FCI L1 Dataset User Guide [FCIL1DUG]

Doc.No. : EUM/MTG/USR/13/719113
Issue : v1C e-signed
Date : 29 June 2015
WBS :

EUMETSAT
Eumetsat-Allee 1, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555
http://www.eumetsat.int

- *Page left blank -*

- *Page left blank -*

- *Page left blank -*

Table of Contents

1 INTRODUCTION

1.1 Scope

This document is a User Guide for FCI Level 1C datasets.

This release is a preliminary version published to accompany the release of FCI Level 1C test data packages. Although the document represents our current best knowledge of the FCI instrument functionality and characteristics, data processing, and output format, it is likely that there will be evolutions in this knowledge in the years up to the launch of the first MTG Imaging platform which will lead to updates in future releases of this document.

In addition, some areas of the document are currently incomplete and these will be detailed and expanded in subsequent releases.

1.2 Acronyms and Definitions

[Information to be added in a later issue]

1.3 Applicable and Reference Documents

1.3.1 Applicable Documents

None

1.3.2 Reference Documents

Acronym	Reference Number	Title
[WMO-386]	http://www.wmo.int/pages/prog/www/ois/Operational_Information/Publications/WMO_386/WMO_386_Vol_I_2009_en.pdf	WMO Manual on the Global Telecommunication System - Volume I. 2009 Edition.
[CF]	http://cfconventions.org/	CF Conventions Document
[NACDD]	https://geo-ide.noaa.gov/wiki/index.php?title=NetCDF_Attribute_Convention_for_Dataset_Discovery	NetCDF Attribute Convention for Dataset Discovery

1.4 Document Structure

The sections of this document present the following information:

Section 1 – An overview of the document.

Section Error! Reference source not found.– A brief introduction to the MTG programme, the MTG platform and the on-board instruments.

Section 3 – The Flexible Combined Image (FCI) instrument hardware and functionality.

Section 4 – The core algorithms used to process data the Level 0 data to Level 1C.

Section 5 – The characteristics of the Level 1C data including the use of reference grids, grouping of spectral channels, and use of quality indicators.

Section 6 – The file naming convention.

Section 7 – Characteristics of the netCDF dataset and the division of the product into chunks.

Section 8 – How to read and extract data from the FCI L1C netCDF files.

Appendix A – A detailed look at the netCDF formats inclduingh complete CDL descriptions.

Appendix B – Discussion of applicable netCDF standards and conventions

Appendix C – Identification of freely available tools for processing, manipulating or displaying these datasets.

2 METEOSAT THIRD GENERATION (MTG)

2.1 The MTG Programme

The Meteosat Third Generation (MTG) programme provides meteorological imagery over Europe and Africa and maintains continuity of the Meteosat programme, continuing and expanding the service provide by Meteosat Second Generation (MSG).

2.2 The MTG Platform

MTG is a twin satellite concept based on 3-axis stabilised platforms. The twin satellites comprise an imaging satellite, MTG-I, and a sounding satellite, MTG-S. Four imaging and two sounding satellites are planned.

The MTG-I payload comprises:

- The Flexible Combined Imager (FCI)
- The Lightning Imager (LI)
- The Data Collection System (DCS)
- Search and Rescue (GEOSAR)

The MTG-S payload comprises:

- The Infrared Sounder (IRS)
- The Sentinel-4 Ultra-violet, Visible and Near-infrared Sounder (UVN)

3 FLEXIBLE COMBINED IMAGER (FCI)

3.1 The FCI Mission

The FCI will provide follow-on services to the Full Disc Scanning Service (FDSS) and Rapid Scanning Service (RSS) currently provided by the Meteosat Second Generation (MSG) Spinning Enhanced Visible and Infrared Imager (SEVIRI). The RSS service provides Local Area Coverage (LAC) corresponding to a quarter disk. The operational coverage will nominally be the LAC 4 area which covers Europe.

Two imagery missions are defined that are combined in the FCI instrument design capabilities: the Full Disc High Spectral resolution Imagery (FDHSI) and the High spatial Resolution Fast Imagery (HRFI) missions.

3.2 Instrument Characteristics

3.2.1 Spectral Channels

The FCI has channels over 16 spectral ranges covering visible to infrared wavelengths.

Spectral Channel	Central Wavelength, λ0	Spectral Width, Δλ0	SSD
VIS 0.4	0.444 µm	0.060 µm	1.0 km
VIS 0.5	0.510 µm	0.040 µm	1.0 km
VIS 0.6	0.640 µm	0.050 µm	1.0 km 0.5 km (HR)
VIS 0.8	0.865 µm	0.050 µm	1.0 km
VIS 0.9	0.914 µm	0.020 µm	1.0 km
NIR 1.3	1.380 µm	0.030 µm	1.0 km
NIR 1.6	1.610 µm	0.050 µm	1.0 km
NIR 2.2	2.250 µm	0.050 µm	1.0 km 0.5 km (HR)
IR 3.8	3.800 µm	0.400 µm	2.0 km 1.0 km (HR)
WV 6.3	6.300 µm	1.000 µm	2.0 km
WV 7.3	7.350 µm	0.500 µm	2.0 km
IR 8.7	8.700 µm	0.400 µm	2.0 km
IR 9.7	9.660 µm	0.300 µm	2.0 km
IR 10.5	10.500 µm	0.700 µm	2.0 km 1.0 km (HR)
IR 12.3	12.300 µm	0.500 µm	2.0 km
IR 13.3	13.300 µm	0.600 µm	2.0 km

Table 1 FCI Spectral Channel Spectral and Spatial Requirements.
The spectral channels VIS 0.6, NIR 2.2, IR 3.8 and IR 10.5 are delivered in both FDHSI and HRFI spatial sampling configurations. The latter is indicated by (HR) in the table.

Spectral Channel	Min. Signal, α_{min}	Max. Signal, α_{max}	Ref. Signal, α_{ref}	SNR
VIS 0.4	0.01	1.20	0.01	>25
VIS 0.5	0.01	1.20	0.01	>25
VIS 0.6	0.01	1.20	0.01	>30 >12 ^{HR}
VIS 0.8	0.01	1.20	0.01	>21
VIS 0.9	0.01	0.80	0.01	>12
NIR 1.3	0.01	0.80	0.01	>40
NIR 1.6	0.01	1.00	0.01	>30
NIR 2.2	0.01	1.00	0.01	>25 >12 ^{HR}
Spectral Channel	**Min. Signal, T_{min}**	**Max. Signal, T_{max}**	**Ref. Signal, T_{ref}**	**NEdT**
IR 3.8	200K 350K	350K Fire range#2	300K 350-Fire range#2	<0.1K <0.2K ^{HR} <1K ^{FIRE}
WV 6.3	165K	270K	250K	<0.3K
WV 7.3	165K	285K	250K	<0.3K
IR 8.7	165K	330K	300K	<0.1K
IR 9.7	165K	310K	250K	<0.3K
IR 10.5	165K	340K	300K	<0.1K <0.2K ^{HR}
IR 12.3	165K	340K	300K	<0.2K
IR 13.3	165K	300K	270K	<0.2K

Table 2 FCI Spectral Channel Radiometric Requirements

Notes:

1. The channels VIS 0.6, NIR 2.2, IR 3.8 and IR 10.5 are delivered in FDHSI sampling and HRFI sampling configurations. The radiometric requirements for the HRFI sampling configuration are indicated by the superscript ^{HR} in the table.
2. For the IR 3.8 spectral channel the radiometric measurement range has been extended to the “Fire range” with reduced radiometric requirements for active fire monitoring indicated by the superscript ^{FIRE} in the table. The fire range is specified to meet the needs for a fire line of temperature 900K, at least 3 km in length and 30m in width on a back ground of 320K.
3. For the FCI, the value α represents the reflectance at the top of atmosphere (TOA) multiplied by the cosine of the solar zenith angle, i.e. $\alpha = \rho.\cos(\theta_s)$ allowing minimum, maximum and reference signals in terms of spectral radiance at the top of atmosphere to be derived for the VNIR spectral channels.
4. Radiometric noise is provided as Signal to Noise Ratio (SNR) for Visible and Near Infrared (VNIR) spectral channels and Noise Equivalent delta Temperature (NEdT) for Infrared spectral channels.

3.2.2 Spectral Response Function (SRF)

Figure 1 shows purely illustrative plots of the theoretical SRFs for each of the FCI spectral channels. These plots do not represent the final SRFs which are still to be measured. Details of how to access a datafile which contains the measured SRF profiles will be added in a later issue of this document.

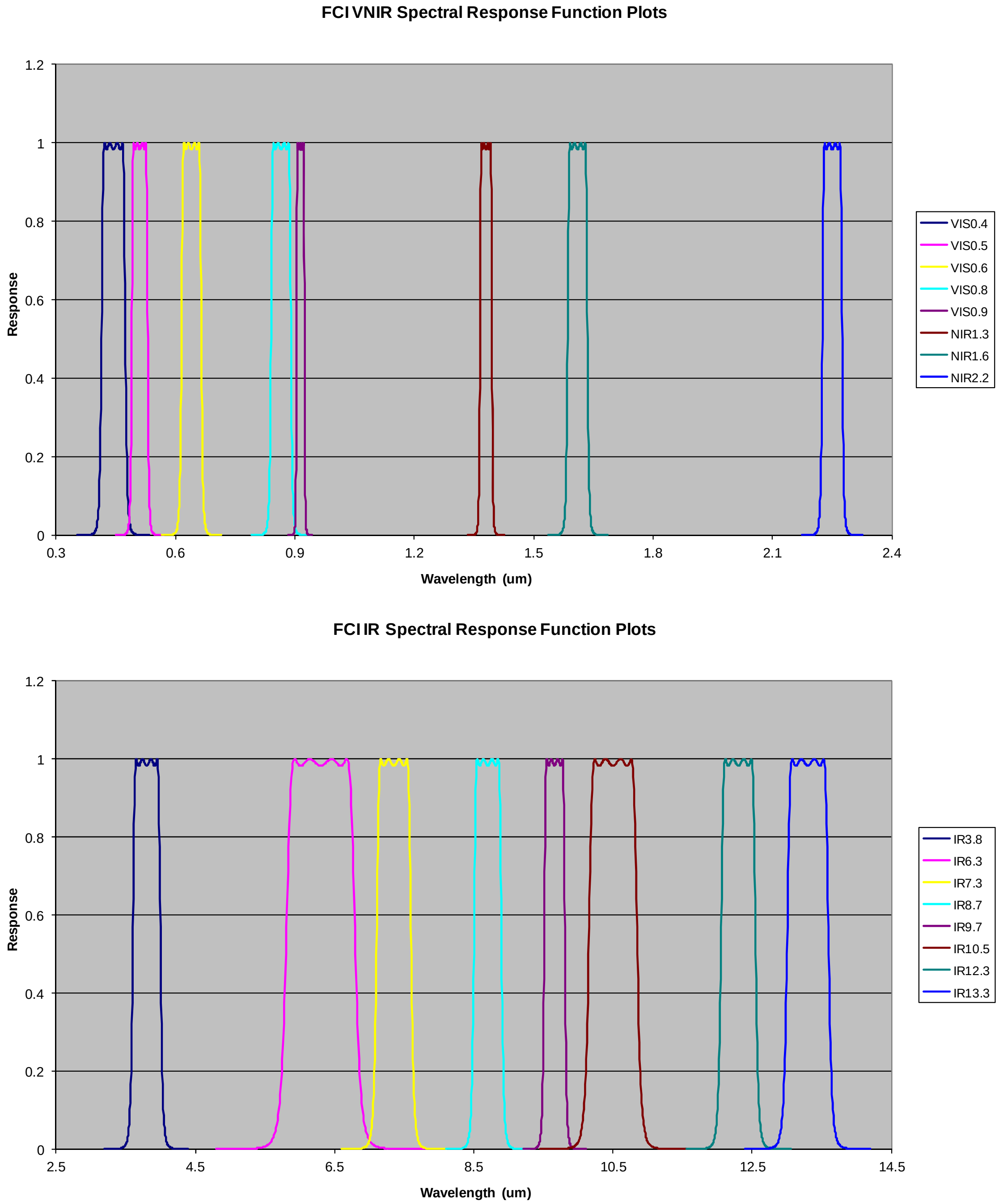


Figure 1 Theoretical FCI Spectral Response Functionsfor illustrative purposes only

3.2.3 Image Acquisition Principle

The FCI data is acquired by scanning the Earth across the detector arrays in an alternating east to west (E-W) and west to east (W-E) direction, with a south to north (S-N) movement between the alternating scans. The band of data collected in a single scan is referred to as a swath. The swaths are numbered from south to north staring from 1. Due to the nature of the scan the level 1b swaths are inclined with respect to the level 1c grid, see Figure 2.

Each swath is 180 km wide (excluding the required overlap) and the time between points at either side of the swath boundary varies between 0 to 20 seconds maximum at the equator. The duration of a swath is approximately 3 seconds duration at the pole and 10 seconds at the equator. The spacecraft performs a yaw flip between summer and winter observation modes, which reverses the detector, but the scan pattern is programmed to remain the same no matter the yaw flip orientation.

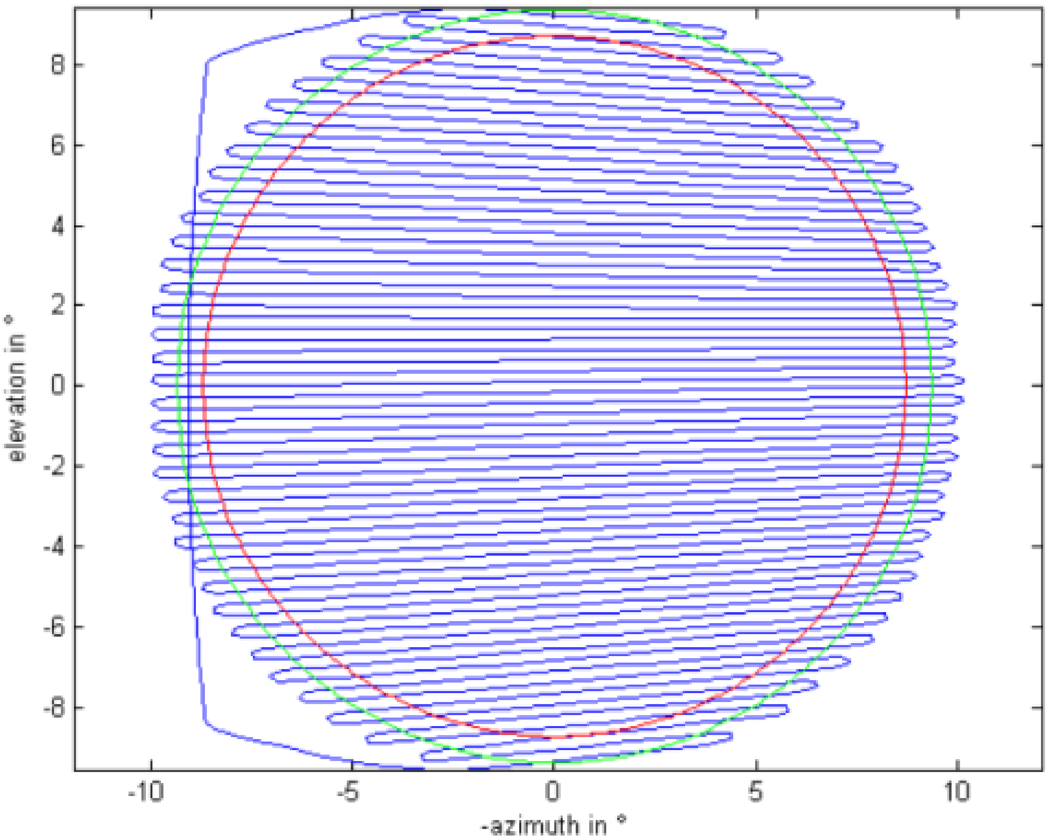


Figure 2 FCI Full Disk Swath Pattern

In nominal operational use, two coverage missions are defined: the full disk coverage (designated in the dataset name as FD) or quarter disk local area coverage (LAC) for Europe (designated as Q4). Each FD or Q4 dataset corresponds to a single FCI repeat cycle.

As noted previously, two imagery missions are defined that are combined in the FCI instrument design capabilities: the Full Disc High Spectral resolution Imagery (FDHSI) mission which has all 16 channels at a 1km SSD for visible and near-infrared channels and 2 km SSD for infrared channels, and the High spatial Resolution Fast Imagery (HRFI) mission which has 4 channels at high-resolution, namely VIS 0.6 and NIR 2.2 at 0.5km SSD and IR 3.8 and IR 10.5 at 1 km SSD.

3.2.4 Focal Plane Arrangement

[Information to be added in a later issue]

3.2.5 On-board Calibration Principle

[Information to be added in a later issue]

3.2.6 Detection Chain

[Information to be added in a later issue]

4 FCI LEVEL 1 PROCESSING ALGORITHMS

4.1 Overview

This Section will describe the core processing steps for genertaing FCI L1C datasets and detail the possible configuraions for these steps.

[Information to be added in a later issue]

[Schematic of overall processing to be added in a later issue]

4.2 Level 0 to Level 1A Processing

[Information to be added in a later issue]

4.3 Level 1A to Level 1B Processing

[Information to be added in a later issue]

4.4 Level 1B to Level 1C Processing

[Information to be added in a later issue]

4.5 Stray-Light Correction

[Information to be added in a later issue]

4.6 INR

[Information to be added in a later issue]

5 CHARACTERISTICS OF THE LEVEL 1C REGISTERED RADIANCE DATASET

5.1 Row and Column Numbering

A row is defined as a line of *spatial samples* or *pixels* running in a (nominal) East to West and West to East direction. The rows are numbered from the south to north starting from 1.

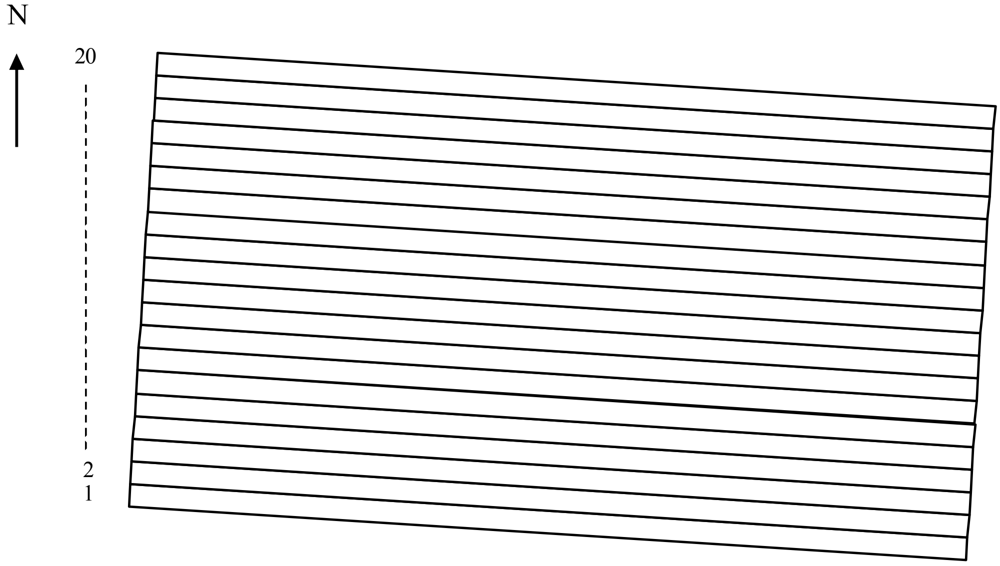


Figure 3: Illustration of row numbering within a **swath**

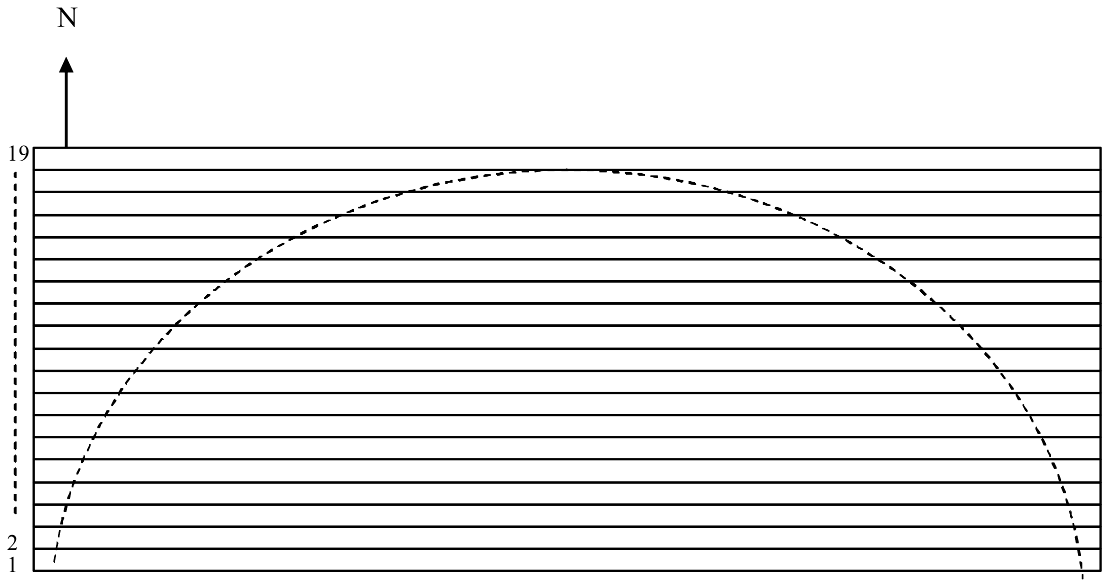


Figure 4: Illustration of row numbering within a L1C rectified **image**

A column is defined a line of *spatial samples* or *pixels* running in a (nominal) South to North direction. The columns are numbered from the west to east starting from 1.

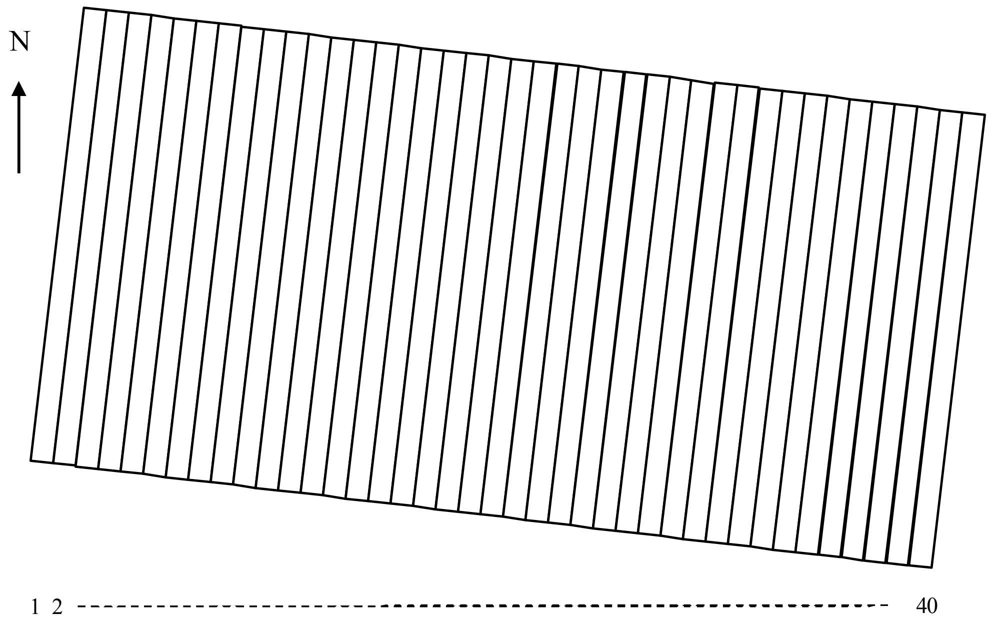


Figure 5: Illustration of column numbering within a **swath**

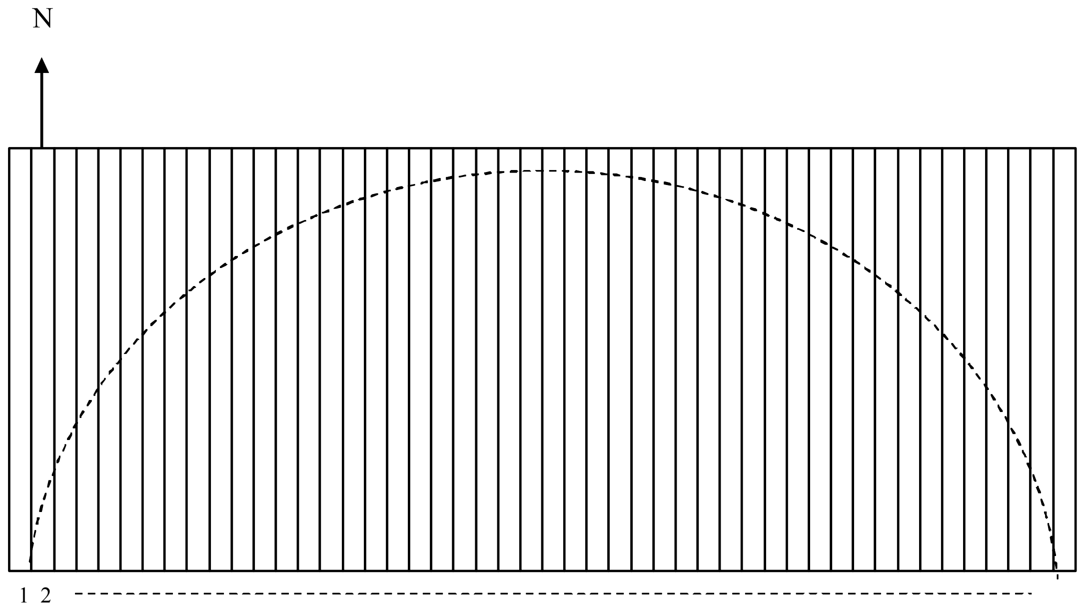


Figure 6: Illustration of column numbering within a rectified **image**

5.2 Level 1C Reference Grid

The reference grid defines the geo-referenced position of the *image pixel centroids* at *level 1c*. The grid steps are equiangular both in satellite azimuth and elevation and equal to the *spatial sampling angle* of the considered channel. The corresponding projected distance at the *sub-satellite point* is the *spatial sampling distance (SSD)*.

