00001 (first ascending node crossing after spacecraft separation).
- The collection number stands for a collection of parameters defining the current product (processor version, auxiliary data, and configuration settings) to ease the interpretation of data products by the end users. The collection number starts at 01.
- The processor version number consists of 6 digits, with the first 2 digits for major updates, the next 2 digits for minor updates and the last 2 digits for new releases, i.e. 010203 for processor version 1.2.3.

7.1.1.4 L1b file name examples

Hereafter (Table 8 and Table 9) some file name examples are provided of the logical file name of the different L1b products. The <instance ID> is not provided for readability.

Radiance products (standard and near real time)	Irradiance products
S5P_OPER_L1B_RA_BD1_<instance ID>	S5P_OPER_L1B_IR_UVN_<instance ID>
S5P_NRTI_L1B_RA_BD1_<instance ID>	S5P_OPER_LIB_IR_SIR_<instance ID>

Table 8: Logical file name examples for radiance and irradiance products

Calibration products	Engineering product
S5P_OPER_L1B_CA_UVN_<instance ID>	S5P_OPER_L1B_ENG_DB_<instance ID>
S5P_OPER_L1B_CA_SIR_<instance ID>	

Table 9: Logical file name examples for calibration products and engineering products

Here is a full example of the physical file name for an L1b radiance product in netCDF format, containing the radiance measurements of Band 1 (of the UVN module):

```
S5P_OPER_L1B_RA_BD1_20151114T112005_20151114T125934_00140_02_010203_20151204T093045.nc
```

7.1.2 L1b header file

The header presents the initial part of a logical file, containing descriptive or configuration control information. The header file (XML) distinguishes a Fixed Header and a Variable Header part. Mandatory elements of the Fixed Header part are defined in EO-FFS and are listed in [RD9].

The Variable Header is specific for each File Type. The EO-FFS suggests some desirable elements that may be included in the variable part, such as a description of the data block type (for L1b: *binary*), the name of input files used, if any, to generate the file and a reference to a formal document describing the format and contents of the data block. The information on input data sets and the documentation on format and contents of the data is available in the LI_Lineage metadata element of the DQ_DataQuality core element (see: [RD9]) of the ISO 19115-2 metadata specification. The approach is to copy the ISO 19139 XML representation of the LI_Lineage element into the Variable Header.

The approach for the TROPOMI L1b products is to include all the required metadata information into the product allowing the automated extraction by dedicated tools of XML formatted metadata records that are fully conformant to the INSPIRE standard [RD14], the OGC standard [RD15], which is adopted by ESA and the ESA standard [AD6]. This means that the metadata are integrated into the product independent of a metadata implementation and that tools are required to produce the standardized metadata representations. The implementation specification, including the header information is provided in section 8.2.

7.2 L1b product data structure

For the TROPOMI L1b products the netCDF-4 enhanced model has been selected as the preferred file format. NetCDF (Network Common Data Form)] [ER18] has been developed by the Unidata Program Center at the University Corporation for Atmospheric Research (UCAR) [ER19] and it is used by many scientists and application developers active in the domains of climatology, meteorology and oceanography. The netCDF-4 format is open standard and has been adopted by the Open Geospatial Consortium (OGC).

NetCDF is a data model for array-oriented scientific data. A freely distributed collection of access libraries implementing support for that data model, and a machine-independent format are available. Together, the interfaces, libraries, and format support the creation, access, and sharing of multi dimensional scientific data. NetCDF is self-documenting, which means it can internally store information used to describe the data. For example, the internal documentation can associate various physical quantities (such as temperature, pressure, and humidity) with spatio-temporal locations (such as points at specific latitudes, longitudes, vertical levels, and times). Three different netCDF formats are supported:

- netCDF classic model format
- netCDF 64-bit offset format
- netCDF enhanced data model format (netCDF-4/HDF5 format)

For all netCDF versions (versions 3.x and 4.x) the classic model is the default format. Compared to the classic model, the enhanced model (starting from version 4) offers some important new features such as support for *groups*, (user-defined) *vlen* (variable length) and *compound types* (structures) and *parallel I/O access*.

Although files written using the classic model have the advantage that they may be read by many applications, the use of the enhanced model, supporting groups and structures in particular, offers significant advantages. By the time TROPOMI has been launched, it is expected that many software products will be upgraded in time to support the features of the enhanced data model. Moreover, processing the L1b products to L2 will require dedicated software to be developed using software libraries that are currently available in several languages and already support these features. In view of the above, the enhanced model is used for all L1b products.

In order to support increased interoperability the L1b products shall also comply with the Climate and Forecast (CF) metadata conventions [RD13]. The CF-conventions provide a definitive description of what the data values found in each netCDF variable represent, and of the spatial and temporal properties of the data, including information about grids, such as grid cell bounds and cell averaging methods. This enables users of files from different sources to decide which variables are comparable, and is a basis for building software applications with powerful data extraction, grid remapping, data analysis, and data visualization capabilities.

For data discovery, the metadata of the L1b products shall follow some of the recommendations of the Attribute Convention for Dataset Discovery (ACDD) [ER20]. This convention describes the recommended netCDF attributes for describing a netCDF dataset for use by discovery systems. Tools, such as provided by THREDDS [ER21], will use these attributes for extracting metadata from datasets, and exporting to Dublin

Core, DIF, ADN, FGDC, ISO 19115 etc. metadata formats. In particular, this allows for the export of geospatial metadata in XML according to the ISO 19139 specification, which provides the XML implementation schema for ISO 19115. In the *"Metadata specification for the TROPOMI L1b products"* [RD9] a comprehensive description of these metadata models and how they are applied to the L1b products are given. Section 8 describes how the metadata is stored in the netCDF file, allowing extraction and exporting to different metadata formats.

NOTE: The L01b products can be read by NetCDF version 4.3.1.1 or higher. It also possible to read the L01b product with HDF5 version 1.8.15-patch1 or higher.

7.2.1 NetCDF File Structure

The file format of the L1b products is structured using groups compliant with the netCDF-4 enhanced model. The group hierarchy is as follows ("/" indicating the root of the groups):

```
/
/global attributes
/MetadataGroup [1]
/MetadataGroup/ISOMetadataGroup [1]
/MetadataGroup/EOPMetadataGroup [1]
/MetadataGroup/ESAMetadataGroup [1]
/ProductGroup [1,*]
/ProductGroup/SensorModeGroup [1,*]
/ProductGroup/SensorModeGroup/ObservationsGroup [1]
/ProductGroup/SensorModeGroup/GeodataGroup [1]
/ProductGroup/SensorModeGroup/InstrumentGroup [1]
/ProcessorGroup [1]
```

In the above schema, for each group is indicated how many occurrences of the particular group are expected/allowed in the parent group ([1,*] meaning 1 or more).

This grouping has several benefits:

- Different metadata groups allow for extraction of metadata into XML documents conforming the different metadata specifications.
- ProductGroups allow the combination of observations made by different sensors into one netCDF file (i.e. Band_1 Radiance, Band_2 Radiance, . . .)
- SensorModeGroups allow the combination of observations made by the same sensor operating in different modes (i.e. standard mode, zoom mode, . . .)
- The various subgroups of the SensorModeGroup allow grouping of measurement data, location data, instrument data, processor data and other, simplifying the access to the relevant information depending on the intended use.
- Comprehensive information about configuration items (typically, algorithm and processor parameters) used in processing the data are stored in a separate ProcessorGroup. This information is not documented in detail here, as it is intended to be used only by experts of the L1b processing team.

7.2.2 Naming conventions

7.2.2.1 Groups

Group names are in upper case and consist of alphanumeric characters and underscores. Spaces are not allowed. The group names for the different groups are defined as follows:

MetadataGroup For all products fixed to: `METADATA`

ISOMetadataGroup For all products fixed to: `ISO_METADATA`

EOPMetadataGroup For all products fixed to: `EOP_METADATA`

ESAMetadataGroup For all products fixed to: `ESA_METADATA`

ProductGroup For radiance products one of the following:

BAND1_RADIANCE | BAND2_RADIANCE | BAND3_RADIANCE | BAND4_RADIANCE | BAND5_RADIANCE | BAND6_RADIANCE | BAND7_RADIANCE | BAND8_RADIANCE

For irradiance products one or more of the following:

BAND1_IRRADIANCE | BAND2_IRRADIANCE | BAND3_IRRADIANCE | BAND4_IRRADIANCE | BAND5_IRRADIANCE | BAND6_IRRADIANCE | BAND7_IRRADIANCE | BAND8_IRRADIANCE

SensorModeGroup For all products one of the following:

STANDARD_MODE | SPECIAL_MODE_%J
where: %J equals to the Instrument Configuration ID modulo 4096 (IcID % 4096); more information on the meaning of the IcID is found in sections 8.39 and 8.40)

There is one STANDARD_MODE group. This means that all measurements taken in the standard mode operation are combined even if the standard operation mode is interleaved with operations of the sensor in a special mode.

ObservationsGroup For all products fixed to: OBSERVATIONS

GeodataGroup For all products fixed to: GEODATA

InstrumentGroup For all products fixed to: INSTRUMENT

ProcessorGroup For all products fixed to: PROCESSOR

7.2.2.2 Variables, attributes and dimensions

All variables and dimensions are written in lower case and consist of alphanumeric characters and underscores. Spaces are not allowed.

Unless specified by CF Conventions or ACDD conventions, attributes are written in lower case and consist of alphanumeric characters and underscores. Spaces are not allowed.

7.2.3 Dimensions and coordinate variables

The spectral radiance measurements are collected as a function of the two dimensions (ground pixels across track and wavelengths) of the detector and of the scans. The corresponding dimensions describing the swath in the netCDF product are named: `ground_pixel`, `spectral_channel` and `scanline`, respectively. For reasons of interoperability the dimension time was added with a fixed size of unity as well as a one-element coordinate variable `time(time)` indicating the reference time of the measurements. This reference time is yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start. The `delta_time(scanline)` variable indicates the time difference with the reference time time(time). Thus combining the information of `time(time)` and `delta_time(scanline)` yields the measurement time for each scanline as UTC time.

Following the recommendations of the CF Conventions with respect to the ordering of dimensions having the interpretations of “date or time” (T), “height or depth” (Z), “latitude” (Y) or “longitude” (X), a logical ordering of the dimensions would be (`time, spectral_channel, scanline, ground_pixel`). However, performance tests have shown that given the preferred way of reading through the data, a relative order of (`time, scanline, ground_pixel, spectral_channel`) is preferable; this latter dimension ordering is therefore selected for the variables.

In case of a swath-type scanning pattern as used by TROPOMI, the `scanline` and `ground_pixel` dimensions cannot be referred to as latitude and longitude because they are on a different grid. However, latitude and longitude information can be stored in auxiliary coordinate variables (here: `latitude(time, scanline, ground_pixel)` and `longitude(time, scanline, ground_pixel)`), which are identified by the `coordinates` attribute. By using this convention, applications will be able to process the latitude and longitudes correctly, allowing, for instance, plotting swath-like measurements on a latitude, longitude grid.

One more dimension is defined in the radiance products: `ncorner`. The dimension `ncorner` has a fixed size of 4 and is used for specifying the corner coordinates of the individual ground pixels. The corner coordinates are specified by the `latitude_bounds(time, scanline, ground_pixel, ncorner)` and `longitude_bounds(time, scanline, ground_pixel, ncorner)` variables, which represent the boundaries of each pixel.

Because during the irradiance measurements the sensors are not imaging the Earth's surface but are measuring the solar irradiance, `pixel` is the preferred name for the across-track dimension. Moreover, after correction for the sun elevation the individual irradiance measurements as function of `scanline` are averaged, which results in just one measurement.

Table 10 lists the typical size of the dimensions for different detectors and bands.

Detector	**UV**		**UVIS**		**NIR**		**SWIR**	
Band	1	2	3	4	5	6	7	8
`time`	1	1	1	1	1	1	1	1
`spectral_channel`	497	497	497	497	497	497	480	480
`scanline`	3246	3246	3246	3246	3246	3246	3246	3246
`ground_pixel` (`pixel`)	77	448	450	450	448	448	215	215

Table 10: Typical NetCDF dimension sizes; The scanline dimension varies between orbits and products. A typical value for this size for a radiance product making observations at the day-side of the Earth is 3246. For irradiance products `scanline=1`. The `ground_pixel` dimension is only present in radiance products.

7.3 L1b products

7.3.1 Radiance products

The following tables (Table 11 to Table 13) list all variables of the radiance products as they appear in the different groups. There is no difference between standard and near-real time products. A detailed description in CDL is provided in sections 8.4 to 8.132. The netCDF base types are defined in Table 50.

ObservationsGroup		
Variable	**Type**	**Description**
time	int	Reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start.
spectral_channel	int	Coordinate variable defining the indices in the spectral dimension
scanline	int	Coordinate variable defining the indices along track
ground_pixel	int	Coordinate variable defining the indices across track
delta_time	int	Time difference with time for each measurement
radiance	float	Measured spectral radiance for each spectral pixel
radiance_error	byte	Estimate of the systematic error (accuracy) of the measured spectral radiance (includes calibration and model errors).
radiance_noise	byte	Estimate of the statistical error (precision) of the measured spectral radiance (includes shot noise and read noise).
small_pixel_radiance	float(*)	Measured spectral radiance for the spectral channel dedicated for the small pixel measurements
quality_level	ubyte	Overall quality assessment information for each (spectral) pixel
spectral_channel_quality	ubyte	Quality assessment information for each (spectral) pixel
detector_row_qualification	ushort	Qualification flag indicating the detector row type or state
detector_column_qualification	ushort	Qualification flag indicating column type or state
measurement_quality	ushort	Overall quality information for a measurement
ground_pixel_quality	ubyte	Quality assessment information for each ground pixel

Table 11: NetCDF variables in the ObservationGroup for radiance products

GeodataGroup		
Variable	**Type**	**Description**
`latitude`	float	Latitude of the center of each ground pixel on the WGS84 reference ellipsoid
`latitude_bounds`	float	The four latitude boundaries of each ground pixel.
`longitude`	float	Longitude of the center of each ground pixel on the WGS84 reference ellipsoid
`longitude_bounds`	float	The four longitude boundaries of each ground pixel.
`satellite_orbit_phase`	float	Relative offset (0.0 ... 1.0) of the measurement in the orbit
`satellite_altitude`	float	The altitude of the spacecraft relative to the WGS84 reference ellipsoid
`satellite_latitude`	float	Latitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
`satellite_longitude`	float	Longitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
`viewing_azimuth_angle`	float	Azimuth angle of the spacecraft measured from the ground pixel WGS84 reference ellipsoid.
`viewing_zenith_angle`	float	Zenith angle of the spacecraft measured from the ground pixel location on the WGS84 reference ellipsoid.
`solar_azimuth_angle`	float	Azimuth angle of the sun measured from the ground pixel location on the WGS84 ellipsoid.
`solar_zenith_angle`	float	Zenith angle of the sun measured from the ground pixel location on the WGS84 reference ellipsoid.
`earth_sun_distance`	float	Distance between the Earth and Sun

Table 12: NetCDF variables in the GeodataGroup for radiance products. [Note: Because of the nature of the information the variables `latitude`, `longitude`, `latitude_bounds` and `longitude_bounds` are placed in the GeodataGroup. However, current software applications might have problems to find the auxiliary coordinate variables (in this case latitude and longitude) listed by the coordinates attribute of a variable in the ObservationsGroup.]

InstrumentGroup		
Variable	**Type**	**Description**
`processing_class`	short	High level identification of the type of measurement, for example earth / radiance, sun / irradiance, WLS calibration, LED calibration, dark current / background, etc.
`instrument_configuration`	compound	Identifier (number) that identifies the (detailed) type of measurement and the configuration of the instrument
`instrument_settings`	compound	All fields that determine the instrument configuration and are relevant for data processing, like exposure tme, binning factors, co-addition period, gain settings, status of calibration unit, etc.
`housekeeping_data`	compound	Fields that describe scanline dependent instrument characteristics, like detector temperatures, etc.
`binning_table`	compound	Contains the binning configuration for all of the instrument configurations used in the group. Not present in SWIR products
`nominal_wavelength`	float	The nominal spectral wavelength for each cross track pixel as a function of the spectral channel.
`nominal_wavelength_error`	float	The nominal spectral wavelength standard deviation for each cross track pixel as a function of the spectral channel.

InstrumentGroup (cont'd)		
Variable	**Type**	**Description**
`sample_cycle`	int	Index of cycle. During one sample cycle an integer number of scanlines is collected
`sample_cycle_length`	int	Length of sample_cycle
`measurement_to_detector_-row_table`	compound	Conversion table from measurement row to begin and end row on detector

Table 13: NetCDF variables in the InstrumentGroup for radiance products

7.3.2 Irradiance products

The following tables (Table 14 to Table16) list all variables of the irradiance products. A detailed description in CDL is provided in sections 8.4 to 8.132. The netCDF base types are defined in Table 50.

ObservationsGroup		
Variable	**Type**	**Description**
`time`	int	Reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start.
`spectral_channel`	int	Coordinate variable defining the indices in the spectral dimension
`scanline`	int	Coordinate variable defining the indices along track
`pixel`	int	Coordinate variable defining the indices across track
`delta_time`	int	Time difference with time for each measurement
`irradiance`	float	Measured spectral irradiance for each spectral pixel
`irradiance_error`	byte	Estimate of the systematic error (accuracy) of the measured spectral radiance (includes calibration and model errors).
`irradiance_noise`	byte	Estimate of the statistical error (precision) of the measured spectral irradiance (includes shot noise and read noise)
`quality_level`	ubyte	Overall quality assessment information for each (spectral) pixel
`spectral_channel_quality`	ubyte	Quality assessment information for each (spectral) pixel
`detector_row_-qualification`	ushort	Qualification flag indicating the detector row type or state
`detector_column_-qualification`	ushort	Qualification flag indicating column type or state
`measurement_quality`	ushort	Overall quality information for a measurement

Table 14: NetCDF variables in the ObservationGroup for irradiance products

GeodataGroup		
Variable	**Type**	**Description**
`earth_sun_distance`	float	Distance between the Earth and Sun

Table 15: NetCDF variables in the GeodataGroup for irradiance products

InstrumentGroup		
Variable	**Type**	**Description**
`calibrated_wavelength`	float	Calibrated wavelength of each spectral pixel

InstrumentGroup (cont'd)		
Variable	**Type**	**Description**
calibrated_wavelength_-error	float	Calibrated wavelength error of each spectral pixel
processing_class	short	High level identification of the type of measurement, for example earth / radiance, sun / irradiance, WLS calibration, LED calibration, dark current / background, etc.
instrument_configuration	compound	Identifier (number) that identifies the (detailed) type of measurement and the configuration of the instrument
instrument_settings	compound	All fields that determine the instrument configuration and are relevant for data processing, like exposure tme, binning factors, co-addition period, gain settings, status of calibration unit, etc.
housekeeping_data	compound	Fields that describe scanline dependent instrument characteristics, like detector temperatures, etc.
binning_table	compound	Contains the binning configuration for all of the instrument configurations used in the group. Not present in SWIR products
nominal_wavelength	float	The nominal spectral wavelength for each cross track pixel as a function of the spectral channel.
nominal_wavelength_error	float	The nominal spectral wavelength standard deviation for each cross track pixel as a function of the spectral channel.
sample_cycle	int	Index of cycle. During one sample cycle an integer number of scanlines is collected
sample_cycle_length	int	Length of sample_cycle
measurement_to_detector_-row_table	compound	Conversion table from measurement row to begin and end row on detector

Table 16: NetCDF variables in the InstrumentGroup for irradiance products

7.3.3 Calibration products

7.3.3.1 NetCDF File Structure

The calibration product has a different NetCDF file structure than the file structure for radiance and irradiance products described in section 7.2.1. The file format of the L1b engineering product is structured using groups compliant with the netCDF-4 enhanced model. The group hierarchy is as follows ("/" indicating the root of the groups):

```
/
/global attributes
/MetadataGroup [1]
/MetadataGroup/ISOMetadataGroup [1]
/MetadataGroup/EOPMetadataGroup [1]
/MetadataGroup/ESAMetadataGroup [1]
/ProductGroup [1,*]
/ProductGroup/SensorModeGroup [1,*]
/ProductGroup/SensorModeGroup/ObservationsGroup [1]
/ProductGroup/SensorModeGroup/GeodataGroup [1]
/ProductGroup/SensorModeGroup/InstrumentGroup [1]
/ProductGroup/SensorModeGroup/QualityAssessmentGroup [1]
/ProcessorGroup [1]
```

7.3.3.2 Naming conventions

All naming conventions for the groups described in section 7.2.2 apply, except for the groups specified hereafter.

