L2 RMIB GERB Products
User Guide

Steven Dewitte Gilles Sadowski Luis Gonzalez

November 25, 2002

Contents

List of Tables

List of Figures

CHANGE RECORD

Issue	Date	Changed by	Reason for change
Version 1	07-6-2000		New document
Version 1.1	08-9-2000	G. Sadowski	Corrections and additions following the recommendations of the review board (δ Review 06-9-2000)
Version 1.2	03-6-2002	L. Gonzalez	Radiance and flux on same file except for ARG
	25-11-2002	L. Gonzalez	Rewriting of chapter "L2 RMIB GERB Products Data Acess" using HDF5 Lite API
	25-11-2002	L. Gonzalez	introduction of MSG7 as possible imager
	25-11-2002	L. Gonzalez	addition of "Summary Thermal Products Confidence", "Summary Solar Products Confidence", "/Radiometry/Longwave Correction/Minimum Correction Value", "/Radiometry/Shortwave Correction/Minimum Correction Value", "/Radiometry/Longwave Correction/Maximum Correction Value", " "/Radiometry/Shortwave Correction/Maximum Correction Value"
	25-11-2002	L. Gonzalez	"Mapped range" has been splitted on "Range" and "Offset"
	25-11-2002	L. Gonzalez	"Surface Type" has been modified

ACRONYMS

API	Application Programming Interface
ARG	Averaged Rectified Geolocated
BARG	Binned Averaged Rectified Geolocated
FTP	File Transfer Protocol
GERB	Geostationary Earth Radiation Budget
HDF	Hierarchical Data Format
HTTP	Hypertext Transfer Protocol
L1.5	Level 1.5
L2	Level 2
MSG	Meteosat Second Generation
NANRG	Non Averaged Non Rectified Geolocated
NCSA	National Center for Super-Computing Applications
RAL	Rutherford-Appleton Laboratory
RMIB	Royal Meteorological Institute of Belgium
ROLSS	RMIB On-Line Short-Term Services
SEVIRI	Spinning Enhanced Visible and Infrared Imager
SHI	Snapshot High Resolution Image
WWW	World Wide Web

1 Purpose of this Document

This guide is intended to assist users of the Level 2 (L2) products derived from the Geostationary Earth Radiation Budget (GERB) instrument data. These L2 products are computed at the Royal Meteorological Institute of Belgium (RMIB). The L2 RMIB GERB Products consist of images of unfiltered radiances and radiative fluxes at the top of the atmosphere, together with some auxiliary information. Unfiltered radiances and radiative fluxes are available for reflected solar radiation and for emitted thermal radiation.
The data format chosen for the L2 RMIB GERB Products is the Hierarchical Data Format (HDF) version 5. HDF is not a straightforward data file format in which data items are accessed from byte location. Instead data items in an HDF file format are accessed by name, using HDF library functions.
Documents detailing the scientific assumptions and algorithms used to derive the L2 RMIB GERB Products are available from our web site:

http://gerb.oma.be/

Any suggestion, information or correction can be sent to:

gerb@oma.be

This document is organised as follows:

- Section 2 gives an overview of each of the various L2 RMIB GERB Products.
- Section 3 provides a detailed description of the logical contents of the products.
- Section 4 describes how a user can write an application to read the L2 RMIB GERB Products.
- Appendix A contains sample code that can help the user devise his own reading programs for the L2 RMIB GERB Products.

2 L2 RMIB GERB Products Files

2.1 Introduction

This section introduces the various L2 RMIB GERB Products. There are eleven (11) of them, differing in contents and/or resolution. Each product is stored in a separate HDF file.

The products are divided in three main types. A product type is defined by a given temporal sampling and spatial resolution.

- An *Averaged Rectified Geolocated* (ARG) product is defined as a 15 minute average obtained by averaging three consecutive GERB measurements. The product has GERB spatial resolution (256x256), geolocated on the Rectified Grid computed by RAL (cf. L1.5 ARG product). The spatial shape of one pixel is the average of the three GERB footprints that have contributed to it.
- A *Snapshot High resolution Image* (SHI) product is defined as a snapshot at imager acquisition times. The product has 3x3 pixels spatial resolution (1237x1237 for SEVIRI and 833x833 for Meteosat).
- A *Binned Averaged Rectified Geolocated* (BARG) product is defined as an average over a time period. The period is an exact imager complete scan period [1]. The first average is computed centered on 00:00 UTC. The product has roughly GERB spatial resolution, i.e. the pixels are geolocated on a rectified grid and are an average of SHI pixels [2] from a high resolution image; the grid resolution is 247x247 for SEVIRI and 277x277 for Meteosat. The spatial shape of one pixel is an exact square at nadir. The dimension of this square is the GERB sampling distance, i.e. close to 50 km.