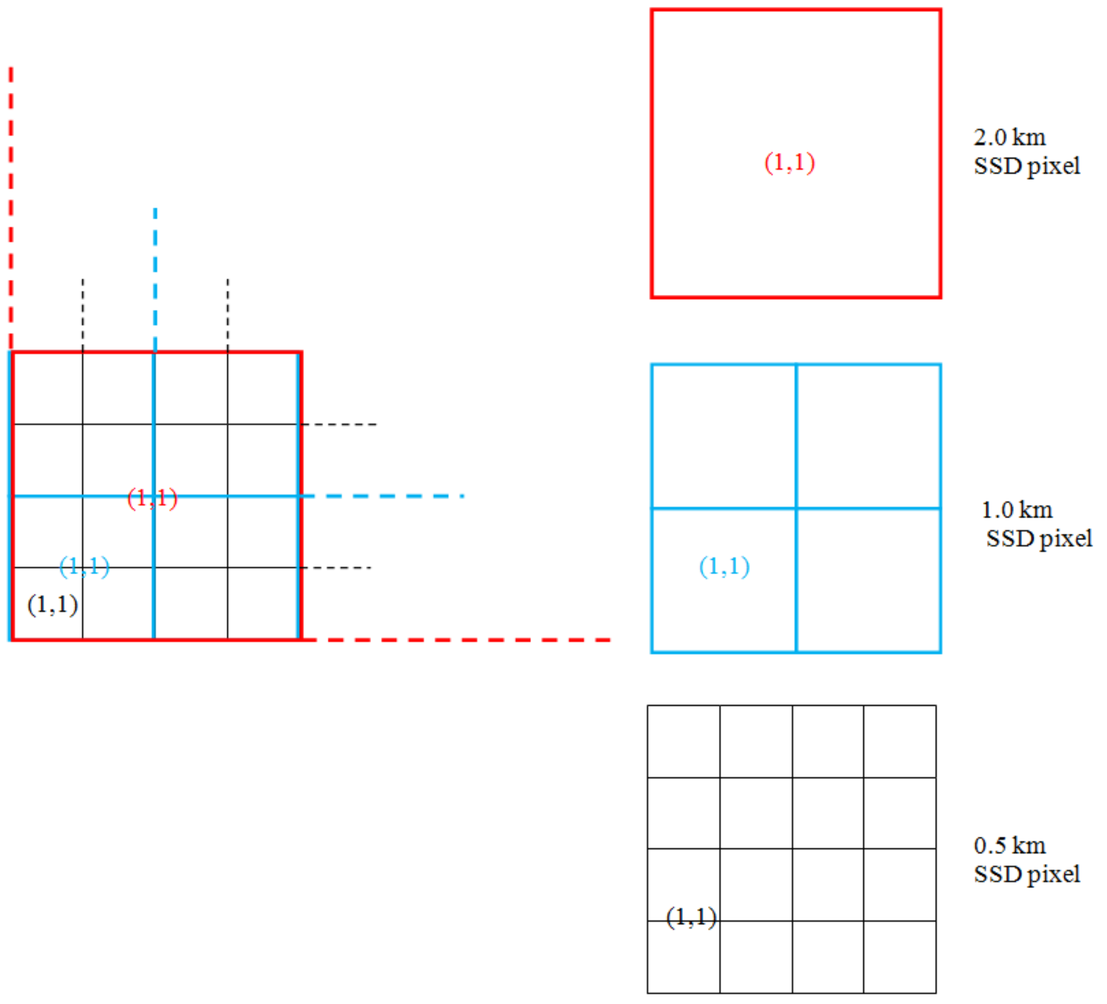


Figure 7: Diagram illustrating the spatial coordination of the three L1C reference grids, starting with the SW corner origin

Figure 7 illustrates how the reference grids for the 3 SSD values are aligned, with the origin pixel at position (1,1) located in the SW corner. Information to generate the FDSS reference grid in the GEOS "Normalized Geostationary Projection" is provided in the dataset, but not the co-ordinates of the points. Other reduced scans (e.g. for RSS) are defined as fixed subsets of the FDSS grid.

The normalized geostationary projection describes the view from a virtual *satellite* to an idealized Earth. The virtual *satellite* is in a geostationary orbit, perfectly located in the Equator plane at the given longitude, λ_D (normally 0 deg) The distance between spacecraft and centre of Earth is given by the *geostationary radius* and the idealized Earth by the *Earth"s reference ellipsoid*. This projection defines the line of sight of each pixel as a vector representing the view from a virtual satellite in geostationary orbit, perfectly located in the Equator plane at the given longitude λ_D. This vector is expressed as a function of two angles called elevation (ϕ_S) and azimuth (λ_S) and defined as follows:

$$\lambda_s = \arctan\left(\frac{r_2}{r_1}\right)$$

$$\phi_s = \arcsin\left(\frac{r_3}{\sqrt{r_1^2 + r_2^2 + r_3^2}}\right)$$

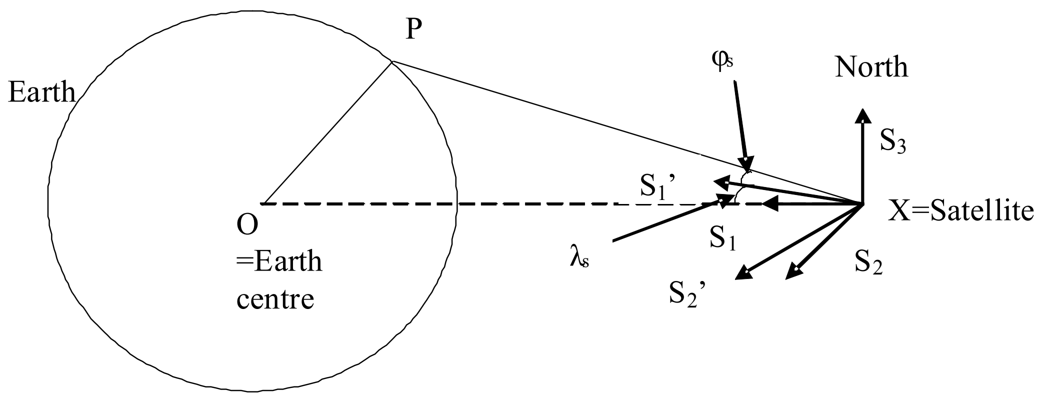


Figure 8: Angular Definition of the Reference Grid

Figure 8 shows the angular definition of the reference grid where:

- the frame (s1,s2,s3) has its origin at the satellite position, (s3) points northwards, and (s1) directs to the centre of the Earth
- the vector r of coordinates (r_1, r_2, r_3) in the frame (s1,s2,s3) is a pixel line of sight vector

In order to geolocate the radiances, the user must first calculate the corresponding azimuth, elevation coordinate for each row and column pixel, and then calculate the corresponding latitude, longitude coordinate from the azimuth, elevation information.

Let (r,c) be the coordinates (row and column) of any pixel of the L1c image. Row and columns are counted increasingly when going from bottom to up (south to north) and left to right (west to east) and beginning at 1. Therefore, the South West corner of a L1c image has coordinates (1,1). For each channel, the correspondence between the row and column position (r, c) and the azimuth and elevation position (λ, φ) is written:

$$\lambda = \lambda_s - (c-1).Azimuth\_Grid\_Sampling$$
$$\varphi = \varphi_s + (r-1).Elevation\_Grid\_Sampling$$

where λ_s and φ_s, depend on the considered channel and Azimuth_grid_sampling and Elevation_grid_sampling are the Azimuth and Elevation sampling angles, respectively.

Note that the E-W scanning angle does not correspond to the standard definition of azimuth, for an observation from the instrument perspective, which runs from negative to positive in a clockwise sense. Instead it runs from negative to positive in an anti-clockwise sense.

The N-S scanning angle corresponds to the standard definition of elevation, for an observation from the instrument perspective.

Channel SSD (in km)	λ_s (degrees)	φ_s (degrees)	Resolution (degrees)	Columns in Full Disk	Rows in Full Disk
0.5	8.914340143	-8.914340143	8.01E-04	22272	22272
1.0	8.913939875	-8.913939875	1.60E-03	11136	11136
2.0	8.913139334	-8.913139334	3.20E-03	5568	5568

Table 3 Values per SSD for the three corresponding reference grids used for FCI L1C

The following definitions are proposed for the L1c LAC products (the row numbers correspond to the full disk row numbering):

LAC type	first row			last row			nb rows in each LAC		
	0,5 km SSD	1 km SSD	2 km SSD	0,5 km SSD	1 km SSD	2 km SSD	0,5 km SSD	1 km SSD	2 km SSD
LAC 1/2 nb 1	0	0	0	11412	5706	2853	11412	5706	2853
LAC 1/2 nb 2	10868	5433	2716	22272	11136	5568	11404	5703	2852
LAC 1/3 nb 1	0	0	0	8398	4199	2099	8398	4199	2099
LAC 1/3 nb 2	7224	3611	1805	14670	7335	3668	7446	3724	1863
LAC 1/3 nb 3	14191	7095	3547	22272	11136	5568	8081	4041	2021
LAC 1/4 nb 1	0	0	0	6850	3425	1713	6850	3425	1713
LAC 1/4 nb 2	5444	2721	1360	11412	5706	2853	5968	2985	1493
LAC 1/4 nb 3	10868	5433	2716	16481	8240	4120	5613	2807	1404
LAC 1/4 nb 4	15714	7856	3928	22272	11136	5568	6558	3280	1640

Table 4 Offset positions and extents of the 4 LAC coverage areas in the 3 full disk reference grids

5.3 Normalized Geostationary Projection

Assuming all trignometric values are in degrees, the transformation from satellite scanning angles (λ_s, φ_s) to geographical coordinates (lon, lat) is given by the inverse projection function:

$$\begin{pmatrix} lon \\ lat \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{s_2}{s_1}\right) + \lambda_D \\ \arctan\left(s_4 \cdot \frac{s_3}{s_{xy}}\right) \end{pmatrix}$$

where:

$$s_1 = h - s_n \cdot \cos(\lambda_s) \cdot \cos(\phi_s)$$

$$s_2 = - s_n \cdot \sin(\lambda_s) \cdot \cos(\phi_s)$$

$$s_3 = s_n \cdot \sin(\phi_s)$$

$$s_4 = \frac{r_{eq}^2}{r_{pol}^2}$$

$$s_5 = \left(h^2 - r_{eq}^2\right)$$

$$s_{xy} = \sqrt{s_1^2 + s_2^2}$$

$$s_n = \frac{h \cdot \cos(\lambda_s) \cdot \cos(\phi_s) - s_d}{\cos^2(\phi_s) + s_4 \cdot \sin^2(\phi_s)}$$

$$s_d = \sqrt{(h \cdot \cos(\lambda_s) \cdot \cos(\phi_s))^2 - (\cos^2(\phi_s) + s_4 \cdot \sin^2(\phi_s)) \cdot s_5}$$

and:

satellite height, h = 42164 km
equatorial Earth radius, r_{eq} = 6378.1690 km
polar Earth radius, r_p = 6356.5838 km

5.4 Spectral Channels

5.5 Repeat Cycle Coverage and Duration

5.6 Timeliness and Availability

5.7 Image Size and Masking

5.8 Radiometric Quality

5.9 Geometric Quality

5.10 Restricted Operations

6 NAMING CONVENTION

All MTG Level 1 products have a WMO-compatible name, following the WMO file naming convention [WMO-386] (cf Attachment II-15 p25 2009 edition)

The filename will consist of the dataset (or product) name with a file_type and a compression field:

(dataset_name) . (file_type) (compression)

Where:

dataset_name is composed of the following fields, separated by underscore symbols, "_":

(pflag)_(productidentifier)_(oflag)_(originator)_(yyyyMMddhhmmss)_(freeformat)

productidentifier is composed of the following fields, separated by commas:

(locationindicator),(datadesignator),(freedescription)

freedescription is composed of the following fields with plus symbol or dash symbol separators:

(spacecraftid)-(data_source)-(processing_level)-(type)-(subtype)-(coverage)-(subsetting)-(component1)-(component2)-(component3)-(purpose)-(format)

freeformat is composed of the following fields, separated by underscore symbols, "_":

(facility_or_tool) _(environment)_(start_time)_(end_time)_
(processing_mode)_(special_compression)_(disposition_mode)_
(accumulation_interval_in_day)

The order of the fields is mandatory.

NOTE: If there is no relevant value within the freeformat section, the field is left out. This can lead to the allowable repetition of underscores.

The following table shows the fully expanded set of name fields in the correct order, with values described for FCI L1C datasets. Following the main table, subsequent subsections describes the allowed values for the selected fields in greater detail.

Name Field	Description	FCI-1C-RAD Values
pflag	WMO mandated	"W"
locationindicator	WMO mandated	"XX-EUMETSAT-Darmstadt"
datadesignator	The type of data with respect to the categories and subcategories defined in [|WMO-386],	"IMG+SAT"

Name Field	Description	FCI-1C-RAD Values
spacecraftid	Spacecraft indicator	“MTIn” for MTG Imager n where n = 1, 2, 3 or 4
data_source	Instrument, platform or SAF	“FCI”
processing_level	Processing Level	“1C”
type	Indentifies the type of data	“RRAD” for rectified radiances
subtype	Identifies a sub-type for the type.	“x”
coverage	Coverage of the full accumulation interval	“FD” for full disk, “Q4” for LAC4
subsetting	Identification of the type of subsetting performed	“x”
component1	Identifies a first level component of the product	“CHK” for chunk “QUICK” for a quick-look file
component2	Identifies a second level component of the product	“BODY” for a body chunk “TRAIL” for a trailer chunk “IMAGE” for an image quick-look
component3	Identifies a third level component of the product	“x” – for No Value FCI Channel Reference – reference to the appropriate FIC channel for a quick-look e.g. VIS06 for Visible 0.6 microns or “RGB01” for RGB-01
purpose	The intended purpose of the dataset. This normally refers to the intended final recipient.	“ARC” for an archival dataset “DIS” for a dissemination dataset
format	The intended encoding format of the dataset.	“NC4E” for netCDF-4 enhanced model “PNG” for a quick-look PNG image
oflag	WMO mandated	“C”
originator	WMO mandated	“EUMT”
yyyyMMddhhmmss	Is the UTC time of the processing, defined as the time of the formatting of the dataset/product by the processor [TBC-EUMETSAT], formatted in Abbreviated Generalised Time format e.g. yyyy = year MM = month dd = day of month hh = hour of day mm = minute of hour ss = second of minute	
facility_or_tool	Facility or tool producing the dataset	“IDPFI” = Instrument Data Processing Facility for MTG-I “GTT” = Generic Test Tool
environment	Ground Segment Environment producing the dataset	“OPE” - Operational
start_time	UTC Time of start of Sensing Data formatted in Abbreviated Generalised Time format (see above).	For the body chunk, this will be the time of the first measurement in the chunk. For a trailer chunk or a quick-look, this is the start time of the first body chunk in the repeat cycle.

Name Field	Description	FCI-1C-RAD Values
end_time	UTC Time of end of Sensing Data formatted in Abbreviated Generalised Time format (see above).	For the body chunk, this will be the time of the last measurement in the chunk. For a trailer chunk or a quick-look, this is the end time of the last body chunk in the repeat cycle.
processing_mode	Identification of the mode of processing	"N" = nominal
special_compression	This field provides identification of a special compression technique that has been applied to one or more variables in the dataset. Special compression does not include the standard netCDF data compression or "deflation" using in-built zlib support which is transparent to the user.	"JLS" = JPEG-LS. Lossless JPEG compression has been applied internally. blank – no special compression
disposition_mode	Shows disposition of the dataset from the perspective of an end-user"s needs.	"O" = operational "T" = testing
accumulation_interval_in_day	4-digit number (right-justified, zero-filled) indicating the current group accumulation interval in the day for this particular dataset. The counter starts at 0001 for the first group accumulation interval at or after midnight and resets for the next group accumulation interval at or after the following midnight. The group accumulation interval for FCI is equivalent to the repeat cycle of the instrument and is used by the archive to associate all chunks of a repeat cycle dataset.	Variable
file_type	Indicator of the encoding format of the data, according to WMO conventions.	".nc" – netCDF ".png" – PNG image
compression	Indicator of compression applied to the dataset as a whole according to WMO conventions (as opposed to the internal compression of variable indicated by the "special_compression" name field).	None

Table 5 Breakdown of the fields in the FCI L1C dataset naming convention

7 STRUCTURE AND PRESENTATION OF THE LEVEL 1C REGISTERED RADIANCE DATASETS

7.1 Overview

An FCI Level 1c rectified radiance dataset consists of a set of files that contain the level 1c science data rectified to a reference grid together with the auxiliary data associated with the processing configuration and the quality assessment of the dataset.

7.2 Coverage Mission and Imagery Mission Datasets

A number of FCI L1C rectified radiance (FCI-1C-RRAD) datasets are available based on coverage and mission.

All datasets will have the same format specification as below.

7.3 Format

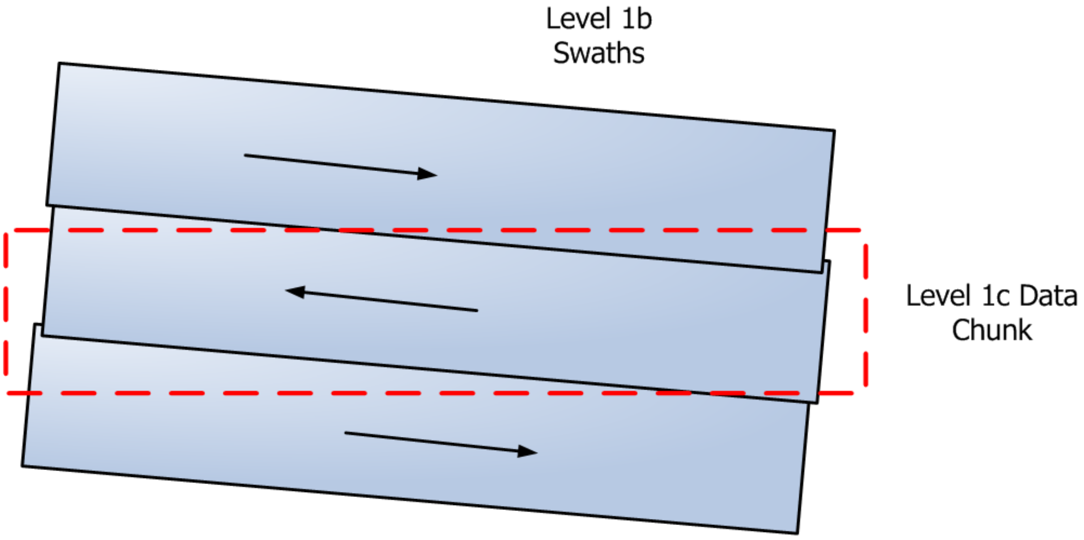
7.3.1 Data Chunks

An FCI-1C-RRAD dataset covers the full repeat cycle and is divided into a series individual files or “chunks” for timely dissemination. These same chunks are sent to the Archive for storage and can be retrieved in this form. The main bulk of the dataset are a series of body chunks that contain the observational data for the repeat cycle. There is also a trailer chunk that contains information applicable to or derived from the complete repeat cycle.

The division of the dataset in this way provides benefits for timely and efficient transfer rates for near real-time dissemination. It also provides a rapid method for retrieving geographically subsetted data from the archive by returning only those chunks that intersect the region of interest.

The body chunks correspond approximately to the size of a FCI swath. Each body chunk will contain about the same number of rows from the reference grid, but the time duration will vary from 4 to 10 seconds in line with the varying duration of the swaths. This will produce 70 body chunks for a full disk repeat cycle and 20 body chunks for a LAC4 repeat cycle.

Note: Level 1b swaths appear tilted when projected onto the reference grid due to the fan shaped scan pattern and may contribute to a number of level 1c body chunks.



7.4 FCI L1C Registered Radiance (FCI-1C-RRAD) Dataset

The FCI Level 1c rectified radiance dataset contains the level 1C science data together with the auxiliary data associated with the processing configuration and the quality assessment of the dataset.

The dataset is represented by different format IDs that can be found as a string in the filename:

FCI-1C-RRAD-FDHSI-CHK-BODY-NC4E
FCI-1C-RRAD-HRFI-CHK-BODY-NC4E
The Level 1c full repeat cycle science data, for the FDHSI or HRFI spectral channels, is divided into a number of L1c body data chunks for dissemination and storage in the data archive. The division of the dataset in this way provides benefits for timely and efficient transfer rates to other environments and for geographically subset retrieval from the archive.

FCI-1C-RRAD-FDHSI-CHK-TRAIL-NC4E
FCI-1C-RRAD-HRFI-CHK-TRAIL-NC4E
The Level 1c trailer, for the FDHSI or HRFI spectral channels, is used to contain information that is calculated at the end of the repeat cycle, e.g. repeat cycle quality metrics, and information that help in the interpretation of the data but would present too large an overhead if transmitted for every L1c Body data chunk, e.g. radiometric noise estimates.

FCI-1C-RRAD-FDHSI-QCK-IMAGE- PNG
FCI-1C-RRAD-HRFI-QCK-IMAGE- PNG
The L1c Quick-look files contain compressed, subsampled images for selected repeat cycle channels that are used for coarse visualisation of the science data, for example for archive

browsing. There may also be RGB images created from a selection of three of the available channels.

7.5 FCI-1C-RRAD Body Chunk

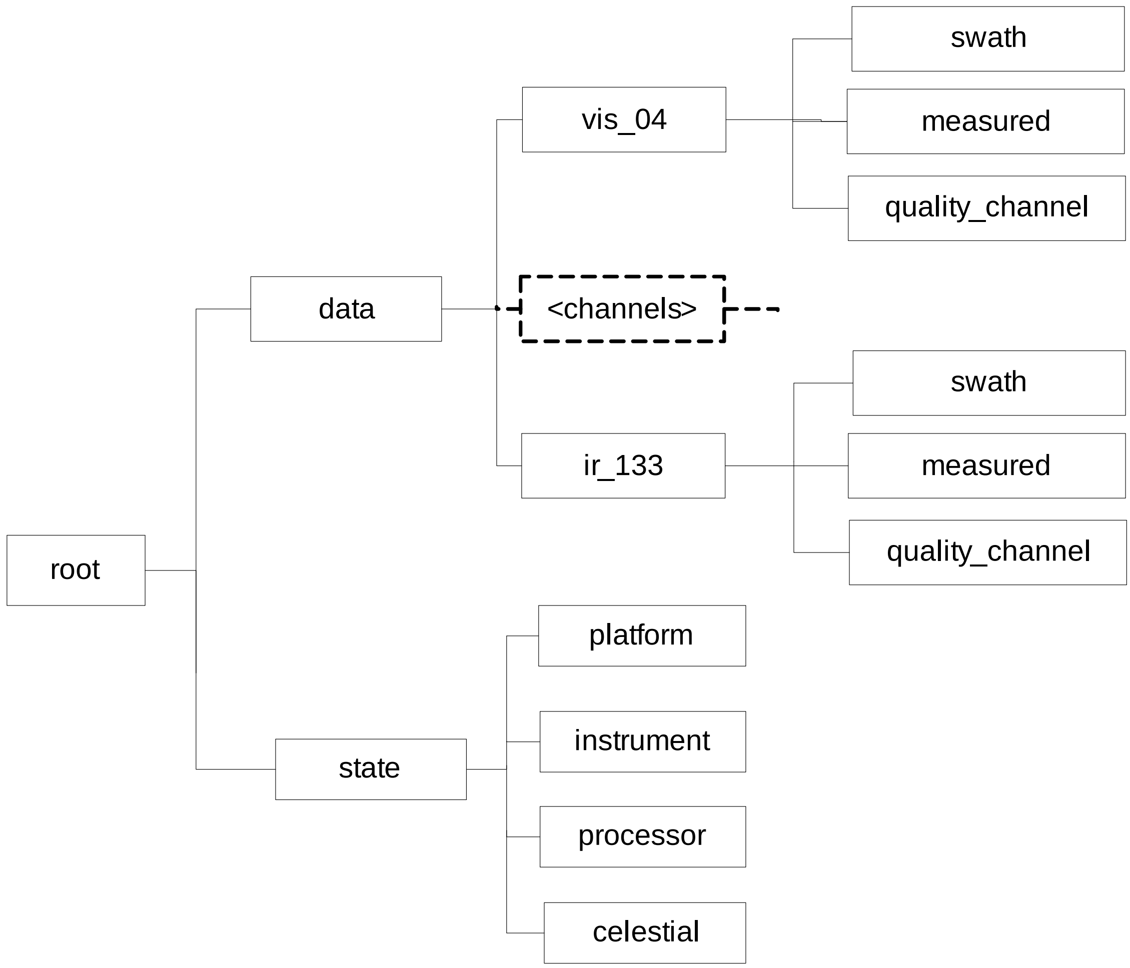


Figure 9 Overview of netCDF groups in the FCI 1C body chunk file for an FDHSI dataset

7.6 Group Overview

Group			Description
Generic Type		netCDF Name	
		root	Root level metadata
		data	Information common to all channels
"channel" groups	FDHSI	vis_04 vis_05 vis_06 vis_08 vis_09 nir_13 nir_16 nir_22 ir_38 wv_63 wv_73 ir_87 ir_97 ir_105 ir_123 ir_133	All "channel" groups share a common generic format and contain information specific to that channel. FDHSI channel groups are found in the FDHSI dataset.
	HRFI	vis_06_hr nir_22_hr ir_38_hr ir_105_hr	HRFI channel groups are found in the HRFI dataset.
		swath	Swath information
		measured	Measured radiances
		quality_channel	Associated quality information specific to a channel
		state	State information
		platform	Satellite state information
		instrument	Instrument state information
		processor	Processor state information
		celestial	Celestial state information

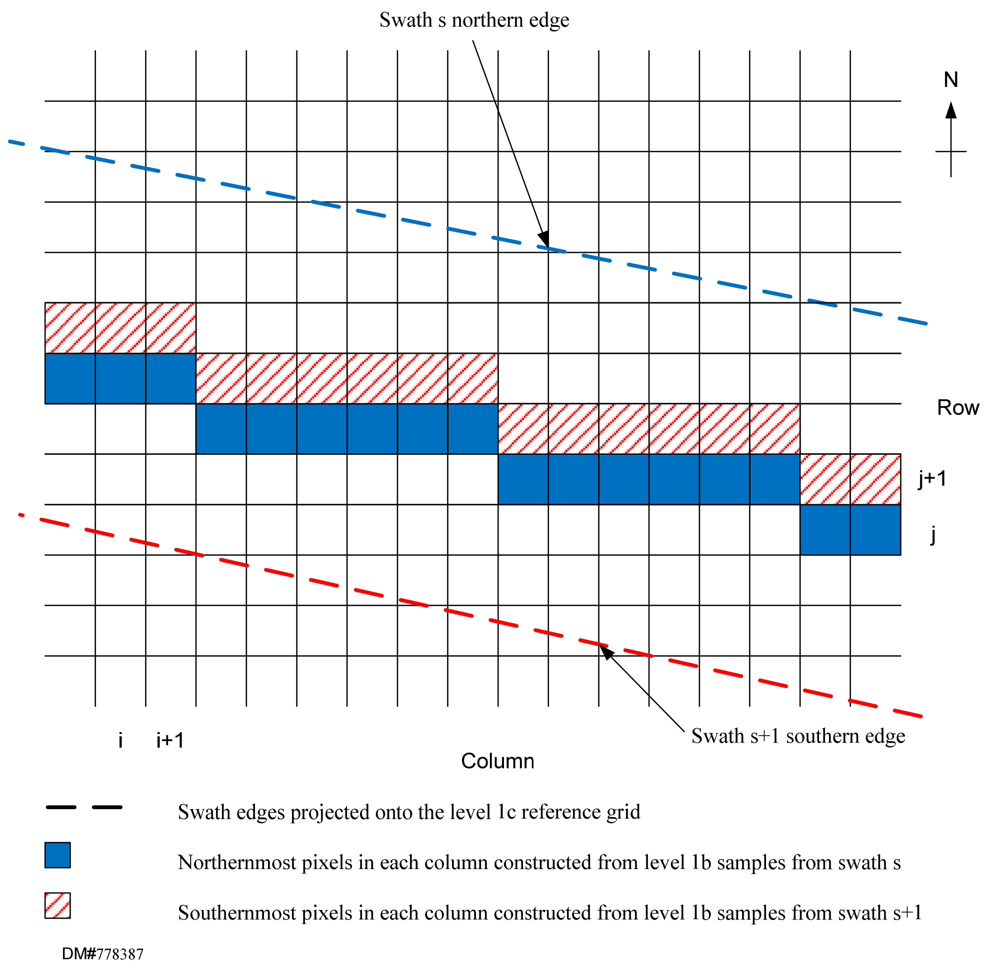
Table 6 Description of the groups in an FCI L1C body chunk

7.7 Channel Subsetting

Channel subsetting may be achieved by selecting which channel groups are delivered in the dataset. As each channel group contains only information specific to that channel, they may be removed from the dataset without affecting its integrity.