ProductGroup For calibration products one of the following:

BAND1_RADIANCE | BAND2_RADIANCE | BAND3_RADIANCE | BAND4_RADIANCE |
BAND5_RADIANCE | BAND6_RADIANCE | BAND7_RADIANCE | BAND8_RADIANCE |

BAND1_IRRADIANCE | BAND2_IRRADIANCE | BAND3_IRRADIANCE | BAND4_IRRADIANCE |
BAND5_IRRADIANCE | BAND6_IRRADIANCE | BAND7_IRRADIANCE | BAND8_IRRADIANCE |

BAND1_CALIBRATION | BAND2_CALIBRATION | BAND3_CALIBRATION | BAND4_CALIBRATION |
BAND5_CALIBRATION | BAND6_CALIBRATION | BAND7_CALIBRATION | BAND8_CALIBRATION

SensorModeGroup For all products the SensorModeGroup name has the format:

%C_MODE_%J
where: %C is the Processing Class Name (in upper case) (see section B);
and %J equals to the Instrument Configuration ID modulo 4096 (IcID % 4096); more information on the meaning of the IcID is found in sections 8.39 and 8.40).
Example: the band 1 calibration product for the white light source measurements is found in the group: /BAND1_CALIBRATION/WLS_MODE_1806/OBSERVATIONS.

QualityAssessmentGroup For all products fixed to: QUALITY_ASSESSMENT

7.3.3.3 Radiance calibration groups

ObservationsGroup		
Variable	**Type**	**Description**
time	int	Reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start.
delta_time	int	Time difference with time for each measurement
detector_row_-qualification	ushort	Qualification flag indicating the detector row type or state
detector_column_-qualification	ushort	Qualification flag indicating column type or state
measurement_quality	ushort	Overall quality information for a measurement
ground_pixel_quality	ubyte	Quality assessment information for each ground pixel
radiance_avg	float	Averaged measured spectral radiance for each spectral pixel of all measurements in the group
radiance_avg_error	float	Average radiance signal error for each spectral pixel of all measurements in the group
radiance_avg_noise	float	Average radiance signal noise for each spectral pixel of all measurements in the group
radiance_avg_spectral_-channel_quality	ubyte	Quality assessment information about a (spectral) pixel in all measurements.
radiance_avg_quality_-level	ubyte	Overall calculated quality assessment information for each (spectral) pixel in the averaged data
radiance_avg_std	float	Average radiance signal standard deviation for each spectral pixel of all measurements in the group
radiance_avg_row	float	Averaged measured spectral radiance value of a single row in a measurement
radiance_avg_col	float	Averaged measured spectral radiance value of a single column in a measurement

ObservationsGroup (cont'd)		
Variable	**Type**	**Description**
radiance_avg_data	float	Averaged measured spectral radiance value of a single measurements
small_pixel_radiance	float(*)	Measured spectral radiance for the spectral channel dedicated for the small pixel measurements
scanline	int	Coordinate variable defining the indices along track
ground_pixel	int	Coordinate variable defining the indices across track
spectral_channel	int	Coordinate variable defining the indices in the spectral dimension

Table 17: NetCDF variables in the ObservationGroup for radiance calibration products

GeodataGroup		
Variable	**Type**	**Description**
satellite_orbit_phase	float	Relative offset (0.0 ... 1.0) of the measurement in the orbit
satellite_altitude	float	The altitude of the spacecraft relative to the WGS84 reference ellipsoid
satellite_latitude	float	Latitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
satellite_longitude	float	Longitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
solar_azimuth_angle	float	Azimuth angle of the sun measured from the ground pixel location on the WGS84 ellipsoid.
solar_zenith_angle	float	Zenith angle of the sun measured from the ground pixel location on the WGS84 reference ellipsoid.
earth_sun_distance	float	Distance between the Earth and Sun
latitude	float	Latitude of the center of each ground pixel on the WGS84 reference ellipsoid
longitude	float	Longitude of the center of each ground pixel on the WGS84 reference ellipsoid
latitude_bounds	float	The four latitude boundaries of each ground pixel.
longitude_bounds	float	The four longitude boundaries of each ground pixel.
viewing_azimuth_angle	float	Azimuth angle of the spacecraft measured from the ground pixel WGS84 reference ellipsoid.
viewing_zenith_angle	float	Zenith angle of the spacecraft measured from the ground pixel location on the WGS84 reference ellipsoid.

Table 18: NetCDF variables in the GeodataGroup for radiance calibration products

InstrumentGroup		
Variable	**Type**	**Description**
processing_class	short	High level identification of the type of measurement, for example earth / radiance, sun / irradiance, WLS calibration, LED calibration, dark current / background, etc.
instrument_configuration	compound	Identifier (number) that identifies the (detailed) type of measurement and the configuration of the instrument
instrument_settings	compound	All fields that determine the instrument configuration and are relevant for data processing, like exposure tme, binning factors, co-addition period, gain settings, status of calibration unit, etc.

InstrumentGroup (cont'd)		
Variable	**Type**	**Description**
housekeeping_data	compound	Fields that describe scanline dependent instrument characteristics, like detector temperatures, etc.
nominal_wavelength	float	The nominal spectral wavelength for each cross track pixel as a function of the spectral channel.
nominal_wavelength_error	float	The nominal spectral wavelength standard deviation for each cross track pixel as a function of the spectral channel.
binning_table	compound	Contains the binning configuration for all of the instrument configurations used in the group. Not present in SWIR products
measurement_to_detector_-row_table	compound	Conversion table from measurement row to begin and end row on detector
sample_cycle	int	Index of cycle. During one sample cycle an integer number of scanlines is collected
sample_cycle_length	int	Length of sample_cycle

Table 19: NetCDF variables in the InstrumentGroup for radiance calibration products

QualityAssessmentGroup		
Variable	**Type**	**Description**
detector_pixel_filling_-histogram	int	Histogram of the detector pixel filling in electrons for each scanline
offset_static_ckd	compound	Detector and electronics offset value, obtained from the calibration key-data
offset_prepostscan_-pixels	compound	Detector and electronics offset value calculated from the detector's pre- and postscan pixels
offset_readout_register	compound	Detector and electronics offset value calculated from the detector's read-out register
offset_overscan_rows	compound	Detector and electronics offset value calculated from the detector's overscan rows
offset_overscan_columns	compound	Detector and electronics offset value calculated from the detector's overscan columns
monitor_smear_observed	float	Observed detector smear values from the masked regions of the detector, for monitoring purposes
monitor_smear_calculated	float	Calculated detector smear values as used for the detector smear correction, for monitoring purposes
monitor_straylight_-observed	float	Observed stray light from the stray light areas on the detector, for monitoring purposes
monitor_straylight_-calculated	float	Calculated stray light, for monitoring purposes
monitor_overscan_rows	float	Signal from the detector's overscan rows, for monitoring purposes
monitor_read_out_-register	float	Spectral channel signal values as read from the read out register
monitor_radiance_-wavelength_shift	float	Wavelenght shift for a small wavelength band around the specified wavelength, for monitoring purposes
monitor_radiance	float	Average radiance of a small wavelength band around the specified wavelength, for monitoring purposes.

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
monitor_gain_alignment_factor	float	Gain alignment factor for the measurement calculated in the GainAlignmentCalculationUVN algorithm. Applied gain alignment factor depends on the settings of this algorithm. Default, the CKD setting of the gain alignment correction factor is used, not the calculated.
percentage_spectral_channels_missing	float	Percentage of spectral channels for which the missing flag is set
percentage_spectral_channels_defective	float	Percentage of spectral channels for which the defective flag is set
percentage_spectral_channels_processing_error	float	Percentage of spectral channels for which the processing error flag is set
percentage_spectral_channels_saturated	float	Percentage of spectral channels for which the saturated flag is set
percentage_spectral_channels_transient	float	Percentage of spectral channels for which the transient flag is set
percentage_spectral_channels_rts	float	Percentage of spectral channels for which the RTS flag is set
percentage_spectral_channels_underflow	float	Percentage of spectral channels for which the underflow flag is set
percentage_spectral_channels_per_scanline_missing	float	Percentage of spectral channels per scanline for which the missing flag is set
percentage_spectral_channels_per_scanline_defective	float	Percentage of spectral channels per scanline for which the defective flag is set
percentage_spectral_channels_per_scanline_processing_error	float	Percentage of spectral channels per scanline for which the processing error flag is set
percentage_spectral_channels_per_scanline_saturated	float	Percentage of spectral channels per scanline for which the saturated flag is set
percentage_spectral_channels_per_scanline_transient	float	Percentage of spectral channels per scanline for which the transient flag is set
percentage_spectral_channels_per_scanline_rts	float	Percentage of spectral channels per scanline for which the RTS flag is set
percentage_spectral_channels_per_scanline_underflow	float	Percentage of spectral channels per scanline for which the underflow flag is set
percentage_scanlines_with_processing_steps_skipped	float	Percentage of scanlines for which one or more processing steps were skipped
percentage_scanlines_with_residual_correction_skipped	float	Percentage of scanlines for which residual correction was skipped
percentage_scanlines_in_south_atlantic_anomaly	float	Percentage of scanlines in the South Atlantic Anomaly (SAA)
percentage_scanlines_in_spacecraft_manoeuvre	float	Percentage of scanlines affected by spacecraft manoeuvres

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
percentage_scanlines_-with_solar_angles_out_-of_nominal_range	float	Percentage of scanlines for which the solar angles are outside the nominal range
percentage_ground_-pixels_descending_side_-orbit	float	Percentage of ground pixels on the descending side of the orbit
percentage_ground_-pixels_geolocation_error	float	Percentage of ground pixels with geolocation error
percentage_ground_-pixels_geometric_-boundary_crossing	float	Percentage of ground pixels that cross a geometric boundary, e.g. dateline crossing
percentage_ground_-pixels_night	float	Percentage of ground pixels for which the night flag is set
percentage_ground_-pixels_solar_eclipse	float	Percentage of ground pixels for which the solar eclipse flag is set
percentage_ground_-pixels_sun_glint	float	Percentage of ground pixels for which the sun glint flag is set
oob_sl_nir_corr_row_avg_-blu_rad	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-blu_rad	float	Calculated oob straylight nir dp factor, blue side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-red_rad	float	Calculated oob straylight nir correction row average, red side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_rad	float	Calculated oob straylight nir dp factor, red side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-red_irr	float	Calculated oob straylight nir correction row average, red side irradiance, for monitoring purposes
oob_sl_nir_dp_factor_-blu_irr	float	Calculated oob straylight nir dp factor, blue side irradiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_irr	float	Calculated oob straylight nir dp factor, red side irradiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-blu_irr	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes

Table 20: NetCDF variables in the QualityAssessmentGroup for radiance calibration products

7.3.3.4 Irradiance calibration groups

ObservationsGroup		
Variable	**Type**	**Description**
time	int	Reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start.
spectral_channel	int	Coordinate variable defining the indices in the spectral dimension
scanline	int	Coordinate variable defining the indices along track
pixel	int	Coordinate variable defining the indices across track
delta_time	int	Time difference with time for each measurement
irradiance	float	Measured spectral irradiance for each spectral pixel

ObservationsGroup (cont'd)		
Variable	**Type**	**Description**
irradiance_error	byte	Estimate of the systematic error (accuracy) of the measured spectral radiance (includes calibration and model errors).
irradiance_noise	byte	Estimate of the statistical error (precision) of the measured spectral irradiance (includes shot noise and read noise)
quality_level	ubyte	Overall quality assessment information for each (spectral) pixel
spectral_channel_quality	ubyte	Quality assessment information for each (spectral) pixel
detector_row_-qualification	ushort	Qualification flag indicating the detector row type or state
detector_column_-qualification	ushort	Qualification flag indicating column type or state
measurement_quality	ushort	Overall quality information for a measurement

Table 21: NetCDF variables in the ObservationGroup for irradiance calibration products

GeodataGroup		
Variable	**Type**	**Description**
satellite_orbit_phase	float	Relative offset (0.0 ... 1.0) of the measurement in the orbit
satellite_altitude	float	The altitude of the spacecraft relative to the WGS84 reference ellipsoid
satellite_latitude	float	Latitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
satellite_longitude	float	Longitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
solar_azimuth_angle	float	Azimuth angle of the sun measured from the instrument
solar_elevation_angle	float	Elevation angle of the sun measured from the instrument.
earth_sun_distance	float	Distance between the Earth and Sun

Table 22: NetCDF variables in the GeodataGroup for irradiance calibration products

InstrumentGroup		
Variable	**Type**	**Description**
processing_class	short	High level identification of the type of measurement, for example earth / radiance, sun / irradiance, WLS calibration, LED calibration, dark current / background, etc.
instrument_configuration	compound	Identifier (number) that identifies the (detailed) type of measurement and the configuration of the instrument
instrument_settings	compound	All fields that determine the instrument configuration and are relevant for data processing, like exposure tme, binning factors, co-addition period, gain settings, status of calibration unit, etc.
housekeeping_data	compound	Fields that describe scanline dependent instrument characteristics, like detector temperatures, etc.
nominal_wavelength	float	The nominal spectral wavelength for each cross track pixel as a function of the spectral channel.
nominal_wavelength_error	float	The nominal spectral wavelength standard deviation for each cross track pixel as a function of the spectral channel.

InstrumentGroup (cont'd)		
Variable	**Type**	**Description**
binning_table	compound	Contains the binning configuration for all of the instrument configurations used in the group. Not present in SWIR products
calibrated_wavelength	float	Calibrated wavelength of each spectral pixel
calibrated_wavelength_-error	float	Calibrated wavelength error of each spectral pixel
measurement_to_detector_-row_table	compound	Conversion table from measurement row to begin and end row on detector
sample_cycle	int	Index of cycle. During one sample cycle an integer number of scanlines is collected
sample_cycle_length	int	Length of sample_cycle

Table 23: NetCDF variables in the InstrumentGroup for irradiance calibration products

QualityAssessmentGroup		
Variable	**Type**	**Description**
detector_pixel_filling_-histogram	int	Histogram of the detector pixel filling in electrons for each scanline
offset_static_ckd	compound	Detector and electronics offset value, obtained from the calibration key-data
offset_prepostscan_-pixels	compound	Detector and electronics offset value calculated from the detector's pre- and postscan pixels
offset_readout_register	compound	Detector and electronics offset value calculated from the detector's read-out register
offset_overscan_rows	compound	Detector and electronics offset value calculated from the detector's overscan rows
offset_overscan_columns	compound	Detector and electronics offset value calculated from the detector's overscan columns
monitor_smear_observed	float	Observed detector smear values from the masked regions of the detector, for monitoring purposes
monitor_smear_calculated	float	Calculated detector smear values as used for the detector smear correction, for monitoring purposes
monitor_straylight_-observed	float	Observed stray light from the stray light areas on the detector, for monitoring purposes
monitor_straylight_-calculated	float	Calculated stray light, for monitoring purposes
monitor_overscan_rows	float	Signal from the detector's overscan rows, for monitoring purposes
monitor_gain_alignment_-factor	float	Gain alignment factor for the measurement calculated in the GainAlignmentCalculationUVN algorithm. Applied gain alignment factor depends on the settings of this algorithm. Default, the CKD setting of the gain alignment correction factor is used, not the calculated.
monitor_read_out_-register	float	Spectral channel signal values as read from the read out register
percentage_spectral_-channels_missing	float	Percentage of spectral channels for which the missing flag is set
percentage_spectral_-channels_defective	float	Percentage of spectral channels for which the defective flag is set

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
percentage_spectral_-channels_processing_-error	float	Percentage of spectral channels for which the processing error flag is set
percentage_spectral_-channels_saturated	float	Percentage of spectral channels for which the saturated flag is set
percentage_spectral_-channels_transient	float	Percentage of spectral channels for which the transient flag is set
percentage_spectral_-channels_rts	float	Percentage of spectral channels for which the RTS flag is set
percentage_spectral_-channels_underflow	float	Percentage of spectral channels for which the underflow flag is set
percentage_spectral_-channels_per_scanline_-missing	float	Percentage of spectral channels per scanline for which the missing flag is set
percentage_spectral_-channels_per_scanline_-defective	float	Percentage of spectral channels per scanline for which the defective flag is set
percentage_spectral_-channels_per_scanline_-processing_error	float	Percentage of spectral channels per scanline for which the processing error flag is set
percentage_spectral_-channels_per_scanline_-saturated	float	Percentage of spectral channels per scanline for which the saturated flag is set
percentage_spectral_-channels_per_scanline_-transient	float	Percentage of spectral channels per scanline for which the transient flag is set
percentage_spectral_-channels_per_scanline_-rts	float	Percentage of spectral channels per scanline for which the RTS flag is set
percentage_spectral_-channels_per_scanline_-underflow	float	Percentage of spectral channels per scanline for which the underflow flag is set
percentage_scanlines_-with_processing_steps_-skipped	float	Percentage of scanlines for which one or more processing steps were skipped
percentage_scanlines_-with_residual_-correction_skipped	float	Percentage of scanlines for which residual correction was skipped
percentage_scanlines_in_-south_atlantic_anomaly	float	Percentage of scanlines in the South Atlantic Anomaly (SAA)
percentage_scanlines_in_-spacecraft_manoeuvre	float	Percentage of scanlines affected by spacecraft manoeuvres
percentage_scanlines_-with_solar_angles_out_-of_nominal_range	float	Percentage of scanlines for which the solar angles are outside the nominal range
oob_sl_nir_dp_factor_-blu_irr	float	Calculated oob straylight nir dp factor, blue side irradiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-blu_irr	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_irr	float	Calculated oob straylight nir dp factor, red side irradiance, for monitoring purposes

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
oob_sl_nir_corr_row_avg_-red_irr	float	Calculated oob straylight nir correction row average, red side irradiance, for monitoring purposes
oob_sl_nir_dp_factor_-blu_rad	float	Calculated oob straylight nir dp factor, blue side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_rad	float	Calculated oob straylight nir dp factor, red side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-red_rad	float	Calculated oob straylight nir correction row average, red side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-blu_rad	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes

Table 24: NetCDF variables in the QualityAssessmentGroup for irradiance calibration products

7.3.3.5 Other calibration groups

ObservationsGroup		
Variable	**Type**	**Description**
time	int	Reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start.
delta_time	int	Time difference with time for each measurement
small_pixel_signal	float(*)	Measured signal for the spectral channel dedicated for the small pixel measurements
quality_level	ubyte	Overall quality assessment information for each (spectral) pixel
spectral_channel_quality	ubyte	Quality assessment information for each (spectral) pixel
detector_row_-qualification	ushort	Qualification flag indicating the detector row type or state
detector_column_-qualification	ushort	Qualification flag indicating column type or state
measurement_quality	ushort	Overall quality information for a measurement
signal	float	Measured signal for the spectral channel
signal_error	byte	Estimate of the systematic error (accuracy) of the measured signal (includes calibration and model errors).
signal_noise	byte	Estimate of the statistical error (precision) of the measured signal (includes shot noise and read noise)
signal_avg	float	Averaged measured spectral signal for each spectral pixel of all measurements in the group
signal_avg_error	float	Average signal error for each spectral pixel of all measurements in the group
signal_avg_noise	float	Average signal noise for each spectral pixel of all measurements in the group
signal_avg_spectral_-channel_quality	ubyte	Quality assessment information about a (spectral) pixel in all measurements.
signal_avg_quality_level	ubyte	Overall calculated quality assessment information for each (spectral) pixel in the averaged data
signal_avg_row	float	Averaged measured spectral signal value of a single row in a measurement

ObservationsGroup (cont'd)		
Variable	**Type**	**Description**
signal_avg_col	float	Averaged measured spectral signal value of a single column in a measurement
signal_avg_data	float	Averaged measured spectral signal value of a single measurement
signal_avg_std	float	Average signal standard deviation for each spectral pixel of all measurements in the group
scanline	int	Coordinate variable defining the indices along track
pixel	int	Coordinate variable defining the indices across track
spectral_channel	int	Coordinate variable defining the indices in the spectral dimension

Table 25: NetCDF variables in the ObservationGroup for calibration products

GeodataGroup		
Variable	**Type**	**Description**
satellite_orbit_phase	float	Relative offset (0.0 ... 1.0) of the measurement in the orbit
satellite_altitude	float	The altitude of the spacecraft relative to the WGS84 reference ellipsoid
satellite_latitude	float	Latitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
satellite_longitude	float	Longitude of the spacecraft sub-satellite point on the WGS84 reference ellipsoid
earth_sun_distance	float	Distance between the Earth and Sun

Table 26: NetCDF variables in the GeodataGroup for calibration products

InstrumentGroup		
Variable	**Type**	**Description**
processing_class	short	High level identification of the type of measurement, for example earth / radiance, sun / irradiance, WLS calibration, LED calibration, dark current / background, etc.
instrument_configuration	compound	Identifier (number) that identifies the (detailed) type of measurement and the configuration of the instrument
instrument_settings	compound	All fields that determine the instrument configuration and are relevant for data processing, like exposure tme, binning factors, co-addition period, gain settings, status of calibration unit, etc.
housekeeping_data	compound	Fields that describe scanline dependent instrument characteristics, like detector temperatures, etc.
nominal_wavelength	float	The nominal spectral wavelength for each cross track pixel as a function of the spectral channel.
nominal_wavelength_error	float	The nominal spectral wavelength standard deviation for each cross track pixel as a function of the spectral channel.
binning_table	compound	Contains the binning configuration for all of the instrument configurations used in the group. Not present in SWIR products
storage_time	float	The time a row has resided in the storage area of the detector during read-out