2.2 Name Convention

The file names of the L2 RMIB GERB Products are made up of several parts, separated by underscore characters, and an extension.

The first part indicates which GERB instrument produced the original raw data. The GERB 1 instrument is identified by

“`G1`”

The second part indicates which imager instrument produced the original raw data. Instrument identifiers are

“`SEV1`”

“`MS7`”

where the first one is for SEVIRI 1 and the last for Meteosat 7 imager.
The next identifier specifies the product type. For Level 2.0 products, this can be

[1] 15’ for SEVIRI and 30’ for Meteosat 7
[2] 5x5 for SEVIRI and 3x3 Meteosat 7

"L20A"

"L20S"

"L20L"

"L20C"

"L20G"

where the last letter, indicates that the product is concerned with either solar radiation "S", thermal radiation "L" or both "A" or geolocation information "G". In addition, the last letter "C" can be used for imager count images (TBC).
The next part(s) specifies the product subtype. This is one of the following:

"15M_50"

"30M_50"

"H"

where "15M"("30M") stands for exact 15(30)-minute bins average, "50" for GERB resolution, and "H" for high resolution.
An optional field indicates if the geographic region covered is different from the full Earth disk. Currently, this can be

"EUROPE"

A reference date is used to identify the successive files. The format of this reference date is <DATE>, i.e. four-digit year, two-digit month, two-digit day, an underscore character, two-digit hour, two-digit minute and two-digit second.
The last part specifies the product version. The first value is

"V001"

Finally, the HDF file format is indicated by the extension

".hdf[.gz]"

with optional gzip compression.

2.3 Products Files

Table 1 lists the name and product type of all eleven L2 RMIB GERB Products files (see section 2.2 about the file naming convention).
Each of the following sections is devoted to one specific L2 RMIB GERB product. The section title indicate the product file name as described in 2.2. For each product, a table presents the logical structure of the data it contains. The first column of the table gives the name of the data (see section 4 on how to use that name to access the data); the number in the second column is the page of this document where the user can find a complete description of the data.

File Name	Type
"<GERB_ID>_<IMAGER_ID>_L20L_<DATE>_<VERSION>.hdf"	ARG
"<GERB_ID>_<IMAGER_ID>_L20S_<DATE>_<VERSION>.hdf"	ARG
"<GERB_ID>_<IMAGER_ID>_L20G_<DATE>_<VERSION>.hdf"	ARG
"<GERB_ID>_<IMAGER_ID>_L20L_H_EUROPE_<DATE>_<VERSION>.hdf"	SHI
"<GERB_ID>_<IMAGER_ID>_L20S_H_EUROPE_<DATE>_<VERSION>.hdf"	SHI
"<GERB_ID>_<IMAGER_ID>_L20G_H_EUROPE_<DATE>_<VERSION>.hdf"	SHI
"<GERB_ID>_<IMAGER_ID>_L20L_15M_50_<DATE>_<VERSION>.hdf"	BARG
"<GERB_ID>_<IMAGER_ID>_L20S_15M_50_<DATE>_<VERSION>.hdf"	BARG
"<GERB_ID>_<IMAGER_ID>_L20G_15M_50_<DATE>_<VERSION>.hdf"	BARG
"<GERB_ID>_<IMAGER_ID>_L20A_H_<DATE>_<VERSION>.hdf"	SHI
"<GERB_ID>_<IMAGER_ID>_L20G_H_<DATE>_<VERSION>.hdf"	SHI

Table 1: L2 RMIB GERB Products

2.3.1 "<GERB_ID>_<IMAGER_ID>_L20L_<DATE>_<VERSION>.hdf"

This is an ARG product. This product contains thermal fluxes and unfiltered radiances for distribution by the GGSPS, defined to be compatible with the GERB filtered radiances derived by the GGSPS.