7.8 Swath Information

The variable swath_boundary indicates which swath has contributed to a given pixel by recording the northernmost row per column of the last pixel to have been created from a particular swath. The column number takes the valid_range of 1 to number_of_columns. In the example in Figure 10 the northernmost pixels in the level 1c data constructed from samples from swaths are indicated by solid blue boxes, thus the row indexing associated with the swath_boundary is as given in Table 7.



Note: The swath edge can take a positive or negative slope when projected in the level 1c grid

Figure 10 Swath boundary appearing in the level 1c grid

Column	Row
i	j+3
i+1	j+3
i+2	j+2
i+3	j+2
...	...
i+7	j+2
i+8	j+1
...	...
i+13	j+1
i+14	j
...	...

Table 7 swath_boundary for example in Figure 10

Once the swath boundary has been identified the remaining information necessary, to identify the acquisition time of the pixels, is the timing to associate with the columns of data within the level 1c grid for that particular swath. The columns and rows for which the first and last valid earth measurements are generated (swath_earth_start_column/row and swath_earth_end_column/row) are identified together with the mean time for their acquisition (swath_earth_start_time and swath_earth_end_time), see Figure 11. Where the mean time is taken as the mean acquisition time of the level 1b samples used to create the pixel. In this instance the time difference, due to the inclination of the swath, between points from each end of a column is taken to be small (~1ms maximum). Additionally the direction of swath acquisition is also recorded (swath_direction). The timing for pixels in between these columns can be derived by linear interpolation between the columns or by using the swath earth start/end pixel positions and the swath_inclination to calculate the timing to sub millisecond accuracy.

Finally information related to the position of the satellite and sun for the swath is provided, together with the swath dependent quality information

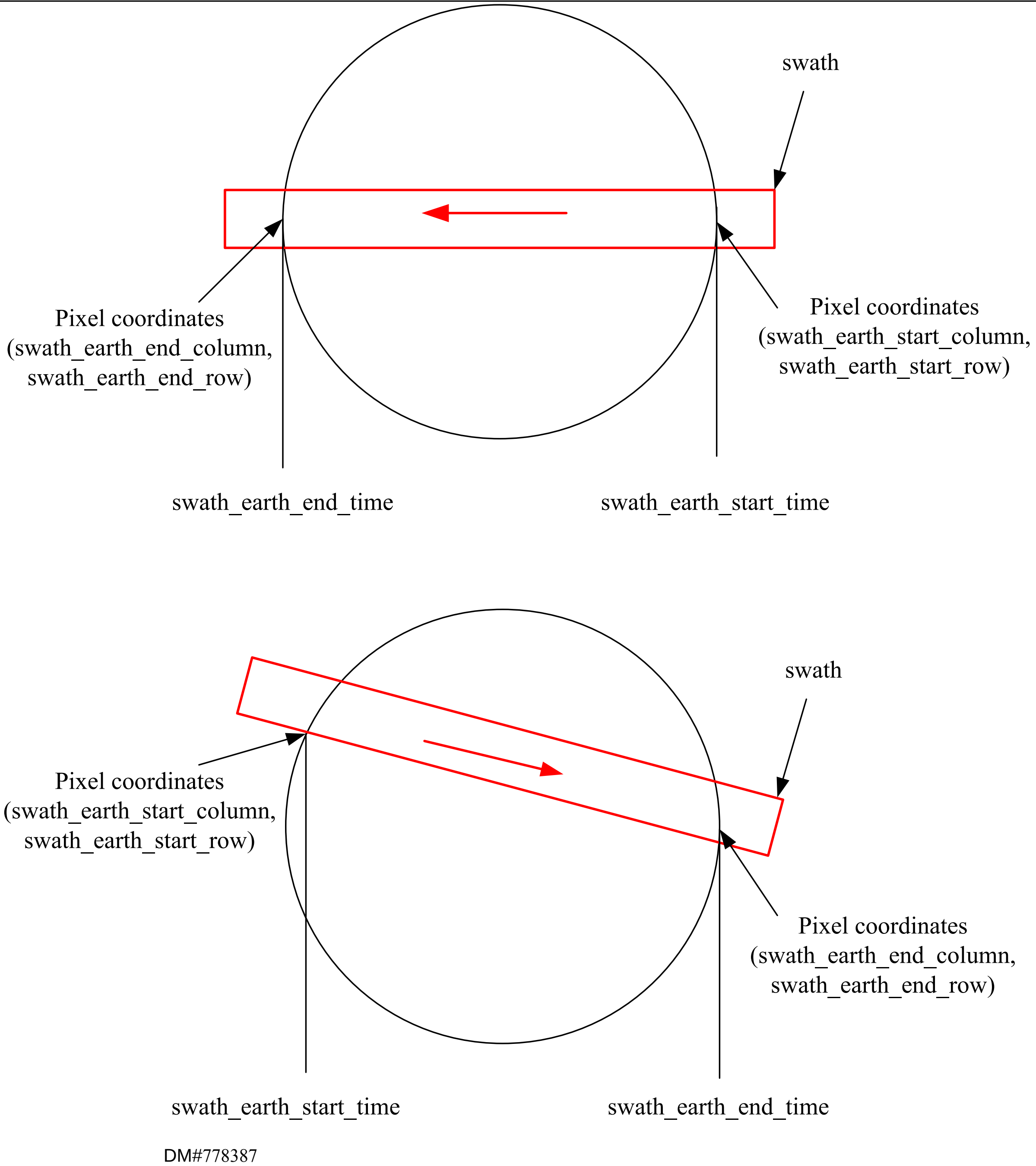


Figure 11 Illustrations of swath times and coordinates stored in the swath group

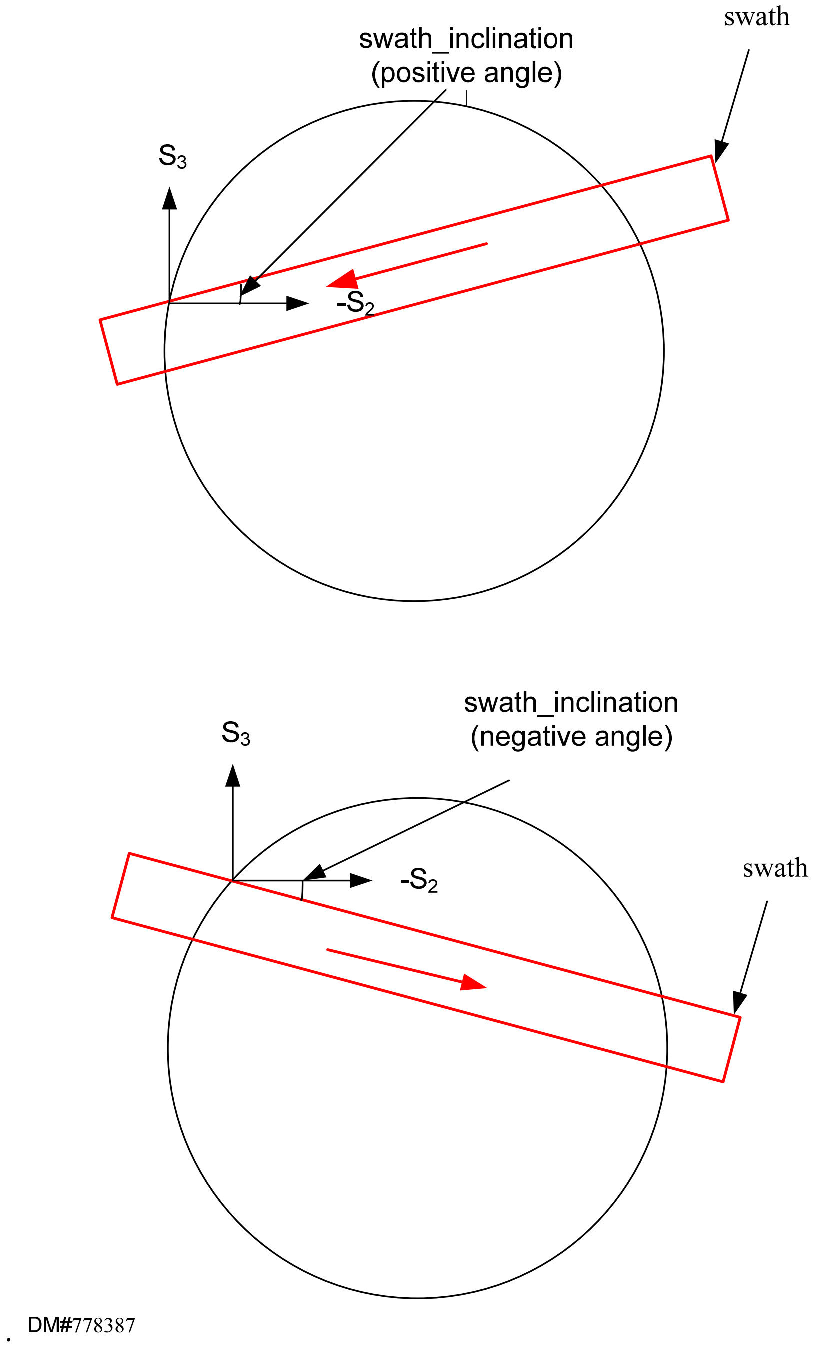


Figure 12 Illustrations of swath inclination with respect to the Normalized Geostationary Projection axes

7.8.1 Radiance Encoding

The 12 (resp. 13 for IR3.8) bits of the netCDF 16-bit integer are used to encode and compress the effective radiance. Attributes scale_factor and add_offset are used to rescale the effective radiance code to $mWm^{-2}sr^{-1}(cm^{-1})^{-1}$.

For IR3.8 additional attributes valid_cold_range, warm_scale_factor and warm_add_offset are used to encode and compress the values above 2^{12} (4096) to cover the extended radiometric range.

_FillValue will be used for data that cannot be produced due to missing level 0 data

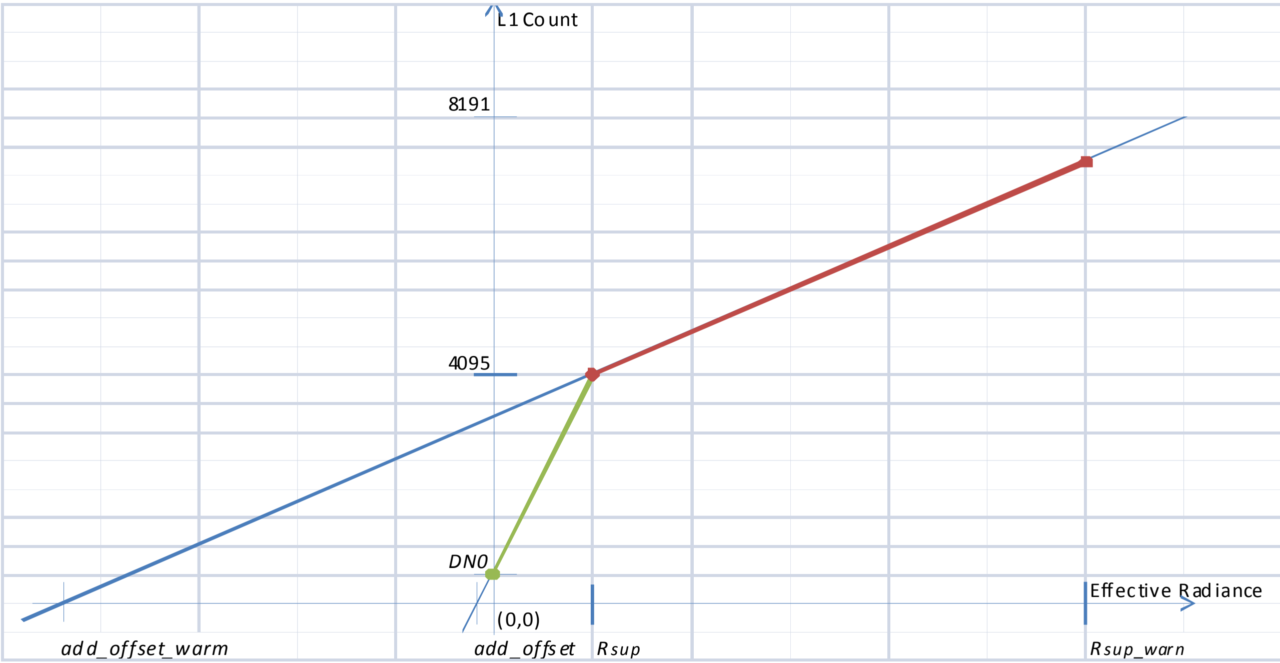


Figure 13 Illustration to show encoding of the combined IR 3.8m channel with offsets and scale factors for the "cold" (green) and "warm" (red) measurements

7.8.2 Pixel Quality

An 8 bit pixel_quality variable, associated with each effective_radiance, is provided as given in the table below.

The usage of the missing_warning flag depends on whether interpolation is applied over missing data (currently not baseline).

Bit	***Name***	***Interpretation***
0	noise_warning	Pixel may be noisy (have a non-nominal noise level) due to a contribution from noisy samples following rectification
1	radiometric_warning	Pixel may have radiometric errors due to a contribution from samples with radiometric errors following rectification. Radiometric

		errors in this sense arise from calibration activities occurring during the repeat cycle that do not impact the calibration of the complete repeat cycle.
2	saturation_warning	Pixel has a contribution from saturated samples following rectification.
3	missing_warning (TBC)	Pixel has a contribution from missing samples following rectification.
4	straylight_warning	Pixel has a contribution from samples with solar stray light contamination (above a set threshold). This may have been corrected depending on the IDPF configuration.
5	straylight_correction_warning	Pixel has a contribution from samples corrected for solar stray light contamination
6	extended_dynamic_range_warning	For the IR3.8 channel only: Pixel has a contribution from samples selected from the FAIR3.8 detector measurements
7	encoding_saturation_warning	Pixel is saturated from the process of encoding into 12-bits (13-bits for FAIR3.8).

Table 8 Description of the quality flags in the pixel_quality variable

7.9 Special Compression

In order to achieve greater compression than allowed by the default netCDF zipping algorithms, disemminated L1C datasets will use lossless Jpeg compression implemented at the HDF layer. Once the releveant decompression module is installed at the user side, decompression will be transparent to the user.

The development of this compression functiuonality is ongoing at the time of this first issue of this document. Further information will be supplied in a future release.

7.10 FCI-1C-RRAD Trailer Chunk

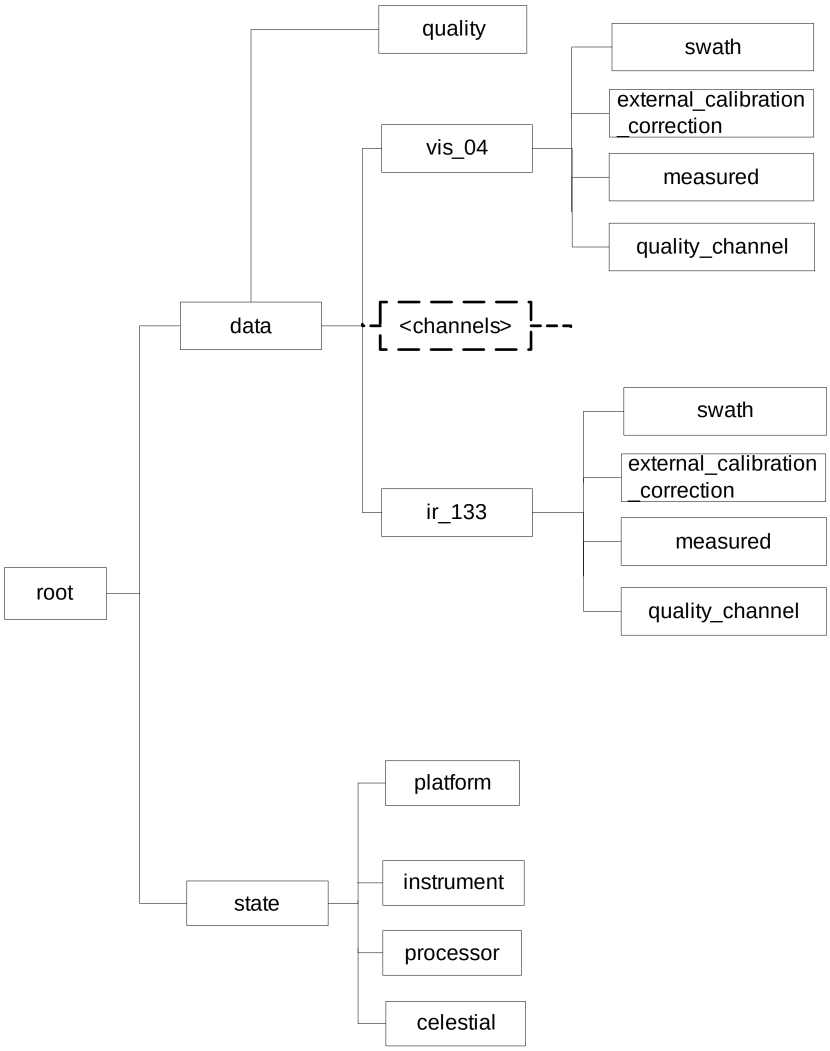


Figure 14 Overview of netCDF groups in the FCI 1C trailer chunk file for an FDHSI dataset

Group			Description
Generic Type		netCDF Name	
		root	Root level metadata
		data	Information common to all channels
"channel" groups	FDHSI	vis_04 vis_05 vis_06 vis_08 vis_09	All "channel" groups share a common generic format and contain information specific to that channel.

		nir_13 nir_16 nir_22 ir_38 wv_63 wv_73 ir_87 ir_97 ir_105 ir_123 ir_133	FDHSI channel groups are found in the FDHSI dataset.
	HRFI	vis_06_hr nir_22_hr ir_38_hr ir_105_hr	HRFI channel groups are found in the HRFI dataset.
		swath	Swath information for the repeat cycle
		measured	Metadata about the measured radiances
		quality	Associated repeat cycle quality information common to all channels
		quality_channel	Associated repeat cycle quality information specific to a channel
		state	State information
		platform	Satellite state information
		instrument	Instrument state information
		processor	Processor state information
		celestial	Celestial state information

Table 9 Description of the groups in an FCI L1C trailer chunk

7.11 FCI-1C-RRAD Quick-Look Image

The FCI-1C-RRAD quick-look images are in PNG format and will be sub-sampled from the full refefence grid, resulting in images that are nominally 500 x 500 pixels in size. The quick-looks are intended for use by the data archive to facilitate visual browsing of the FCC-1C-RRAD datasets. The size and selection of quick-looks is configurable, but it is posisble to produce a quick-look for each FCI spectral channel, as well as RGB images based on a selection of 3 suitable FCI channels. The final set of quick-looks will be chosen to provide the optimal information presentation when browing the data arhcive.

8 FCI L1 DATASET USAGE

8.1 Reconstructing Reference Grids

Pixel-related data (radiances and pixel quality flags) do not have associated geolocation coordinate variables (in order to reduce the size of the product). However, the geolocation information may be calcualted and associated to the varibales using the equations given in Section 5.2.

8.2 Unpacking Coded Radiances

Radiances are stored in a compressed form as integer values with associated offsets and scale factors as per the standard see CF conventions [CF]. However, the extended 3.8 channel has additional set of offset and scale factors that have been used to compress the data into 13 bits. These need to be unpacked as per the explanation in Section 7.8.1.

8.3 Effective Radiance Unit Conversion

Radiances are stored in the FCI L1C dataset ("effective_radiance") with units of mW.m-2.sr-1.(cm-1)-1.

The variable "radiance_unit_conversion_coefficient" contains the coefficient that can be used to convert effective radiance units from per cm^-1 to per wavelength in microns.

8.4 Converting to Effective Radiance to Brightness Temperature for IR Channels

The relationship between the band-average spectral radiance per wavenumber $\overline{L_\nu}$ and the effective brightness temperature T_{eff} can be analytically approximated as:

$$B_\sigma\left(\nu_c, a \cdot T_{eff} + b\right) \approx \overline{L_\nu}$$

Where ν_c is a representative wavenumber.

So the effective brightness temperature T_{eff} can be computed as follows:

$$T_{eff} = \frac{c_2 \cdot \nu_c}{a \cdot \ln\left(1 + \frac{c_1 \cdot \nu_c^{\,3}}{\overline{L_\nu}}\right)} - \frac{b}{a}$$

The set of coefficients $\{\nu_c, a, b\}$, corresponding to a given spectral response function, are found by regression over the required range of temperatures. Constants c_1 = 2hc2 and c_2 = hc/k are radiation constants where c, h, and k are the speed of light, Planck, and Boltzmann constant respectively

The variable "central_wavelength_actual" contains the wavelength corresponding to the representative wavenumber, ν_c.

The variable "radiance_to_bt_conversion_coefficients" contains the conversion coefficients a and b for IR channels. It is sized to zero for visible channels and set to the _FillValue.

The variable "radiance_to_bt_conversion_constants" contains the constants c_1 and c_2 for IR channels. It is sized to zero for visible channels and set to the _FillValue..

8.5 Converting to Effective Radiance to Reflectance for VNIR Channels

The variable "channel_effective_solar_irradiance" contains the channel effective solar irradiance at 1 AU to be used in the derivation of the reflectance for VNIR spectral channels. The variable is set to _FillValue for IR spectral channels. netCDF Formats.

8.6 Solar zenith Angle Calculation

[Information to be added in a later issue]

8.7 Radiometric Noise Assessment

[Information to be added in a later issue]

8.8 Radiometric Accuracy Assessment

[Information to be added in a later issue]

8.9 Recombining Chunks

As noted in Section 7, each FCI Level 1 repeat cycle dataset (either FDHSI or HRFI) is distributed as a set of multiple netCDF files referred to as chunks. There are 2 types of chunks: "body" and "trailer" . Typically, a FDHSI product will consist of 70 body chunks, and a HRFI product about 20 body chunks. Both products have final trailer chunk contating repeat cycle-based information.

The start_position_column, start_position_row, end_position_column and end_position_row variables may be used to locate the pixel-based data (radiances and pixel-quality flags) in each chunk with the correct position in the Level 1C reference grid (see Section 5.2 and Figure 15).