InstrumentGroup (cont'd)		
Variable	**Type**	**Description**
measurement_to_detector_-row_table	compound	Conversion table from measurement row to begin and end row on detector
sample_cycle	int	Index of cycle. During one sample cycle an integer number of scanlines is collected
sample_cycle_length	int	Length of sample_cycle

Table 27: NetCDF variables in the InstrumentGroup for calibration products

QualityAssessmentGroup		
Variable	**Type**	**Description**
detector_pixel_filling_-histogram	int	Histogram of the detector pixel filling in electrons for each scanline
offset_static_ckd	compound	Detector and electronics offset value, obtained from the calibration key-data
offset_prepostscan_-pixels	compound	Detector and electronics offset value calculated from the detector's pre- and postscan pixels
offset_readout_register	compound	Detector and electronics offset value calculated from the detector's read-out register
offset_overscan_rows	compound	Detector and electronics offset value calculated from the detector's overscan rows
offset_overscan_columns	compound	Detector and electronics offset value calculated from the detector's overscan columns
monitor_smear_observed	float	Observed detector smear values from the masked regions of the detector, for monitoring purposes
monitor_smear_calculated	float	Calculated detector smear values as used for the detector smear correction, for monitoring purposes
monitor_straylight_-observed	float	Observed stray light from the stray light areas on the detector, for monitoring purposes
monitor_straylight_-calculated	float	Calculated stray light, for monitoring purposes
monitor_overscan_rows	float	Signal from the detector's overscan rows, for monitoring purposes
monitor_radiance_-wavelength_shift	float	Wavelenght shift for a small wavelength band around the specified wavelength, for monitoring purposes
monitor_read_out_-register	float	Spectral channel signal values as read from the read out register
monitor_radiance	float	Average radiance of a small wavelength band around the specified wavelength, for monitoring purposes.
monitor_gain_alignment_-factor	float	Gain alignment factor for the measurement calculated in the GainAlignmentCalculationUVN algorithm. Applied gain alignment factor depends on the settings of this algorithm. Default, the CKD setting of the gain alignment correction factor is used, not the calculated.
percentage_spectral_-channels_missing	float	Percentage of spectral channels for which the missing flag is set
percentage_spectral_-channels_defective	float	Percentage of spectral channels for which the defective flag is set
percentage_spectral_-channels_processing_-error	float	Percentage of spectral channels for which the processing error flag is set

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
percentage_spectral_-channels_saturated	float	Percentage of spectral channels for which the saturated flag is set
percentage_spectral_-channels_transient	float	Percentage of spectral channels for which the transient flag is set
percentage_spectral_-channels_rts	float	Percentage of spectral channels for which the RTS flag is set
percentage_spectral_-channels_underflow	float	Percentage of spectral channels for which the underflow flag is set
percentage_spectral_-channels_per_scanline_-missing	float	Percentage of spectral channels per scanline for which the missing flag is set
percentage_spectral_-channels_per_scanline_-defective	float	Percentage of spectral channels per scanline for which the defective flag is set
percentage_spectral_-channels_per_scanline_-processing_error	float	Percentage of spectral channels per scanline for which the processing error flag is set
percentage_spectral_-channels_per_scanline_-saturated	float	Percentage of spectral channels per scanline for which the saturated flag is set
percentage_spectral_-channels_per_scanline_-transient	float	Percentage of spectral channels per scanline for which the transient flag is set
percentage_spectral_-channels_per_scanline_-rts	float	Percentage of spectral channels per scanline for which the RTS flag is set
percentage_spectral_-channels_per_scanline_-underflow	float	Percentage of spectral channels per scanline for which the underflow flag is set
percentage_scanlines_-with_processing_steps_-skipped	float	Percentage of scanlines for which one or more processing steps were skipped
percentage_scanlines_-with_residual_-correction_skipped	float	Percentage of scanlines for which residual correction was skipped
percentage_scanlines_in_-south_atlantic_anomaly	float	Percentage of scanlines in the South Atlantic Anomaly (SAA)
percentage_scanlines_in_-spacecraft_manoeuvre	float	Percentage of scanlines affected by spacecraft manoeuvres
percentage_scanlines_-with_solar_angles_out_-of_nominal_range	float	Percentage of scanlines for which the solar angles are out-side the nominal range
oob_sl_nir_corr_row_avg_-blu_irr	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-blu_rad	float	Calculated oob straylight nir dp factor, blue side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-red_irr	float	Calculated oob straylight nir correction row average, red side irradiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_rad	float	Calculated oob straylight nir dp factor, red side radiance, for monitoring purposes

QualityAssessmentGroup (cont'd)		
Variable	**Type**	**Description**
oob_sl_nir_dp_factor_-blu_irr	float	Calculated oob straylight nir dp factor, blue side irradiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-red_rad	float	Calculated oob straylight nir correction row average, red side radiance, for monitoring purposes
oob_sl_nir_corr_row_avg_-blu_rad	float	Calculated oob straylight nir correction row average, blue side radiance, for monitoring purposes
oob_sl_nir_dp_factor_-red_irr	float	Calculated oob straylight nir dp factor, red side irradiance, for monitoring purposes

Table 28: NetCDF variables in the QualityAssessmentGroup for calibration products

7.3.4 Engineering product

The engineering products are input for the calibration processors who will use these products for generating updates to the calibration key data and for generating trending and monitoring products. The L1b engineering product is only intended for calibration and monitoring purposes. All instrument information needed or relevant for L2 processing will be contained within the radiance and irradiance products. As such, the engineering product is expected to be used by experts investigating and troubleshooting instrument performance anomalies. For that reason, only a high level description of the product is provided here. However, this description together with the detailed information contained in the netCDF and the Command and Telemetry Handbook [RD16] will allow expert users to retrieve the relevant engineering data.

7.3.4.1 NetCDF File Structure

The engineering product has a different NetCDF file structure than the file structure for radiance and irradiance products described in section 7.2.1. The file format of the L1b engineering product is structured using groups compliant with the netCDF-4 enhanced model. The group hierarchy is as follows ("/" indicating the root of the groups):

```
/
/global attributes
/MetadataGroup [1]
/MetadataGroup/ISOMetadataGroup [1]
/MetadataGroup/EOPMetadataGroup [1]
/MetadataGroup/ESAMetadataGroup [1]
/DetectorGroup [4]
/DetectorGroup/BandGroup [2]
/DetectorGroup/DetectorHousekeepingGroup [1]
/MeasurementSetGroup [1]
/NominalHouseKeepingGroup/EventsGroup [1]
/NominalHouseKeepingGroup/MechanismGroup [1]
/NominalHouseKeepingGroup/HeatersGroup [1]
/NominalHouseKeepingGroup/LEDInformationGroup [1]
/NominalHouseKeepingGroup/OBDHGroup [1]
/NominalHouseKeepingGroup/SoftwareConfigurationGroup [1]
/NominalHouseKeepingGroup/TemperaturesGroup [1]
/NominalHouseKeepingGroup/VersionInformationGroup [1]
/NominalHouseKeepingGroup/VoltagesGroup [1]
/ProcessorGroup [1]
/AncillaryDataGroup [1]
/SatelliteInformationGroup[1]
```

All groups in the schema listed above are always present in the netCDF file. The relation between detector type, detector number and bands can be found in Table 10. In the L0 product instrument parameters are available in engineering data packages. The L01b processor extracts all the parameters from these data packages and groups them in variables which are then stored in the netCDF engineering product. The variables are stored in different groups; the groups and the variables they contain are described in the following sections.

7.3.4.2 Naming conventions

Except for the group names, all naming conventions described in section 7.2.2 apply. Group names are in upper case and consist of alphanumeric characters and underscores. Spaces are not allowed. The group names for the different groups are defined as follows:

DetectorGroup fixed to: DETECTOR1 | DETECTOR2 | DETECTOR3 | DETECTOR4

BandGroup fixed to: BAND1 | BAND2 | BAND3 | BAND4 | BAND5 | BAND6 | BAND7 | BAND8

DetectorHousekeepingGroup fixed to: DETECTOR_HK

MeasurementSetGroup fixed to: MSMTSET

NominalHouseKeepingGroup fixed to: NOMINAL_HK

EventsGroup fixed to: EVENTS

MechanismGroup fixed to: MECHANISMS

HeatersGroup fixed to: HEATERS

LEDInformationGroup fixed to LED_DATA

OBDHGroup fixed to OBDH_DATA

SoftwareConfigurationGroup fixed to SW_CFG

TemperaturesGroup fixed to TEMPERATURES

VersionInformationGroup fixed to VERSION_INFO

VoltagesGroup fixed to VOLTAGES

SatelliteInformationGroup fixed to SATELLITE_INFO

AncillaryDataGroup fixed to ANCILLARY_DATA

7.3.4.3 Engineering product groups

The following tables (Table 29 to Table 47) list all variables in the engineering product as they appear in the different groups. A detailed description is outside the scope of this document.

DetectorGroup UVN detector		
Variable	**Type**	**Description**
afe_common_config	compound	Extracted AFE setting common for both bands on the detector
afe_reg_vals	compound	Raw AFE register values from which the AFE parameters are extracted
clock	compound	Clock information of the detector electronics
ft_table	compound	Frame Transfer synchronization parameters as stored in the DEM
heater_cfg	compound	Heater settings for the detector
misc	compound	Miscellaneous parameters for the detector that don't fit in other groups
timing	componud	Detector specific timing parameters. Partly extracted from ft_table parameters

Table 29: NetCDF variables in the DetectorHousekeepingGroup of the engineering product for UVN detectors (detector1 - detector3)

DetectorGroup SWIR detector		
Variable	**Type**	**Description**
ft_table	compound	Frame Transfer synchronization parameters as stored in the DEM
swir_settings	compound	Extracted detector settings which are valid for both bands read from the SWIR Detector
timing	componud	Detector specific timing parameters. Partly extracted from ft_table parameters

Table 30: NetCDF variables in the DetectorGroup of the engineering product for the SWIR detector (detector4)

BandGroup UVN detector		
Variable	**Type**	**Description**
afe_band_cfg	compound	Band specific settings extracted from the AFE registers.
readout_cfg	compound	Read-out settings for the band as used by the ISM.

Table 31: NetCDF variables in the BandGroup for the engineering product for the UVN detectors (detector1-detector3)

BandGroup SWIR detector		
Variable	**Type**	**Description**
readout_cfg	compound	Read-out settings for the band as used by the ISM.

Table 32: NetCDF variables in the BandGroup for the engineering product for the SWIR detector (detector4)

DetectorHousekeepingGroup UVN detector		
Variable	**Type**	**Description**
dem_cntrs	compound	DEM counter values for the detector
power_info	compound	Current and voltage values specific for the detector
spare_info	compound	Spare values in detector Housekeeping data
stat_info	compound	Extracted status values specific for the detector
temperature_info	compound	Temperature values specific for the detector
version_info	compound	DEM firmware version information for the detector

Table 33: NetCDF variables in the DetectorHousekeepingGroup for the engineering product for the UVN detectors (detector1-detector3)

DetectorHousekeepingGroup SWIR detector		
Variable	**Type**	**Description**
adc_info	compound	Extracted current SWIR ADC values
dem_cntrs	compound	DEM counter values for the detector
err_cntrs	compound	SWIR specific error counters
heater_data	compound	Information about SWIR detector internal heater settings
power_info	compound	Current and voltage values specific for the detector
stat_info	compound	Extracted status values specific for the detector
temperature_info	compound	Temperature values specific for the detector
tmtc_info	compound	Extracted TMTC counter values

DetectorHousekeepingGroup SWIR detector (cont'd)		
Variable	**Type**	**Description**
misc	compound	Miscellaneous parameters for the detector that don't fit in other groups

Table 34: NetCDF variables in the DetectorHousekeepingGroup for the engineering product for the SWIR detector (detector4)

MeasurementSetGroup		
Variable	**Type**	**Description**
msmtset	compound	Measurement set information of all engineering data packages, like processing class, instrument configuration and selected DEM and ISM tables

Table 35: NetCDF variables in the MeasurementSetGroup for the engineering product

EventsGroup		
Variable	**Type**	**Description**
events	compound	General event information extracted from the housekeeping data
processing_events	compound	Processing event information extracted from the housekeeping data

Table 36: NetCDF variables in the EventsGroup for the engineering product

MechanismsGroup		
Variable	**Type**	**Description**
fmm	compound	Status information about the folding mirror mechanism on the instrument
difm	compound	Status information about the diffuser mechanism on the instrument

Table 37: NetCDF variables in the FoldingMirrorsGroup for the engineering product

HeatersGroup		
Variable	**Type**	**Description**
heater_data	compound	Settings and status information about the heaters on the instrument
peltier_info	compound	Settings and status information about the peltier elements on the instrument

Table 38: NetCDF variables in the HeatersGroup for the engineering product

LEDInformationGroup		
Variable	**Type**	**Description**
led_data	compound	Status and voltage information about the LEDs on the instrument

Table 39: NetCDF variables in the LEDInformationGroup for the engineering product

OBDHGroup		
Variable	**Type**	**Description**
`obdh_data`	compound	Onboard data handling data parameters

Table 40: NetCDF variables in the OBDHGroup for the engineering product

SoftwareConfigurationGroup		
Variable	**Type**	**Description**
`params`	compound	Software configuration parameters of the instrument.

Table 41: NetCDF variables in the SoftwareConfigurationGroup for the engineering product

TemperaturesGroup		
Variable	**Type**	**Description**
`hires_temperatures`	compound	Calculated temperatures of the high-resolution temperature sensors on the instrument
`instr_temperatures`	compound	Calculated temperatures of the instrument temperature sensor on the instrument
`named_temperatures`	compound	Calculated temperatures of named sensors in the engineering data
`reference_thermistors`	compound	Calculated resistor values of the reference thermistors

Table 42: NetCDF variables in the TemperaturesGroup for the engineering product

VersionInformationGroup		
Variable	**Type**	**Description**
`info`	compound	Version information of onboard software of the instrument

Table 43: NetCDF variables in the VersionInformationGroup for the engineering product

VoltagesGroup		
Variable	**Type**	**Description**
`detector1_voltages`	compound	Voltages measured for detector1
`detector2_voltages`	compound	Voltages measured for detector2
`detector3_voltages`	compound	Voltages measured for detector3
`detector4_voltages`	compound	Voltages measured for detector4
`instrument_voltages`	compound	Instrument voltages

Table 44: NetCDF variables in the VoltagesGroup for the engineering product

ProcessorGroup		
Variable	**Type**	**Description**
`job_configuration`	string array	Joborder used for generating this l1b file
`algorithm_configuration`	string array	Algorithm table used for generating this l1b file

VoltagesGroup (cont'd)		
Variable	**Type**	**Description**
processing_configuration	string array	Processing configuration used for generating this l1b file

Table 45: NetCDF variables in the ProcessorGroup for the engineering product

AncillaryDataGroup		
Variable	**Type**	**Description**
aocs_data	compound	AOCS values of the satellite platform
attitude_data	compound	Attitude data in of the satellite platform
gps_satellite_data	compound	GPS satellite data of the satellite platform
navigation_data	compound	Navigation data of the satellite platform
propagated_gps_pos_data	compound	Propagated GPS position of the satellite platform
star_tracker_-configuration	compound	Star tracker configuration of the satellite platform
temperatures	compound	Temperatures measured on the satellite platform

Table 46: NetCDF variables in the AncillaryDataGroup for the engineering product

SatelliteInformationGroup		
Variable	**Type**	**Description**
satellite_pos	compound	Instrument position information calculated by the L1b processor.

Table 47: NetCDF variables in the ProcessorGroup for the engineering product

8 TROPOMI L1b product specification

8.1 NetCDF4 global attributes

In the “Metadata specification for the TROPOMI L1b products” [RD9] it is discussed how metadata content can be provided by the use of global attributes, thereby facilitating the discovery and understanding of the dataset. The CF-Metadata conventions [RD13] and the Attribute Conventions for Dataset Discovery [ER20] recommend a comprehensive set of attributes to be included as metadata elements. However, for TROPOMI L1b products it was decided to create specific metadata groups in which INSPIRE (ISO), ESA EOP and ESA FFS related metadata information is stored. Many of the metadata attributes proposed by CF-Metadata Conventions and ACDD overlap with the ISO 19115-2 standard and hence the same information can be found in the metadata groups.

In view of the above, only a very limited set of metadata elements is included as global attributes. These attributes provide a convenient way to users of the data products to retrieve quickly some basic information. In Table 48 a list is presented of metadata items included as global attributes in the netCDF product file.

Attribute	ISO mapping	Remark
`Conventions`		fixed: “CF-1.6”
`title`	MI_Metadata.identificationInfo.citation.title	
`summary`	MI_Metadata.identificationInfo.abstract	
`institution`	MI_Metadata.identificationInfo.pointOfContact.organisationName	
`time_coverage_start`	MI_Metadata.identificationInfo.extent.temporalElement.beginPosition	UTC time (start of measurements)
`time_coverage_end`	MI_Metadata.identificationInfo.extent.temporalElement.endPosition	UTC time (end of measurements)
`time_reference`		UTC time (reference time = “yyyy-mm-ddT00:00:00Z”)
`orbit`		orbit number at which measurements of the data granule start
`orbit_begin_icid`	First begin trigger icid encountered in L0 data	
`orbit_end_icid`	Last end trigger icid encountered in L0 data	
`orbit_type`	(E2) Orbit type	"UNKN" is not known
`orbit_type_id`	(E2) orbit type id	65535 if not kown
`institution`		fixed: “KNMI”
`processor_version`	Version of the L01b processor used for generating product	
`library_information`	Version information of the libraries used by L01b processor	

Table 48: Global attributes.
Remark 1: UTC times are in expressed in the ISO 8601 format (i.e. YYYY-MM-DDThh:mm:ssZ).
Remark 2: The values of `time_coverage_start` and `time_coverage_end` truncated to integer seconds refer to the actual start and end of the measurements, i.e. the measurment time of the first and last scanline, respectively. Therefore, these times do not correspond to the times used in the filename of the product, where the start and end of the orbit (or data slice) are used instead (see section 7.1.1.3).
Remark 3: In case there are no scanlines in the processed orbit the values for `time_coverage_start` and `time_coverage_end` are equal to “NULL”.
Remark 4: For the definition of the reference time see section 8.4.

8.2 Metadata specification

The netCDF file will have one metadata group (named METADATA) which is a container for specific metadata groups containing metadata information required to produce INSPIRE conformant [RD14], ESA EOP conformant [RD15] and ESA FFS conformant [AD6] XML formatted metadata records. These three specific metadata groups named ISO_METADATA, EOP_METADATA and ESA_METADATA, are structured in subgroups containing only attributes.

The structure of the groups reflects the structure of the particular metadata model, i.e. the groups correspond largely with the major metadata objects of the model. Whenever applicable, the groups contain an attribute with name="objectType" with a value equal to the corresponding object (including namespace) from the metadata model. This approach follows the groups-of-groups approach suggested by [ER22]. In addition, the attributes containing the relevant metadata information are given the same name as the corresponding element of the metadata model.

Details on the metadata can be found in [RD9].

Attribute	Description
`ancillary_variables`	Attribute to express relationship with other variables; For example, to relate instrument data with associated measures of uncertainty.
`bounds`	The name of the variable that contains the vertices of the cell boundaries. Used to relate the variable to a coordinate variable.
`coordinates`	Indicates the spatiotemporal coordinate variables that are needed to geo-locate the data. Contains full path when coordinate variables are not in the same group.
`comment`	Miscellaneous information about the variable or methods used to produce it
`flag_meanings`	The flag_meanings attribute is a string whose value is a blank separated list of descriptive words or phrases, one for each flag value.
`flag_values`	The flag_values attribute is the same type as the variable to which it is attached, and contains a list of the possible flag values
`long_name`	A long descriptive name describing the content of the variable
`standard_name`	A standardized name describing the content of the variable
`units`	A character string that specifies the units used for the variable's data (required for all variables that represent dimensional quantities, except for boundary variables)
`valid_max`	The maximum valid value for the variable
`valid_min`	The minimum valid value for the variable
`_FillValue`	The FillValue attribute specifies the fill value used for missing or undefined data

Table 49: Description of variable attributes

8.3 Fill values

The CF convention recommends to use the _FillValue attribute (or to use the default values) to assign a specific value to NetCDF variables in case of undefined or missing data. The `_FillValue` depends on the data type of the variable. The following table (Table 50) lists the values used for the various base data types. In the sections hereafter, the `_FillValue` attribute will only be present in de CDL descriptions if it is different from the default value.