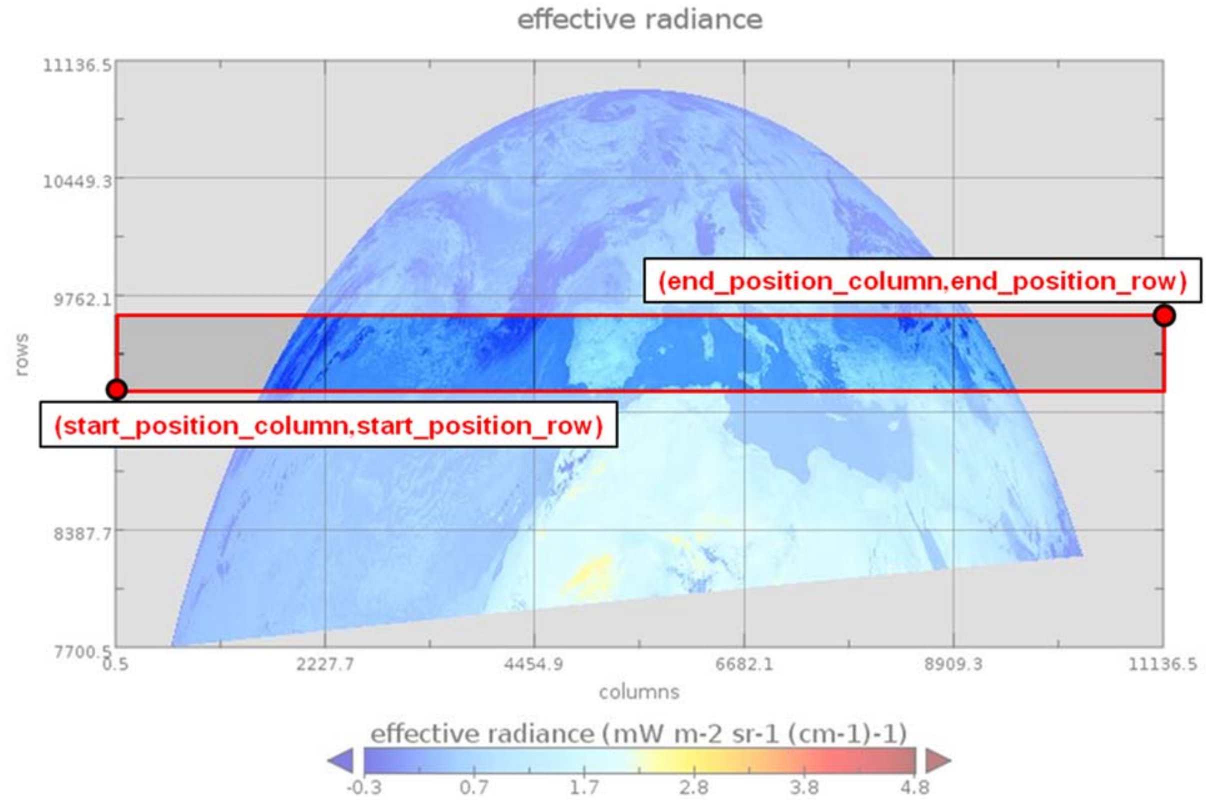


Figure 15 Illustration of the location of a typical body chunk within a LAC 4 repeat cycle dataset

In addition, each of pixel-based variables are linked to row and column 2D coordinate variables (as per the CF conventions [CF]) that contain the position of the pixel in the reference grid. These coordinate variables can also be used to locate the chunk within the reference grid and should allow CF-aware tools to combine the chunks into a complete repeat cycle image. However, at the time of this issue, this functionality appears to be available only for geolocated datasets.

The user currently has three paths to recombine the chunks into a complete repeat cycle image for each channel:

1. Create arrays based on the correct-sized reference grid for each channel and copy the pixel data into the correct area of the grid based upoin either the associated corner coordinate varaibles or the linked 2D coordinate varaible in the coordinate attribute.
2. Extend the method of option (1) by also geolocating the reference grid by clauclating the relevant latitude and logitude variables and associating them as 2D coordinate variables with the pixel data. This may require the creation of a new netCDF file on disk or, if supported by the netCDF libraries, as a netCDF object in memory.
3. Calculating the geolocation information of the reference grid (as per option (2)) but writing out each chunk as a netCDF file with geolocated pixel information and then

usong a CF-aware tool such as Panoply (see Appendix B) to read the and display the set of chunks.

APPENDIX A FORMAT DESCRIPTIONS

Format descriptions in this section are currently given in CDL. In future, these may be replaced or supplemented by nCML descriptions.

A.1 Body Chunk

A.1.1 CDL Description

```
group: root{ // Generic

        :Conventions = "";  // To be determined
        :title  = product_name;
        :summary = ""; //see relevant format specification document
        :keywords = ""; //see relevant format specification document
        :history = "original generated file";
        :institution = "EUMETSAT"; // For datasets generated at EUMETSAT
        :location_indicator = "";//
        :data_designator  = "";
        :spacecraft = "";
        :product_id = "";
        :processing_level = "";
        :baseline_version = "";
        :release_version = "";
        :processor_version = "";
        :algorithm_version = "";
        :format_version = "";
        :sensing_start = "YYYYMMDDhhmmss";
        :end_time = "YYYYMMDDhhmmss";
        :processing_mode  = "";
        :special_compression = "";
        :subsetting  = "";
        :disposition_mode  = "";
        :source = "";
        :facility_or_tool = "";
        :environment = "";
        :references = "";
```

```
        :comment = "";
        :processing_time = "";
        :group_tag = "";
        :accumulation_interval_in_day = 1;
        :count_in_accumulation_interval = 1;
        :instrument_configuration_id = 0;
        :instrument_configuration_id_version = 0;
        :subsettable_groups = "";
        :subsettable_groups_present = "";
        :mtg_name="";
        :alternative_name="";
        :purpose="";
        :format="";
        :geospatial_lat_min = 0;
        :geospatial_lat_max = 0;
        :geospatial_lon_min = 0;
        :geospatial_lon_max = 0;

        types:
                byte enum boolean {false = 0, true =1}
                byte enum trilean {false = 0, true =1, undefined =2}
          ubyte enum reference_grid_type (500m = 0, 1km = 1, 2km = 2);
    ubyte enum swath_direction_type(EastWest = 0, WestEast = 1);

        dimensions:
                number_of_l0_channels = runtime_value or configured_value;
                number_of_llc_channels = runtime_value or configured_value;
                number_of_reference_grids  = 2;

        variables:
                string l1c_channels_present(number_of_l1c_channels);

                channels_present:long_name = "Level 1c spectral channels present in dataset"

        boolean timeliness_non_nominal;
                        timeliness_non_nominal:long_name = "Timeliness non-nominal warning flag";

        group: data{

                group: vis_06{   // Only one example channel group shown
                        : long_name="FCI FDHSI Visible 0.6 micron channel";
                        : subsettable = "yes";
```

```
dimensions:
        number_of_rows = configured_value;
        number_of_columns = configured_value;

variables:
        string channel_srf_identifier;
                channel_srf_identifier:long_name="Channel Spectral Response Function identifier";

        string channel_mtf_identifier;
                channel_mtf_identifier:long_name="Channel Modulation Transfer Function identifier";

        ushort channel_srf_version;
                channel_srf_file:long_name="Channel Spectral Response Function identifier version";

        ushort channel_mtf_version;
                channel_mtf_file:long_name="Channel Modulation Transfer Function identifier version";

        float central_wavelength_specified;
                central_wavelength_specified:long_name="Specified central wavelength of channel";
                central_wavelength_specified:units="micrometres";

        float spectral_width_specified;
                spectral_width_specified:long_name="Specified spectral width of channel";
                spectral_width_specified:units=" micrometres";

        float central_wavelength_actual;
                central_wavelength_actual:long_name="Actual central wavelength of channel";
                central_wavelength_actual:units="micrometres";

        float spectral_width_actual;
                spectral_width_actual:long_name="Actual FWHM spectral width of channel";
                spectral_width_actual:units=" micrometres";

        reference_grid_type reference_grid;
                reference_grid:long_name="Reference grid used for this channel";

        group:measured{

          dimensions:
                    number_of_radiance_to_bt_conversion_coefficients = configured_value;  // default = 3 for IR channels, 0 for VNIR channels;
                    number_of_radiance_to_bt_conversion_constants = configured_value;  // default = 2 for IR channels, 0 for VNIR channels;

          variables:
```

```
ushort start_position_row;
        start_position_row:long_name= "Row index of the pixel closest to the origin of the reference grid";

ushort start_position_column;
        start_position_column:long_name= "Column index of the pixel closest to the origin of the reference grid";

ushort end_position_row;
        end_position_row:long_name= "Row index of the pixel furthest from the origin of the reference grid";

ushort end_position_column;
        end_position_column:long_name= "Column index of the pixel furthest from the origin of the reference grid";

ushort effective_radiance(number_of_rows, number_of_columns);
        effective_radiance:long_name = "Effective radiance";
        effective_radiance:standard_name = "effective_radiance_in_wavenumber"
        effective_radiance:units = "mW.m-2.sr-1.(cm-1)-1";
        effective_radiance:_FillValue = NC_FILL_SHORT;
        effective_radiance:valid_range = 0, 4095;  // 0,8191 for IR3.8
        effective_radiance:valid_cold_range = 0, 4095;
        effective_radiance:scale_factor = runtime_value;
        effective_radiance:add_offset = runtime_value;
        effective_radiance:warm_scale_factor = runtime_value;
        effective_radiance:warm_add_offset = runtime_value;
        effective_radiance:ancillary_variables = "pixel_quality";

ubyte pixel_quality(number_of_rows, number_of_columns);
        pixel_quality:long_name = "Pixel quality flags";
        pixel_quality:standard_name = "effective_radiance_in_wavenumber status_flag";
        pixel_quality:valid_range = 0b, 255b;
        pixel_quality:flag_masks = 1b, 2b, 4b, 8b, 16b, 32b, 64b, 128b;
        pixel_quality:flag_meanings = "noise_warning
                                        radiometric_warning
                                        saturation_warning
                                        missing_warning
                                        straylight_warning
                                        straylight_correction_warning
                                        extended_dynamic_range_warning
                                        encoding_saturation_warning";

float radiance_unit_conversion_coefficient;
        radiance_unit_conversion_coefficient:long_name = "Coefficient used to convert effective radiance units from per cm^-1 to per wavelength in micron";
        radiance_unit_conversion_coefficient:unit = "(cm-1).micron"
```

```
                    float radiance_to_bt_conversion_coefficients(number_of_radiance_to_bt_conversion_coefficients)
                        radiance_to_bt_conversion_coefficients:longname = "Radiance to brightness temperature conversion coefficients";
                        radiance_to_bt_conversion_coefficients:_FillValue = NC_FILL_FLOAT;

                    float radiance_to_bt_conversion_constants(number_of_radiance_to_bt_conversion_constants)
                        radiance_to_bt_conversion_constants:longname = "Constants used to convert effective radiance to brightness temperature";
                        radiance_to_bt_conversion_constants:_FillValue = NC_FILL_FLOAT;

                    float channel_effective_solar_irradiance;
                        channel_solar_effective_irradiance: longname = "Channel integrated solar irradiance at 1AU";
                        channel_solar_effective_irradiance: units = "mW.m-2.(cm-1)-1";
                        channel_solar_effective_irradiance:_FillValue = NC_FILL_FLOAT;

} // measured

group: swath{
    :long_name = "Swath related information";

    dimensions:
        number_of_swaths = runtime_value; // configured at runtime
        number_of_swath_boundaries = runtime_value; // configured at runtime

    variables:

        ushort swath_boundary(number_of_swaths_boundaries, number_of_columns);
            swath_boundary:long_name = "Swath northern edge boundary"
            swath_boundary:_FillValue =  NC_FILL_USHORT;
            swath_boundary:valid_range = 1, configured_value;

        swath_direction_type swath_direction(number_of_swaths);
            swath_direction:long_name = "Scan direction of swath";

        double swath_earth_start_time(number_of_swaths);
            swath_earth_start_time:long_name  = "Time in UTC of first Earth measurement in the swath";
            swath_earth_start_time:standard_name = "time";
            swath_earth_start_time:units  = "seconds since 2000-01-01 00:00:00.0";
            swath_earth_start_time:precision = "1 millisecond";
            swath_earth_start_time:_FillValue =  NC_FILL_DOUBLE;

        double swath_earth_end_time(number_of_swaths);
            swath_earth_end_time:long_name  = "Time in UTC of last Earth measurement in the swath";
            swath_earth_end_time:standard_name = "time";
            swath_earth_end_time:units  = "seconds since 2000-01-01 00:00:00.0";
```

```
        swath_earth_end_time:precision = "1 millisecond";
        swath_earth_end_time:_FillValue =  NC_FILL_DOUBLE;

ushort swath_earth_start_column(number_of_swaths);
        swath_earth_start_column:long_name = "Column with first Earth measurement";
        swath_earth_start_column:_FillValue =  NC_FILL_USHORT;

ushort swath_earth_end_column(number_of_swaths);
        swath_earth_end_column:long_name = "Column with last Earth measurement";
        swath_earth_end_column:_FillValue =  NC_FILL_USHORT;

ushort swath_earth_start_row(number_of_swaths);
        swath_earth_start_column:long_name = "Row with first Earth measurement";
        swath_earth_start_column:_FillValue =  NC_FILL_USHORT;

ushort swath_earth_end_row(number_of_swaths);
        swath_earth_end_column:long_name = "Row with last Earth measurement";
        swath_earth_end_column:_FillValue =  NC_FILL_USHORT;

double swath_inclination(number_of_swaths);
        swath_inclination:long_name = "Inclination of the swath relative to the level 1c grid";
        swath_inclination:unit="degrees"
        swath_inclination:_FillValue =  NC_FILL_DOUBLE;

// Satellite and Solar Positions

double swath_earth_start_subsatellite_point_longitude(number_of_swaths);
        swath_earth_start_subsatellite_point_longitude:long_name = "Sub-satellite longitude at swath_earth_start_time";
        swath_earth_start_subsatellite_point_ longitude:units = "degrees";
        swath_earth_start_subsatellite_point_longitude:_FillValue =  NC_FILL_DOUBLE;

double swath_earth_start_subsatellite_point_latitude(number_of_swaths);
        swath_earth_start_subsatellite_point_latitude:long_name = "Sub-satellite latitude at swath_earth_start_time";
        swath_earth_start_subsatellite_point_ latitude:units = "degrees";
        swath_earth_start_subsatellite_point_ latitude:_FillValue =  NC_FILL_DOUBLE;

double swath_earth_end_subsatellite_point_longitude(number_of_swaths);
        swath_earth_end_subsatellite_point_longitude:long_name = "Sub-satellite longitude at swath_earth_end_time";
        swath_earth_end_subsatellite_point_longitude:units = "degrees";
        swath_earth_end_subsatellite_point_longitude:_FillValue =  NC_FILL_DOUBLE;

double swath_earth_end_subsatellite_point_latitude(number_of_swaths);
        swath_earth_end_subsatellite_point_latitude:long_name = "Sub-satellite latitude at swath_earth_end_time";
```

```
                    swath_earth_end_subsatellite_point_latitude:units = "degrees";
                    swath_earth_end_subsatellite_point_latitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_start_subsolar_point_longitude(number_of_swaths);
                    swath_earth_start_subsolar_point_longitude:long_name = "Sub-solar longitude at swath_earth_start_time";
                    swath_earth_start_subsolar_point_longitude:units = "degrees";
                    swath_earth_start_subsolar_point_longitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_start_subsolar_point_latitude(number_of_swaths);
                    swath_earth_start_subsolar_point_latitude:long_name = "Sub-solar latitude at swath_earth_start_time";
                    swath_earth_start_subsolar_point_latitude:units = "degrees";
                    swath_earth_start_subsolar_point_latitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_end_subsolar_point_longitude(number_of_swaths);
                    swath_earth_end_subsolar_point_longitude:long_name = "Sub-solar longitude at swath_earth_end_time";
                    swath_earth_end_subsolar_point_longitude:units = "degrees";
                    swath_earth_end_subsolar_point_longitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_end_subsolar_point_longitude(number_of_swaths);
                    swath_earth_end_subsolar_point_longitude:long_name = "Sub-solar longitude at swath_earth_end_time";
                    swath_earth_end_subsolar_point_longitude:units = "degrees";
                    swath_earth_end_subsolar_point_longitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_start_satellite_altitude(number_of_swaths);
                    swath_earth_start_satellite_altitude:long_name = "Satellite altitude at swath_earth_start_time";
                    swath_earth_start_satellite_altitude:unit = "km";
                    swath_earth_start_satellite_altitude:_FillValue =  NC_FILL_DOUBLE;

          double swath_earth_end_satellite_altitude(number_of_swaths);
                    swath_earth_end_satellite_altitude:long_name = "Satellite altitude at swath_earth_end_time";
                    swath_earth_end_satellite_altitude:unit = "km";
                    swath_earth_end_satellite_altitude:_FillValue =  NC_FILL_DOUBLE;

          double sun_earth_distance (number_of_swaths);
                    sun_earth_distance:long_name = "Distance from the centre of the sun to the centre of the earth at the swath midpoint";
                    sun_earth_distance:units = "km";
                    sun_earth_distance:_FillValue = NC_FILL_DOUBLE;

}// swath

group:quality_channel{
          :long_name = "Quality indicators applicable to a particular channel for the data chunk"
```

```
variables:
        uint number_of_expected_earth_pixels;
                number_of_expected_earth_pixels:long_name = "Number of expected Earth pixels";

        uint number_of_masked_pixels  ;
                number_of_masked_pixels:long_name = "Number of masked pixels";

        uint number_of_missing_earth_pixels;
                number_of_missing_earth_pixels:long_name = "Number of missing Earth pixels";

        uint number_of_oversaturated_earth_pixels;
                number_of_oversaturated_earth_pixels:long_name = "Number of over-saturated Earth pixels";

        uint number_of_undersaturated_earth_pixels;
                number_of_missing_earth_pixels:long_name = "Number of under-saturated Earth pixels";

        uint number_noise_warning_pixels;
                number_of_noise_warning_pixels:long_name = "Number of Earth pixels with noise_warning flag set";

        uint number_of_radiometric_warning_pixels;
                number_of_radiometric_warning_pixels:long_name = "Number of Earth pixels with radiometric_warning flag set";

        uint number_of_saturation_warning_pixels;
                number_of_saturation_warning_pixels:long_name = "Number of Earth pixels with saturation_warning flag set";

        uint number_of_missing_warning_pixels;
                number_of_missing_warning_pixels:long_name = "Number of Earth pixels with missing_warning flag set";

        uint number_of_straylight_warning_pixels;
                number_of_straylight_warning_pixels:long_name = "Number of Earth pixels with straylight_warning flag set";

        uint number_of_straylight_correction_warning_pixels;
                number_of_straylight_correction_warning_pixels:long_name = "Number of Earth pixels with straylight_warning flag set";

        uint number_of_extended_dynamic_range_warning_pixels; // value >0 only to IR3.8 fire channel
                number_of_extended_dynamic_range_warning_pixels:long_name = "Number of Earth pixels with extended_dynamic_range_warning flag set ";

        uint number_of_encoding_saturation_warning_pixels;
                number_of_encoding_saturation_warning_pixels:long_name = "Number of Earth pixels with encoding_saturation_warning flag set ";


} // quality_channel
```

```
        }// vis_o6

} // data

group: state{

        group:instrument{

                types:
                        uint enum fci_mode_type (Observation = 0, Decontamination = 1, Refocusing = 2, VNIR_Calibration = 3); //TBC based on Level 0 definition

                variables:
                        fci_mode_type fci_mode;
                                fci_mode:long_name = "Mode of FCI instrument";

                        string level0_channels(number_of_l0_channels );
                                channels_present:long_name = "FCI level 0 data channels"

                        double repeat_cycle_start_time;
                                repeat_cycle_start_time:long_name = "Start time in UTC of repeat cycle";
                                repeat_cycle_start_time:standard_name = "time";
                                repeat_cycle_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                                repeat_cycle_start_time:precision = "1 millisecond";
                                repeat_cycle_start_time:_FillValue =  NC_FILL_DOUBLE;

                        ushort repeat_sequence_counter;
                                repeat_sequence_counter:long_name = "Repeat sequence counter";

                        ushort repeat_cycle_counter;
                                repeat_cycle_counter:long_name = "Repeat cycle counter since the last transition to operational mode";

                        ushort repeat_cycle_counter_in_repeat_sequence;
                                repeat_cycle_counter_in_repeat_sequence:long_name = "Repeat cycle counter in the current repeat sequence";

                        ushort repeat_sequence_id;
                                repeat_sequence_id:long_name = "Repeat sequence ID";

                        ushort repeat_cycle_type;
                                repeat_cycle_type:long_name = "Repeat cycle type";

                        ushort scan_law_id;
                                scan_law_id = "Scan Law Identifier";
```

```
        boolean channel_on(number_of_l0_channels);
            channel_on:long_name = "Channel active";

        double last_decontamination_start_time;
            last_decontamination_start_time:long_name = "Start time in UTC of most recent decontamination";
            last_decontamination_start_time:standard_name = "time";
            last_decontamination_start_time:units = "seconds since 2000-01-01 00:00:00.0";
            last_decontamination_start_time:precision = "1 millisecond";
            last_decontamination_start_time:_FillValue =  NC_FILL_DOUBLE;

        double last_decontamination_end_time;
            last_decontamination_end_time:long_name = "End time in UTC of most recent decontamination";
            last_decontamination_end_time:standard_name = "time";
            last_decontamination_end_time:units = "seconds since 2000-01-01 00:00:00.0";
            last_decontamination_end_time:precision = "1 millisecond";
            last_decontamination_end_time:_FillValue =  NC_FILL_DOUBLE;

        double last_detection_chain_parameter_change_time(number_of_l0_channels );
            last_detection_chain_parameter_change_time:long_name = "Time in UTC of last change in the detection chain parameters";
            last_detection_chain_parameter_change_time:standard_name = "time";
            last_detection_chain_parameter_change_time:units = "seconds since 2000-01-01 00:00:00.0";
            last_detection_chain_parameter_change_time:precision = "1 millisecond";
            last_detection_chain_parameter_change_time:_FillValue =  NC_FILL_DOUBLE;

        double last_heated_black_body_calibration_time;
            last_heated_black_body_calibration_time:long_name = "Time in UTC of last heated black body calibration";
            last_heated_black_body_calibration_time:standard_name = "time";
            last_heated_black_body_calibration_time:units = "seconds since 2000-01-01 00:00:00.0";
            last_heated_black_body_calibration_time:precision = "1 millisecond";
            last_heated_black_body_calibration_time:_FillValue =  NC_FILL_DOUBLE;

        double last_mnd_calibration_time;
            last_mnd_calibration_time:long_name = "Time in UTC of last metallic neutral density calibration";
            last_mnd_calibration_time:standard_name = "time";
            last_mnd_calibration_time:units = "seconds since 2000-01-01 00:00:00.0";
            last_mnd_calibration_time:precision = "1 millisecond";
            last_mnd_calibration_time:_FillValue =  NC_FILL_DOUBLE;

    }//instrument

    group:processor{
```

```
dimensions:
        auxiliary_datasets;

        types:

                byte enum auxiliary_dataset_status{OK = 0, out_of_validity =1, missing =2};

                ubyte enum resampling_method_type ( // Options [TBD]
                                        TruncatedShannon8=0,
                                        TruncatedShannon16=1,
                                        BiCubicSpline=2,
                                        NUFTTiteration=3
                                        NearestNeighbour =4);

                ubyte enum weighting_function_type ( // Options [TBD]
                                None = 0,
                                Kaiser=1,
                                Hamming = 2);

                ubyte enum projection_type(Geostationary = 0); // Options [TBD]

        variables:

                string auxiliary_dataset_identifier(number_of_auxiliary_datasets);
                        auxiliary_dataset_identifier:long_name = "Identifier for auxiliary dataset"

                auxiliary_dataset_status auxiliary_dataset_status(number_of_auxiliary_datasets);
                        auxiliary_dataset_state:long_name = "Status of  auxiliary dataset"

                // Processing settings

                boolean radiance_linearization_enabled(number_of_l0_channels);
                        radiance_linearization_enabled:long_name = "Radiance linearization enabled in this dataset";

                boolean detector_equalization_enabled(number_of_l0_channels);
                        detector_equalization_enabled:long_name = "Detector equalization enabled in this dataset";

                boolean mtf_adaptation_enabled(number_of_1c_channels);
                        mtf_adaptation_enabled:long_name = "MTF adaptation enabled in this dataset";

                boolean straylight_correction_enabled(number_of_10_channels);
                        straylight_correction_enabled:long_name = "Straylight correction enabled in this dataset";
```

```
resampling_method_type resampling_method;
        resampling_method:long_name = "Selected resampling method";

weighting_function_type weighting_function;
        weighting_function:long_name = "Weighting function used with the selected resampling method";

// Processing history

boolean radiometric_warning(number_of_1c_channels);
        radiometric_warning.long_name = "Radiometric calibration warning for the complete repeat cycle";

boolean geometric_warning(number_of_1c_channels);
        geometric_warning.long_name = "Geometric processing warning for the complete repeat cycle";

// Reference Grid Parameters

reference_grid_type reference_grid(number_of_reference_grids);;
        reference_grid:long_name="Reference grid identifier";

string reference_grid_identifier(number_of_reference_grids);
        reference_grid_indentifier:long_name = "Filename from which reference grid parameters have been read";

string reference_grid_earth_model;
        reference_grid_earth_model:long_name = "Earth model used for reference grid";

ushort reference_grid_version(number_of_reference_grids);
        reference_grid_version:long_name = "Version of reference grid parameters ";

projection_type reference_grid_projection;
        reference_grid_projection:long_name = "Projection used for reference grid";

double projection_origin_longitude;
        projection_origin_longitude:long_name = "Longitude of projection origin";
        projection_origin_longitude:units = "degrees";
        projection_origin_longitude:_FillValue = NC_FILL_DOUBLE;

double projection_origin_latitude;
        projection_origin_latitude:long_name = "Latitude of projection origin";
        projection_origin_latitude:units = "degrees";
        projection_origin_latitude:_FillValue = NC_FILL_DOUBLE;

double reference_altitude;
        altitude:long_name = "Satellite reference altitude";
```

```
                altitude:units = "metres"

            float reference_grid_spatial_sampling_angle_ns(number_of_reference_grids);
                reference_grid_spatial_sampling_angle_ns:long_name = "Spatial sampling angle for each reference grid in North-South direction";
                reference_grid_spatial_sampling_angle_ns:units = "radians"

            float reference_grid_spatial_sampling_angle_ew(number_of_reference_grids);
                reference_grid_spatial_sampling_angle_ew:long_name = "Spatial sampling angle for each reference grid in East-West direction";
                reference_grid_spatial_sampling_angle_ew:units = "radians"

            double earth_polar_radius;
                earth_polar_radius:long_name = "Earth polar radius";
                earth_polar_radius:units = "metres"

            double earth_equatorial_radius;
                earth_equatorial_radius:long_name = "Earth equatorial radius";
                earth_equatorial_radius:units = "metres"

            uint reference_grid_number_of_columns(number_of_reference_grids);
                reference_grid_number_of_columns:long_name = "Number of columns in reference grid";

            uint reference_grid_number_of_rows(number_of_reference_grids);
                reference_grid_number_of_rows:long_name = "Number of rows in reference grid";

            double azimuth_angle_at_reference_grid_origin;
                azimuth_angle_at_reference_grid_origin:long_name = "Azimuth angle from the GEOS projection origin to the centre of the first reference grid column";
                azimuth_angle_at_reference_grid_origin:units="radians";

            double elevation_angle_at_reference_grid_origin;
                elevation_angle_at_reference_grid_origin:long_name = "Elevation angle from the GEOS projection origin to the centre of the first reference grid row ";
                elevation_angle_at_reference_grid_origin:units="radians";

} // processor

group: celestial{

        dimensions:
            sunglint_time = runtime_value;  // number of sunglint parameters

            moon_shadow_time = runtime_value; // number of solar eclipse parameters

        variables:
```

```
boolean sun_eclipse_by_earth;
        sun_eclipse_by_earth:lone_name ="Sun eclipse by Earth in this dataset";

boolean sun_eclipse_by_moon;
        sun_eclipse_by_moon:lone_name ="Sun eclipse by Moon in this dataset";

double eclipse_start_time;
        eclipse_start_time:long_name = "Start time of eclipse";
        eclipse_start_time:units = "seconds since 2000-01-01 00:00:00.0";
        eclipse_start_time:precision = "1 millisecond";
        eclipse_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

double eclipse_end_time;
        eclipse_end_time:long_name = "End time of eclipse";
        eclipse_end_time:units = "seconds since 2000-01-01 00:00:00.0";
        eclipse_end_time:precision = "1 millisecond";
        eclipse_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

boolean moon_shadow;
        moon_shadow:long_name = "Moon shadow on the Earth occurs in this dataset";

boolean sunglint;
        sunglint:long_name = "Sun glint possible in this dataset";

double sunglint_time(sunglint_time);
        sunglint_time:long_name = "Time in UTC of sunglint parameters";
        sunglint_time:standard_name = "time";
        sunglint_time:units = "seconds since 2000-01-01 00:00:00.0";
        sunglint_time:precision = "1 millisecond";
        sunglint_time:_FillValue = NC_FILL_DOUBLE;

double sunglint_centre_latitude(sunglint_time);
        sunglint_centre_latitude:units = "degrees";
        sunglint_centre_latitude:long_name = "Latitude of centre of sunglint circle";
        sunglint_centre_latitude:_FillValue = NC_FILL_DOUBLE;

double sunglint_centre_longitude(sunglint_time);
        sunglint_centre_longitude:units = "degrees";
        sunglint_centre_longitude:long_name = "Longitude of centre of sunglint circle";
        sunglint_centre_longitude:_FillValue = NC_FILL_DOUBLE;

double sunglint_radius(sunglint_time);
        sunglint_radius:units = "degrees";
```

```
                sunglint_radius:long_name = "Radius of sunglint circle";
                sunglint_radius:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_time(moon_shadow_time);
                moon_shadow_time:units = "seconds since 2000-01-01 00:00:00.0";
                moon_shadow_time:standard_name = "time";
                moon_shadow_time:long_name = "Time in UTC of moon shadow on the Earth parameters";
                moon_shadow_time:precision = "1 millisecond";
                moon_shadow_time:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_centre_latitude(moon_shadow_time);
                moon_shadow_centre_latitude:units = "degrees";
                moon_shadow_centre_latitude:long_name = "Latitude of centre of the moon shadow circle";
                moon_shadow_centre_latitude:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_centre_longitude(moon_shadow_time);
                moon_shadow_centre_longitude:units = "degrees";
                moon_shadow_centre_longitude:long_name = "Longitude of centre of moon shadow circle";
                moon_shadow_centre_longitude:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_penumbra_radius(moon_shadow_time);
                moon_shadow_penumbra_penumbra_radius:units = "degrees";
                moon_shadow_penumbra_radius:long_name = "Radius of moon shadow penumbra circle";
                moon_shadow_penumbra_radius:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_umbra_radius(moon_shadow_time);
                moon_shadow_umbra_radius:units = "degrees";
                moon_shadow_umbra_radius:long_name = "Radius of moon shadow umbra circle";
                moon_shadow_umbra_radius:_FillValue = NC_FILL_DOUBLE;

} //celestial

group: platform{

        types:
            byte enum manoeuvre_type(None = 0, NSSK = 1, EWSK = 2, SR = 3, MU = 4);  // TBC

            byte enum reference_frame(GCRF=1, EME2000=2, ITRF2008=3, TDR = 4 TEME = 5, TOD = 6 , RTN =7);

        variables:
            double recent_manoeuvre_time_window;
                recent_manoeuvre_time_window:long_name = "Recent manoeuvre time window";
                recent_manoeuvre_time_window: title = "Time window to search for a manoeuvre that starts before or during this dataset";
```

```
        recent_manoeuvre_time_window:units="seconds";
        recent_manoeuvre_time_window:_FILL_VALUE = NC_FILL_DOUBLE;

boolean recent_maneouvre_found
        recent_maneouvre_found:long_name = "Recent or current manoeuvre found";
        recent_maneouvre_found:title = "Recent or current manoeuvre found in the recent manoeuvre time window";

manoeuvre_type recent_manoeuvre_type;
        recent_manoeuvre_type:long_name = "Type of recent manoeuvre";

double recent_manoeuvre_start_time;
        recent_manoeuvre_start_time:long_name = "Start time of recent manoeuvre";
        recent_manoeuvre_start_time:units = "seconds since 2000-01-01 00:00:00.0";
        recent_manoeuvre_start_time:precision = "1 millisecond";
        recent_manoeuvre_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

double recent_manoeuvre_end_time;
        recent_manoeuvre_end_time:long_name = "End time of recent manoeuvre";
        recent_manoeuvre_end_time:units = "seconds since 2000-01-01 00:00:00.0";
        recent_manoeuvre_end_time:precision = "1 millisecond";
        recent_manoeuvre_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

reference_frame recent_manoeuvre_reference_frame;
        reference_frame:long_name = "Reference frame for manoeuvre paramaters";

double recent_manoeuvre_delta_vx
        recent_manoeuvre_delta_vx:long_name = "X component delta v for recent manoeuvre";
        recent_manoeuvre_delta_vx:units = "m/s";

double recent_manoeuvre_delta_vy
        recent_manoeuvre_delta_vy:long_name = "Y component delta v for recent manoeuvre";
        recent_manoeuvre_delta_vy:units = "m/s";

double recent_manoeuvre_delta_vz
        recent_manoeuvre_delta_vz:long_name = "Z component delta v for recent manoeuvre";
        recent_manoeuvre_delta_vz:units = "m/s";

double recent_manoeuvre_spacecraft_delta_mass;
        recent_manoeuvre_spacecraft_delta_mass:long_name = "Delta spacecraft mass for recent manoeuvre";
        recent_manoeuvre_delta_v:units = "g";

double upcoming_manoeuvre_time_window;
        upcoming_manoeuvre_time_window:long_name = "upcoming manoeuvre time window";
```

```
			upcoming_manoeuvre_time_window:title = "Time window to search for a manoeuvre that starts after this dataset";
			upcoming_manoeuvre_time_window:units="seconds";
			upcoming_manoeuvre_time_window:_FILL_VALUE = NC_FILL_DOUBLE;

		boolean upcoming_maneouvre_found
			upcoming_maneouvre_found:long_name = "Upcoming manoeuvre found";
			upcoming_maneouvre_found:title = "Upcoming manoeuvre found in the upcoming manoeuvre time window";

		manoeuvre_type upcoming_manoeuvre_type;
			upcoming_manoeuvre_type:long_name = "Type of upcoming manoeuvre";

		double upcoming_manoeuvre_start_time;
			upcoming_manoeuvre_start_time:long_name = "Start time of upcoming manoeuvre";
			upcoming_manoeuvre_start_time:units = "seconds since 2000-01-01 00:00:00.0";
			upcoming_manoeuvre_start_time:precision = "1 millisecond";
			upcoming_manoeuvre_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

		double upcoming_manoeuvre_end_time;
			upcoming_manoeuvre_end_time:long_name = "End time of upcoming manoeuvre";
			upcoming_manoeuvre_end_time:units = "seconds since 2000-01-01 00:00:00.0";
			upcoming_manoeuvre_end_time:precision = "1 millisecond";
			upcoming_manoeuvre_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

		reference_frame upcoming_manoeuvre_reference_frame;
			reference_frame:long_name = "Reference frame for manoeuvre paramaters";

		double upcoming_manoeuvre_delta_vx
			upcoming_manoeuvre_delta_vx:long_name = "X component delta v for upcoming manoeuvre";
			upcoming_manoeuvre_delta_vx:units = "m/s";

		double upcoming_manoeuvre_delta_vy
			upcoming_manoeuvre_delta_vy:long_name = "Y component delta v for upcoming manoeuvre";
			upcoming_manoeuvre_delta_vy:units = "m/s";

		double upcoming_manoeuvre_delta_vz
			upcoming_manoeuvre_delta_vz:long_name = "Z component delta v for upcoming manoeuvre";
			upcoming_manoeuvre_delta_vz:units = "m/s";

		double upcoming_manoeuvre_spacecraft_delta_mass;
			upcoming_manoeuvre_spacecraft_delta_mass:long_name = "Delta spacecraft mass for upcoming manoeuvre";
			upcoming_manoeuvre_delta_v:units = "g";

	} //platform
```

```
        } // state
} //  root
```

A.1.2 Variable Description

A.1.1.1 Root Group

Name	Description
Group Attributes	
Conventions	Actual values are [TBD] but an example of multiple conventions would be: `"CF-1.6, Unidata Dataset Discovery v1.0"`
title	Dataset/product name formatted as set out in Section 6 .
summary	Set to "Flexible Combined Imager (FCI) Level 1c Rectified Radiance dataset - body data chunk"
keywords	Set to "MTG FCI Rectified Radiance"
history	"original generated file"
institution	`"EUMETSAT".` This field may be extended with other values should datasets/products be generated in other locations.
location_indicator	As per the dataset name field" location_indicator" in Table 5
data_designator	As per the dataset name field "data_designator" in Table 5
spacecraft	As per the dataset name field "spacecraft" in Table 5
data_source	As per the dataset name field "data_source" in Table 5
processing_level	As per the dataset name field "level" in Table 5
coverage	As per the dataset name field "coverage" in Table 5
type	As per the dataset name field "type" in Table 5
subtype	As per the dataset name field "subtype" in Table 5
component1	As per the dataset name field "component1" in Table 5
component2	As per the dataset name field "component2" in Table 5
component3	As per the dataset name field "component3" in Table 5
baseline_version	Baseline version [TBC] The baseline version will reference of all other version numbers. Assumes processor_version is not sufficient for this.
release_version	Release version. [TBC] Used to tag datasets that can be considered to have a contiguous consistency sufficient for example, for consideration as a climate set.
processor_version	Processor version. Currently assumes a single processor version number suffices for the relevant IDPF or L2PP. Currently undefined if processor version also includes configuration of static auxiliary data and processor switch configuration, etc.
algorithm_version	Algorithm version [TBC] Currently unclear how this would be used and it may be

Name	Description
	redundant with processor_version.
format_version	Format version of the dataset/product.
start_time	As per the dataset name field “start_time” in Table 5
end_time	As per the dataset name field “end_time” in Table 5
processing_mode	As per the dataset name field “processing_mode” in Table 5
special_compression	As per the dataset name field “special_compression” in Table 5
subsetting	Identification of the type of subsetting performed.
disposition_mode	As per the dataset/product name field “disposition_mode” in Table 5
source	As particularised in the relevant dataset/product format specification, an array of strings of the form: (PROCESSOR_FULL_NAME) (CONFIGURATION_FILE_NAME)* (INPUT_DATASET/PRODUCT_NAME)* where the asterisks indicate zero or more instances. Note: it is intended that users of the dataset/product can determine from the Source attribute the version of the processing software and algorithm and the configuration data used to create the dataset/product, as well as the datasets/products that were inputs to its creation. Details [TBD]
facility_or_tool	As per the dataset name field “facility_or_tool” in Table 5
environment	As per the dataset name field “environment” in Table 5
references	“www.eumetsat.int” Note: It is intended that users of the dataset/product can access published, web-based references describing the data and the methods used to produce it at this address.
comment	Unless otherwise specified in the relevant dataset/product format specification, “None.”
processing_time	UTC time of processing formatted in Abbreviated Generalised Time format and defined as the time of the formatting of the dataset/product by the processor.
group_tag	String that represents a grouping of datasets that allows chunks and quick-looks to be linked together. The string has the format: YYYY_DDD_NNNN where YYYY = the year value of the “start_time” field DDD = day in year value of the “start_time” field, left padded with zeroes: 001 = Jan 1st, etc. NNNN = copy of the “accumulation_interval_in_day” field
accumulation_interval_in_day	As per the dataset/product name field “accumulation_interval_in_day” in Table 5

Name	Description
count_in_accumulation_interval	Cumulative count of the dataset chunk in the accumulation interval. Resets when the accumulation_interval_in_day value changes. The counter increments for each created chunk in an accumulation interval. It does not increment when a chunk is not created due to missing parent data. It allows the receiver of the data to check that no data was lost during dissemination.
instrument_configuation_id	Value of the "instrument configuration identifier" from the level 0 data ICU-I auxiliary data
instrument_configuation_id_version	Value of the "instrument configuration identifier version" from the level 0 data ICU-I auxiliary data
mtg_name	String field containing the MTG WMO-convention name for the file
alternative_name	String field containing a possible alternative name for the file (e.g. Sentinel-4 naming convention)
purpose	As per the dataset/product name field "purpose" in Table 5
format	As per the dataset/product name field "format" in Table 5
geospatial_lat_min	Geospatial_lat_min specifies the southernmost latitude covered by the dataset.
geospatial_lat_max	Geospatial_lat_max specifies the northernmost latitude covered by the dataset.
geospatial_lon_min	Geospatial_lon_min specifies the westernmost longitude covered by the dataset.
geospatial_lon_max	Geospatial_lon_max specifies the easternmost longitude covered by the dataset.
timeliness_ non_nominal	Timeliness does not meet the SRD/EURD requirements for this dataset. Any test not performed defaults to nominal result.
Types	
boolean	There is no boolean type in netCDF. This enumerated type at root level can be used by all datasets/products.
trilean	For situations where an undefined state is also required.
reference_grid_type	Identifies the reference grid used by the channel in terms of the Spatial Sampling Distance (SSD) at nadir
swath_direction_type	Identified the direction of swath acquisition from East to West or West to East.
Dimensions	
number_of_l0_channels	Number of data channels delivered by the FCI instrument used to create the level 1c data [17 if all channels are present, otherwise set at according to the channels available from the instrument]
number_of_l1c_channels	Number of spectral channels present in the originally generated dataset [16 if all FDHSI channels are present, 4 if all HRFI channels are present, otherwise set at according to the selected/available channels]

Name	Description
number_of_reference_grids	Number of reference grid used by the channels [default 2]. Note although 3 different grid exist for the FCI there are only 2 per mission (FDHSI/HRFI)
Variables	
l1c_channels_present	Level 1c spectral channels present in the originally generated dataset. Selection from FDHSI (“VIS0.4”, “VIS0.5”, “VIS0.6”, “VIS0.8”, “VIS0.9”, “NIR1.3”, “NIR1.6”, “NIR2.2”, “IR3.8”, “WV6.3”, “WV7.3”,”IR8.7”,”IR9.7”, “IR10.5”, “IR12.3”, “IR13.3”) HRFI (VIS0.6_HR”, NIR2.2_HR”, “IR3.8_HR”, “IR10.5_HR”)

A.1.1.2 Data Group

No content.

A.1.1.3 Generic Channel Group

Name	Description
Group Attributes	
long_name	A string uniquely identifying the channel wavelength and resolution e.g. “FCI FDHSI Visible 0.6 micron channel”
subsettable	Group can be included or excluded from the dataset according to configured selection
Types	
Dimensions	
number_of_rows	The number of rows in the Level 1c Body data chunk.
number_of_columns	The number of columns in the Level 1c Body data chunk this will equal either 5568, 11136 or 22272 data points depending on the channel.
Variables	
channel_srf_identifier	Identifier for the SRF for this channel.
channel_mtf_identifier	Identifier for the MTF for this channel.
channel_srf_version	Version number of the SRF for this channel.
channel_mtf_version	Version number of the MTF for this channel.
central_wavelength_specified	Specified central wavelength
spectral_width_specified	Specified spectral width
central_wavelength_actual	Actual (measured) central wavelength
spectral_width_actual	Actual (measured) spectral width
reference_grid	Indicates the reference grid used for rectification for this channel/SSD

A.1.1.4 Measured Group

Name	Description
Group Attributes	
Types	
Dimensions	
number_of_radiance_to_bt_conversion_coefficients	Number of coefficients used in the radiance to brightness temperature conversion formula
number_of_radiance_to_bt_conversion_constants	Number of constants used in the radiance to brightness temperature conversion formula
Variables	
start_position_row	Row index of the pixel in the effective_radiance array closest to the origin of the reference grid (most southerly row)
start_position_column	Column index of the pixel in the effective_radiance array closest to the origin of the reference grid (most easterly column)
end_position_row	Row index of the pixel in the effective_radiance array furthest from the origin of the reference grid (most northerly row)
end_position_column	Column index of the pixel in the effective_radiance array furthest from the origin of the reference grid (most westerly column)
effective_radiance	The effective radiance at each pixel. **NOTE:** For the IR_3.8 and IR_3.8_HR channels, the effective radiance is stored in a 16 bit integer but the merging of the extended radiometric range observations, aimed at fire radiance measurements, requires a different offset and gain to be applied to the data above the upper value in valid_cold_range. Masked pixels are set to the _FillValue
pixel_quality	Pixel quality flags
radiance_unit_conversion_coefficient	Conversion coefficients to convert radiance units from $mW.m^{-2}.sr^{-1}.(cm^{-1})^{-1}$ to $mW.m^{-2}.sr^{-1}.\mu m^{-1}$.
radiance_to_bt_conversion_coefficients	Conversion coefficients to convert radiance to brightness temperature to be used in the calculation of brightness temperature for IR spectral channels. Variable is set to _FillVaue for VNIR spectral channels.
radiance_to_bt_conversion_constants	Conversion constants to convert radiance to brightness temperature to be used in the calculation of brightness temperature for IR spectral channels. Variable is set to _FillVaue for VNIR spectral channels.
channel_effective_solar_irra	Channel effective solar irradiance at 1 AU to be used in

Name	Description
diance	the derivation of the reflectance for VNIR spectral channels. Variable is set to _FillVaue for IR spectral channels.

A.1.1.5 Swath Group

Name	Description
Group Attributes	
long_name	Group description " Swath related information"
Types	
Dimensions	
number_of_swaths	Number of swaths in the dataset
number_of_swath_boundaries	Number of swaths boundaries in the dataset
Variables	
swath_boundary	The northernmost row per column of the last pixel to have been created from a particular swath
swath_direction	Direction of data acquisition for the current swath (East to West or West to East)
swath_earth_start_time	Start time in UTC of the first valid earth data from a swath provided in the level 1c dataset
swath_earth_end_time	End time in UTC of the first valid earth data from a swath provided in the level 1c dataset
swath_earth_start_column	Column position in the reference grid of the first valid earth data from a swath provided in the level 1c dataset
swath_earth_end_column	Column position in the reference grid of the last valid earth data from a swath provided in the level 1c dataset
swath_earth_start_row	Row position in the reference grid of the first valid data from a swath provided in the level 1c dataset
swath_earth_end_row	Row position in the reference grid of the last valid data from a swath provided in the level 1c dataset
swath_inclination	Inclination of the swath relative to the level 1c grid
swath_earth_start_subsatellite _point_longitude	Position of the subsatellite point at the swath_earth_start_time in latitude and longitude
swath_earth_start_subsatellite _point_latitude	Position of the subsatellite point at the swath_earth_start_time in latitude and longitude
swath_earth_end_subsatellite_point_longitude	Position of the subsatellite point at the swath_earth_end_time in latitude and longitude
swath_earth_end_subsatellite_point_latitude	Position of the subsatellite point at the swath_earth_end_time in latitude and longitude
swath_earth_start_subsolar_	Position of the sun at the swath_earth_start_time

Name	Description
point_longitude	represented in Earth-Centred Earth-Fixed Frame (TBC)
swath_earth_start_subsolar_point_latitude	Position of the sun at the swath_earth_start_time represented in Earth-Centred Earth-Fixed Frame (TBC)
swath_earth_end_subsolar_point_longitude	Position of the sun at the swath_earth_end_time represented in Earth-Centred Earth-Fixed Frame (TBC)
swath_earth_end_subsolar_point_latitude	Position of the sun at the swath_earth_end_time represented in Earth-Centred Earth-Fixed Frame (TBC)
swath_earth_start_satellite_altitude	Satellite altitude at swath_earth_start_time
swath_earth_end_satellite_altitude	Satellite altitude at swath_earth_end_time
sun_earth_distance_start	Distance from the centre of the sun to the centre of the earth at the swath midpoint

A.1.1.6 Quality_Channel Group

Name	Description
Group Attributes	
long_name	Description of group "Quality indicators applicable to a particular channel for the data chunk"
Variables	
number_of_expected_earth_pixels	Number of earth pixels that are expected be in the nominal dataset
number_of_masked_pixels	Number of space pixels that have been masked.
number_of_missing_earth_pixels	Number of Earth pixels flagged as missing
number_of_oversaturated_earth_pixels	Number of earth pixels that are over-saturated in dataset (set to saturated flag value)
number_of_undersaturated_earth_pixels	Number of earth pixels that are under-saturated in dataset (set to under-saturated flag value)
number_noise_warning_pixels	Number of Earth pixels with noise_warning flag set
number_radiometric_warning_pixels	Number of Earth pixels with radiometric_warning flag set
number_of_saturation_warning_pixels	Number of Earth pixels with saturation_warning flag set
number_of_missing_warning_pixels	Number of Earth pixels with missing_warning flag set
number_of_straylight_warning_pixels	Number of Earth pixels with straylight_warning flag set
number_of_straylight_correction_warning_pixels	Number of Earth pixels with straylight_correction_warning flag set

Name	Description
number_of_extended_dynamic_range_warning_pixels	Number of Earth pixels with extended_dynamic_range_warning flag set
number_of_encoding_saturation_warning_pixels	Number of Earth pixels with encoding_saturation_warning flag set

A.1.1.7 State Group

No content.