8.4 Variable: `time`

The variable `time(time)` is the reference time of the measurements. The reference time is set to yyyy-mm-ddT00:00:00Z UTC, where yyyy-mm-dd is the day on which the measurements of a particular data granule start. The `delta_time(scanline)` variable (see section 8.5) indicates the time difference with the reference time `time(time)`. Thus combining the information of `time(time)` and `delta_time(scanline)` yields the measurement time for each scanline as UTC time. The variable `time(time)` does (intentionally) not include any leap seconds, to make the conversion from `time(time)` and `delta_time(scanline)` to an UTC time easier.

The reference `time(time)` corresponds to the global attribute `time_reference` which is an UTC time specified as an ISO 8601 date.

Type	Storage	_FillValue
byte	8-bit signed integer	-127
ubyte	8-bit unsigned integer	255
short	16-bit signed integer	-32767
ushort	16-bit unsigned integer	65535
int	32-bit signed integer	-2147483647
float	32-bit floating point	9.9692099683868690e+36 (hex: 0x1.ep+122)
double	64-bit floating point	9.9692099683868690e+36 (hex: 0x1.ep+122)
float(*)	32-bit floating point(*)	9.9692099683868690e+36 (hex: 0x1.ep+122)

Table 50: NetCDF type definitions and fill values. Remark 1: The base type for a VLEN type (Variable Length Array) is indicated as type(*), i.e. float(*), short(*), etc. Remark 2: In order to avoid rounding errors, it is recommended to programmers to use the hexadecimal notation when specifying the above fill values for float and double types.

Variable	Storage type	Units
`time`	int	seconds

CDL

```
int time(time) ;
time:long_name = "reference start time of measurement" ;
time:standard_name = "time" ;
time:units = "seconds since 2010-01-01 00:00:00" ;
time:comment = "Reference time of the measurements.  The reference time is
set to yyyy-mm-ddT00:00:00 UTC, where yyyy-mm-dd is the day on which the
measurements of a particular data granule start." ;
```

Remarks The time is UTC seconds since UTC2010-01-01 00:00:00. The UTC time defined by this variable time corresponds to the global attribute `time_reference`, which is a UTC time specified as an ISO 8601 (i.e. YYYY-MM-DDThh:mm:ssZ).

Table 51: CDL definition `time` variable

8.5 Variable: `delta_time`

The `delta_time(scanline)` variable indicates the time difference with the reference time `time(time)` (see section 8.4). Thus combining the information of `time(time)` and `delta_time(scanline)` yields the measurement time for each scanline as UTC time. The UTC time derived for the first scanline corresponds to the global attribute `time_coverage_start`. Similarly, the UTC time derived for the last scanline corresponds to global attribute `time_coverage_end`. One scanline measurement is the result of adding independent measurements during one co-addition period. The time attributed to the scanline measurement is equal to the center time of the co-addition period defined by the first and last sample in this co-addition.

Variable	Storage type	Units
`delta_time`	int	ms

CDL

```
int delta_time(time,scanline) ;
delta_time:long_name = "offset from the reference start time of
measurement" ;
delta_time:units = "ms" ;
delta_time:comment = "Time difference with time for each measurement" ;
```

Remarks

Table 52: CDL definition `delta_time` variable

8.6 Variable: `ground_pixel`

The coordinate variable `ground_pixel` refers to the across-track dimension of the measurement. The spectral radiance measurements are collected as a function of the two-dimensions (ground pixels across track and wavelengths), of the detector and of the scans. The corresponding dimensions describing the swath in the netCDF product are named: `ground_pixel`, `spectral_channel` and `scanline`, respectively.

	Variable	Storage type	Units
	`ground_pixel`	int	none

CDL	`int ground_pixel(ground_pixel) ;`
	`ground_pixel:long_name = "across track dimension index" ;`
	`ground_pixel:units = "1" ;`
	`ground_pixel:comment = "This dimension variable defines the indices across track; index starts at 0" ;`		

Remarks	Coordinate variable; The `ground_pixel` ordering is from west to east, i.e. a higher index corresponds to a higher longitude value during the ascending part of the orbit.		

Table 53: CDL definition `ground_pixel` variable

8.7 Variable: `pixel`

The coordinate variable `pixel` refers to the across-track dimension of the measurement. Because during the irradiance measurements the sensors are not imaging the Earth's surface but are measuring the solar irradiance, `pixel` is the preferred name (rather than `ground_pixel`) for the across-track dimension.

	Variable	Storage type	Units
	`pixel`	int	none

CDL	`int pixel(pixel) ;`
	`pixel:long_name = "across track dimension index" ;`
	`pixel:units = "1" ;`
	`pixel:comment = "This dimension variable defines the indices across track; index starts at 0" ;`		

Remarks	Coordinate variable. The `pixel` ordering corresponds to the `ground_pixel` order in the radiance products, which is from west to east, i.e. a higher index in corresponds to a higher longitude value during the ascending part of the orbit		

Table 54: CDL definition `pixel` variable

8.8 Variable: `scanline`

The coordinate variable `scanline` refers to the along-track dimension of the measurement. Scanline numbering starts a 0 for each product. (Thus: the scanline value of 0 is not related to a 'fixed' time but to the first measurement in the product.)

	Variable	Storage type	Units
	scanline	int	none
CDL	int scanline(scanline) ; scanline:long_name = "along track dimension index" ; scanline:units = "1" ; scanline:comment = "This dimension variable defines the indices along track; index starts at 0" ;		
Remarks	Coordinate variable. The scanlines are time-ordered; meaning that "earlier" measurements come before "later" measurements		

Table 55: CDL definition `scanline` variable

8.9 Variable: `spectral_channel`

	Variable	Storage type	Units
	spectral_channel	int	none
CDL	int spectral_channel(spectral_channel) ; spectral_channel:long_name = "wavelength dimension index" ; spectral_channel:units = "1" ; spectral_channel:comment = "This dimension variable defines the indices spectral dimension; index starts at 0" ;		
Remarks	Coordinate variable; The spectral channels are ordered by increasing wavelength, i.e. a higher index corresponds to a higher wavelength value.		

Table 56: CDL definition `spectral_channel` variable

8.10 Variable: `radiance`

TROPOMI measures the light radiated from and reflected by the Earth's surface and atmosphere in a given direction. The *spectral radiance* is a measure of the rate of the energy received per unit area an per unit of the solid angle as a function of wavelength and is expressed in SI units W.m^{-2}.nm^{-1}.sr^{-1}. Because TROPOMI actually measures the rate of photons per unit area and the exact wavelength is not known the *spectral photon radiance* is provided in the L1b product. The spectral photon radiance is expressed with SI units mol.s^{-1}.m^{-2}.nm^{-1}.sr^{-1} using the amount of photons. [2] In addition, the spectral photon radiance provided is normalized to the Earth-Sun distance of 1AU. [3] If the Earth spectral radiance is denoted by S_{earth}, the wavelength by λ and the Earth-Sun distance by R, then the Earth spectral radiance normalized at 1AU is given by:

$$S_{\text{earth}}(R_{\text{AU}},\lambda) = \left(\frac{R}{R_{\text{AU}}}\right)^2 S_{\text{earth}}(R,\lambda)\ , \tag{1}$$

where R_{AU} is the Earth-Sun distance equal to 1AU. Similarly, the spectral photon radiance is normalized using the factor $\left(\frac{R}{R_{\text{AU}}}\right)^2$.

[2] 1 Mole (unit symbol *mol*) corresponds to Avogadro's number N_A and is equal to 6.02214129.10^{23} photons or N_A = 6.02214129.10^{23} mol^{-1}.
[3] 1 Astronomical Unit (AU) =149,597,870,700 meters

	Variable	Storage type	Units
	radiance	float	mol.s^{-1}.m^{-2}.nm^{-1}.sr^{-1}
CDL	float radiance(time,scanline,ground_pixel,spectral_channel) ; radiance:long_name = "spectral photon radiance" ; radiance:units = "mol.s-1.m-2.nm-1.sr-1" ; radiance:coordinates = "longitude latitude" ; radiance:ancillary_variables = "radiance_noise radiance_error quality_level spectral_channel_quality ground_pixel_quality" ; radiance:comment = "Measured spectral radiance for each spectral pixel" ;		
Remarks	There is no standard_name for photon radiance as measured by sensors on board satellites. In line with the standard_name for radiance that has been suggested by the cf-satellite user community on the Unidata mailing list, toa_outgoing_spectral_photon_radiance is suggested here.		

Table 57: CDL definition radiance variable

8.11 Variable: radiance_noise

The radiance noise (and similarly the error and the irradiance noise and error) is represented as a 10 times the base-10 logarithmic value of the ratio between the radiance and the random error. The representation of the errors in dB is assumed to be accurate and precise. Using a byte type has a considerable contribution as to limiting the final product file size.

	Variable	Storage type	Units
	radiance_noise	byte	none
CDL	byte radiance_noise(time,scanline,ground_pixel,spectral_channel) ; radiance_noise:long_name = "spectral photon radiance noise, one standard deviation" ; radiance_noise:units = "1" ; radiance_noise:coordinates = "longitude latitude" ; radiance_noise:comment = "The radiance_noise is a measure for the one standard deviation random error of the radiance measurement; it is expressed in decibel (dB), i.e. 10 times the base-10 logarithmic value of the ratio between the radiance and the random error." ;		
Remarks			

Table 58: CDL definition radiance_noise variable

8.12 Variable: `radiance_error`

	Variable	Storage type	Units
	`radiance_error`	byte	none

CDL	`byte radiance_error(time,scanline,ground_pixel,spectral_channel) ;`
	`radiance_error:long_name = "spectral photon radiance error, one standard deviation" ;`
	`radiance_error:units = "1" ;`
	`radiance_error:coordinates = "longitude latitude" ;`
	`radiance_error:comment = "The radiance_error is a measure for the one standard deviation error of the bias of the radiance measurement; it is expressed in decibel (dB), i.e. 10 times the base-10 logarithmic value of the ratio between the radiance and the estimation error." ;`		

Remarks			

Table 59: CDL definition `radiance_error` variable

8.13 Variable: `irradiance`

Every 15 orbits - approximately once every calendar day - TROPOMI will be commanded to perform a solar irradiance measurement. Irradiance is a measurement of solar power and is defined as the rate at which solar energy falls onto a surface. Similar to the spectral radiance, the *spectral irradiance* is the irradiance as function of wavelength. The SI units of spectral irradiance are $\mathrm{W.m^{-2}.nm^{-1}}$. However, like the case of the radiance variable, the L1b product provides the *spectral photon irradiance* with SI units $\mathrm{mol.s^{-1}.m^{-2}.nm^{-1}}$. Also the spectral photon irradiance is normalized to the Earth-Sun distance of 1 AU by applying a factor $\left(\frac{R}{R_{\mathrm{AU}}}\right)^2$ (see Equation 1).

	Variable	Storage type	Units
	`irradiance`	float	$\mathrm{mol.s^{-1}.m^{-2}.nm^{-1}}$

CDL	`float irradiance(time,scanline,pixel,spectral_channel) ;`
	`irradiance:long_name = "spectral photon irradiance" ;`
	`irradiance:units = "mol.s-1.m-2.nm-1" ;`
	`irradiance:ancillary_variables = "irradiance_noise irradiance_error quality_level spectral_channel_quality" ;`
	`irradiance:comment = "Measured spectral irradiance for each spectral pixel" ;`		

Remarks	There is no `standard_name` for spectral photon irradiance as measured by sensors on board satellites. In line with the `standard_name` for radiance that has been suggested by the cf-satellite user community on the Unidata mailing list, `toa_incoming_spectral_photon_irradiance` is suggested here.		

Table 60: CDL definition `irradiance` variable

8.14 Variable: `irradiance_noise`

Variable	Storage type	Units
irradiance_noise	byte	none

CDL

```
byte irradiance_noise(time,scanline,pixel,spectral_channel) ;
irradiance_noise:long_name = "spectral photon irradiance noise, one
standard deviation" ;
irradiance_noise:units = "1" ;
irradiance_noise:comment = "The irradiance_noise is a measure for the
one standard deviation random error of the irradiance measurement; it is
expressed in decibel (dB), i.e.  10 times the base-10 logarithmic value of
the ratio between the irradiance and the random error." ;
```

Remarks

Table 61: CDL definition `irradiance_noise` variable

8.15 Variable: `irradiance_error`

Variable	Storage type	Units
irradiance_error	byte	none

CDL

```
byte irradiance_error(time,scanline,pixel,spectral_channel) ;
irradiance_error:long_name = "spectral irradiance error, one standard
deviation" ;
irradiance_error:units = "1" ;
irradiance_error:comment = "The irradiance_error is a measure for the one
standard deviation error of the bias of the irradiance measurement; it is
expressed in decibel (dB), i.e.  10 times the base-10 logarithmic value of
the ratio between the irradiance and the estimation error." ;
```

Remarks

Table 62: CDL definition `irradiance_error` variable

8.16 Variable: `small_pixel_radiance`

One configurable detector pixel, in every row, for both detector output chains, i.e., two columns per detector, is not co-added and is stored separately for every exposure/co-addition of an image. The data for these 'small-pixel columns' are included in the science data and provide information on a higher spatial resolution than the data for other columns, which may be useful for certain studies. Thus for a given wavelength, the small_pixel_-radiance is the measurement of the spectral photon radiance expressed with SI units mol.s^{-1}.m^{-2}.nm^{-1}.sr^{-1}. The small_pixel_radiance is normalized to the Earth-Sun distance of 1 AU by applying a factor $\left(\frac{R}{R_{\mathrm{AU}}}\right)^2$ (see Equation 1).

	Variable	Storage type	Units
	small_pixel_radiance	float(*)	mol.s^{-1}.m^{-2}.nm^{-1}.sr^{-1}
CDL	types: float(*) small_pixel_radiance_type ; small_pixel_radiance_type small_pixel_radiance(time,scanline,ground_pixel) ; small_pixel_radiance:long_name = "small pixel photon radiance" ; small_pixel_radiance:units = "mol.s-1.m-2.nm-1.sr-1" ; small_pixel_radiance:_FillValue = 0x1.ep+122 ; small_pixel_radiance:coordinates = "longitude latitude" ; small_pixel_radiance:comment = "Measured spectral radiance for the spectral channel dedicated for the small pixel measurements" ;		
Remarks	small_pixel_type is a netCDF VLEN type There is no standard_name for photon radiance as measured by sensors on board satellites. In line with the standard_name for radiance that has been suggested by the cf-satellite user community on the Unidata mailing list, toa_outgoing_spectral_photon_radiance is suggested here.		

Table 63: CDL definition small_pixel_radiance variable

8.17 Variable: spectral_channel_quality

	Variable	Storage type	Units
	spectral_channel_quality	ubyte	none
CDL	ubyte spectral_channel_quality(time,scanline,ground_pixel,spectral_channel) ; spectral_channel_quality:long_name = "spectral channel quality flag" ; spectral_channel_quality:valid_min = 0 ; spectral_channel_quality:valid_max = 254 ; spectral_channel_quality:coordinates = "longitude latitude" ; spectral_channel_quality:flag_values = 0UB, 1UB, 2UB, 8UB, 16UB, 32UB, 64UB, 128UB ; spectral_channel_quality:flag_meanings = no_error, missing, bad_pixel, processing_error, saturated, transient, rts, underflow ; spectral_channel_quality:comment = "Quality assessment information for each (spectral) pixel" ;		
Remarks			

Table 64: CDL definition spectral_channel_quality variable

8.18 Variable: `ground_pixel_quality`

	Variable	Storage type	Units
	ground_pixel_quality	ubyte	none
CDL	ubyte ground_pixel_quality(time,scanline,ground_pixel) ; ground_pixel_quality:long_name = "ground pixel quality flag" ; ground_pixel_quality:valid_min = 0 ; ground_pixel_quality:valid_max = 254 ; ground_pixel_quality:coordinates = "longitude latitude" ; ground_pixel_quality:flag_values = 0UB, 1UB, 2UB, 4UB, 8UB, 16UB, 128UB ; ground_pixel_quality:flag_meanings = no_error, solar_eclipse, sun_glint_-possible, descending, night, geo_boundary_crossing, geolocation_error ; ground_pixel_quality:comment = "Quality assessment information for each ground pixel" ;		
Remarks			

Table 65: CDL definition `ground_pixel_quality` variable

8.19 Variable: `quality_level`

The L1b variable `quality_level` is used to provide an overall indication of L1b data quality. Typically, to assign a quality level to a data product, *Quality Indicators (QIs)* are needed, in particular at each stage of the data processing chain - from collection and processing to delivery. A QI should provide sufficient information to allow all users to evaluate a product's suitability for their particular application. These QIs are provided to the users in the variable `spectral_channel_quality` (covering e.g. transient) and the variable `ground_pixel_quality` (covering e.g. solar eclipse). A QI is stored in a binary format, representing an on/off mode. Whenever a bit for a specific QI is set, this QI negatively influenced the determination of the `quality_level`.

The value for the overall quality is obtained by multiplying the quality indicators (ranging from 0 to 1) of the individual algorithms applied in the L01b processing chain. This product is then multiplied by hundred. Thus the maximum quality level is equal to 100; each processing algorithm might introduce a degradation which ultimately can result in the worst quality level equal to 0.

	Variable	Storage type	Units
	quality_level	ubyte	none
CDL	ubyte quality_level(time,scanline,ground_pixel,spectral_channel) ; quality_level:long_name = "qualiy level of spectral channel" ; quality_level:valid_min = 0 ; quality_level:valid_max = 100 ; quality_level:coordinates = "longitude latitude" ; quality_level:comment = "Overall quality assessment information for each (spectral) pixel" ;		
Remarks			

Table 66: CDL definition `quality_level` variable

8.20 Variable: `measurement_quality`

	Variable	Storage type	Units
	`measurement_quality`	ushort	none

CDL	`ushort measurement_quality(time,scanline) ;`
	`measurement_quality:long_name = "measurement quality flag" ;`
	`measurement_quality:valid_min = 0 ;`
	`measurement_quality:valid_max = 65534 ;`
	`measurement_quality:coordinates = "longitude latitude" ;`
	`measurement_quality:flag_values = 0US, 1US, 2US, 16US, 32US, 128US, 256US, 4096US ;`
	`measurement_quality:flag_meanings = no_error, proc_skipped, no_residual, saa, spacecraft_manoeuvre, sub_grp, irr_out_of_range, sub_group ;`
	`measurement_quality:comment = "Overall quality information for a measurement" ;`		

Remarks	Extended description: • no_error: No measurement qualification • proc_skipped: One or more processing steps (algorithms) where skipped • no residual: No residual correction applied because no correction values where found • Measurement was obtained while spacecraft was in South Atlantic Anomaly • Measurement was obtained during spacecraft manoeuvre • Measurement was flagged as sub-group • Measurement outside nominal elevation / azimuth range • Measurement was flagged as sub-group by subgroup algorithm		

Table 67: CDL definition `measurement_quality` variable

8.21 Variable: `detector_row_qualification`

Variable	Storage type	Units
`detector_row_qualification`	ushort	none

CDL

```
ushort detector_row_qualification(time,scanline,ground_pixel) ;
detector_row_qualification:long_name = "Detector row qualification flags" ;
detector_row_qualification:valid_min = 0 ;
detector_row_qualification:valid_max = 65534 ;
detector_row_qualification:flag_values = 0US, 1US, 2US, 4US, 8US, 16US,
256US, 4096US, 8192US ;
detector_row_qualification:flag_meanings = no_qualification, uvn_ror,
uvn_dump, uvn_covered, uvn_overscan, uvn_higain, swir_reference, gen_-
transistion, gen_non_illuminated ;
detector_row_qualification:comment = "Qualification flag indicating row
type or state" ;
```

Remarks Extended description:

- no_qualification: No row qualification
- uvn_ror: UVN detector specific, row is read-out register (ROR)
- uvn_dump: UVN detector specific, row is read using dump gate setting
- uvn_covered: UVN detector specific, row is covered on detector
- uvn_overscan: UVN detector specific, over-scan row
- uvn_higain: UVN detector specific, row is read using high gain output
- swir_reference: SWIR detector specific, row is reference line
- Row is transition row on detector
- Row is not illuminated by spectrometer output

Table 68: CDL definition `detector_row_qualification` variable

8.22 Variable: detector_column_qualification

Variable	Storage type	Units
detector_column_qualification	ushort	none

CDL

```
ushort detector_column_qualification(time,scanline,spectral_channel) ;
detector_column_qualification:long_name = "Detector column qualification
flags" ;
detector_column_qualification:valid_min = 0 ;
detector_column_qualification:valid_max = 65534 ;
detector_column_qualification:flag_values = 0US, 1US, 16US, 32US, 64US,
256US, 512US, 1024US, 2048US ;
detector_column_qualification:flag_meanings = no_qualification, skipped,
uvn_odd, uvn_prepost, uvn_overscan, swir_adc0, swir_adc1, swir_adc2, swir_-
adc3 ;
detector_column_qualification:comment = "Qualification flag indicating
column indicating column type or state" ;
```

Remarks Extended description:

- no_qualification: No column qualification
- skipped: Column was not read and therefore contains fill values
- uvn_odd: UVN detector specific, pixels in the column took the odd ADC path
- uvn_prepost: UVN detector specific, pixels in the column are pre- or post-scan pixels
- uvn_overscan: UVN detector specific, column is an over-scan column
- swir_adc0: SWIR detector specific, pixels in the column used ADC0
- swir_adc1: SWIR detector specific, pixels in the column used ADC1
- swir_adc2: SWIR detector specific, pixels in the column used ADC2
- swir_adc3: SWIR detector specific, pixels in the column used ADC3

Table 69: CDL definition `detector_column_qualification` variable

8.23 Variable: calibrated_wavelength

The `nominal_wavelength` (section 8.45) provides for each ground pixel the wavelength measured by a spectral channel and is defined by the design parameters of the instrument. The wavelength values as provided by the `nominal_wavelength` are based on the Calibration Key Data (CKD) which are input to the L01b processing (section 5).

During the measurements the actual measured wavelength will vary from the nominal one and a calibration step is required to correct for this effect. For radiance products this calibration is applied as part of the L2 processing, because it involves atmospheric corrections which are only available at that product level. Therefore, the `calibrated_wavelength` is not part of the L1b radiance product.

For the L1b irradiance products the `calibrated_wavelength` is available. As part of the L01b processing the spectral information obtained from the irradiance measurements is compared with a reference solar spectrum. From this comparison a calibrated set of wavelengths is derived which provides a per pixel best estimate for the wavelength actually measured by each individual spectral channel.

	Variable	Storage type	Units
	calibrated_wavelength	float	nm
CDL	float calibrated_wavelength(time,pixel,spectral_channel) ; calibrated_wavelength:long_name = "spectral channel calibrated wavelength" ; calibrated_wavelength:standard_name = "radiation_wavelength" ; calibrated_wavelength:units = "nm" ; calibrated_wavelength:comment = "Calibrated wavelength of each spectral pixel" ;		
Remarks	The calibrated_wavelength provides for each pixel the wavelength measured by a spectral channel and is defined by the design parameters of the instrument.		

Table 70: CDL definition calibrated_wavelength variable

8.24 Variable: calibrated_wavelength_error

	Variable	Storage type	Units
	calibrated_wavelength_error	float	nm
CDL	float calibrated_wavelength_error(time,pixel,spectral_channel) ; calibrated_wavelength_error:long_name = "spectral channel calibrated wavelength error" ; calibrated_wavelength_error:standard_name = "radiation_wavelength standard_-error" ; calibrated_wavelength_error:units = "nm" ; calibrated_wavelength_error:comment = "Standard deviation on the calibrated wavelength of each spectral pixel" ;		
Remarks	The calibrated_wavelength provides for each pixel the standard deviation on the wavelength measured by a spectral channel and is defined by the design parameters of the instrument.		

Table 71: CDL definition calibrated_wavelength_error variable

8.25 Variable: latitude

	Variable	Storage type	Units
	latitude	float	degrees north
CDL	float latitude(time,scanline,ground_pixel) ; latitude:long_name = "pixel center latitude" ; latitude:standard_name = "latitude" ; latitude:units = "degrees_north" ; latitude:valid_min = -90.f ; latitude:valid_max = 90.f ; latitude:bounds = "latitude_bounds" ; latitude:comment = "Latitude of the center of each ground pixel on the WGS84 reference ellipsoid" ;		
Remarks	Latitude, longitude coordinates for the ground pixel center and the ground pixel corners are calculated at the WGS84 ellipsoid. In principle, the information provided in the GeodataGroup allows to calculate these coordinates at arbitrary altitudes.		

Table 72: CDL definition latitude variable

8.26 Variable: `longitude`

	Variable	Storage type	Units
	`longitude`	float	degrees east

CDL	`float longitude(time,scanline,ground_pixel) ;`
	`longitude:long_name = "pixel center longitude" ;`
	`longitude:standard_name = "longitude" ;`
	`longitude:units = "degrees_east" ;`
	`longitude:valid_min = -180.f ;`
	`longitude:valid_max = 180.f ;`
	`longitude:bounds = "longitude_bounds" ;`
	`longitude:comment = "Longitude of the center of each ground pixel on the WGS84 reference ellipsoid" ;`		

Remarks	Latitude, longitude coordinates for the ground pixel center and the ground pixel corners are calculated at the WGS84 ellipsoid. In principle, the information provided in the GeodataGroup allows to calculate these coordinates at arbitrary altitudes.		

Table 73: CDL definition `longitude` variable

8.27 Variable: `latitude_bounds`

The four corner points of the ground pixels are calculated as an interpolation between the centre coordinates (`longitude, latitude`) of adjacent pixels and lines. The variable `latitude_bounds` provides the latitude value of these corner points.

	Variable	Storage type	Units
	`latitude_bounds`	float	degrees north

CDL	`float latitude_bounds(time,scanline,ground_pixel,ncorner) ;`
	`latitude_bounds:units = "degrees_north" ;`
	`latitude_bounds:comment = "The four latitude boundaries of each ground pixel." ;`		

Remarks	CF-Convention: Since a boundary variable is considered to be part of a coordinate variable's metadata, it is not necessary to provide it with attributes such as `long_name` and `units`. Using a right-handed coordinate system, the ordering of the bounds is anti-clockwise on the longitude-latitude surface seen from above. Latitude, longitude coordinates for the ground pixel center and the ground pixel corners are calculated at the WGS84 ellipsoid. In principle, the information provided in the GeodataGroup allows to calculate these coordinates at arbitrary altitudes.		

Table 74: CDL definition `latitude_bounds` variable

8.28 Variable: `longitude_bounds`

The four corner points of the ground pixels are calculated as an interpolation between the centre coordinates (`longitude, latitude`) of adjacent pixels and lines. The variable `longitude_bounds` provides the longitude value of these corner points.

	Variable	Storage type	Units
	longitude_bounds	float	degrees east
CDL	float longitude_bounds(time,scanline,ground_pixel,ncorner) ; longitude_bounds:units = "degrees_east" ; longitude_bounds:comment = "The four longitude boundaries of each ground pixel." ;		
Remarks	CF-Convention: Since a boundary variable is considered to be part of a coordinate variable's metadata, it is not necessary to provide it with attributes such as long_name and units. Using a right-handed coordinate system, the ordering of the bounds is anti-clockwise on the longitude-latitude surface seen from above. Latitude, longitude coordinates for the ground pixel center and the ground pixel corners are calculated at the WGS84 ellipsoid. In principle, the information provided in the GeodataGroup allows to calculate these coordinates at arbitrary altitudes.		

Table 75: CDL definition longitude_bounds variable

8.29 Variable: solar_zenith_angle

	Variable	Storage type	Units
	solar_zenith_angle	float	degree
CDL	float solar_zenith_angle(time,scanline,ground_pixel) ; solar_zenith_angle:long_name = "solar zenith angle" ; solar_zenith_angle:standard_name = "solar_zenith_angle" ; solar_zenith_angle:units = "degree" ; solar_zenith_angle:valid_min = 0.f ; solar_zenith_angle:valid_max = 180.f ; solar_zenith_angle:coordinates = "longitude latitude" ; solar_zenith_angle:comment = "Solar zenith angle at the ground pixel location on the reference ellipsoid. Angle is measured away from the vertical. ESA definition of day side: SZA less the 92 degrees" ;		
Remarks			

Table 76: CDL definition solar_zenith_angle variable

8.30 Variable: solar_elevation_angle

	Variable	Storage type	Units
	solar_elevation_angle	float	degree
CDL	float solar_elevation_angle(time,scanline) ; solar_elevation_angle:long_name = "solar elevation angle" ; solar_elevation_angle:units = "degree" ; solar_elevation_angle:valid_min = -90.f ; solar_elevation_angle:valid_max = +90.f ; solar_elevation_angle:comment = "Solar elevation angle measured from the Sun port on instrument. Angle is measured from the YZ-plane towards the X-axis (=nominal Sun LOS) of the Sun Port reference frame." ;		
Remarks	This variable is only present in the irradiance calibration product		

Table 77: CDL definition solar_elevation_angle variable

8.31 Variable: `solar_azimuth_angle`

Level-2 data processors need information on the lines of sight from the ground pixel position to the spacecraft and to the Sun, in the topocentric reference frame. These are defined by the solar azimuth ϕ_0 and zenith θ_0 angles for the incident sunlight, and spacecraft azimuth ϕ and zenith θ angles for the scattered sunlight With these angles the level-2 data processors can for instance determine the scattering angle Θ. For a complete description see the section on Geometrical algorithms" in [RD12].

	Variable	Storage type	Units
	`solar_azimuth_angle`	float	degree

CDL

```
float solar_azimuth_angle(time,scanline,ground_pixel) ;
solar_azimuth_angle:long_name = "solar azimuth angle" ;
solar_azimuth_angle:standard_name = "solar_azimuth_angle" ;
solar_azimuth_angle:units = "degree" ;
solar_azimuth_angle:valid_min = -180.f ;
solar_azimuth_angle:valid_max = 180.f ;
solar_azimuth_angle:coordinates = "longitude latitude" ;
solar_azimuth_angle:comment = "Solar azimuth angle at the ground pixel
location on the reference ellipsoid.  Angle is measured clockwise from the
North (East = +90, South = -+180, West = -90)" ;
```

Remarks

Table 78: CDL definition `solar_azimuth_angle` variable

8.32 Variable: `viewing_zenith_angle`

	Variable	Storage type	Units
	`viewing_zenith_angle`	float	degree

CDL

```
float viewing_zenith_angle(time,scanline,ground_pixel) ;
viewing_zenith_angle:long_name = "viewing zenith angle" ;
viewing_zenith_angle:standard_name = "platform_zenith_angle" ;
viewing_zenith_angle:units = "degree" ;
viewing_zenith_angle:valid_min = 0.f ;
viewing_zenith_angle:valid_max = 180.f ;
viewing_zenith_angle:coordinates = "longitude latitude" ;
viewing_zenith_angle:comment = "Zenith angle of the satellite at the ground
pixel location on the reference ellipsoid.  Angle is measured away from the
vertical." ;
```

Remarks

Table 79: CDL definition `viewing_zenith_angle` variable

8.33 Variable: viewing_azimuth_angle

Variable	Storage type	Units
viewing_azimuth_angle	float	degree

CDL
```
float viewing_azimuth_angle(time,scanline,ground_pixel) ;
viewing_azimuth_angle:long_name = "viewing azimuth angle" ;
viewing_azimuth_angle:standard_name = "platform_azimuth_angle" ;
viewing_azimuth_angle:units = "degree" ;
viewing_azimuth_angle:valid_min = -180.f ;
viewing_azimuth_angle:valid_max = 180.f ;
viewing_azimuth_angle:coordinates = "longitude latitude" ;
viewing_azimuth_angle:comment = "Azimuth angle of the satellite at the
ground pixel location on the reference ellipsoid.  Angle is measured
clockwise from the North (East = +90, South = -+180, West = -90)" ;
```

Remarks

Table 80: CDL definition viewing_azimuth_angle variable

8.34 Variable: satellite_latitude

Variable	Storage type	Units
satellite_latitude	float	degrees north

CDL
```
float satellite_latitude(time,scanline) ;
satellite_latitude:long_name = "sub-satellite latitude" ;
satellite_latitude:units = "degrees_north" ;
satellite_latitude:valid_min = -90.f ;
satellite_latitude:valid_max = 90.f ;
satellite_latitude:comment = "Latitude of the spacecraft sub-satellite
point on the WGS84 reference ellipsoid" ;
```

Remarks

Table 81: CDL definition satellite_latitude variable

8.35 Variable: satellite_longitude

Variable	Storage type	Units
satellite_longitude	float	degrees east

CDL
```
float satellite_longitude(time,scanline) ;
satellite_longitude:units = "degrees_east" ;
satellite_longitude:valid_min = -180.f ;
satellite_longitude:valid_max = 180.f ;
satellite_longitude:comment = "Longitude of the spacecraft sub-satellite
point on the WGS84 reference ellipsoid" ;
```

Remarks

Table 82: CDL definition satellite_longitude variable

8.36 Variable: satellite_altitude

	Variable	Storage type	Units
	satellite_altitude	float	m
CDL	float satellite_altitude(time,scanline) ; satellite_altitude:long_name = "satellite altitude" ; satellite_altitude:units = "m" ; satellite_altitude:valid_min = 700000.f ; satellite_altitude:valid_max = 900000.f ; satellite_altitude:comment = "The altitude of the spacecraft relative to the WGS84 reference ellipsoid" ;		
Remarks			

Table 83: CDL definition satellite_altitude variable

8.37 Variable: satellite_orbit_phase

The orbit phase is defined as $1/(2\pi)$ times the angle in radians traversed by the spacecraft since spacecraft midnight as seen from the center of the Earth. Spacecraft midnight is the point on the night side of the Earth where the spacecraft crosses the orbital plane of the Earth about the Sun. This makes the orbit phase a quantity that runs from 0 to 1, while the spacecraft moves between each spacecraft midnight.

	Variable	Storage type	Units
	satellite_orbit_phase	float	none
CDL	float satellite_orbit_phase(time,scanline) ; satellite_orbit_phase:long_name = "fractional satellite orbit phase" ; satellite_orbit_phase:units = "1" ; satellite_orbit_phase:valid_min = -0.02f ; satellite_orbit_phase:valid_max = 1.02f ; satellite_orbit_phase:comment = "Relative offset (0.0 ... 1.0) of the measurement in the orbit" ;		
Remarks	CF-Convention: The conforming unit for quantities that represent fractions, or parts of a whole, is "1".		

Table 84: CDL definition satellite_orbit_phase variable

8.38 Variable: earth_sun_distance

	Variable	Storage type	Units
	earth_sun_distance	float	astronomical unit
CDL	float earth_sun_distance(time) ; earth_sun_distance:long_name = "distance between the earth and the sun" ; earth_sun_distance:units = "astronomical_unit" ; earth_sun_distance:valid_min = 0.98f ; earth_sun_distance:valid_max = 1.02f ; earth_sun_distance:comment = "1 au equals 149,597,870,700 meters" ;		
Remarks			

Table 85: CDL definition earth_sun_distance variable

8.39 Variable: processing_class

Different operating modes of the system and the derived L01B products are described by three parameters: the Processing Class, the Instrument Configuration ID (IcID) and Instrument Configuration Version (IcVersion). The concept for these three parameters is taken from the OMI mission:

- The Processing Class defines the type of measurement at a very high level. Contrary to the IcIDs, the set of processing classes is (fairly) static. The advantage of this, is that it is possible to create new IcIDs and as long as these can use an existing processing class, it is not required to update the L01b to support that IcID. Examples of processing classes are Earth_radiance, Sun_irradiance, DLED, WLS, Dark, Background, . . . For a complete overview of valid processing classes see Appendix B.
- The Instrument Configuration ID defines the type of measurement and its purposes. The number of Instrument Configuration IDs will increase over the mission as new types of measurements are created / used;
- The Instrument Configuration Version allows to differentiate between multiple versions for a specific IcID.

Each Processing Class and each IcID corresponds to a number. The numbers for Processing Class, IcID and IcVersion are set in the instrument by the instrument operations team for each measurement.

	Variable	Storage type	Units
	processing_class	short	none
CDL	short processing_class(time,scanline) ; processing_class:long_name = "processing class" ; processing_class:valid_min = 0 ; processing_class:valid_max = 255 ; processing_class:comment = "The processing_class defines the type of measurement at a very high level. Contrary to Instrument Configuration IDs, only a limited, fixed set of processing classes is identified. Examples of processing classes are Earth_radiance, Sun_irradiance, CLED, WLS, Dark, Background, ...;" ;		
Remarks	For a complete overview of valid processing classes see Appendix B.		

Table 86: CDL definition processing_class variable

8.40 Variable: instrument_configuration

The TROPOMI instrument has many configurable parameters. For example, the exposure time, co-addition period, gains and (for UVN-DEMs) the binning factors can be varied. As a result, the instrument can be operated in many different modes or configurations. Each combination of instrument settings is referred to as instrument configuration and is identified by an instrument configuration ID, a number in the range [1,65535]. This instrument configuration ID, or IcID, is primarily used by the instrument, where it identifies an entry in the instrument configuration tables. On ground, the IcID is used to determine the intended purpose of a measurement and is used in the L01b data processing to determine the processing path.

For an IcID, it is possible to have multiple versions, identified by the instrument configuration version or IcVersion. The combination of IcID and IcVersion uniquely identifies the set of configuration settings of the instrument. At a given time, only one IcVersion of an IcID can be active within the instrument. The IcVersion allows to have multiple versions of a measurement with the same purpose, but with different settings. As a result of, for example, instrument degradation, it may be required to change the settings for a measurement. In that case, it is not necessary to create a new IcID, instead the same IcID can be using with a new IcVersion.

	Variable	Storage type	Units
	instrument_configuration	n/a	none
CDL	types: instrument_configuration_type { int icid ; short ic_version ; } ; instrument_configuration_type instrument_configuration(time,scanline) ; instrument_configuration:long_name = "instrument configuration, IcID and IcVersion" ; instrument_configuration:comment = "The Instrument Configuration ID defines the type of measurement and its purposes. The number of Instrument Configuration IDs will increase over the mission as new types of measurements are created / used; The Instrument Configuration Version allows to differentiate between multiple versions for a specific IcID." ;		
Remarks			

Table 87: CDL definition `instrument_configuration` variable

8.41 Variable: `instrument_settings`

The `instrument_settings` variable contains all the instrument settings that are relevant for data processing. Due to the UVN and SWIR modules having different instrument configuration parameters, `instrument_settings` is defined differently for UVN and SWIR products. The instrument settings are given for each Instrument Configuration ID and version contained in the product.

8.41.1 UVN product: instrument_settings

Variable	Storage type	Units
instrument_settings	instrument_settings_type	none

CDL

```
types:
compound instrument_settings_type {
int ic_id ;
short ic_version ;
short ic_set ;
short ic_idx ;
short processing_class ;
float master_cycle_period ;
float coaddition_period ;
float exposure_time ;
float msmt_mcp_ft_offset ;
float msmt_ft_msmt_start_offset ;
float msmt_duration ;
float flush_duration ;
short nr_coadditions ;
short cds_gain ;
float pga_gain ;
float dac_offset ;
int master_cycle_period_us ;
int coaddition_period_us ;
int exposure_time_us ;
int exposure_period_us ;
short small_pixel_column ;
short stop_column_read ;
short start_column_coad ;
short stop_column_coad ;
short pga_gain_code ;
short dac_offset_code ;
ubyte clock_mode ;
ubyte clipping ;
}; // instrument_settings_type
variables:
instrument_settings_type instrument_settings(nsettings) ;
```

Remarks

Table 88: CDL definition instrument_settings variable

field	type	unit	description
ic_id	int	1	Instrument configuration ID; number that uniquely specifies a type of measurement. The combination of the icid and icversion uniquely identifies a specific instrument configuration
ic_version	short	1	Instrument configuration version; version number for the instrument configuration ID. The combination of the icid and icversion uniquely identifies a specific instrument configuration
ic_set	short	1	Instrument configuration set of which the instrument configuration ID is part.
ic_idx	short	1	Index of the instrument configuration ID in the instrument configuration set

field	type	unit	description
processing_class	short	1	The processing_class defines the type of measurement at a very high level. Contrary to Instrument Configuration IDs, only a limited, fixed set of processing classes is identified. Examples of processing classes are Earth_radiance, Sun_irradiance, CLED, WLS, Dark, Background, ..
master_cycle_period	float	s	Measurement master cycle period in seconds; must be a multiple of the coaddition period.
coaddition_period	float	s	Co-addition period in seconds; the time interval covered by all the co-additions in the measurement. In case no flushing is used, this is equal to the number of co-additions multiplied by the exposure time. In case flushing is used, this is equal to the number of co-additions multiplied by the sum of the exposure time and the flushing time
exposure_time	float	s	The exposure time in seconds for a single (unco-added) frame.
msmt_mcp_ft_offset	float	s	Offset between Master clock pulse and frame trigger starting measurement
msmt_ft_msmt_start_offset	float	s	Offset between FT and start of exposure
msmt_duration	float	s	Delta between start of first exposure in a measurement and end of last exposure in a measurement
flush_duration	float	s	Duration of the flush period of a measurement
nr_coadditions	short	1	The number of co-additions.
cds_gain	short	1	The CDS V/V gain, based on design parameters, either 1x or 2x.
pga_gain	float	1	The AFE PGA V/V gain, based on design parameters.
dac_offset	float	V	The AFE DAC offset in V, based on design parameters.
master_cycle_period_us	int	us	Measurement master cycle period in microseconds; must be a multiple of the coaddition period. Note: Contrary to the master_cycle_period, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable
coaddition_period_us	int	us	Co-addition period in microseconds; the time interval covered by all the co-additions in the measurement. In case no flushing is used, this is equal to the number of co-additions multiplied by the exposure time. In case flushing is used, this is equal to the number of co-additions multiplied by the sum of the exposure time and the flushing time. Note: Contrary to the coaddition_period, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable
exposure_time_us	int	us	The exposure time in microseconds for a single (unco-added) frame. Note: Contrary to the exposure_time, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable.
exposure_period_us	int	us	The interval between two consecutive exposures
small_pixel_column	short	1	Setting (code) for the AFE PGA
stop_column_read	short	1	Setting (code) for the AFE DAC
start_column_coad	short	1	Column for which the data are downlinked for all co-addition

field	type	unit	description
stop_column_coad	short	1	The number of columns from the detector that are read-out
pga_gain_code	short	1	First column that is being co-added
dac_offset_code	short	1	Laststst column that is being co-added
clock_mode	ubyte	1	CCD Clocking mode; 0 = normal, 1 = reverse, 2 =static, 3 = CTE, 4 = invalid
clipping	ubyte	1	Data clipping position

Table 89: Fields in the instrument_settings variable.