A.1.1.8 Instrument Group

Name	Description
Types	
fci_mode_type	FCI Mode. Note that in decontamination mode only the Visible channels are generated. For refocusing mode the level 1c product is generated only if Earth targets are used and the data is not disseminated. VNIR calibration does not generate earth view data during the' blind' LAC the other 4 LACs in the 10 minute cycle are disseminated, but are not expected to meet geometric performance.
Variables	
fci_mode	Mode of the FCI instrument from the level 0 ICU-I auxiliary data
level0_channels	Array of strings indicating the FCI data channels delivered in the level 0 data ("FDVIS0.4", "FDVIS0.5", "HRVIS0.6", "FDVIS0.8", "FDVIS0.9", "FDNIR1.3", "FDNIR1.6", "HRNIR2.2", "HRIR3.8", "FAIR3.8", "FDIR6.3", "FDIR7.3","FDIR8.7","FDIR9.7", "HRIR10.5", "FDIR12.3", "FDIR13.3")
repeat_cycle_start_time	Conversion of the "repeat_cycle_start_time" from the level 0 ICU-I auxiliary data into UTC
repeat_sequence_counter	Copy of the "repeat sequence counter" from the level 0 ICU-I auxiliary data
repeat_cycle_counter	Copy of the "repeat cycle counter since the last transition to operational mode" from the level 0 ICU-I auxiliary data
repeat_cycle_counter_in_repeat_sequence	Copy of the "repeat cycle counter in repeat sequence" from the level 0 ICU-I auxiliary data
repeat_sequence_id	Copy of the "repeat sequence identifier" from the level 0 ICU-I auxiliary data
repeat_cycle_type	Copy of the "repeat cycle type" from the level 0 ICU-I auxiliary data, giving the current repeat cycle type
scan_law_id	Copy of the "Scan law id" from the level 0 ICU-I

Name	Description
	auxiliary data
channel_on	TRUE if the spectral channel is switched on and active
last_decontamination_start_time	Start time in UTC of most recent decontamination
last_decontamination_end_time	End time in UTC of most recent decontamination
last_detection_chain_parameter_change_time	Time in UTC of the last change in detection chain parameters, corresponding to the start of the repeat cycle when the parameters were activated
last_heated_black_body_calibration_time	Time in UTC of the last heated black body calibration for the IR spectral channels, corresponding to the start of the calibration data acquisition
last_mnd_calibration_time	Time in UTC of the last metallic neutral density calibration for the VNIR spectral channels, corresponding to the start of the calibration data acquisition

A.1.1.9 Platform Group

Name	Description
Enumerated Types	
manoeuvre_type	NSSK = North-South Station Keeping, EWSK = East-West Station Keeping, SR = Station Relocation, MU = Momentum Unloading.
reference_frame	Options are GCRF, EME2000, ITRF2008, TDR, TEME, TOD and RTN.
Variables	
recent_manoeuvre_time_window	Window of time prior to dataset start that is searched for a recent manoeuvre. (Default of configured value).
recent_maneouvre_found	Boolean to indicate if a recent manoeuvre was found (Derived by comparing manoeuvre time range to L1 dataset time range as described above)
recent_manoeuvre_type	Type of manoeuvre (Source of data TBD)
recent_manoeuvre_start_time	Start time for manoeuvre (derived from iOrbitParametersOPM. MAN_EPOCH_IGNITION)
recent_manoeuvre_end_time	End time for manoeuvre (derived from iOrbitParametersOPM. MAN_DURATION)

Name	Description
recent_manoeuvre_reference_frame	Reference frame for manoeuvre parameters (derived from iOrbitParametersOPM. MAN_REF_FRAME)
recent_manoeuvre_delta_vx	X component of the velocity increment (derived from iOrbitParametersOPM. MAN_DV_1)
recent_manoeuvre_delta_vy	Y component of the velocity increment (derived from iOrbitParametersOPM. MAN_DV_2)
recent_manoeuvre_delta_vz	Z component of the velocity increment (derived from iOrbitParametersOPM. MAN_DV_3)
recent_manoeuvre_spacecraft_delta_mass	Change in spacecraft mass (derived from iOrbitParametersOPM. MAN_DELTA_MASS)
upcoming_manoeuvre_time_window	Windows of time that is searched post dataset end for an upcoming manoeuvre. (Upcoming manoeuvre parameters have the same source as the recent manoeuvre parameters)
upcoming_maneouvre_found	Boolean to indicate if an upcoming manoeuvre was found
upcoming_manoeuvre_type	Type of manoeuvre
upcoming_manoeuvre_start_time	Start time for manoeuvre
upcoming_manoeuvre_end_time	End time for manoeuvre
upcoming_manoeuvre_reference_frame	Reference frame for manoeuvre parameters
upcoming_manoeuvre_delta_vx	X component of the velocity increment
upcoming_manoeuvre_delta_vy	Y component of the velocity increment
upcoming_manoeuvre_delta_vz	Z component of the velocity increment
upcoming_manoeuvre_spacecraft_delta_mass	Change in spacecraft mass

A.1.1.10 Processor Group

Name	Description
Dimensions	
auxiliary_datasets	Number of auxiliary datasets used by the processor
Types	
auxiliary_dataset_status	Possible states for an auxiliary dataset 0 = OK 1 = dataset was used but was out of its stated validity time 2 = auxiliary dataset was not available [TBC – not clear what identifier can be provided if file

Name	Description
	is not available].
resampling_method_type	Resampling method applied to the level 1b samples to create the level 1c dataset
weighting_function_type	Weighting function used with the selected resampling method.
projection_type	Projection for the reference grid
Variables	
auxiliary_dataset_identifier	Unique identifier for the auxiliary dataset. If available, the filename should be used. If the auxiliary file was not available, the file name template should be stated, with unknown values such as times set to the correct length of lower case x characters.
auxiliary_dataset_status	Status of the auxiliary dataset
radiance_linearization_enabled	TRUE if radiance linearization has been applied to the dataset
detector_equalization_enabled	TRUE if detector equalization has been applied to the dataset
mtf_adaptation_enabled	TRUE if MTF adaption has been applied to the dataset
straylight_correction_enabled	TRUE if stray light correction has been applied to the dataset
resampling_method	Resampling method applied to the level 1b samples to create the level 1c dataset
weighting_function	Weighting function used with the selected resampling method.
radiometric_warning	Radiometric calibration in the previous repeat cycles has led to a potential problem in the calibration of the channel data for the complete repeat cycle, e.g. a missing black body calibration.
geometric_warning	The geometric processing in the previous repeat cycles has not allowed the update of the INR state vector the required accuracy to allow current repeat cycle measurements to be guaranteed.
reference_grid	Identifies which of the three SSD-based grids the parameters are associated with.
reference_grid_identifier	Filename from which reference grid parameters have been read
reference_grid_version	Version number of the set of reference grid parameters. A change in version number between datasets implies the grid must be recalculated.
reference_grid_earth_model	Earth model used for reference grid
reference_grid_projection	Projection used for reference grid

Name	Description
projection_origin_longitude	Longitude of projection origin
projection_origin_latitude	Latitude of projection origin
reference_altitude	Satellite reference altitude
reference_grid_spatial_sampling_angle_ns	Spatial sampling angle for each reference grid in North-South direction
reference_grid_spatial_sampling_angle_ew	Spatial sampling angle for each reference grid in East-West direction
earth_polar_radius	Earth polar radius
earth_equatorial_radius	Earth equatorial radius
reference_grid_number_of_columns	Number of columns in reference grid
reference_grid_number_of_rows	Number of rows in reference grid
azimuth_angle_at_reference_grid_origin	Azimuth angle from the GEOS projection origin to the centre of the first reference grid column
elevation_angle_at_reference_grid_origin	Elevation angle from the GEOS projection origin to the centre of the first reference grid row

A.1.1.11 Celestial Group

Name	Description
Dimensions	
sunglint_time	Number times for which the sunglint parameters are provided
moon_shadow_time	Number times for which the solar eclipse parameters are provided
Variables	
moon_shadow	TRUE if the moon shadow on the Earth occurs in this dataset within the FCI field of regard during the repeat cycle
sunglint	TRUE if sunglint is possible within the FCI field of regard during the repeat cycle
sunglint_time	Array of times in UTC for the following sunglint parameters
sunglint_centre_latitude	Latitude of centre of sunglint circle
sunglint_centre_longitude	Longitude of centre of sunglint circle
sunglint_radius	Radius of sunglint circle
moon_shadow_time	Array of times in UTC for the following moon shadow parameters
moon_shadow_centre_latitude	Latitude of centre of moon shadow circle
moon_shadow_centre_longitude	Longitude of centre of moon shadow circle
moon_shadow_penumbra_radius	Radius of moon shadow penumbra circle

Name	Description
moon_shadow_umbra_radius	Radius of moon shadow umbra circle
sun_eclipse_by_earth	If TRUE indicates an eclipse of the sun by the earth, as viewed by the satellite, occurred during dataset acquisition
sun_eclipse_by_moon	If TRUE indicates an eclipse of the sun by the moon, as viewed by the satellite, occurred during dataset acquisition
eclipse_start_time	Start time of eclipse
eclipse_end_time	End time of eclipse

A.2 Trailer Chunk

A.2.1 CDL Description

```
group: root{ // Generic

        :Conventions = "";  // To be determined
        :title  = product_name;
        :summary = ""; //see relevant format specification document
        :keywords = ""; //see relevant format specification document
        :history = "original generated file";
        :institution = "EUMETSAT"; // For datasets generated at EUMETSAT
        :location_indicator = "";//
        :data_designator  = "";
        :spacecraft = "";
        :product_id = "";
        :processing_level = "";
        :baseline_version = "";
        :release_version = "";
        :processor_version = "";
        :algorithm_version = "";
        :format_version = "";
        :sensing_start = "YYYYMMDDhhmmss";
        :end_time = "YYYYMMDDhhmmss";
        :processing_mode  = "";
        :special_compression = "";
        :subsetting  = "";
        :disposition_mode  = "";
        :source = "";
        :facility_or_tool = "";
        :environment = "";
        :references = "";
        :comment = "";
        :processing_time = "";
        :group_tag = "";
```

```
        :accumulation_interval_in_day = 1;
        :count_in_accumulation_interval = 1;
        :instrument_configuration_id = 0;
        :instrument_configuration_id_version = 0;
        :subsettable_groups = "";
        :subsettable_groups_present = "";
        :mtg_name="";
        :alternative_name="";
        :purpose="";
        :format="";
        :geospatial_lat_min = 0;
        :geospatial_lat_max = 0;
        :geospatial_lon_min = 0;
        :geospatial_lon_max = 0;

        types:
                byte enum boolean {false = 0, true =1}
                byte enum trilean {false = 0, true =1, undefined =2}
          ubyte enum reference_grid_type (500m = 0, 1km = 1, 2km = 2);
    ubyte enum swath_direction_type(EastWest = 0, WestEast = 1);

        dimensions:
                number_of_l0_channels = runtime_value or configured_value;
                number_of_llc_channels = runtime_value or configured_value;
                number_of_reference_grids  = 2;

        variables:
   string l1c_channels_present(number_of_l1c_channels);
     channels_present:long_name = "Level 1c spectral channels present in dataset"

        boolean timeliness_non_nominal;
                        timeliness_non_nominal:long_name = "Timeliness non-nominal warning flag";

        group: data{

                group: vis_06{   // Only one example channel group shown
                        : long_name="FCI FDHSI Visible 0.6 micron channel";
                        : subsettable = "yes";

                        dimensions:
                                number_of_rows = configured_value;
                                number_of_columns = configured_value;
```

```
variables:
        string channel_srf_identifier;
                channel_srf_identifier:long_name="Channel Spectral Response Function identifier";

        string channel_mtf_identifier;
                channel_mtf_identifier:long_name="Channel Modulation Transfer Function identifier";

        ushort channel_srf_version;
                channel_srf_file:long_name="Channel Spectral Response Function identifier version";

        ushort channel_mtf_version;
                channel_mtf_file:long_name="Channel Modulation Transfer Function identifier version";

        float central_wavelength_specified;
                central_wavelength_specified:long_name="Specified central wavelength of channel";
                central_wavelength_specified:units="micrometres";

        float spectral_width_specified;
                spectral_width_specified:long_name="Specified spectral width of channel";
                spectral_width_specified:units=" micrometres";

        float central_wavelength_actual;
                central_wavelength_actual:long_name="Actual central wavelength of channel";
                central_wavelength_actual:units="micrometres";

        float spectral_width_actual;
                spectral_width_actual:long_name="Actual FWHM spectral width of channel";
                spectral_width_actual:units=" micrometres";

        reference_grid_type reference_grid;
                reference_grid:long_name="Reference grid used for this channel";

        group:measured{

                dimensions:
                        number_of_radiometric_noise_lut_steps  = configured_value;  // default = 1024
                        number_of_radiometric_accuracy_lut_steps = configured_value;  // default = 1024
                        number_of_radiance_to bt_conversion_coefficients = configured_value;  // default = 3 for IR channels, 0 for VNIR channels;
                        number_of_radiance_to bt_conversion_constants = configured_value;  // default = 2 for IR channels, 0 for VNIR channels;

                variables:

                        ushort radiometric_noise_lut_noise(number_of_radiometric_noise_lut_steps);
```

```
        radiometric_noise_lut_noise:long_name = "Look-up-table for the radiometric noise applicable to the effective radiance – radiometric noise";
        radiometric_noise_lut_noise::standard_name = "effective_radiance_in_wavenumber standard_error"
        radiometric_noise_lut_noise:units = "mW.m-2.sr-1.(cm-1)-1";
        radiometric_noise_lut_noise:_FillValue = NC_FILL_SHORT;
        radiometric_noise_lut_noise:valid_range = 0, configured_value;
        radiometric_noise_lut_noise:scale_factor = runtime_value;
        radiometric_noise_lut_noise:add_offset = runtime_value;
        radiometric_noise_lut_noise:ancillary_variables = "radiometric_noise_lut_radiance";

ushort radiometric_noise_lut_radiance(number_of_radiometric_noise_lut_steps);
        radiometric_noise_lut_radiance:long_name = "Look-up-table for the radiometric noise applicable to the effective radiance – radiance";
        radiometric_noise_lut_radiance:units = "mW.m-2.sr-1.(cm-1)-1";
        radiometric_noise_lut_radiance:_FillValue = NC_FILL_SHORT;
        radiometric_noise_lut_radiance:valid_range = 0, 4095;  // 0,8191 for IR3.8
        radiometric_noise_lut_radiance:valid_cold_range = 0, 4095;
        radiometric_noise_lut_radiance:scale_factor = runtime_value;
        radiometric_noise_lut_radiance:add_offset = runtime_value;
        radiometric_noise_lut_radiance:warm_scale_factor = runtime_value;
        radiometric_noise_lut_radiance:warm_add_offset = runtime_value;

ushort radiometric_accuracy_lut_accuracy(number_of_radiometric_accuracy_lut_steps);
        radiometric_accuracy_lut_accuracy:long_name = "Look-up-table for the radiometric accuracy applicable to the effective radiance – radiometric accuracy";

        radiometric_accuracy_lut_accuracy:units = "mW.m-2.sr-1.(cm-1)-1";
        radiometric_accuracy_lut_accuracy:_FillValue = NC_FILL_SHORT;
        radiometric_accuracy_lut_accuracy:valid_range = 0, configured_value;
        radiometric_accuracy_lut_accuracy:scale_factor = runtime_value;
        radiometric_accuracy_lut_accuracy:add_offset = runtime_value;
        radiometric_accuracy_lut_accuracy:ancillary_variables = "radiometric_accuracy_lut_radiance";

ushort radiometric_accuracy_lut_radiance(number_of_radiometric_accuracy_lut_steps);
        radiometric_accuracy_lut_radiance:long_name = "Look-up-table for the radiometric noise applicable to the effective radiance – radiance";
        radiometric_accuracy_lut_radiance:units = "mW.m-2.sr-1.(cm-1)-1";
        radiometric_accuracy_lut_radiance:_FillValue = NC_FILL_SHORT;
        radiometric_accuracy_lut_radiance:valid_range = 0, 4095;  // 0,8191 for IR3.8
        radiometric_accuracy_lut_radiance:valid_cold_range = 0, 4095;
        radiometric_accuracy_lut_radiance:scale_factor = runtime_value;
        radiometric_accuracy_lut_radiance:add_offset = runtime_value;
        radiometric_accuracy_lut_radiance:warm_scale_factor = runtime_value;
        radiometric_accuracy_lut_radiance:warm_add_offset = runtime_value;

float radiance_unit_conversion_coefficient;
        radiance_unit_conversion_coefficient:long_name = "Coefficient used to convert effective radiance units from per cm^-1 to per wavelength in micron";
```

```
                    radiance_unit_conversion_coefficient:unit = "(cm-1).micron"

          float radiance_to_bt_conversion_coefficients(number_of_radiance_to bt_conversion_coefficients)
                    radiance_to_bt_conversion_coefficients:longname = "Radiance to brightness temperature conversion coefficients";
                    radiance_to_bt_conversion_coefficients:_FillValue = NC_FILL_FLOAT;

          float radiance_to bt_conversion_constants(number_of_radiance_to bt_conversion_constants)
                    radiance_to bt_conversion_constants:longname = "Constants used to convert effective radiance to brightness temperature";
                    radiance_to_bt_conversion_constants:_FillValue = NC_FILL_FLOAT;

          float channel_effective_solar_irradiance;
                    channel_solar_effective_irradiance: longname = "Channel integrated solar irradiance at 1AU";
                    channel_solar_effective_irradiance: units = "mW.m-2.(cm-1)-1";
                    channel_solar_effective_irradiance:_FillValue = NC_FILL_FLOAT;

} // measured

group: swath{
          :long_name = "Swath related information";

          dimensions:
                    number_of_swaths = runtime_value; // configured at runtime
                    number_of_swath_boundaries = runtime_value; // configured at runtime

          variables:

                    // Swath quality data

                    uint number_of_samples(number_of_swaths);
                              number_of_samples:long_name = "Number of Earth samples";

                    uint number_of_missing_samples(number_of_swaths);
                              number_of_missing_samples:long_name = "Number of missing Earth samples";

                    uint number_of_oversaturated_samples(number_of_swaths);
                              number_of_oversaturated_samples:long_name = "Number of over-saturated Earth samples";

                    uint number_of_undersaturated_samples(number_of_swaths);
                              number_of_undersaturated_samples:long_name = "Number of under-saturated Earth samples";

                    uint number_of_extended_dynamic_range_samples(number_of_swaths);// value >0 only to IR3.8 fire channel
                              number_of_extended_dynamic_range_samples:long_name = "Number of extended dynamic range samples";
```

```
        boolean swath_coverage_compliance (number_of_swaths);
            swath_coverage_compliance:long_name = “Compliance for Swath coverage”;

        boolean swath_overlap_compliance (number_of_swath_boundaries);
            swath_overlap_compliance:long_name = “Compliance for Swath overlap”;

        double interswath_navigation_error(number_of_swath_boundaries);
            interswath_navigation_error:long_name:long_name = “Inter-swath navigation error between adjacent swaths”;

        boolean interswath_navigation_compliance(number_of_swath_boundaries);
            interswath_navigation_compliance:long_name = “Compliance for Inter-swath navigation error between adjacent swaths”;

        double sun_earth_distance (number_of_swaths);
            sun_earth_distance:long_name = "Distance from the centre of the sun to the centre of the earth at the swath midpoint";
            sun_earth_distance:units = "km";

}// swath

group: quality_channel{
    :long_name = "Quality indicators applicable to a particular channel for the repeat cycle"

    types:
        ubyte enum channel_status_type(NOMINAL=0, NON-NOMINAL=1);

    variables:
        channel_status_type channel_status;
            channel_status:long_name = "Status of channel";

        uint number_of_expected_earth_pixels;
            number_of_expected_earth_pixels:long_name = "Number of expected Earth pixels";

        uint number_of_masked_pixels;
            number_of_masked_pixels:long_name = "Number of masked pixels";

        uint number_of_missing_earth_pixels;
            number_of_missing_earth_pixels:long_name = "Number of missing Earth pixels";

        uint number_of_oversaturated_earth_pixels;
            number_of_oversaturated_earth_pixels:long_name = "Number of over-saturated Earth pixels";

        uint number_of_undersaturated_earth_pixels;
            number_of_missing_earth_pixels:long_name = "Number of under-saturated Earth pixels";
```

```
uint number_of_noise_warning_pixels;
        number_of_noise_warning_pixels:long_name = "Number of Earth pixels with noise_warning flag set";

uint number_of_radiometric_warning_pixels;
        number_of_radiometric_warning_pixels:long_name = "Number of Earth pixels with radiometric_warning flag set";

uint number_of_saturation_warning_pixels;
        number_of_saturation_warning_pixels:long_name = "Number of Earth pixels with saturation_warning flag set";

uint number_of_missing_warning_pixels;
        number_of_missing_warning_pixels:long_name = "Number of Earth pixels with missing_warning flag set";

uint number_of_straylight_warning_pixels;
        number_of_straylight_warning_pixels:long_name = "Number of Earth pixels with straylight_warning flag set";

uint number_of_straylight_correction_warning_pixels;
        number_of_straylight_correction_warning_pixels:long_name = "Number of Earth pixels with straylight_warning flag set";

uint number_of_extended_dynamic_range_warning_pixels; // value >0 only to IR3.8 fire channel
        number_of_extended_dynamic_range_warning_pixels:long_name = "Number of Earth pixels with extended_dynamic_range_warning flag set ";

uint number_of_encoding_saturation_warning_pixels;
        number_of_encoding_saturation_warning_pixels:long_name = "Number of Earth pixels with encoding_saturation_warning flag set ";

float percentage_coverage_achieved;
        percentage_coverage_achieved:long_name = "Percentage Earth coverage achieved";
        percentage_coverage_achieved:units = "percent";

boolean completeness_compliance;
        completeness_compliance:long_name = "Image completeness compliance";

boolean accuracy_compliance;
        accuracy_compliance:long_name = "Image accuracy compliance";

boolean coverage_compliance;
        coverage_compliance:long_name = "Coverage compliance";

boolean radiometric_restricted_zone_applied;
        radiometric_restricted_zone_applied:long_name = "Pixels in the solar radiometric restricted zone present in image";

boolean sse_compliance;
        sse_compliance:long_name = "Compliance to spatial sampling error requirement";
```

```
                float absolute_pixel_position_knowledge_error;
                        absolute_pixel_position_knowledge_error:long_name = "Absolute pixel position knowledge error";

                boolean absolute_pixel_position_knowledge_error_compliance;
                        absolute_pixel_position_knowledge_error_compliance:long_name = "Compliance to absolute pixel position knowledge error requirement";

                float absolute_pixel_position_knowledge_error_500;
                        absolute_pixel_position_knowledge_error_500:long_name = "Absolute pixel position knowledge error within a 500x500 pixel imagette";

                boolean absolute_pixel_position_knowledge_error_500_compliance;
                        absolute_pixel_position_knowledge_error_500_compliance:long_name = "Compliance to absolute pixel position knowledge error within a 500x500 pixel imagette
requirement";

                float absolute_pixel_position_knowledge_error_100;
                        absolute_pixel_position_knowledge_error_100:long_name = "Absolute pixel position knowledge error within a 100x100 pixel imagette ";

                boolean absolute_pixel_position_knowledge_error_100_compliance;
                        absolute_pixel_position_knowledge_error_100_compliance:long_name = "Compliance to absolute pixel position knowledge error within a 100x100 pixel imagette
requirement";

                float relative_pixel_position_knowledge_error;
                        relative_pixel_position_knowledge_error:long_name = "Relative pixel position knowledge error";
                        relative_pixel_position_knowledge_error:title  = "Pixel position knowledge error relative to previous repeat cycle";

                boolean relative_pixel_position_knowledge_error_compliance;
                        relative_pixel_position_knowledge_error_compliance:long_name = "Compliance to relative pixel position knowledge error to previous repeat cycle requirement";

                boolean radiometric_noise_compliance;
                        radiometric_noise_compliance:long_name = "Compliance to radiometric noise requirement";

                boolean noise_power_spectral_density_compliance;
                        noise_power_spectral_density_compliance:long_name = "Compliance to noise power spectral density requirement";

                boolean radiometric_accuracy_compliance;
                        radiometric_accuracy_compliance:long_name = "Compliance to radiometric noise requirement";

        } // quality_channel

        group: external_calibration_coefficicients{
                :long_name = “Calibration coefficients for the FCI generated calibration derived from external means”
                :subsettable = “yes”;

                dimensions:
```

```
                    number_of_external_calibration_coefficents = configured_value;
                    min_max = 2;

                variables:
                    string external_calibration_coefficicients_filename;
                        external_calibration_coefficicients_filename:long_name = "Name of the External Calibration Coefficient data file";

                    double external_calibration_coefficicients_update_time;
                        external_calibration_coefficients_update_time:long_name = "Time in UTC the External Calibration Coefficient data was last updated";
                        external_calibration_coefficients_update_time:standard_name = "time";
                        external_calibration_coefficients_update_time:units  = “seconds since 2000-01-01 00:00:00.0”;
                        external_calibration_coefficients_update_time:precision = “1 millisecond”;
                        external_calibration_coefficients_update_time:_FillValue =  NC_FILL_DOUBLE;

                    double validity_period(min_max);
                        validity_period:long_name = "Period in UTC over which the External Calibration Coefficients are valid";
                        validity_period:standard_name = "time";
                        validity_period:units  = “seconds since 2000-01-01 00:00:00.0”;
                        validity_period:precision = “1 millisecond”;
                        validity_period:_FillValue =  NC_FILL_DOUBLE;

                    boolean external_calibration_coefficicients_valid;
                        external_calibration_coefficicients_valid:long_name = “External calibration coefficients valid for the current repeat cycle”;

                    double external_calibration_coefficients(number_of_external_calibration_coefficents);
                        external_calibration_coefficients:long_name = "External calibration coefficients";

                    double external_calibration_covariance_matrix(number_of_external_calibration_coefficents, number_of_external_calibration_coefficents);
                        external_calibration_covariance:long_name = "External calibration covariance matrix";
                        external_calibration_covariance:_FillValue = NC_FILL_DOUBLE;

                    double radiance_validity_range(min_max);
                        validity_radiance_range:long_name = “Range of randiance over which external calibration coefficients are valid”;
                        validity_radiance_range:units=” mW.m-2.sr-1.(cm-1)-1”;
                        validity_radiance_range:_FillValue = NC_FILL_DOUBLE;

            } // external_calibration_coefficients


    }// vis_o6

    group: quality{
```

```
:long_name = "Quality indicators at data level";

dimensions:
        channel_pair = 2;
        number_of_icdt_channel_pairs = configured_value; // FDHSI default = 28, HRFI default = 2
        number_of_icdt_temperatures = configured_value; //default = 5
        number_of_rppke_channel_pairs = configured_value; //FDHSI default = 64, HRFI default = 4
        number_of_icra_channel_pairs = configured_value; // FDHSI default = 56, HRFI default = 2

variables:

        boolean geometric_restricted_zone_earth_applicable;
                geometric_restricted_zone_earth _applicable:long_name = "Geometric restricted operations due to a sun eclipse by Earth from satellite during the dataset";

        double geometric_restricted_zone_earth_start_time;
                geometric_restricted_zone_earth_start_time:long_name = "Start in UTC of geometric restricted operations due to a sun eclipse by Earth from satellite";
                geometric_restricted_zone_earth_start_time:standard_name = "time";
                geometric_restricted_zone_earth_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                geometric_restricted_zone_earth_start_time:precision = “1 millisecond”;
                geometric_restricted_zone_earth_start_time:_FillValue = NC_FILL_DOUBLE;

        double geometric_restricted_zone_earth_end_time;
                geometric_restricted_zone_earth_end_time:long_name = "End in UTC of geometric restricted operations due to a sun eclipse by Earth from satellite";
                geometric_restricted_zone_earth_end_time:standard_name = "time";
                geometric_restricted_zone_earth_end_time:units = "seconds since 2000-01-01 00:00:00.0";
                geometric_restricted_zone_earth_end_time:precision = “1 millisecond”;
                geometric_restricted_zone_earth_end_time:_FillValue = NC_FILL_DOUBLE;

        boolean geometric_restricted_zone_moon_applicable;
                geometric_restricted_zone_moon_applicable:long_name = "Geometric restricted operations due to a sun eclipse by moon from satellite during the dataset ";

        double geometric_restricted_zone_moon_start_time;
                geometric_restricted_zone_moon_start_time:long_name = “Start in UTC of geometric restricted operations due to a sun eclipse by moon from satellite";
                geometric_restricted_zone_moon_start_time:standard_name = "time";
                geometric_restricted_zone_moon_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                geometric_restricted_zone_moon_start_time:precision = “1 millisecond”;
                geometric_restricted_zone_moon_start_time:_FillValue = NC_FILL_DOUBLE;

        double geometric_restricted_zone_moon_end_time;
                geometric_restricted_zone_moon_end_time:long_name = “End in UTC of geometric restricted operations due to a sun eclipse by moon from satellite";
                geometric_restricted_zone_moon_end_time:standard_name = "time";
                geometric_restricted_zone_moon_end_time:units = "seconds since 2000-01-01 00:00:00.0";
                geometric_restricted_zone_moon_end_time:precision = “1 millisecond”;
```

```
        geometric_restricted_zone_moon_end_time:_FillValue = NC_FILL_DOUBLE;

ubyte icdt_channel_pairs_id(number_of_icdt_channel_pairs,channel_pair);
        icdt_channel_pairs_id:long_name = "IR channel pairs evaluated for ICdT";

float icdt_temperatures(number_of_icdt_channel_pairs,number_of_icdt_temperatures);
        icdt_temperatures:long_name = "Interchannel temperature difference evaluation temperatures";
        icdt_temperatures:units = "Kelvin";
        icdt_temperatures:_FillValue = NC_FILL_FLOAT;

float icdt(number_of_icdt_channel_pairs,number_of_icdt_temperatures);
        icdt:long_name = "Interchannel temperature difference";
        icdt:unit = "Kelvin";
        icdt:_FillValue = NC_FILL_FLOAT;
        icdt:ancillary_variables = "icdt_temperatures icdt_channel_pairs_id";

boolean icdt_compliance;
        icdt_channels_compliance:long_name = "Compliance for interchannel temperature difference";

ubyte rppke_channel_pairs_id(number_of_rppke_channel_pairs,channel_pair);
        rppke_channel_pairs_id:long_name = "Channel pairs evaluated for relative pixel position knowledge error between channels";

float rppke_between_channels(number_of_rppke_channel_pairs);
        rppke_between_channels:long_name = "Relative pixel position knowledge error between channels";
        rppke_between_channels:_FillValue = NC_FILL_FLOAT;

boolean rppke_between_channels_compliance;
        rppke_between_channels_compliance:long_name = "Compliance for relative pixel position knowledge error between channels";

ubyte icra_channel_pairs_id(number_of_icra_channel_pairs,channel_pair);
        icra_channel_pairs_id:long_name = "Channel pairs evaluated for ICRA";

float icra(number_of_icra_channel_pairs,channel_pair);
        icra:long_name = "Inter-channel co-registration accuracy";
        icra:_FillValue = NC_FILL_FLOAT;

boolean icra_compliance;
        icra_compliance:long_name = "Compliance for inter-channel co-registration accuracy";

boolean repeat_cycle_start_compliance;
        repeat_cycle_start_compliancer:long_name = "Compliance for repeat cycle start";

boolean repeat_cycle_duration_compliance;
```

```
                        repeat_cycle_duration_compliance:long_name = "Compliance for repeat cycle duration";

                boolean swath_timeliness_compliance;
                        swath_timeliness_compliance:long_name = "Compliance for swath timeliness";

}//quality


} // data

group: state{

        group:instrument{

                types:
                        uint enum fci_mode_type (Observation = 0, Decontamination = 1, Refocusing = 2, VNIR_Calibration = 3); //TBC based on Level 0 definition

                variables:
                        fci_mode_type fci_mode;
                                fci_mode:long_name = "Mode of FCI instrument";

                        string level0_channels(number_of_l0_channels );
                                channels_present:long_name = "FCI level 0 data channels"

                        double repeat_cycle_start_time;
                                repeat_cycle_start_time:long_name = "Start time in UTC of repeat cycle";
                                repeat_cycle_start_time:standard_name = "time";
                                repeat_cycle_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                                repeat_cycle_start_time:precision = "1 millisecond";
                                repeat_cycle_start_time:_FillValue =  NC_FILL_DOUBLE;

                        ushort repeat_sequence_counter;
                                repeat_sequence_counter:long_name = "Repeat sequence counter";

                        ushort repeat_cycle_counter;
                                repeat_cycle_counter:long_name = "Repeat cycle counter since the last transition to operational mode";

                        ushort repeat_cycle_counter_in_repeat_sequence;
                                repeat_cycle_counter_in_repeat_sequence:long_name = "Repeat cycle counter in the current repeat sequence";

                        ushort repeat_sequence_id;
                                repeat_sequence_id:long_name = "Repeat sequence ID";
```

```
ushort repeat_cycle_type;
        repeat_cycle_type:long_name = "Repeat cycle type";

ushort scan_law_id;
        scan_law_id = "Scan Law Identifier";

boolean channel_on(number_of_l0_channels);
        channel_on:long_name = "Channel active";

double last_decontamination_start_time;
        last_decontamination_start_time:long_name = "Start time in UTC of most recent decontamination";
        last_decontamination_start_time:standard_name = "time";
        last_decontamination_start_time:units = "seconds since 2000-01-01 00:00:00.0";
        last_decontamination_start_time:precision = "1 millisecond";
        last_decontamination_start_time:_FillValue =  NC_FILL_DOUBLE;

double last_decontamination_end_time;
        last_decontamination_end_time:long_name = "End time in UTC of most recent decontamination";
        last_decontamination_end_time:standard_name = "time";
        last_decontamination_end_time:units = "seconds since 2000-01-01 00:00:00.0";
        last_decontamination_end_time:precision = "1 millisecond";
        last_decontamination_end_time:_FillValue =  NC_FILL_DOUBLE;

double last_detection_chain_parameter_change_time(number_of_l0_channels );
        last_detection_chain_parameter_change_time:long_name = "Time in UTC of last change in the detection chain parameters";
        last_detection_chain_parameter_change_time:standard_name = "time";
        last_detection_chain_parameter_change_time:units = "seconds since 2000-01-01 00:00:00.0";
        last_detection_chain_parameter_change_time:precision = "1 millisecond";
        last_detection_chain_parameter_change_time:_FillValue =  NC_FILL_DOUBLE;

double last_heated_black_body_calibration_time;
        last_heated_black_body_calibration_time:long_name = "Time in UTC of last heated black body calibration";
        last_heated_black_body_calibration_time:standard_name = "time";
        last_heated_black_body_calibration_time:units = "seconds since 2000-01-01 00:00:00.0";
        last_heated_black_body_calibration_time:precision = "1 millisecond";
        last_heated_black_body_calibration_time:_FillValue =  NC_FILL_DOUBLE;

double last_mnd_calibration_time;
        last_mnd_calibration_time:long_name = "Time in UTC of last metallic neutral density calibration";
        last_mnd_calibration_time:standard_name = "time";
        last_mnd_calibration_time:units = "seconds since 2000-01-01 00:00:00.0";
        last_mnd_calibration_time:precision = "1 millisecond";
```

```
                        last_mnd_calibration_time:_FillValue =  NC_FILL_DOUBLE;

}//instrument

group:processor{

dimensions:
        auxiliary_datasets;

        types:

                byte enum auxiliary_dataset_status{OK = 0, out_of_validity =1, missing =2};

                ubyte enum resampling_method_type ( // Options [TBD]
                                        TruncatedShannon8=0,
                                        TruncatedShannon16=1,
                                        BiCubicSpline=2,
                                        NUFTTiteration=3
                                        NearestNeighbour =4);

                ubyte enum weighting_function_type ( // Options [TBD]
                                None = 0,
                                Kaiser=1,
                                Hamming = 2);

                ubyte enum projection_type(Geostationary = 0); // Options [TBD]

        variables:

                string auxiliary_dataset_identifier(number_of_auxiliary_datasets);
                        auxiliary_dataset_identifier:long_name = "Identifier for auxiliary dataset"

                auxiliary_dataset_status auxiliary_dataset_status(number_of_auxiliary_datasets);
                        auxiliary_dataset_state:long_name = "Status of  auxiliary dataset"

                // Processing settings

                boolean radiance_linearization_enabled(number_of_l0_channels);
                        radiance_linearization_enabled:long_name = "Radiance linearization enabled in this dataset";

                boolean detector_equalization_enabled(number_of_l0_channels);
                        detector_equalization_enabled:long_name = "Detector equalization enabled in this dataset";
```

```
boolean mtf_adaptation_enabled(number_of_1c_channels);
        mtf_adaptation_enabled:long_name = "MTF adaptation enabled in this dataset";

boolean straylight_correction_enabled(number_of_10_channels);
        straylight_correction_enabled:long_name = "Straylight correction enabled in this dataset";

resampling_method_type resampling_method;
        resampling_method:long_name = "Selected resampling method";

weighting_function_type weighting_function;
        weighting_function:long_name = "Weighting function used with the selected resampling method";

// Processing history

boolean radiometric_warning(number_of_10_channels);
        radiometric_warning.long_name = "Radiometric calibration warning for the complete repeat cycle";

boolean geometric_warning(number_of_1c_channels);
        geometric_warning.long_name = "Geometric processing warning for the complete repeat cycle";

// Reference Grid Parameters

reference_grid_type reference_grid(number_of_reference_grids);;
        reference_grid:long_name="Reference grid identifier";

string reference_grid_identifier(number_of_reference_grids);
        reference_grid_indentifier:long_name = "Filename from which reference grid parameters have been read";

string reference_grid_earth_model;
        reference_grid_earth_model:long_name = "Earth model used for reference grid";

ushort reference_grid_version(number_of_reference_grids);
        reference_grid_version:long_name = "Version of reference grid parameters ";

projection_type reference_grid_projection;
        reference_grid_projection:long_name = "Projection used for reference grid";

double projection_origin_longitude;
        projection_origin_longitude:long_name = "Longitude of projection origin";
        projection_origin_longitude:units = "degrees";
        projection_origin_longitude:_FillValue = NC_FILL_DOUBLE;

double projection_origin_latitude;
```

```
            projection_origin_latitude:long_name = "Latitude of projection origin";
            projection_origin_latitude:units = "degrees";
            projection_origin_latitude:_FillValue = NC_FILL_DOUBLE;

        double reference_altitude;
            altitude:long_name = "Satellite reference altitude";
            altitude:units = "metres"

        float reference_grid_spatial_sampling_angle_ns(number_of_reference_grids);
            reference_grid_spatial_sampling_angle_ns:long_name = "Spatial sampling angle for each reference grid in North-South direction";
            reference_grid_spatial_sampling_angle_ns:units = "radians"

        float reference_grid_spatial_sampling_angle_ew(number_of_reference_grids);
            reference_grid_spatial_sampling_angle_ew:long_name = "Spatial sampling angle for each reference grid in East-West direction";
            reference_grid_spatial_sampling_angle_ew:units = "radians"

        double earth_polar_radius;
            earth_polar_radius:long_name = "Earth polar radius";
            earth_polar_radius:units = "metres"

        double earth_equatorial_radius;
            earth_equatorial_radius:long_name = "Earth equatorial radius";
            earth_equatorial_radius:units = "metres"

        uint reference_grid_number_of_columns(number_of_reference_grids);
            reference_grid_number_of_columns:long_name = "Number of columns in reference grid";

        uint reference_grid_number_of_rows(number_of_reference_grids);
            reference_grid_number_of_rows:long_name = "Number of rows in reference grid";

        double azimuth_angle_at_reference_grid_origin;
            azimuth_angle_at_reference_grid_origin:long_name = "Azimuth angle from the GEOS projection origin to the centre of the first reference grid column";
            azimuth_angle_at_reference_grid_origin:units="radians";

        double elevation_angle_at_reference_grid_origin;
            elevation_angle_at_reference_grid_origin:long_name = "Elevation angle from the GEOS projection origin to the centre of the first reference grid row ";
            elevation_angle_at_reference_grid_origin:units="radians";

} // processor

group: celestial{

    dimensions:
```

```
        sunglint_time = runtime_value;  // number of sunglint parameters

        moon_shadow_time = runtime_value; // number of solar eclipse parameters

variables:

        boolean sun_eclipse_by_earth;
                sun_eclipse_by_earth:lone_name ="Sun eclipse by Earth in this dataset";

        boolean sun_eclipse_by_moon;
                sun_eclipse_by_moon:lone_name ="Sun eclipse by Moon in this dataset";

        double eclipse_start_time;
                eclipse_start_time:long_name = "Start time of eclipse";
                eclipse_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                eclipse_start_time:precision = "1 millisecond";
                eclipse_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

        double eclipse_end_time;
                eclipse_end_time:long_name = "End time of eclipse";
                eclipse_end_time:units = "seconds since 2000-01-01 00:00:00.0";
                eclipse_end_time:precision = "1 millisecond";
                eclipse_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

        boolean moon_shadow;
                moon_shadow:long_name = "Moon shadow on the Earth occurs in this dataset";

        boolean sunglint;
                sunglint:long_name = "Sun glint possible in this dataset";

        double sunglint_time(sunglint_time);
                sunglint_time:long_name = "Time in UTC of sunglint parameters";
                sunglint_time:standard_name = "time";
                sunglint_time:units = "seconds since 2000-01-01 00:00:00.0";
                sunglint_time:precision = "1 millisecond";
                sunglint_time:_FillValue = NC_FILL_DOUBLE;

        double sunglint_centre_latitude(sunglint_time);
                sunglint_centre_latitude:units = "degrees";
                sunglint_centre_latitude:long_name = "Latitude of centre of sunglint circle";
                sunglint_centre_latitude:_FillValue = NC_FILL_DOUBLE;

        double sunglint_centre_longitude(sunglint_time);
```

```
                    sunglint_centre_longitude:units = "degrees";
                    sunglint_centre_longitude:long_name = "Longitude of centre of sunglint circle";
                    sunglint_centre_longitude:_FillValue = NC_FILL_DOUBLE;

            double sunglint_radius(sunglint_time);
                    sunglint_radius:units = "degrees";
                    sunglint_radius:long_name = "Radius of sunglint circle";
                    sunglint_radius:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_time(moon_shadow_time);
                    moon_shadow_time:units = "seconds since 2000-01-01 00:00:00.0";
                    moon_shadow_time:standard_name = "time";
                    moon_shadow_time:long_name = "Time in UTC of moon shadow on the Earth parameters";
                    moon_shadow_time:precision = "1 millisecond";
                    moon_shadow_time:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_centre_latitude(moon_shadow_time);
                    moon_shadow_centre_latitude:units = "degrees";
                    moon_shadow_centre_latitude:long_name = "Latitude of centre of the moon shadow circle";
                    moon_shadow_centre_latitude:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_centre_longitude(moon_shadow_time);
                    moon_shadow_centre_longitude:units = "degrees";
                    moon_shadow_centre_longitude:long_name = "Longitude of centre of moon shadow circle";
                    moon_shadow_centre_longitude:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_penumbra_radius(moon_shadow_time);
                    moon_shadow_penumbra_penumbra_radius:units = "degrees";
                    moon_shadow_penumbra_radius:long_name = "Radius of moon shadow penumbra circle";
                    moon_shadow_penumbra_radius:_FillValue = NC_FILL_DOUBLE;

            double moon_shadow_umbra_radius(moon_shadow_time);
                    moon_shadow_umbra_radius:units = "degrees";
                    moon_shadow_umbra_radius:long_name = "Radius of moon shadow umbra circle";
                    moon_shadow_umbra_radius:_FillValue = NC_FILL_DOUBLE;

} //celestial


group: platform{

        types:
            byte enum manoeuvre_type(None = 0, NSSK = 1, EWSK = 2, SR = 3, MU = 4);  // TBC
```

```
        byte enum reference_frame(GCRF=1, EME2000=2, ITRF2008=3, TDR = 4 TEME = 5, TOD = 6 , RTN =7);

variables:
        double recent_manoeuvre_time_window;
                recent_manoeuvre_time_window:long_name = "Recent manoeuvre time window";
                recent_manoeuvre_time_window: title = "Time window to search for a manoeuvre that starts before or during this dataset";
                recent_manoeuvre_time_window:units="seconds";
                recent_manoeuvre_time_window:_FILL_VALUE = NC_FILL_DOUBLE;

        boolean recent_maneouvre_found
                recent_maneouvre_found:long_name = "Recent or current manoeuvre found";
                recent_maneouvre_found:title = "Recent or current manoeuvre found in the recent manoeuvre time window";

        manoeuvre_type recent_manoeuvre_type;
                recent_manoeuvre_type:long_name = "Type of recent manoeuvre";

        double recent_manoeuvre_start_time;
                recent_manoeuvre_start_time:long_name = "Start time of recent manoeuvre";
                recent_manoeuvre_start_time:units = "seconds since 2000-01-01 00:00:00.0";
                recent_manoeuvre_start_time:precision = "1 millisecond";
                recent_manoeuvre_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

        double recent_manoeuvre_end_time;
                recent_manoeuvre_end_time:long_name = "End time of recent manoeuvre";
                recent_manoeuvre_end_time:units = "seconds since 2000-01-01 00:00:00.0";
                recent_manoeuvre_end_time:precision = "1 millisecond";
                recent_manoeuvre_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

        reference_frame recent_manoeuvre_reference_frame;
                reference_frame:long_name = "Reference frame for manoeuvre paramaters";

        double recent_manoeuvre_delta_vx
                recent_manoeuvre_delta_vx:long_name = "X component delta v for recent manoeuvre";
                recent_manoeuvre_delta_vx:units = "m/s";

        double recent_manoeuvre_delta_vy
                recent_manoeuvre_delta_vy:long_name = "Y component delta v for recent manoeuvre";
                recent_manoeuvre_delta_vy:units = "m/s";

        double recent_manoeuvre_delta_vz
                recent_manoeuvre_delta_vz:long_name = "Z component delta v for recent manoeuvre";
                recent_manoeuvre_delta_vz:units = "m/s";
```

```
double recent_manoeuvre_spacecraft_delta_mass;
        recent_manoeuvre_spacecraft_delta_mass:long_name = "Delta spacecraft mass for recent manoeuvre";
        recent_manoeuvre_delta_v:units = "g";

double upcoming_manoeuvre_time_window;
        upcoming_manoeuvre_time_window:long_name = "upcoming manoeuvre time window";
        upcoming_manoeuvre_time_window:title = "Time window to search for a manoeuvre that starts after this dataset";
        upcoming_manoeuvre_time_window:units="seconds";
        upcoming_manoeuvre_time_window:_FILL_VALUE = NC_FILL_DOUBLE;

boolean upcoming_maneouvre_found
        upcoming_maneouvre_found:long_name = "Upcoming manoeuvre found";
        upcoming_maneouvre_found:title = "Upcoming manoeuvre found in the upcoming manoeuvre time window";

manoeuvre_type upcoming_manoeuvre_type;
        upcoming_manoeuvre_type:long_name = "Type of upcoming manoeuvre";

double upcoming_manoeuvre_start_time;
        upcoming_manoeuvre_start_time:long_name = "Start time of upcoming manoeuvre";
        upcoming_manoeuvre_start_time:units = "seconds since 2000-01-01 00:00:00.0";
        upcoming_manoeuvre_start_time:precision = "1 millisecond";
        upcoming_manoeuvre_start_time:_FILL_VALUE = NC_FILL_DOUBLE;

double upcoming_manoeuvre_end_time;
        upcoming_manoeuvre_end_time:long_name = "End time of upcoming manoeuvre";
        upcoming_manoeuvre_end_time:units = "seconds since 2000-01-01 00:00:00.0";
        upcoming_manoeuvre_end_time:precision = "1 millisecond";
        upcoming_manoeuvre_end_time:_FILL_VALUE = NC_FILL_DOUBLE;

reference_frame upcoming_manoeuvre_reference_frame;
        reference_frame:long_name = "Reference frame for manoeuvre paramaters";

double upcoming_manoeuvre_delta_vx
        upcoming_manoeuvre_delta_vx:long_name = "X component delta v for upcoming manoeuvre";
        upcoming_manoeuvre_delta_vx:units = "m/s";

double upcoming_manoeuvre_delta_vy
        upcoming_manoeuvre_delta_vy:long_name = "Y component delta v for upcoming manoeuvre";
        upcoming_manoeuvre_delta_vy:units = "m/s";

double upcoming_manoeuvre_delta_vz
        upcoming_manoeuvre_delta_vz:long_name = "Z component delta v for upcoming manoeuvre";
```

```
                    upcoming_manoeuvre_delta_vz:units = "m/s";

                double upcoming_manoeuvre_spacecraft_delta_mass;
                    upcoming_manoeuvre_spacecraft_delta_mass:long_name = "Delta spacecraft mass for upcoming manoeuvre";
                    upcoming_manoeuvre_delta_v:units = "g";

        } //platform

    } // state

} //  root
```