8.41.2 SWIR product: instrument_settings

Variable	Storage type	Units
instrument_settings	instrument_settings_type	none

CDL

```
types:
compound instrument_settings_type {
int ic_id ;
short ic_version ;
short ic_set ;
short ic_idx ;
short processing_class ;
float master_cycle_period ;
float coaddition_period ;
float exposure_time ;
float msmt_mcp_ft_offset ;
float msmt_ft_msmt_start_offset ;
float msmt_duration ;
float reset_time ;
short nr_coadditions ;
int master_cycle_period_us ;
int coaddition_period_us ;
int exposure_time_us ;
int exposure_period_us ;
short small_pixel_column ;
short stop_column_read ;
short start_column_coad ;
short stop_column_coad ;
uint int_hold ;
ushort int_delay ;
ubyte clipping ;
}; // instrument_settings_type
variables:
instrument_settings_type instrument_settings(nsettings) ;
```

Remarks

Table 90: CDL definition instrument_settings variable

field	type	unit	description
ic_id	int	1	Instrument configuration ID; number that uniquely specifies a type of measurement. The combination of the icid and icversion uniquely identifies a specific instrument configuration

field	type	unit	description
ic_version	short	1	Instrument configuration version; version number for the instrument configuration ID. The combination of the icid and icversion uniquely identifies a specific instrument configuration
ic_set	short	1	Instrument configuration set of which the instrument configuration ID is part.
ic_idx	short	1	Index of the instrument configuration ID in the instrument configuration set
processing_class	short	1	The processing_class defines the type of measurement at a very high level. Contrary to Instrument Configuration IDs, only a limited, fixed set of processing classes is identified. Examples of processing classes are Earth_radiance, Sun_irradiance, CLED, WLS, Dark, Background, ..
master_cycle_period	float	s	Measurement master cycle period in seconds; must be a multiple of the coaddition period.
coaddition_period	float	s	Co-addition period in seconds; the time interval covered by all the co-additions in the measurement. In case no flushing is used, this is equal to the number of co-additions multiplied by the exposure time. In case flushing is used, this is equal to the number of co-additions multiplied by the sum of the exposure time and the flushing time
exposure_time	float	s	The exposure time in seconds for a single (unco-added) frame.
msmt_mcp_ft_offset	float	s	Offset between Master clock pulse and frame trigger starting measurement
msmt_ft_msmt_start_offset	float	s	Offset between FT and start of measurement
msmt_duration	float	s	Delta between start of first exposure in a measurement and end of last exposure in a measurement
reset_time	float	s	Reset time between exposures
nr_coadditions	short	1	The number of co-additions.
master_cycle_period_us	int	us	Measurement master cycle period in microseconds; must be a multiple of the coaddition period. Note: Contrary to the master_cycle_period, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable
coaddition_period_us	int	us	Co-addition period in microseconds; the time interval covered by all the co-additions in the measurement. In case no flushing is used, this is equal to the number of co-additions multiplied by the exposure time. In case flushing is used, this is equal to the number of co-additions multiplied by the sum of the exposure time and the flushing time. Note: Contrary to the coaddition_period, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable
exposure_time_us	int	us	The exposure time in microseconds for a single (unco-added) frame. Note: Contrary to the exposure_time, which is stored as a float, this field is stored as a long and therefore exactly representable and comparable.
exposure_period_us	int	us	The interval between two consecutive exposures

field	type	unit	description
small_pixel_column	short	1	Column for which the data are downlinked for all co-addition
stop_column_read	short	1	The number of columns from the detector that are read-out
start_column_coad	short	1	First column that is being co-added
stop_column_coad	short	1	Latest column that is being co-added
int_hold	uint	1	INT_HOLD code
int_delay	ushort	1	INT_DELAY code
clipping	ubyte	1	Data clipping position

Table 91: Fields in the instrument_settings variable.

8.42 Variable: binning_table

Variable	Storage type	Units
binning_table	binning_table_type	none

CDL

```
types:
compound binning_table_type {
short size ;
short binning_factor ;
short gain ;
short detector_start_row ;
short detector_stop_row ;
short measurement_start_row ;
short measurement_stop_row ;
}; // binning_table_type
variables:
binning_table_type binning_table(nsettings, nbinningregions) ;
```

Remarks

Table 92: CDL definition binning_table variable

field	type	unit	description
size	short	1	Number of rows in the area before binning / read-out
binning_factor	short	1	Binning factor for the area; 0 if rows are skipped
gain	short	1	CCD output gain for the area (0 = dump, 1 = 1x, 2 = 2x)
detector_start_row	short	1	Start row of the binning area on the detector
detector_stop_row	short	1	Stop row of the binning area on the detector; the stop row is exclusive (i.e. up to, but not including)
measurement_start_row	short	1	Start row of the binning area in the measurement. Set to -1 in case the area is skipped. Reflects the rows that are actually written to the output, in case a subset of the data is written.
measurement_stop_row	short	1	Stop row of the binning area in the measurement; the stop row is exclusive (i.e. up to, but not including). Set to -1 in case the area is skipped. Reflects the rows that are actually written to the output, in case a subset of the data is written.

Table 93: Fields in the binning_table variable.

8.43 Variable: housekeeping_data

Variable	Storage type	Units
housekeeping_data	housekeeping_data_type	none

CDL

```
types:
compound housekeeping_data_type {
float temp_det1 ;
float temp_det2 ;
float temp_det3 ;
float temp_det4 ;
float data_offset_s ;
float temp_tss_up_neg_x ;
float temp_tss_up_neg_y ;
float temp_tss_up_pos_x ;
float temp_tss_up_pos_y ;
float temp_tss_up_mid ;
float temp_tss_low_mid ;
float temp_low_uvn_obm ;
float temp_up_uvn_obm ;
float temp_obm_swir ;
float temp_obm_solar_baffle ;
float temp_cu_sls_stim ;
float temp_obm_swir_grating ;
float temp_obm_swir_if ;
float temp_pelt_cu_sls1 ;
float temp_pelt_cu_sls2 ;
float temp_pelt_cu_sls3 ;
float temp_pelt_cu_sls4 ;
float temp_pelt_cu_sls5 ;
ubyte difm_status ;
ubyte fmm_status ;
ubyte det1_led_status ;
ubyte det2_led_status ;
ubyte det3_led_status ;
ubyte det4_led_status ;
ubyte common_led_status ;
ubyte sls1_status ;
ubyte sls2_status ;
ubyte sls3_status ;
ubyte sls4_status ;
ubyte sls5_status ;
ubyte wls_status ;
ubyte filler_char1 ;
float swir_vdet_bias ;
}; // housekeeping_data_type
variables:
housekeeping_data_type housekeeping_data(time, scanline) ;
```

Remarks

Table 94: CDL definition housekeeping_data variable

field	type	unit	description
temp_det1	float	K	Temperature of the detector 1
temp_det2	float	K	Temperature of the detector 2

field	type	unit	description
temp_det3	float	K	Temperature of the detector 3
temp_det4	float	K	Temperature of the detector 4
data_offset_s	float	s	Offset time to measurement time of housekeeping data
temp_tss_up_neg_x	float	K	TSS Upper surface Mid -X side temperature
temp_tss_up_neg_y	float	K	TSS Upper surface Mid -Y side temperature
temp_tss_up_pos_x	float	K	TSS Upper surface Mid +X side temperature
temp_tss_up_pos_y	float	K	TSS Upper surface Mid +Y side temperature
temp_tss_up_mid	float	K	TSS Upper surface middle temperature
temp_tss_low_mid	float	K	TSS lower surface middle temperature
temp_low_uvn_obm	float	K	Temperature of the lower UVN OBM
temp_up_uvn_obm	float	K	Temperature of the upper UVN OBM
temp_obm_swir	float	K	Temperature of the SWIR OBM
temp_obm_solar_baffle	float	K	Temperature of the OBM Solar baffle
temp_cu_sls_stim	float	K	Temperature of the OBM CU SLS stimuli
temp_obm_swir_grating	float	K	Temperature of the SWIR grating
temp_obm_swir_if	float	K	Temperature of the OBM at SWIR interface
temp_pelt_cu_sls1	float	K	Temperature of the Peltier Control Calibration unit for SLS1
temp_pelt_cu_sls2	float	K	Temperature of the Peltier Control Calibration unit for SLS2
temp_pelt_cu_sls3	float	K	Temperature of the Peltier Control Calibration unit for SLS3
temp_pelt_cu_sls4	float	K	Temperature of the Peltier Control Calibration unit for SLS4
temp_pelt_cu_sls5	float	K	Temperature of the Peltier Control Calibration unit for SLS5
difm_status	ubyte	1	DIFM status; 0 UNKNOWN, 1 WLS, CLED_QVD2 2, SUN_QVD2 3, SLS 4, CLED_QVD2 5, SUN_QVD1 6, OSCILATING 7
fmm_status	ubyte	1	FMM status; UNKNOWN 0, NADIR_VIEW 1, CALIBRATION 2
det1_led_status	ubyte	1	Led of detector 1 on (1) or off (0)
det2_led_status	ubyte	1	Led of detector 2 on (1) or off (0)
det3_led_status	ubyte	1	Led of detector 3 on (1) or off (0)
det4_led_status	ubyte	1	Led of detector 1 on (1) or off (0)
common_led_status	ubyte	1	Common led on (1) or off (0)
sls1_status	ubyte	1	Led SLS1 on (1) or off (0)
sls2_status	ubyte	1	Led SLS2 on (1) or off (0)
sls3_status	ubyte	1	Led SLS3 on (1) or off (0)
sls4_status	ubyte	1	Led SLS4 on (1) or off (0)
sls5_status	ubyte	1	Led SLS5 on (1) or off (0)
wls_status	ubyte	1	Led WLS on (1) or off (0)
filler_char1	ubyte	1	Filler byte for alignment
swir_vdet_bias	float	V	Bias voltage of SWIR detector

Table 95: Fields in the housekeeping_data variable.

8.44 Variable: measurement_to_detector_row_table in engineering product

Variable	Storage type	Units
measurement_to_detector_row_table	msmt_to_det_row_table_type	none

CDL

```
types:
compound msmt_to_det_row_table_type {
short det_start_row ;
short det_end_row ;
}; // msmt_to_det_row_table_type
variables:
msmt_to_det_row_table_type measurement_to_detector_row_table(time, scanline,
ground_pixel) ;
```

Remarks

Table 96: CDL definition measurement_to_detector_row_table variable

field	type	unit	description
det_start_row	short	1	Detector start row for measurement row
det_end_row	short	1	Detector end row for measurement row

Table 97: Fields in the measurement_to_detector_row_table variable.

8.45 Variable: nominal_wavelength

The nominal_wavelength provides for each ground pixel the wavelength measured by a spectral channel and is defined by the design parameters of the instrument. The wavelength values as provided by the nominal_wavelength are based on the Calibration Key Data (CKD) which are input to the L01b processing (section 5). See also the discussion on calibrated_wavelength in section 8.23.

Variable	Storage type	Units
nominal_wavelength	float	nm

CDL

```
float nominal_wavelength(time,ground_pixel,spectral_channel) ;
nominal_wavelength:long_name = "spectral channel nominal wavelength" ;
nominal_wavelength:standard_name = "radiation_wavelength" ;
nominal_wavelength:units = "nm" ;
nominal_wavelength:comment = "The nominal spectral wavelength for each
cross track pixel as a function of the spectral channel." ;
```

Remarks The nominal_wavelength provides for each pixel the wavelength measured by a spectral channel and is defined by the design parameters of the instrument.

The values mentioned for valid_min and valid_max apply to the Band1 product and serve as an example. The valid values for all products are listed in Table 1 in section 4.2

Table 98: CDL definition nominal_wavelength variable

8.46 Variable: nominal_wavelength_error

Variable	Storage type	Units
nominal_wavelength_error	float	nm

CDL

```
float nominal_wavelength_error(time,ground_pixel,spectral_channel) ;
nominal_wavelength_error:long_name = "spectral channel nominal wavelength
error" ;
nominal_wavelength_error:standard_name = "radiation_wavelength standard_-
error" ;
nominal_wavelength_error:units = "nm" ;
nominal_wavelength_error:comment = "The nominal spectral wavelength error
for each cross track pixel as a function of the spectral channel." ;
```

Remarks The nominal_wavelength_error provides for each pixel the standard deviation wavelength measured by a spectral channel and is defined by the design parameters of the instrument.

The values mentioned for valid_min and valid_max apply to the Band1 product and serve as an example. The valid values for all products are listed in Table 1 in section 4.2

Table 99: CDL definition nominal_wavelength_error variable

8.47 Variable: sample_cycle

The concept of *"sample cycle"* has been introduced to allow for comparison of the different radiance products (i.e. bands). In principle, the eight products all can have different co-addition periods, i.e. the time period in which independent measurements are added in order to reduce the data rate as well as to increase the signal–to-noise ratio. The number of independent measurements is depending on the integration time which differs for each band, but is fixed for a specific instrument configuration.

For all bands measurements start at the same time but because the co-addition time may be different the scanlines may have a different time stamp. However, after a period of length sample_cycle_length the measurement cycle is repeated and again the measurements start at the same time. Thus, within the sample_cycle a fixed number (for a certain instrument configuration) of scanlines is collected, which differ for each radiance product. However, the sample_cycle index is the same for all these products.

Variable	Storage type	Units
sample_cycle	int	none

CDL

```
int sample_cycle(time,scanline) ;
sample_cycle:long_name = "sample cycle" ;
sample_cycle:units = "1" ;
sample_cycle:comment = "sample_cycle provides a sample_cycle index for each
scanline; index starts at 0" ;
```

Remarks One unique set of sample_cycle indexes is applicable to all radiance products (i.e. bands) originating from the same orbit.

Table 100: CDL definition sample_cycle variable

8.48 Variable: sample_cycle_length

Variable	Storage type	Units
sample_cycle_length	int	ms

CDL
```
int sample_cycle_length(time,scanline) ;
sample_cycle_length:long_name = "length of sample cycle" ;
sample_cycle_length:units = "ms" ;
sample_cycle_length:comment = "Length of sample_cycle" ;
```

Remarks

Table 101: CDL definition sample_cycle_length variable

8.49 Variable: monitor_straylight_observed

Variable	Storage type	Units
monitor_straylight_observed	float	electron.s-1

CDL
```
float monitor_straylight_observed(time,scanline,dual_dim,spectral_channel)
;
monitor_straylight_observed:units = "electron.s-1" ;
monitor_straylight_observed:comment = "Observed stray light from the stray
light areas on the detector, for monitoring purposes" ;
```

Remarks

Table 102: CDL definition monitor_straylight_observed variable

8.50 Variable: offset_readout_register

Variable	Storage type	Units
offset_readout_register	n/a	none

CDL
```
types:  datapoint_type {
double value ;
double error ;
}
offset_readout_register(time,scanline,ccd_gain,parity) ;
offset_readout_register:comment = "Detector and electronics offset value
calculated from the detector's read-out register" ;
```

Remarks Only available for UVN bands

Table 103: CDL definition offset_readout_register variable

8.51 Variable: irradiance_avg

	Variable	Storage type	Units
	irradiance_avg	float	mol.s^{-1}.m^{-2}.nm^{-1}
CDL	float irradiance_avg(time,pixel,spectral_channel) ; irradiance_avg:units = "mol.s-1.m-2.nm-1" ; irradiance_avg:ancillary_variables = "irradiance_avg_noise irradiance_avg_-error" ; irradiance_avg:comment = "Averaged measured spectral irradiance for each spectral pixel of all measurements in the group" ;		
Remarks			

Table 104: CDL definition irradiance_avg variable

8.52 Variable: irradiance_avg_noise

	Variable	Storage type	Units
	irradiance_avg_noise	float	none
CDL	float irradiance_avg_noise(time,pixel,spectral_channel) ; irradiance_avg_noise:comment = "Average irradiance signal noise for each spectral pixel of all measurements in the group" ;		
Remarks			

Table 105: CDL definition irradiance_avg_noise variable

8.53 Variable: irradiance_avg_error

	Variable	Storage type	Units
	irradiance_avg_error	float	none
CDL	float irradiance_avg_error(time,pixel,spectral_channel) ; irradiance_avg_error:comment = "Average irradiance signal error for each spectral pixel of all measurements in the group" ;		
Remarks			

Table 106: CDL definition irradiance_avg_error variable

8.54 Variable: irradiance_avg_quality_level

Variable	Storage type	Units
irradiance_avg_quality_level	ubyte	none

CDL
```
ubyte irradiance_avg_quality_level(time,pixel,spectral_channel) ;
irradiance_avg_quality_level:long_name = "qualiy level of spectral channel"
;
irradiance_avg_quality_level:valid_min = 0 ;
irradiance_avg_quality_level:valid_max = 100 ;
irradiance_avg_quality_level:comment = "Overall calculated quality
assessment information for each (spectral) pixel in the averaged data" ;
```

Remarks

Table 107: CDL definition irradiance_avg_quality_level variable

8.55 Variable: irradiance_avg_std

Variable	Storage type	Units
irradiance_avg_std	float	none

CDL
```
float irradiance_avg_std(time,pixel,spectral_channel) ;
irradiance_avg_std:comment = "Average irradiance signal standard deviation
for each spectral pixel of all measurements in the group" ;
```

Remarks

Table 108: CDL definition irradiance_avg_std variable

8.56 Variable: irradiance_avg_spectral_channel_quality

Variable	Storage type	Units
irradiance_avg_spectral_channel_-quality	ubyte	none

CDL
```
ubyte irradiance_avg_spectral_channel_quality(time,pixel,spectral_channel)
;
irradiance_avg_spectral_channel_quality:long_name = "spectral channel
quality flag" ;
irradiance_avg_spectral_channel_quality:valid_min = 0 ;
irradiance_avg_spectral_channel_quality:valid_max = 254 ;
irradiance_avg_spectral_channel_quality:flag_values = 0UB, 1UB, 2UB, 8UB,
16UB, 32UB, 64UB, 128UB ;
irradiance_avg_spectral_channel_quality:flag_meanings = no_error, missing,
bad_pixel, processing_error, saturated, transient, rts, underflow ;
irradiance_avg_spectral_channel_quality:comment = "Quality assessment
information about a (spectral) pixel in all measurements." ;
```

Remarks Flags of measurements ignored by the averaging algorithms are present.

Table 109: CDL definition irradiance_avg_spectral_channel_quality variable

8.57 Variable: irradiance_avg_col

	Variable	Storage type	Units
	irradiance_avg_col	float	none
CDL	float irradiance_avg_col(time,scanline,pixel) ;		
Remarks			

Table 110: CDL definition irradiance_avg_col variable

8.58 Variable: radiance_avg

	Variable	Storage type	Units
	radiance_avg	float	mol.s^{-1}.m^{-2}.nm^{-1}.sr^{-1}
CDL	float radiance_avg(time,ground_pixel,spectral_channel) ; radiance_avg:units = "mol.s-1.m-2.nm-1.sr-1" ; radiance_avg:coordinates = "longitude latitude" ; radiance_avg:ancillary_variables = "radiance_avg_noise radiance_avg_error" ; radiance_avg:comment = "Averaged measured spectral radiance for each spectral pixel of all measurements in the group" ;		
Remarks	There is no standard_name for spectral photon radiance as measured by sensors on board satellites. In line with the standard_name for radiance that has been suggested by the cf-satellite user community on the Unidata mailing list, toa_outgoing_spectral_photon_radiance is suggested here.		