A.2.2 Variable Description

A.1.1.12 Root Group

As per Appendix A.1.1.1.

A.1.1.13 Data Group

As per Appendix A.1.1.2.

A.1.1.14 Generic Channel Group

As per Appendix A.1.1.3

Name	Description
Dimensions	
number_of_rows	The number of rows in the Level 1c repeat cycle dataset this will equal either 5568, 11136 or 22272 data points for a full disc scan depending on the channel. For reduced coverage scans this will be according to the configured area.

A.1.1.15 Measured Group

Name	Description
Dimensions	
number_of_radiometric_noise_lut_steps	Number of steps in the radiometric noise Look Up Table (LUT). Configured value. Default = 1024
number_of_radiometric _accuracy_lut_steps	Number of steps in the radiometric accuracy Look Up Table (LUT). Configured value. Default = 1024
number_of_radiance_to bt_conversion_coefficients	Number of coefficients used in the radiance to brightness temperature conversion formula
number_of_radiance_to bt_conversion_constants	Number of constants used in the radiance to brightness temperature conversion formula
Variables	
radiometric_noise_lut_noise	The radiometric noise Look Up Table (LUT) provides the output of a radiometric noise model for each of the effective radiance code words given in the radiometric_noise_lut_radiance variable.

Name	Description
	The LUT is an average for all the positions in the image and does not include stray light effects. The LUT is updated based on the measurements of deep space, black body and MND. The LUT consists of a series of points giving as the radiometric noise for a given radiance. The points may not necessarily be equally space in terms of radiance. To find the radiometric noise for a given effective_radiance value the LUT may need to be interpolated. The attribute valid_range for radiometric_accuracy_lut_radiance is the same as that for effective_radiance in the FCI Level 1c Body.
radiometric_noise_lut_radiance	See radiometric_noise_lut_noise description
radiometric_accuracy_lut_accuracy	The radiometric accuracy Look Up Table (LUT) provides the output of a radiometric accuracy model for each of the effective radiance code words given in the radiometric_accuracy_lut_radiance variable. The LUT is an average for all the positions in the image and does not include stray light effects. The LUT is updated based on the measurements of deep space, black body and MND. The LUT consists of a series of points giving as the radiometric accuracy for a given radiance. The points may not necessarily be equally space in terms of radiance. To find the radiometric accuracy for a given effective_radiance value the LUT may need to be interpolated. The attribute valid_range for radiometric_accuracy_lut_radiance is the same as that for effective_radiance in the FCI Level 1c Body.
radiometric_accuracy_lut_radiance	See radiometric_accuracy_lut_accuracy description
radiance_unit_conversion_coefficient	Conversion coefficients to convert radiance units. Details of use to be given in User Guide.
radiance_to_bt_conversion_coefficients	Conversion coefficients to convert radiance to brightness temperature to be used in the calculation of brightness temperature for IR spectral channels. Variable is set to _FillVaue for VNIR spectral channels.

Name	Description
radiance_to bt_conversion_constants	Conversion constants to convert radiance to brightness temperature to be used in the calculation of brightness temperature for IR spectral channels. Variable is set to _FillVaue for VNIR spectral channels.
channel_effective_solar_irradiance	Channel effective solar irradiance at 1 AU to be used in the derivation of the reflectance for VNIR spectral channels. Variable is set to _FillVaue for IR spectral channels.

A.1.1.16 Swath Group

Name	Description
Group Attributes	
long_name	Group description " Swath related information"
Dimensions	
number_of_swaths	Number of swaths in the dataset
number_of_swath_boundaries	Number of swaths boundaries in the dataset
Variables	
number_of_samples	Number of earth samples identified for the swath in the input level 0 dataset, i.e. samples whose line-of-sight intersects the earth ellipsoid.
number_of_missing_samples	Number of earth samples that are missing from the input level 0 dataset
number_of_oversaturated_samples	Number of earth samples that are over-saturated in the input level 0 dataset
number_of_undersaturated_samples	Number of earth samples that are under-saturated in the input level 0 dataset
number_of_extended_dynamic_range_samples	Number of earth samples for IR3.8 input level 0 datasets where fire radiometric range samples have replaced those from the normal radiometric range due to saturation of the normal radiometric range. Set to zero for all other spectral channels.
swath_coverage_compliance	Compliance to swath coverage requirement [SRD] FCI-05330 for the swath between the current and last repeat cycles. TRUE indicates compliance.
swath_overlap_compliance	Compliance to swath overlap requirement [SRD] FCI-05300 for current swath to next swath in the northerly direction. TRUE indicates compliance.
interswath_navigation_error	Calculated interswath navigation error evaluated at 95.45% confidence level
interswath_navigation_compliance	Compliance to interswath navigation error requirement [SRD] FCI-06470 for current swath to next swath in the northerly direction. TRUE indicates compliance.