Table 111: CDL definition radiance_avg variable

8.59 Variable: radiance_avg_error

	Variable	Storage type	Units
	radiance_avg_error	float	none
CDL	float radiance_avg_error(time,ground_pixel,spectral_channel) ; radiance_avg_error:coordinates = "longitude latitude" ; radiance_avg_error:comment = "Average radiance signal error for each spectral pixel of all measurements in the group" ;		
Remarks			

Table 112: CDL definition radiance_avg_error variable

8.60 Variable: radiance_avg_noise

Variable	Storage type	Units
radiance_avg_noise	float	none

CDL
```
float radiance_avg_noise(time,ground_pixel,spectral_channel) ;
radiance_avg_noise:coordinates = "longitude latitude" ;
radiance_avg_noise:comment = "Average radiance signal noise for each
spectral pixel of all measurements in the group" ;
```

Remarks

Table 113: CDL definition radiance_avg_noise variable

8.61 Variable: radiance_avg_quality_level

Variable	Storage type	Units
radiance_avg_quality_level	ubyte	none

CDL
```
ubyte radiance_avg_quality_level(time,ground_pixel,spectral_channel) ;
radiance_avg_quality_level:long_name = "qualiy level of spectral channel" ;
radiance_avg_quality_level:valid_min = 0 ;
radiance_avg_quality_level:valid_max = 100 ;
radiance_avg_quality_level:coordinates = "longitude latitude" ;
radiance_avg_quality_level:comment = "Overall calculated quality assessment
information for each (spectral) pixel in the averaged data" ;
```

Remarks

Table 114: CDL definition radiance_avg_quality_level variable

8.62 Variable: radiance_avg_spectral_channel_quality

Variable	Storage type	Units
radiance_avg_spectral_channel_quality	ubyte	none

CDL
```
ubyte radiance_avg_spectral_channel_quality(time,ground_pixel,spectral_-
channel) ;
radiance_avg_spectral_channel_quality:long_name = "spectral channel quality
flag" ;
radiance_avg_spectral_channel_quality:valid_min = 0 ;
radiance_avg_spectral_channel_quality:valid_max = 254 ;
radiance_avg_spectral_channel_quality:coordinates = "longitude latitude" ;
radiance_avg_spectral_channel_quality:flag_values = 0UB, 1UB, 2UB, 8UB,
16UB, 32UB, 64UB, 128UB ;
radiance_avg_spectral_channel_quality:flag_meanings = no_error, missing,
bad_pixel, processing_error, saturated, transient, rts, underflow ;
radiance_avg_spectral_channel_quality:comment = "Quality assessment
information about a (spectral) pixel in all measurements." ;
```

Remarks Flags of measurements ignored by the averaging algorithms are present.

Table 115: CDL definition radiance_avg_spectral_channel_quality variable

8.63 Variable: radiance_avg_std

	Variable	Storage type	Units
	radiance_avg_std	float	none
CDL	float radiance_avg_std(time,ground_pixel,spectral_channel) ; radiance_avg_std:coordinates = "longitude latitude" ; radiance_avg_std:comment = "Average radiance signal standard deviation for each spectral pixel of all measurements in the group" ;		
Remarks			

Table 116: CDL definition radiance_avg_std variable

8.64 Variable: radiance_avg_row

	Variable	Storage type	Units
	radiance_avg_row	float	none
CDL	float radiance_avg_row(time,scanline,spectral_channel) ; radiance_avg_row:comment = "Averaged measured spectral radiance value of a single row in a measurement" ;		
Remarks			

Table 117: CDL definition radiance_avg_row variable

8.65 Variable: radiance_avg_data

	Variable	Storage type	Units
	radiance_avg_data	float	none
CDL	float radiance_avg_data(time,scanline) ; radiance_avg_data:comment = "Averaged measured spectral radiance value of a single measurements" ;		
Remarks			

Table 118: CDL definition radiance_avg_data variable

8.66 Variable: percentage_ground_pixels_geolocation_error

	Variable	Storage type	Units
	percentage_ground_pixels_geolocation_-error	float	none
CDL	float percentage_ground_pixels_geolocation_error(time) ; percentage_ground_pixels_geolocation_error:comment = "Percentage of ground pixels with geolocation error" ;		
Remarks			

Table 119: CDL definition percentage_ground_pixels_geolocation_error variable

8.67 Variable: percentage_spectral_channels_rts

Variable	Storage type	Units
percentage_spectral_channels_rts	float	none

CDL
```
float percentage_spectral_channels_rts(time) ;
percentage_spectral_channels_rts:comment = "Percentage of spectral channels
for which the RTS flag is set" ;
```

Remarks

Table 120: CDL definition percentage_spectral_channels_rts variable

8.68 Variable: percentage_spectral_channels_per_scanline_transient

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_transient	float	none

CDL
```
float percentage_spectral_channels_per_scanline_transient(time,scanline) ;
percentage_spectral_channels_per_scanline_transient:comment = "Percentage
of spectral channels per scanline for which the transient flag is set" ;
```

Remarks

Table 121: CDL definition percentage_spectral_channels_per_scanline_transient variable

8.69 Variable: oob_sl_nir_corr_row_avg_blu_irr

Variable	Storage type	Units
oob_sl_nir_corr_row_avg_blu_irr	float	electron.s-1

CDL
```
float oob_sl_nir_corr_row_avg_blu_irr(time,scanline,dual_dim,spectral_-
channel) ;
oob_sl_nir_corr_row_avg_blu_irr:units = "electron.s-1" ;
oob_sl_nir_corr_row_avg_blu_irr:comment = "Calculated oob straylight nir
correction row average, blue side radiance, for monitoring purposes" ;
```

Remarks

Table 122: CDL definition oob_sl_nir_corr_row_avg_blu_irr variable

8.70 Variable: oob_sl_nir_dp_factor_blu_irr

Variable	Storage type	Units
oob_sl_nir_dp_factor_blu_irr	float	electron.s-1.nm-1

CDL
```
float oob_sl_nir_dp_factor_blu_irr(time,scanline,fiber) ;
oob_sl_nir_dp_factor_blu_irr:units = "electron.s-1.nm-1" ;
oob_sl_nir_dp_factor_blu_irr:comment = "Calculated oob straylight nir dp
factor, blue side irradiance, for monitoring purposes" ;
```

Remarks

Table 123: CDL definition oob_sl_nir_dp_factor_blu_irr variable

8.71 Variable: oob_sl_nir_corr_row_avg_red_irr

Variable	Storage type	Units
oob_sl_nir_corr_row_avg_red_irr	float	electron.s-1

CDL

```
float oob_sl_nir_corr_row_avg_red_irr(time,scanline,dual_dim,spectral_-
channel) ;
oob_sl_nir_corr_row_avg_red_irr:units = "electron.s-1" ;
oob_sl_nir_corr_row_avg_red_irr:comment = "Calculated oob straylight nir
correction row average, red side irradiance, for monitoring purposes" ;
```

Remarks

Table 124: CDL definition oob_sl_nir_corr_row_avg_red_irr variable

8.72 Variable: oob_sl_nir_dp_factor_red_irr

Variable	Storage type	Units
oob_sl_nir_dp_factor_red_irr	float	electron.s-1.nm-1

CDL

```
float oob_sl_nir_dp_factor_red_irr(time,scanline,fiber) ;
oob_sl_nir_dp_factor_red_irr:units = "electron.s-1.nm-1" ;
oob_sl_nir_dp_factor_red_irr:comment = "Calculated oob straylight nir dp
factor, red side irradiance, for monitoring purposes" ;
```

Remarks

Table 125: CDL definition oob_sl_nir_dp_factor_red_irr variable

8.73 Variable: oob_sl_nir_corr_row_avg_blu_rad

Variable	Storage type	Units
oob_sl_nir_corr_row_avg_blu_rad	float	electron.s-1

CDL

```
float oob_sl_nir_corr_row_avg_blu_rad(time,scanline,dual_dim,spectral_-
channel) ;
oob_sl_nir_corr_row_avg_blu_rad:units = "electron.s-1" ;
oob_sl_nir_corr_row_avg_blu_rad:comment = "Calculated oob straylight nir
correction row average, blue side radiance, for monitoring purposes" ;
```

Remarks

Table 126: CDL definition oob_sl_nir_corr_row_avg_blu_rad variable

8.74 Variable: oob_sl_nir_dp_factor_blu_rad

Variable	Storage type	Units
oob_sl_nir_dp_factor_blu_rad	float	electron.s-1.nm-1

CDL

```
float oob_sl_nir_dp_factor_blu_rad(time,scanline,fiber) ;
oob_sl_nir_dp_factor_blu_rad:units = "electron.s-1.nm-1" ;
oob_sl_nir_dp_factor_blu_rad:comment = "Calculated oob straylight nir dp
factor, blue side radiance, for monitoring purposes" ;
```

Remarks

Table 127: CDL definition oob_sl_nir_dp_factor_blu_rad variable

8.75 Variable: oob_sl_nir_corr_row_avg_red_rad

Variable	Storage type	Units
oob_sl_nir_corr_row_avg_red_rad	float	electron.s-1

CDL

```
float oob_sl_nir_corr_row_avg_red_rad(time,scanline,dual_dim,spectral_-
channel) ;
oob_sl_nir_corr_row_avg_red_rad:units = "electron.s-1" ;
oob_sl_nir_corr_row_avg_red_rad:comment = "Calculated oob straylight nir
correction row average, red side radiance, for monitoring purposes" ;
```

Remarks

Table 128: CDL definition oob_sl_nir_corr_row_avg_red_rad variable

8.76 Variable: oob_sl_nir_dp_factor_red_rad

Variable	Storage type	Units
oob_sl_nir_dp_factor_red_rad	float	electron.s-1.nm-1

CDL

```
float oob_sl_nir_dp_factor_red_rad(time,scanline,fiber) ;
oob_sl_nir_dp_factor_red_rad:units = "electron.s-1.nm-1" ;
oob_sl_nir_dp_factor_red_rad:comment = "Calculated oob straylight nir dp
factor, red side radiance, for monitoring purposes" ;
```

Remarks

Table 129: CDL definition oob_sl_nir_dp_factor_red_rad variable

8.77 Variable: solar_azimuth_angle_irr_cal

	Variable	Storage type	Units
	solar_azimuth_angle	float	degree

CDL
```
float solar_azimuth_angle(time,scanline) ;
solar_azimuth_angle:long_name = "solar azimuth angle" ;
solar_azimuth_angle:standard_name = "solar_azimuth_angle" ;
solar_azimuth_angle:units = "degree" ;
solar_azimuth_angle:valid_min = -180.f ;
solar_azimuth_angle:valid_max = 180.f ;
solar_azimuth_angle:comment = "Azimuth angle of the sun measured from the
instrument" ;
```

Remarks

Table 130: CDL definition solar_azimuth_angle variable

8.78 Variable: irradiance_avg_data

	Variable	Storage type	Units
	irradiance_avg_data	float	none

CDL
```
float irradiance_avg_data(time,scanline) ;
irradiance_avg_data:comment = "Averaged measured spectral irradiance value
of a single measurements" ;
```

Remarks

Table 131: CDL definition irradiance_avg_data variable

8.79 Variable: solar_azimuth_angle_rad_cal

	Variable	Storage type	Units
	solar_azimuth_angle	float	degree

CDL
```
float solar_azimuth_angle(time,scanline,ground_pixel) ;
solar_azimuth_angle:long_name = "solar azimuth angle" ;
solar_azimuth_angle:standard_name = "solar_azimuth_angle" ;
solar_azimuth_angle:units = "degree" ;
solar_azimuth_angle:valid_min = -180.f ;
solar_azimuth_angle:valid_max = 180.f ;
solar_azimuth_angle:coordinates = "longitude latitude" ;
solar_azimuth_angle:comment = "Solar azimuth angle at the ground pixel
location on the reference ellipsoid.  Angle is measured clockwise from the
North (East = +90, South = -+180, West = -90)" ;
```

Remarks

Table 132: CDL definition solar_azimuth_angle variable

8.80 Variable: signal_avg

Variable	Storage type	Units
signal_avg	float	none

CDL
```
float signal_avg(time,pixel,spectral_channel) ;
signal_avg:ancillary_variables = "signal_avg_noise signal_avg_error" ;
signal_avg:comment = "Averaged measured spectral signal for each spectral
pixel of all measurements in the group" ;
```

Remarks Unit differs between groups

Table 133: CDL definition signal_avg variable

8.81 Variable: signal_avg_error

Variable	Storage type	Units
signal_avg_error	float	none

CDL
```
float signal_avg_error(time,pixel,spectral_channel) ;
signal_avg_error:comment = "Average signal error for each spectral pixel of
all measurements in the group" ;
```

Remarks Unit differs between groups

Table 134: CDL definition signal_avg_error variable

8.82 Variable: signal_avg_noise

Variable	Storage type	Units
signal_avg_noise	float	none

CDL
```
float signal_avg_noise(time,pixel,spectral_channel) ;
signal_avg_noise:comment = "Average signal noise for each spectral pixel of
all measurements in the group" ;
```

Remarks Unit differs between groups

Table 135: CDL definition signal_avg_noise variable

8.83 Variable: signal_avg_quality_level

Variable	Storage type	Units
signal_avg_quality_level	ubyte	none

CDL
```
ubyte signal_avg_quality_level(time,pixel,spectral_channel) ;
signal_avg_quality_level:long_name = "qualiy level of spectral channel" ;
signal_avg_quality_level:valid_min = 0 ;
signal_avg_quality_level:valid_max = 100 ;
signal_avg_quality_level:comment = "Overall calculated quality assessment
information for each (spectral) pixel in the averaged data" ;
```

Remarks

Table 136: CDL definition signal_avg_quality_level variable

8.84 Variable: signal_avg_spectral_channel_quality

Variable	Storage type	Units
signal_avg_spectral_channel_quality	ubyte	none

CDL
```
ubyte signal_avg_spectral_channel_quality(time,pixel,spectral_channel) ;
signal_avg_spectral_channel_quality:long_name = "spectral channel quality
flag" ;
signal_avg_spectral_channel_quality:valid_min = 0 ;
signal_avg_spectral_channel_quality:valid_max = 254 ;
signal_avg_spectral_channel_quality:flag_values = 0UB, 1UB, 2UB, 8UB, 16UB,
32UB, 64UB, 128UB ;
signal_avg_spectral_channel_quality:flag_meanings = no_error, missing, bad_-
pixel, processing_error, saturated, transient, rts, underflow ;
signal_avg_spectral_channel_quality:comment = "Quality assessment
information about a (spectral) pixel in all measurements." ;
```

Remarks Flags of measurements ignored by the averaging algorithms are present.

Table 137: CDL definition signal_avg_spectral_channel_quality variable

8.85 Variable: signal_avg_std

Variable	Storage type	Units
signal_avg_std	float	none

CDL
```
float signal_avg_std(time,pixel,spectral_channel) ;
signal_avg_std:comment = "Average signal standard deviation for each
spectral pixel of all measurements in the group" ;
```

Remarks Unit differs between groups

Table 138: CDL definition signal_avg_std variable

8.86 Variable: signal_avg_data

Variable	Storage type	Units
signal_avg_data	float	none

CDL
```
float signal_avg_data(time,scanline) ;
signal_avg_data:comment = "Averaged measured spectral signal value of a
single measurement" ;
```

Remarks

Table 139: CDL definition signal_avg_data variable

8.87 Variable: signal_avg_row

	Variable	Storage type	Units
	signal_avg_row	float	none
CDL	float signal_avg_row(time,scanline,spectral_channel) ; signal_avg_row:comment = "Averaged measured spectral signal value of a single row in a measurement" ;		
Remarks			

Table 140: CDL definition signal_avg_row variable

8.88 Variable: signal_avg_col

	Variable	Storage type	Units
	signal_avg_col	float	none
CDL	float signal_avg_col(time,scanline,pixel) ; signal_avg_col:comment = "Averaged measured spectral signal value of a single column in a measurement" ;		
Remarks			

Table 141: CDL definition signal_avg_col variable

8.89 Variable: small_pixel_signal

	Variable	Storage type	Units
	small_pixel_signal	float(*)	none
CDL	types: float(*) small_pixel_signal_type ; small_pixel_signal_type small_pixel_signal(time,scanline,pixel) ; small_pixel_signal:long_name = "small pixel photon signal" ; small_pixel_signal:_FillValue = 0x1.ep+122 ; small_pixel_signal:comment = "Measured signal for the spectral channel dedicated for the small pixel measurements" ;		
Remarks			

Table 142: CDL definition small_pixel_signal variable

8.90 Variable: percentage-spectral_channels_per_scanline_rts

	Variable	Storage type	Units
	percentage_spectral_channels_per_-scanline_rts	float	none
CDL	float percentage_spectral_channels_per_scanline_rts(time,scanline) ; percentage_spectral_channels_per_scanline_rts:comment = "Percentage of spectral channels per scanline for which the RTS flag is set" ;		
Remarks			

Table 143: CDL definition percentage_spectral_channels_per_scanline_rts variable

8.91 Variable: percentage-scanlines_with_processing_steps_skipped

Variable	Storage type	Units
percentage_scanlines_with_processing_-steps_skipped	float	none

CDL
```
float percentage_scanlines_with_processing_steps_skipped(time) ;
percentage_scanlines_with_processing_steps_skipped:comment = "Percentage of
scanlines for which one or more processing steps were skipped" ;
```

Remarks

Table 144: CDL definition percentage_scanlines_with_processing_steps_skipped variable

8.92 Variable: percentage_scanlines_with_residual_correction_skipped

Variable	Storage type	Units
percentage_scanlines_with_residual_-correction_skipped	float	none

CDL
```
float percentage_scanlines_with_residual_correction_skipped(time) ;
percentage_scanlines_with_residual_correction_skipped:comment = "Percentage
of scanlines for which residual correction was skipped" ;
```

Remarks

Table 145: CDL definition percentage_scanlines_with_residual_correction_skipped variable

8.93 Variable: percentage_ground_pixels_descending_side_orbit

Variable	Storage type	Units
percentage_ground_pixels_descending_-side_orbit	float	none

CDL
```
float percentage_ground_pixels_descending_side_orbit(time) ;
percentage_ground_pixels_descending_side_orbit:comment = "Percentage of
ground pixels on the descending side of the orbit" ;
```

Remarks

Table 146: CDL definition percentage_ground_pixels_descending_side_orbit variable

8.94 Variable: percentage_spectral_channels_per_scanline_defective

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_defective	float	none

CDL
```
float percentage_spectral_channels_per_scanline_defective(time,scanline) ;
percentage_spectral_channels_per_scanline_defective:comment = "Percentage
of spectral channels per scanline for which the defective flag is set" ;
```

Remarks

Table 147: CDL definition percentage_spectral_channels_per_scanline_defective variable

8.95 Variable: percentage_scanlines_in_spacecraft_manoeuvre

Variable	Storage type	Units
percentage_scanlines_in_spacecraft_-manoeuvre	float	none

CDL
```
float percentage_scanlines_in_spacecraft_manoeuvre(time) ;
percentage_scanlines_in_spacecraft_manoeuvre:comment = "Percentage of
scanlines affected by spacecraft manoeuvres" ;
```

Remarks

Table 148: CDL definition percentage_scanlines_in_spacecraft_manoeuvre variable

8.96 Variable: monitor_straylight_calculated

Variable	Storage type	Units
monitor_straylight_calculated	float	electron.s-1

CDL
```
float monitor_straylight_calculated(time,scanline,dual_dim,spectral_channel)
;
monitor_straylight_calculated:units = "electron.s-1" ;
monitor_straylight_calculated:comment = "Calculated stray light, for
monitoring purposes" ;
```

Remarks

Table 149: CDL definition monitor_straylight_calculated variable

8.97 Variable: monitor_radiance_wavelength_shift

Variable	Storage type	Units
monitor_radiance_wavelength_shift	float	none

CDL
```
float monitor_radiance_wavelength_shift(time,scanline,pixel) ;
monitor_radiance_wavelength_shift:comment = "Wavelenght shift for a small
wavelength band around the specified wavelength, for monitoring purposes" ;
```

Remarks The name of the variable in the output file is monitor_radiance_wavelength_shift_*xxxx*nm where *xxxx* is the center wavelength value. The center wavelength and the bandwidth around the center wavelength can be found in the variable attributes center_wavelength and wavelength_bandwidth

Table 150: CDL definition monitor_radiance_wavelength_shift variable

8.98 Variable: monitor_gain_alignment_factor

Variable	Storage type	Units
monitor_gain_alignment_factor	float	none

CDL

```
float monitor_gain_alignment_factor(time,scanline) ;
monitor_gain_alignment_factor:comment = "Gain alignment factor for the
measurement calculated in the GainAlignmentCalculationUVN algorithm.
Applied gain alignment factor depends on the settings of this algorithm.
Default, the CKD setting of the gain alignment correction factor is used,
not the calculated." ;
```

Remarks

Table 151: CDL definition monitor_gain_alignment_factor variable

8.99 Variable: measurement_to_detector_row_table

Variable	Storage type	Units
measurement_to_detector_row_table	n/a	none

CDL

```
types:  msmt_to_det_row_table_type {
short detector_start_row ;
short detector_end_row ;
}
measurement_to_detector_row_table(time,scanline,ground_pixel) ;
measurement_to_detector_row_table:comment = "Conversion table from
measurement row to begin and end row on detector" ;
```

Remarks

Table 152: CDL definition measurement_to_detector_row_table variable

8.100 Variable: signal

Variable	Storage type	Units
signal	float	none

CDL

```
float signal(time,scanline,pixel,spectral_channel) ;
signal:long_name = "spectral photon signal" ;
signal:ancillary_variables = "signal_noise signal_error quality_level
spectral_channel_quality" ;
signal:comment = "Measured signal for each spectral pixel" ;
```

Remarks

Table 153: CDL definition signal variable

8.101 Variable: signal_error

Variable	Storage type	Units
signal_error	byte	none

CDL
```
byte signal_error(time,scanline,pixel,spectral_channel) ;
signal_error:long_name = "spectral photon signal error" ;
signal_error:units = "1" ;
signal_error:comment = "The signal_error is a measure for the one standard
deviation error of the bias of the measurement signal; it is expressed in
decibel (dB), i.e.  10 times the base-10 logarithmic value of the ratio
between the signal and the estimation error." ;
```

Remarks

Table 154: CDL definition signal_error variable

8.102 Variable: signal_noise

Variable	Storage type	Units
signal_noise	byte	none

CDL
```
byte signal_noise(time,scanline,pixel,spectral_channel) ;
signal_noise:long_name = "spectral photon signal noise, one standard
deviation" ;
signal_noise:units = "1" ;
signal_noise:comment = "The signal_noise is a measure for the one standard
deviation random error of the measurement signal; it is expressed in
decibel (dB), i.e.  10 times the base-10 logarithmic value of the ratio
between the signal and the random error." ;
```

Remarks

Table 155: CDL definition signal_noise variable

8.103 Variable: percentage_ground_pixels_night

Variable	Storage type	Units
percentage_ground_pixels_night	float	none

CDL
```
float percentage_ground_pixels_night(time) ;
percentage_ground_pixels_night:comment = "Percentage of ground pixels for
which the night flag is set" ;
```

Remarks

Table 156: CDL definition percentage_ground_pixels_night variable

8.104 Variable: percentage_spectral_channels_transient

Variable	Storage type	Units
percentage_spectral_channels_-transient	float	none

CDL
```
float percentage_spectral_channels_transient(time) ;
percentage_spectral_channels_transient:comment = "Percentage of spectral
channels for which the transient flag is set" ;
```

Remarks

Table 157: CDL definition percentage_spectral_channels_transient variable

8.105 Variable: offset_prepostscan_pixels

Variable	Storage type	Units
offset_prepostscan_pixels	n/a	none

CDL
```
types:  datapoint_type {
double value ;
double error ;
}
offset_prepostscan_pixels(time,scanline,ccd_gain,parity) ;
offset_prepostscan_pixels:comment = "Detector and electronics offset value
calculated from the detector's pre- and postscan pixels" ;
```

Remarks Only available for UVN bands

Table 158: CDL definition offset_prepostscan_pixels variable

8.106 Variable: percentage_spectral_channels_per_scanline_saturated

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_saturated	float	none

CDL
```
float percentage_spectral_channels_per_scanline_saturated(time,scanline) ;
percentage_spectral_channels_per_scanline_saturated:comment = "Percentage
of spectral channels per scanline for which the saturated flag is set" ;
```

Remarks

Table 159: CDL definition percentage_spectral_channels_per_scanline_saturated variable

8.107 Variable: monitor_smear_calculated

Variable	Storage type	Units
monitor_smear_calculated	float	electron

CDL
```
float monitor_smear_calculated(time,scanline,spectral_channel) ;
monitor_smear_calculated:units = "electron" ;
monitor_smear_calculated:comment = "Calculated detector smear values as
used for the detector smear correction, for monitoring purposes" ;
```

Remarks Only available for UVN bands

Table 160: CDL definition monitor_smear_calculated variable

8.108 Variable: radiance_avg_col

Variable	Storage type	Units
radiance_avg_col	float	none

CDL
```
float radiance_avg_col(time,scanline,ground_pixel) ;
radiance_avg_col:comment = "Averaged measured spectral radiance value of a
single column in a measurement" ;
```

Remarks

Table 161: CDL definition radiance_avg_col variable

8.109 Variable: offset_overscan_rows

Variable	Storage type	Units
offset_overscan_rows	n/a	none

CDL
```
types:  datapoint_type {
double value ;
double error ;
}
offset_overscan_rows(time,scanline,ccd_gain,parity) ;
offset_overscan_rows:comment = "Detector and electronics offset value
calculated from the detector's overscan rows" ;
```

Remarks Only available for UVN bands

Table 162: CDL definition offset_overscan_rows variable

8.110 Variable: percentage_spectral_channels_per_scanline_underflow

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_underflow	float	none

CDL

```
float percentage_spectral_channels_per_scanline_underflow(time,scanline) ;
percentage_spectral_channels_per_scanline_underflow:comment = "Percentage
of spectral channels per scanline for which the underflow flag is set" ;
```

Remarks

Table 163: CDL definition percentage_spectral_channels_per_scanline_underflow variable

8.111 Variable: offset_overscan_columns

Variable	Storage type	Units
offset_overscan_columns	n/a	none

CDL

```
types:  datapoint_type {
double value ;
double error ;
}
offset_overscan_columns(time,scanline,ccd_gain,parity) ;
offset_overscan_columns:comment = "Detector and electronics offset value
calculated from the detector's overscan columns" ;
```

Remarks Only available for UVN bands

Table 164: CDL definition offset_overscan_columns variable

8.112 Variable: percentage_scanlines_with_solar_angles_out_of_nominal_range

Variable	Storage type	Units
percentage_scanlines_with_solar_-angles_out_of_nominal_range	float	none

CDL

```
float percentage_scanlines_with_solar_angles_out_of_nominal_range(time) ;
percentage_scanlines_with_solar_angles_out_of_nominal_range:comment =
"Percentage of scanlines for which the solar angles are outside the nominal
range" ;
```

Remarks

Table 165: CDL definition percentage_scanlines_with_solar_angles_out_of_nominal_range variable

8.113 Variable: small_pixel_irradiance

	Variable	Storage type	Units
	small_pixel_irradiance	float(*)	none

CDL

```
types:  float(*) small_pixel_irradiance_type ;
small_pixel_irradiance_type small_pixel_irradiance(time,scanline,pixel) ;
small_pixel_irradiance:long_name = "small pixel photon signal" ;
small_pixel_irradiance:_FillValue = 0x1.ep+122 ;
small_pixel_irradiance:comment = "Measured signal for the spectral channel
dedicated for the small pixel measurements" ;
```

Remarks

Table 166: CDL definition small_pixel_irradiance variable

8.114 Variable: monitor_overscan_rows

	Variable	Storage type	Units
	monitor_overscan_rows	float	none

CDL

```
float monitor_overscan_rows(time,scanline,spectral_channel) ;
monitor_overscan_rows:comment = "Signal from the detector's overscan rows,
for monitoring purposes" ;
```

Remarks

Table 167: CDL definition monitor_overscan_rows variable

8.115 Variable: detector_pixel_filling_histogram

	Variable	Storage type	Units
	detector_pixel_filling_histogram	int	none

CDL

```
int detector_pixel_filling_histogram(time,scanline,nbins) ;
detector_pixel_filling_histogram:comment = "Histogram of the detector pixel
filling in electrons for each scanline" ;
```

Remarks Only available for UVN bands

Table 168: CDL definition detector_pixel_filling_histogram variable

8.116 Variable: offset_static_ckd

Variable	Storage type	Units
offset_static_ckd	n/a	none

CDL

```
types:  datapoint_type {
double value ;
double error ;
}
offset_static_ckd(time,scanline,ccd_gain,parity) ;
offset_static_ckd:comment = "Detector and electronics offset value,
obtained from the calibration key-data" ;
```

Remarks Only available for UVN bands

Table 169: CDL definition offset_static_ckd variable

8.117 Variable: percentage_spectral_channels_defective

Variable	Storage type	Units
percentage_spectral_channels_-defective	float	none

CDL

```
float percentage_spectral_channels_defective(time) ;
percentage_spectral_channels_defective:comment = "Flags of measurements
ignored by the averaging algorithms are present." ;
```

Remarks

Table 170: CDL definition percentage_spectral_channels_defective variable

8.118 Variable: percentage_spectral_channels_missing

Variable	Storage type	Units
percentage_spectral_channels_missing	float	none

CDL

```
float percentage_spectral_channels_missing(time) ;
percentage_spectral_channels_missing:comment = "Percentage of spectral
channels for which the missing flag is set" ;
```

Remarks

Table 171: CDL definition percentage_spectral_channels_missing variable

8.119 Variable: percentage_scanlines_in_south_atlantic_anomaly

Variable	Storage type	Units
percentage_scanlines_in_south_-atlantic_anomaly	float	none

CDL
```
float percentage_scanlines_in_south_atlantic_anomaly(time) ;
percentage_scanlines_in_south_atlantic_anomaly:comment = "Percentage of
scanlines in the South Atlantic Anomaly (SAA)" ;
```

Remarks

Table 172: CDL definition percentage_scanlines_in_south_atlantic_anomaly variable

8.120 Variable: storage_time

Variable	Storage type	Units
storage_time	float	s

CDL
```
float storage_time(nsettings,pixel) ;
storage_time:long_name = "Storage time" ;
storage_time:units = "s" ;
storage_time:comment = "The time a row has resided in the storage area of
the detector during read-out" ;
```

Remarks

Table 173: CDL definition storage_time variable

8.121 Variable: percentage_spectral_channels_per_scanline_processing_error

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_processing_error	float	none

CDL
```
float percentage_spectral_channels_per_scanline_processing_-
error(time,scanline) ;
percentage_spectral_channels_per_scanline_processing_error:comment =
"Percentage of spectral channels per scanline for which the processing
error flag is set" ;
```

Remarks

Table 174: CDL definition percentage_spectral_channels_per_scanline_processing_error variable

8.122 Variable: percentage_spectral_channels_saturated

Variable	Storage type	Units
percentage_spectral_channels_-saturated	float	none

CDL
```
float percentage_spectral_channels_saturated(time) ;
percentage_spectral_channels_saturated:comment = "Percentage of spectral
channels for which the saturated flag is set" ;
```

Remarks

Table 175: CDL definition percentage_spectral_channels_saturated variable

8.123 Variable: irradiance_avg_row

Variable	Storage type	Units
irradiance_avg_row	float	none

CDL
```
float irradiance_avg_row(time,scanline,spectral_channel) ;
irradiance_avg_row:comment = "Averaged measured spectral irradiance value
of a single row in a measurement" ;
```

Remarks

Table 176: CDL definition irradiance_avg_row variable

8.124 Variable: percentage_spectral_channels_processing_error

Variable	Storage type	Units
percentage_spectral_channels_-processing_error	float	none

CDL
```
float percentage_spectral_channels_processing_error(time) ;
percentage_spectral_channels_processing_error:comment = "Percentage of
spectral channels for which the processing error flag is set" ;
```

Remarks

Table 177: CDL definition percentage_spectral_channels_processing_error variable

8.125 Variable: percentage_spectral_channels_underflow

Variable	Storage type	Units
percentage_spectral_channels_-underflow	float	none

CDL
```
float percentage_spectral_channels_underflow(time) ;
percentage_spectral_channels_underflow:comment = "Percentage of spectral
channels for which the underflow flag is set" ;
```

Remarks

Table 178: CDL definition percentage_spectral_channels_underflow variable

8.126 Variable: monitor_read_out_register

Variable	Storage type	Units
monitor_read_out_register	float	none

CDL
```
float monitor_read_out_register(time,scanline,spectral_channel) ;
monitor_read_out_register:comment = "Spectral channel signal values as read
from the read out register" ;
```

Remarks Only available for UVN bands

Table 179: CDL definition monitor_read_out_register variable

8.127 Variable: percentage_ground_pixels_sun_glint

Variable	Storage type	Units
percentage_ground_pixels_sun_glint	float	none

CDL
```
float percentage_ground_pixels_sun_glint(time) ;
percentage_ground_pixels_sun_glint:comment = "Percentage of ground pixels
for which the sun glint flag is set" ;
```

Remarks

Table 180: CDL definition percentage_ground_pixels_sun_glint variable

8.128 Variable: monitor_radiance

Variable	Storage type	Units
monitor_radiance	float	none

CDL
```
float monitor_radiance(time,scanline,pixel) ;
monitor_radiance:comment = "Average radiance of a small wavelength band
around the specified wavelength, for monitoring purposes." ;
```

Remarks The name of the variable in the output file is monitor_radiance_*xxxx*nm where *xxxx* is the center wavelength value. The center wavelength and the bandwidth around the center wavelength can be found in the variable attributes center_wavelength and wavelength_bandwidth

Table 181: CDL definition monitor_radiance variable

8.129 Variable: percentage_ground_pixels_geometric_boundary_crossing

Variable	Storage type	Units
percentage_ground_pixels_geometric_-boundary_crossing	float	none

CDL
```
float percentage_ground_pixels_geometric_boundary_crossing(time) ;
percentage_ground_pixels_geometric_boundary_crossing:comment = "Percentage
of ground pixels that cross a geometric boundary, e.g.  dateline crossing"
;
```

Remarks

Table 182: CDL definition percentage_ground_pixels_geometric_boundary_crossing variable

8.130 Variable: monitor_smear_observed

Variable	Storage type	Units
monitor_smear_observed	float	electron

CDL
```
float monitor_smear_observed(time,scanline,dual_dim,spectral_channel) ;
monitor_smear_observed:units = "electron" ;
monitor_smear_observed:comment = "Observed detector smear values from the
masked regions of the detector, for monitoring purposes" ;
```

Remarks Only available for UVN bands

Table 183: CDL definition monitor_smear_observed variable

8.131 Variable: percentage_spectral_channels_per_scanline_missing

Variable	Storage type	Units
percentage_spectral_channels_per_-scanline_missing	float	none

CDL
```
float percentage_spectral_channels_per_scanline_missing(time,scanline) ;
percentage_spectral_channels_per_scanline_missing:comment = "Percentage of
spectral channels per scanline for which the missing flag is set" ;
```

Remarks

Table 184: CDL definition percentage_spectral_channels_per_scanline_missing variable

8.132 Variable: percentage_ground_pixels_solar_eclipse

Variable	Storage type	Units
percentage_ground_pixels_solar_-eclipse	float	none

CDL
```
float percentage_ground_pixels_solar_eclipse(time) ;
percentage_ground_pixels_solar_eclipse:comment = "Percentage of ground
pixels for which the solar eclipse flag is set" ;
```

Remarks

Table 185: CDL definition percentage_ground_pixels_solar_eclipse variable

Appendix A Estimated product size

Table 186 lists the estimated product sizes for the eight different standard radiance products. This estimation is based on the netCDF product definition as presented in this document, No compression has been applied. The baseline for the granule size of the standard products is one orbit; no sliced products, i.e. products covering a part of the orbit are foreseen.

Near-real time products (NRT) cover approximately one data slice of one total data downlink volume (see chapter 6) rather than one orbit. Because the standard products and NRT products are based on the same netCDF product definition, the product sizes of these slices can be estimated from Table 186 taking into account that the file size is proportional to the number of scanlines.

Detector	**UV**		**UVIS**		**NIR**		**SWIR**	
Band	1	2	3	4	5	6	7	8
`spectral_channel`	497	497	497	497	497	497	480	480
`scanline`	3246	3246	3246	3246	3246	3246	3246	3246
`ground_pixel`	77	448	450	450	448	448	215	215
Product size (GByte)	1.0	5.6	5.7	5.7	5.7	5.7	2.6	2.6

Table 186: Estimated product size of radiance products; these sizes largely depend on the size of the dimensions `spectral_channel`, `scanline` and `ground_pixel`.

Estimated product sizes for the irradiance, calibration and engineering products are presented in the table below (Table 187). For all products both the average data volumes and typical product size are presented. Especially for irradiance products there is a substantial difference between these, as the irradiance products are only generated for a selection of orbits.

Product	**Average Product size (GByte)**	**Typical Product size (GByte)**
Irradiance UVN	0.003	0.030
Irradiance SWIR	0.0006	0.006
Calibration UVN	19.59	17.52
Calibration SWIR	3.42	3.07
Engineering	0.06	0.06

Table 187: Estimated product size irradiance, calibration and engineering products

Appendix B Processing classes

Class	Name	Definition
		Undefined
0	Undefined	Value to indicate that a processing class was explicitly not set
		Nominal modes
1	Earth_radiance	Nominal earth radiance measurement
2	Earth_radiance_special	Earth radiance special mode. Can be used for special radiance measurements that have a special purpose (e.g. specific campaigns, geolocation validation) or require special handling (e.g. zoom modes)
3	Solar_irradiance	Nominal solar irradiance measurement
4	Solar_irradiance_special	Solar Irradiance special mode. Can be used for special irradiance measurements that have a special purpose (e.g. back-up diffuser) or require special handling (e.g. zoom modes)
5-15	-	Reserved for future use
		In-flight calibration modes
16	DLED	Detector LED measurement
17	CLED	Common LED measurement
18	WLS	White Light Source measurement
19	SLS	Spectral Line Source measurement
20	Dark	Dark current measurement
21	Background	Background measurement
22	CTE	UVN CTE measurement (using ClkDrvAb = 1)
23	No_clock	UVN no clocking measurement (using ClkDrvAll = 1)
24	Reverse_clock	UVN reverse clocking measurement (using reverse clocking timing for RiseR* and FallR*)
25	Storage	UVN CCD Storage section characterization measurement
26	Flush	Detector flush mode
27	Orbit_identification	Special IcID used for identification of the different orbit types.
28	RTS	Measurement for identification of pixels that have Random Telegraph Signal (RTS) behaviour
29	-	Reserved for future use
30	-	Reserved for future use
31	Background_radiance	Background measurement for an earth radiance measurement
32	Background_radiance_special	Background measurement for an earth radiance special measurement
33	Background_irradiance	Background measurement for a solar irradiance measurement
34	Background_irradiance_special	Background measurement for a solar irradiance special measurement
35-39	-	Reserved for future use
40	Electronics_cal_offset	Measurement for calibration of the electronics offset
41	Electronics_cal_gain	Measurement for calibration of the electronics gain
42	Electronics_cal_linearity	Measurement for calibration of the electronics (non-)linearity
43-63	-	Reserved for future use
		Test modes
64	ICU_test	ICU test mode
65	DEM_test	DEM test mode
66	Functional_test	Instrument functional test

Class	Name	Definition
67	Processor_test	Data processor software test
68	Auto_optimization	Automated optimization measurement
69-95	-	Reserved for future use
		Modes for specific processing
96	Discard	Discard / ignore data
97	Process_BU	Process data up-to binary units (i.e. no processing)
98	Process_electrons	Process data up-to electrons
99	Process_electron_flux	Process data up-to electrons per second
100	Process_photon_flux	Process data up-to photons per second (similar to Earth radiance)
101	Process_upto_binning	Process data up-to binning factor correction
102-127	-	Reserved for future use
		On-ground calibration modes
128	OCAL	Generic on-ground calibration processing, nominal mode
129	OCAL_special	Generic on-ground calibration processing, special mode
130-200	-	Reserved for future use
201	OCAL_radiance	Nominal on-ground calibration radiance measurement
202	OCAL_radiance_special	On-ground calibration radiance special mode
203	OCAL_irradiance	Nominal on-ground calibration irradiance measurement
204	OCAL_irradiance_special	On-ground calibration irradiance special mode
205-215	-	Reserved for future use
216	OCAL_DLED	On-ground calibration detector LED measurement
217	OCAL_CLED	On-ground calibration common LED measurement
218	OCAL_WLS	On-ground calibration White Light Source measurement
219	OCAL_SLS	On-ground calibration Spectral Line Source measurement
220	OCAL_Dark	On-ground calibration dark current measurement
221	OCAL_Background	On-ground calibration background measurement
222	OCAL_CTE	On-ground calibration UVN CTE measurement (using ClkDrvAb = 1)
223	OCAL_No_clock	On-ground calibration UVN no clocking measurement (using ClkDrvAll = 1)
224	OCAL_Reverse_clock	On-ground calibration UVN reverse clocking measurement (using reverse clocking timing for RiseR* and FallR*)
225	OCAL_Storage	On-ground calibration UVN CCD Storage section characterization measurement
226	OCAL_Flush	On-ground calibration detector flush mode
227-255	-	Reserved for future use
		Unused
Values 256 through 32767 are not used.		

Table 188: Processing classes