Name	Description
sun_earth_distance_start	Distance from the centre of the sun to the centre of the earth at the swath midpoint

A.1.1.17 Quality Group

Name	Description
Group Attributes	
long_name	Group description "Quality indicators at data level"
Dimensions	
channel_pair	Dimension for a channel pair
number_of_icdt_channel_pairs	Number of channel pairs evaluated for the Inter-channel temperature difference (ICdT) requirement [FDHSI default = 28, HRFI default = 2]
number_of_icdt_temperatures	Number of temperatures evaluated for the Inter-channel temperature difference (ICdT) requirement [Default 5]
number_of_rppke_channel_pairs	Number of channel pairs evaluated for the Relative Pixel Position Error (RPPKE) between channels (RPPKE) [FDHSI default = 64, HRFI default = 4]
number_of_icra_channel_pairs	Number of channel pairs evaluated for the HRFI Inter-channel co-registration accuracy(ICRA) [FDHSI default = 56, HRFI default = 2]
Variables	
geometric_restricted_zone_earth_applicable	TRUE indicates that the sun is in the geometric restricted zone associated to the time around the eclipse of the sun by the Earth when viewed from the satellite, during a portion of the repeat cycle and geometric requirements relaxation applies ([SRD] FCI-06620)
geometric_restricted_zone_earth_start_time	Time in UTC when the sun enters the geometric restricted zone associated to the time around the eclipse of the sun by the Earth when viewed from the satellite
geometric_restricted_zone_earth_end_time	Time in UTC when the sun leaves the geometric restricted zone associated to the time around the eclipse of the sun by the Earth when viewed from the satellite
geometric_restricted_zone_moon_applicable	TRUE indicates that the sun is in the geometric restricted zone associated to the time around the eclipse of the sun by the moon when viewed from the satellite, during a portion of the repeat cycle and geometric requirements relaxation applies ([SRD] FCI-06620)
geometric_restricted_zone_moon_start_time	Time in UTC when the sun enters the geometric restricted zone associated to the time around the eclipse of the sun by the moon when viewed from the satellite
geometric_restricted_zone_moon_end_time	Time in UTC when the sun leaves the geometric restricted zone associated to the time around the eclipse of the sun by the moon when viewed from the satellite

Name	Description
icdt_channel_pairs_id	Identification of channel pairs evaluated for interchannel temperature different (ICdT). The first dimension corresponds to the same dimension as the dimension for the variable icdt, the second dimension identifies a channel pair corresponding to an index selection from the variable l1c_channels_present
icdt_temperatures	Temperatures selected for the interchannel temperature different (ICdT) evaluation
icdt	Values of interchannel temperature different (ICdT) matching the channel pairs in icdt_channel_pairs_id. NC_FILL_FLOAT indicates an unavailable comparison.
icdt_compliance	Overall status of compliance to the interchannel temperature different (ICdT) between infrared channels. TRUE indicates compliance. ([SRD] FCI-05870)
rppke_channel_pairs_id	Identification of channel pairs evaluated for relative pixel position knowledge error (RPPKE). The first dimension corresponds to the same dimension as the dimension for the variable rppke_between_channels, the second dimension identifies a channel pair corresponding to an index selection from the variable l1c_channels_present
rppke_between_channels	Values of relative pixel position knowledge error (RPPKE) between channels of dissimilar sampling distance. NC_FILL_FLOAT indicates an unavailable comparison.
rppke_between_channels_compliance	Overall status of compliance to the relative pixel position knowledge error between channels. TRUE indicates compliance. ([SRD] FCI-06560)
icra_channel_pairs_id	Identification of channel pairs evaluated for the interchannel co-registration accuracy (ICRA). The first dimension corresponds to the same dimension as the dimension for the variable icra, the second dimension identifies a channel pair corresponding to an index selection from the variable l1c_channels_present
icra	Values of HRFI interchannel corregistration accuracy between channels of the same sampling distance. NC_FILL_ FLOAT indicates an unavailable comparison
icra_compliance	Overall status of interchannel corregistration accuracy between channels. TRUE indicates compliance. ([SRD] FCI-06530)
repeat_cycle_start_compliance	The repeat cycle has started within the allowed margins of the required time. TRUE indicates compliance.([SRD] FCI-05420)
repeat_cycle_duration_com	The repeat cycle has the correct duration within the

Name	Description
pliance	allowed limits. TRUE indicates compliance.
repeat_cycle_timing_compliance	The repeat cycle has the correct timing within the allowed limits. TRUE indicates compliance. ([SRD] FCI-05450)
swath_timeliness_compliance	All swath data meets the timeliness requirement ([SRD] FCI-05480)

A.1.1.18 Quality_Channel Group

Name	Description
Group Attributes	
long_name	Group description "Quality indicators applicable to a particular channel for the repeat cycle"
Types	
channel_status_type	Status of the channel. NOMINAL, NON-NOMINAL]
Variables	
channel_status	Status of the channel. Defined by the compliance status of the channels to its overall requirements.
number_of_expected_earth_pixels	Number of earth pixels that are expected be in the nominal dataset
number_of_masked_pixels	Number of space pixels that have been masked.
number_of_missing_earth_pixels	Number of Earth pixels flagged as missing
number_of_oversaturated_earth_pixels	Number of earth pixels that are over-saturated in dataset (set to saturated flag value)
number_of_undersaturated_earth_pixels	Number of earth pixels that are under-saturated in dataset (set to under-saturated flag value)
number_of_noise_warning_pixels	Number of Earth pixels with noise_warning flag set
number_of_radiometric_warning_pixels	Number of Earth pixels with radiometric_warning flag set
number_of_saturation_warning_pixels	Number of Earth pixels with saturation_warning flag set
number_of_missing_warning_pixels	Number of Earth pixels with missing_warning flag set
number_of_straylight_warning_pixels	Number of Earth pixels with straylight_warning flag set
number_of_straylight_correction_warning_pixels	Number of Earth pixels with straylight_correction_warning flag set
number_of_extended_dynamic_range_warning_pixels	Number of Earth pixels with extended_dynamic_range_warning flag set
number_of_encoding_saturation_warning_pixels	Number of Earth pixels with encoding_saturation_warning flag set
percentage_coverage_achie	Percentage of earth coverage achieved.

Name	Description
ved	
completeness_compliance	Flag to indicate if image has passed the completeness requirement ([SRD] FCI-05360). TRUE indicates compliance.
accuracy_compliance	Flag to indicate if image has passed the accuracy requirement ([SRD] FCI-05390). TRUE indicates compliance
coverage_compliance	Status of compliance to the coverage requirement ([SRD] FCI-05270). TRUE indicates compliance.
radiometric_restricted_zone_applied	If TRUE indicates that some of the pixels created during the repeat cycle lie within the radiometric restricted zone around the sun ([SRD] FCI-06650), implying a relaxation in the radiometric requirements for those pixels.
sse_compliance	Status of compliance to the spatial sampling error requirement for the whole image ([SRD] FCI-06170). TRUE indicates compliance.
absolute_pixel_position_knowledge_error	Estimate of the absolute pixel position knowledge error for the whole image
absolute_pixel_position_knowledge_error_compliance	Status of compliance to the absolute pixel position knowledge error requirement for the whole image ([SRD] FCI-06380). TRUE indicates compliance.
absolute_pixel_position_knowledge_error_500	Estimate of the absolute pixel position knowledge error within 500 by 500 pixel imagette
absolute_pixel_position_knowledge_error_compliance_500	Status of compliance to the absolute pixel position knowledge error requirement within 500 by 500 pixel imagettes ([SRD] FCI-06410). TRUE indicates compliance.
absolute_pixel_position_knowledge_error_100	Estimate of the absolute pixel position knowledge error within 100 by 100 pixel imagette
absolute_pixel_position_knowledge_error_compliance_100	Status of compliance to the absolute pixel position knowledge error requirement within 100 by 100 pixel imagettes ([SRD] FCI-06440). TRUE indicates compliance.
relative_pixel_position_knowledge_error	Estimate of the relative pixel position knowledge error relative to last repeat cycle
relative_pixel_position_knowledge_error_compliance	Status of compliance to the relative pixel position knowledge error requirement relative to last repeat cycle requirement ([SRD] FCI-06500). TRUE indicates compliance.
radiometric_noise_compliance	Status of compliance to the radiometric noise requirement ([SRD] FCI-05690). TRUE indicates compliance.
noise_power_spectral_density_compliance	Status of compliance to the noise power spectral density requirement ([SRD] FCI-05720). TRUE indicates compliance.

Name	Description
radiometric_accuracy_compliance	Status of compliance to the radiometric accuracy requirement ([SRD] FCI-05840). TRUE indicates compliance.

A.1.1.19 External_Calibration_Coefficients Group

Name	Description
Group Attributes	
long_name	Description of group "Calibration coefficients for FCI generated calibration derived from external means"
subsettable	Group can be included or excluded from the dataset according to configured selection
Dimensions	
number_of_external_calibration_coefficients	Number of polynomial correction coefficients
min_max	Dimensions for defining minimum and maximum of a range
Variables	
external_calibration_filenames	File name from which External Calibration Coefficients were read
external_calibration_update_time	Time in UTC of the last update of External Calibration Correction parameters
validity_period	Minimum and maximum times in UTC over which the External Calibration Coefficients are valid
external_calibration_coefficients_valid	External calibration coefficients valid for the current repeat cycle
external_calibration_coefficients	Coefficients for External Calibration Correction polynomial correction with first value = 0^{th} order coefficient, second value = 1^{st} order coefficient, etc.
external_calibration_covariance matrix	Covariance matrix for the External Calibration Correction polynomial correction coefficients
radiance_validity_range	Minimum and maximum radiance for which the External Calibration Correction coefficients are valid

A.1.1.20 State Group

As per Appendix A.1.1.7.

A.1.1.21 Instrument Group

As per Appendix A.1.1.8.

A.1.1.22 Platform Group

As per Appendix A.1.1.9.

A.1.1.23 Processor Group

As per Appendix A.1.1.10.

A.1.1.24 Celestial Group

As per Appendix A.1.1.11.

APPENDIX B NETCDF AND APPLICABLE STANDARDS AND CONVENTIONS

B.1 netCDF

The FCI L1C datasets are netCDF-4 files and use the enhanced data model. In addition, they utilise the Hierarchical Data Format version 5 (HDF5) as the storage layer and so can also be read as HDF-5 files.

Use of the enhanced netCDF-4 data model allows groups to be created to aid with the natural collection of various data and the subsetting of channels. In additions, enumerated variable types allow flags to be defined once and used throughout the dataset.

Also, the use of the HDF-5 data layer allows the use of the additional compression functionality as described in Section 7.9.

B.2 CF Conventions

The current Climate and Forecast Conventions (CF 1.6) are applicable to version 3 of the netCDF data model. As such, the FCI L1C datasets cannot conform terms of the conventions although they do try to follow the spirit of the conventioons as far as possible. The draft CF 1.7 document is also netCDF-3 specific, but there are plnas to create a CF-2 document to cover the enhanced netCDF-4 model.

B.3 NetCDF Attribute Convention for Dataset Discovery

The table below shows the conformance of the MTG products to the NetCDF Attribute Convention for Dataset Discovery [NACDD]. The datasets are conformant with all the Highly Recommended attributes and the majority of the recoemmended attributes that are applicable to the datasets.

Attribute	Description	Present	Contents
Highly Recommended			
title	A short phrase or sentence describing the dataset. In many discovery systems, the title will be displayed in the results list from a search, and therefore should be human readable and reasonable to display in a list of such names. This attribute is also recommended by the NetCDF Users Guide and the CF conventions.	Yes	product_name
summary	A paragraph describing the dataset, analogous to an abstract for a paper.	Yes	Should be set per FS document

keywords	A comma-separated list of key words and/or phrases. Keywords may be common words or phrases, terms from a controlled vocabulary (GCMD is often used), or URIs for terms from a controlled vocabulary (see also "keywords_vocabulary" attribute).	Yes	Should be set per FS document
Conventions	A comma-separated list of the conventions that are followed by the dataset. For files that follow this version of ACDD, include the string 'ACDD-1.3'. (This attribute is described in the NetCDF Users Guide.)	Yes	
Recommended			
id	An identifier for the data set, provided by and unique within its naming authority. The combination of the "naming authority" and the "id" should be globally unique, but the id can be globally unique by itself also. IDs can be URLs, URNs, DOIs, meaningful text strings, a local key, or any other unique string of characters. The id should not include white space characters.	No	
naming_authority	The organization that provides the initial id (see above) for the dataset. The naming authority should be uniquely specified by this attribute. We recommend using reverse-DNS naming for the naming authority; URIs are also acceptable. Example: 'edu.ucar.unidata'.	No	
history	Provides an audit trail for modifications to the original data. This attribute is also in the NetCDF Users Guide: 'This is a character array with a line for each invocation of a program that has modified the dataset. Well-behaved generic netCDF applications should append a line containing: date, time of day, user name, program name and command arguments.' To include a more complete description you can append a reference to an ISO Lineage entity; see NOAA EDM ISO Lineage guidance.	Yes	"original generated file"

source	The method of production of the original data. If it was model-generated, source should name the model and its version. If it is observational, source should characterize it. This attribute is defined in the CF Conventions. Examples: 'temperature from CTD #1234'; 'world model v.0.1'.	Yes	As particularised in the relevant dataset/product format specification, an array of strings of the form: (PROCESSOR_FULL_NAME) (CONFIGURATION_FILE_NAME)* (INPUT_DATASET/PRODUCT_NAME)* where the asterisks indicate zero or more instances.
processing_level	A textual description of the processing (or quality control) level of the data.	Yes	As per the dataset name field “level” in Table 3 1
comment	Miscellaneous information about the data, not captured elsewhere. This attribute is defined in the CF Conventions.	Yes	Unless otherwise specified in the relevant dataset/product format specification, “None.”
acknowledgement	A place to acknowledge various types of support for the project that produced this data.	No	
license	Provide the URL to a standard or specific license, enter "Freely Distributed" or "None", or describe any restrictions to data access and distribution in free text.	No	
standard_name_vocabulary	The name and version of the controlled vocabulary from which variable standard names are taken. (Values for any standard_name attribute must come from the CF Standard Names vocabulary for the data file or product to comply with CF.) Example: 'CF Standard Name Table v27'.	No	
date_created	The date on which this version of the data was created. (Modification of values implies a new version, hence this would be assigned the date of the most recent values modification.) Metadata changes are not considered when assigning the date_created. The ISO 8601:2004 extended date format is recommended, as described in the	No	However, we do have "processing_time" = the time of the formatting of the dataset/product by the processor.

	Attribute Content Guidance section.		
creator_name	The name of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	No	
creator_email	The email address of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	No	
creator_url	The URL of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	No	
institution	The name of the institution principally responsible for originating this data. This attribute is recommended by the CF convention.	Yes	"EUMETSAT". This field may be extended with other values should datasets/products be generated in other locations.
project	The name of the project(s) principally responsible for originating this data. Multiple projects can be separated by commas, as described under Attribute Content Guidelines. Examples: 'PATMOS-X', 'Extended Continental Shelf Project'.	No	
publisher_name	The name of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	No	
publisher_email	The email address of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	No	

publisher_url	The URL of the person (or other entity specified by the publisher_type attribute) responsible for publishing the data file or product to users, with its current metadata and format.	No	
geospatial_bounds	Describes the data's 2D or 3D geospatial extent in OGC's Well-Known Text (WKT) Geometry format (reference the OGC Simple Feature Access (SFA) specification). The meaning and order of values for each point's coordinates depends on the coordinate reference system (CRS). The ACDD default is 2D geometry in the EPSG:4326 coordinate reference system. The default may be overridden with geospatial_bounds_crs and geospatial_bounds_vertical_crs (see those attributes). EPSG:4326 coordinate values are latitude (decimal degrees_north) and longitude (decimal degrees_east), in that order. Longitude values in the default case are limited to the [-180, 180) range. Example: 'POLYGON ((40.26 -111.29, 41.26 -111.29, 41.26 -110.29, 40.26 -110.29, 40.26 -111.29))'.	No	
geospatial_bounds_crs	The coordinate reference system (CRS) of the point coordinates in the geospatial_bounds attribute. This CRS may be 2-dimensional or 3-dimensional, but together with geospatial_bounds_vertical_crs, if that attribute is supplied, must match the dimensionality, order, and meaning of point coordinate values in the geospatial_bounds attribute. If geospatial_bounds_vertical_crs is also present then this attribute must only specify a 2D CRS. EPSG CRSs are strongly recommended. If this attribute is not specified, the CRS is assumed to be EPSG:4326. Examples: 'EPSG:4979' (the 3D WGS84 CRS), 'EPSG:4047'.	No	

geospatial_bounds_vertical_crs	The vertical coordinate reference system (CRS) for the Z axis of the point coordinates in the geospatial_bounds attribute. This attribute cannot be used if the CRS in geospatial_bounds_crs is 3-dimensional; to use this attribute, geospatial_bounds_crs must exist and specify a 2D CRS. EPSG CRSs are strongly recommended. There is no default for this attribute when not specified. Examples: 'EPSG:5829' (instantaneous height above sea level), "EPSG:5831" (instantaneous depth below sea level), or 'EPSG:5703' (NAVD88 height).	No	
geospatial_lat_min	Describes a simple lower latitude limit; may be part of a 2- or 3-dimensional bounding region. Geospatial_lat_min specifies the southernmost latitude covered by the dataset.	No	
geospatial_lat_max	Describes a simple upper latitude limit; may be part of a 2- or 3-dimensional bounding region. Geospatial_lat_max specifies the northernmost latitude covered by the dataset.	No	
geospatial_lon_min	Describes a simple longitude limit; may be part of a 2- or 3-dimensional bounding region. geospatial_lon_min specifies the westernmost longitude covered by the dataset. See also geospatial_lon_max.	No	
geospatial_lon_max	Describes a simple longitude limit; may be part of a 2- or 3-dimensional bounding region. geospatial_lon_max specifies the easternmost longitude covered by the dataset. Cases where geospatial_lon_min is greater than geospatial_lon_max indicate the bounding box extends from geospatial_lon_max, through the longitude range discontinuity meridian (either the antimeridian for -180:180 values, or Prime Meridian for 0:360 values), to geospatial_lon_min; for example, geospatial_lon_min=170 and geospatial_lon_max=-175 incorporates 15 degrees of longitude (ranges 170 to 180 and -180 to -175).	No	

geospatial_vertical_min	Describes the numerically smaller vertical limit; may be part of a 2- or 3-dimensional bounding region. See geospatial_vertical_positive and geospatial_vertical_units.	No	
geospatial_vertical_max	Describes the numerically larger vertical limit; may be part of a 2- or 3-dimensional bounding region. See geospatial_vertical_positive and geospatial_vertical_units.	No	
geospatial_vertical_positive	One of 'up' or 'down'. If up, vertical values are interpreted as 'altitude', with negative values corresponding to below the reference datum (e.g., under water). If down, vertical values are interpreted as 'depth', positive values correspond to below the reference datum. Note that if geospatial_vertical_positive is down ('depth' orientation), the geospatial_vertical_min attribute specifies the data's vertical location furthest from the earth's center, and the geospatial_vertical_max attribute specifies the location closest to the earth's center.	No	
time_coverage_start	Describes the time of the first data point in the data set. Use the ISO 8601:2004 date format, preferably the extended format as recommended in the Attribute Content Guidance section.	No	But we do have start_time
time_coverage_end	Describes the time of the last data point in the data set. Use ISO 8601:2004 date format, preferably the extended format as recommended in the Attribute Content Guidance section.	No	But we do have end_time
time_coverage_duration	Describes the duration of the data set. Use ISO 8601:2004 duration format, preferably the extended format as recommended in the Attribute Content Guidance section.	No	
time_coverage_resolution	Describes the targeted time period between each value in the data set. Use ISO 8601:2004 duration format, preferably the extended format as recommended in the Attribute Content Guidance section.	No	
	Suggested		

creator_type	Specifies type of creator with one of the following: 'person', 'group', 'institution', or 'position'. If this attribute is not specified, the creator is assumed to be a person.	No	
creator_institution	The institution of the creator; should uniquely identify the creator's institution. This attribute's value should be specified even if it matches the value of publisher_institution, or if creator_type is institution.	No	
publisher_type	Specifies type of publisher with one of the following: 'person', 'group', 'institution', or 'position'. If this attribute is not specified, the publisher is assumed to be a person.	No	
publisher_institution	The institution that presented the data file or equivalent product to users; should uniquely identify the institution. If publisher_type is institution, this should have the same value as publisher_name.	No	
program	The overarching program(s) of which the dataset is a part. A program consists of a set (or portfolio) of related and possibly interdependent projects that meet an overarching objective. Examples: 'GHRSST', 'NOAA CDR', 'NASA EOS', 'JPSS', 'GOES-R'.	No	
contributor_name	The name of any individuals, projects, or institutions that contributed to the creation of this data. May be presented as free text, or in a structured format compatible with conversion to ncML (e.g., insensitive to changes in whitespace, including end-of-line characters).	No	
contributor_role	The role of any individuals, projects, or institutions that contributed to the creation of this data. May be presented as free text, or in a structured format compatible with conversion to ncML (e.g., insensitive to changes in whitespace, including end-of-line characters). Multiple roles should be presented in the same order and number as the names in contributor_names.	No	

geospatial_lat_units	Units for the latitude axis described in "geospatial_lat_min" and "geospatial_lat_max" attributes. These are presumed to be "degree_north"; other options from udunits may be specified instead.	No	
geospatial_lat_resolution	Information about the targeted spacing of points in latitude. Recommend describing resolution as a number value followed by the units. Examples: '100 meters', '0.1 degree'	No	
geospatial_lon_units	Units for the longitude axis described in "geospatial_lon_min" and "geospatial_lon_max" attributes. These are presumed to be "degree_east"; other options from udunits may be specified instead.	No	
geospatial_lon_resolution	Information about the targeted spacing of points in longitude. Recommend describing resolution as a number value followed by units. Examples: '100 meters', '0.1 degree'	No	
geospatial_vertical_units	Units for the vertical axis described in "geospatial_vertical_min" and "geospatial_vertical_max" attributes. The default is EPSG:4979 (height above the ellipsoid, in meters); other vertical coordinate reference systems may be specified. Note that the common oceanographic practice of using pressure for a vertical coordinate, while not strictly a depth, can be specified using the unit bar. Examples: 'EPSG:5829' (instantaneous height above sea level), 'EPSG:5831' (instantaneous depth below sea level).	No	
geospatial_vertical_resolution	Information about the targeted vertical spacing of points. Example: '25 meters'	No	
date_modified	The date on which the data was last modified. Note that this applies just to the data, not the metadata. The ISO 8601:2004 extended date format is recommended, as described in the Attributes Content Guidance section.	No	

date_issued	The date on which this data (including all modifications) was formally issued (i.e., made available to a wider audience). Note that these apply just to the data, not the metadata. The ISO 8601:2004 extended date format is recommended, as described in the Attributes Content Guidance section.	No	
date_metadata_modified	The date on which the metadata was last modified. The ISO 8601:2004 extended date format is recommended, as described in the Attributes Content Guidance section.	No	
product_version	Version identifier of the data file or product as assigned by the data creator. For example, a new algorithm or methodology could result in a new product_version.	No	We have other version control fields
keywords_vocabulary	If you are using a controlled vocabulary for the words/phrases in your "keywords" attribute, this is the unique name or identifier of the vocabulary from which keywords are taken. If more than one keyword vocabulary is used, each may be presented with a prefix and a following comma, so that keywords may optionally be prefixed with the controlled vocabulary key. Example: 'GCMD:GCMD Keywords, CF:NetCDF COARDS Climate and Forecast Standard Names'.	No	
platform	Name of the platform(s) that supported the sensor data used to create this data set or product. Platforms can be of any type, including satellite, ship, station, aircraft or other. Indicate controlled vocabulary used in platform_vocabulary.	No	We have "spacecraft"
platform_vocabulary	Controlled vocabulary for the names used in the "platform" attribute.	No	
instrument	Name of the contributing instrument(s) or sensor(s) used to create this data set or product. Indicate controlled vocabulary used in instrument_vocabulary.	No	We have "data source"
instrument_vocabulary	Controlled vocabulary for the names used in the "instrument" attribute.	No	

cdm_data_type	The data type, as derived from Unidata's Common Data Model Scientific Data types and understood by THREDDS. (This is a THREDDS "dataType", and is different from the CF NetCDF attribute 'featureType', which indicates a Discrete Sampling Geometry file in CF.)	No	
metadata_link	A URL that gives the location of more complete metadata. A persistent URL is recommended for this attribute.	No	
references	Published or web-based references that describe the data or methods used to produce it. Recommend URIs (such as a URL or DOI) for papers or other references. This attribute is defined in the CF conventions.	Yes	"www.eumetsat.int" Note: It is intended that users of the dataset/product can access published, web-based references describing the data and the methods used to produce it at this address.
Highly Recommended Variable Attributes			
long_name	A long descriptive name for the variable (not necessarily from a controlled vocabulary). This attribute is recommended by the NetCDF Users Guide, the COARDS convention, and the CF convention.	Usually Yes	
standard_name	A long descriptive name for the variable taken from a controlled vocabulary of variable names. We recommend using the CF convention and the variable names from the CF standard name table. This attribute is recommended by the CF convention.	Usually Yes	
units	The units of the variable's data values. This attribute value should be a valid udunits string. The "units" attribute is recommended by the NetCDF Users Guide, the COARDS convention, and the CF convention.	Usually Yes	
coverage_content_type	An ISO 19115-1 code to indicate the source of the data (image, thematicClassification, physicalMeasurement, auxiliaryInformation,	No	

	qualityInformation, referenceInformation, modelResult, or coordinate).		

APPENDIX C NETCDF TOOLS

C.1 Overview

The MTG netCDF datasets make use of a number of features of the enhanced netCDF-4 data model, including groups, unsigned integer data types and enumerated data types. Not all netCDF tools are capable of utilizing enhanced netCDF-4 datasets. However, the netCDF-4 files also use HDF-5 as the data layer, and so the datasets may also be examined with HDF-5 tools.

This Appendix lists freely available tools that are known to be compatible with the MTG netCDF-4 datasets.

This is not an exhaustive list as other tools and libraries may also be compatible with the enhanced netCDF-4 model, or may be updated to be so in future.

C.2 netCDF Libraries and Tools

The latest version of the netCDF (Network Common Data Form) libraries should be installed. At a minimum, netCDF 4.3.3.1 is required for writing datasets in the MTG format.

netCDF libraries are being developed by Unidata, a member of the UCAR Comunity Programs. Libraries can be downloaded from their webpage:

`http://www.unidata.ucar.edu`

The netCDF distribution provides a number of command line tools for looking at the structure and contents of netCDF datasets.

HDF-5 and gzip need to be installed before netCDF.

C.2.1 gzip

Gzip is used as the internal compression tool for the MTG netCDF-4 datasets. The gzip libraries need to be installed before installing HDF-5.

C.2.2 HDF-5

HDF-5 (Hierarchical Data Format, version 5) is used as the storage layer for the MTG netCDF-4 datasets. The HDF-5 libraries need to be compiled before installing netCDF-4.

HDF 5 is being developed by The HDF Group, The latest libraries can be downloaded from their webpage:

https://www.hdfgroup.org

C.2.3 FCI Decompressor

Details are [TBD]. This is the EUMETSAT-provided tool required to decompress disseminated FCI-1C-RRAD files that have radiances compressed using charLS compression.

C.3 Panoply

Panoply is a freely available, cross-platform java application that provides as GUI for browsing and plotting geo-gridded and other arrays from netCDF datasets. It can also handle other formats such as GRIB, HDF, etc. It is supported by NASA and is available from:

http://www.giss.nasa.gov/tools/panoply/

As it is implemented in Java, it provides the same GUI in different operating systems and does not require administrative or root privileges to install.

It can display the CDL description as well as images, and makes use of many of the CF conventions. For instance, it converts integer counts from the effective_radiance variable to float numbers in the images using the scale_factor and offset variable attributes.

C.4 HDFView

HDFView is a freely available, cross-platform java application with a GUI for browsing and editing HDF4 and HDF5 files.

It is available from:

http://www.hdfgroup.org/products/java/hdfview/