Table 2: Hierarchical structure of the product "L20L"

Access Path Name	Page
/Summary Thermal Products Confidence	24
/Duplication Flag	24
/File Creation Time	25
/File Name	25
/Radiation Type Identifier	25
/GERB/	28
/GERB/Instrument Identifier	28
/GGSPS/	28
/GGSPS/L1.5 NANRG File Name	29
/GGSPS/L1.5 NANRG Product Version	29
/Geolocation/	29
/Geolocation/Geolocation File Name	32
/Geolocation/Line of Sight North-South Speed	32
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36

continued on next page

continued from previous page	
/Geolocation/Short Wave Image 1/	37
/Geolocation/Short Wave Image 1/Histogram of Line of Sight East-West Positions	37
/Geolocation/Short Wave Image 2/	37
/Geolocation/Short Wave Image 2/Histogram of Line of Sight East-West Positions	38
/Geolocation/Short Wave Image 3/	38
/Geolocation/Short Wave Image 3/Histogram of Line of Sight East-West Positions	38
/Geolocation/Total Image 1/	39
/Geolocation/Total Image 1/Histogram of Line of Sight East-West Positions	39
/Geolocation/Total Image 2/	39
/Geolocation/Total Image 2/Histogram of Line of Sight East-West Positions	40
/Geolocation/Total Image 3/	40
/Geolocation/Total Image 3/Histogram of Line of Sight East-West Positions	40
/Geolocation/Nominal Satellite Longitude	36
/Imager/	41
/Imager/Instrument Identifier	41
/Imager/Type	41
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Radiometry/	42
/Radiometry/Longwave Correction	43
/Radiometry/Spectral Regression Parameters	46
/Radiometry/Thermal Flux	47
/Radiometry/Thermal Radiance	47
/Radiometry/A Values (per GERB detector cell)/	47
/Radiometry/C Values (per GERB detector cell)/	48
/Scene Identification/	48
/Scene Identification/Thermal Angular Dependency Models Set Version	51
/Times/	52
/Times/First GERB Packet	52
/Times/Last GERB Packet	53
/Times/End of Integration (per column)/	53
/Times/Start of Integration (per column)/	54

2.3.2 "<GERB_ID>_<IMAGER_ID>_L20S_<DATE>_<VERSION>.hdf"

This is an ARG product. This product contains solar fluxes and unfiltered radiances for distribution by the GGSPS, defined to be compatible with the GERB filtered radiances derived by the GGSPS.

Table 3: Hierarchical structure of the product "L20S"

Access Path Name	Page
/Summary Solar Products Confidence	24
/Duplication Flag	24
/File Creation Time	25
/File Name	25
/Radiation Type Identifier	25
/GERB/	28
/GERB/Instrument Identifier	28
/GGSPS/	28
/GGSPS/L1.5 NANRG File Name	29
/GGSPS/L1.5 NANRG Product Version	29
/Geolocation/	29
continued on next page	

continued from previous page

continued on next page

continued from previous page

/Times/First GERB Packet	52
/Times/Last GERB Packet	53
/Times/End of Integration (per column)/	53
/Times/Start of Integration (per column)/	54

2.3.3 "<GERB_ID>_<IMAGER_ID>_L20G_<DATE>_<VERSION>.hdf"

This product contains the geolocation (latitude, longitude) for the pixels of the "L20L" and "L20S" products.

Table 4: Hierarchical structure of the product "L20G"

Access Path Name	Page
/File Creation Time	25
/File Name	25
/GERB/	28
/GERB/Instrument Identifier	28
/Geolocation/	29
/Geolocation/Latitude	30
/Geolocation/Longitude	31
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Times/	52
/Times/First GERB Packet	52

2.3.4 "<GERB_ID>_<IMAGER_ID>_L20L_H_EUROPE_<DATE>_<VERSION>.hdf"

This is a SHI product, defined in a window over Europe. This product contains thermal radiances and fluxes, defined to be compatible with the SEVIRI radiance images. It can e.g. be used together with SEVIRI derived cloud products.

Table 5: Hierarchical structure of the product "L20L_H_EUROPE"

Access Path Name	Page
/Summary Thermal Products Confidence	24
/File Creation Time	25
/File Name	25
/Radiation Type Identifier	25
/Angles/	25

continued on next page

continued from previous page	
/Angles/Viewing Zenith	27
/Angles/Relative Azimuth	26
/GERB/	28
/GERB/Instrument Identifier	28
/GGSPS/	28
/GGSPS/L1.5 NANRG File Name	29
/GGSPS/L1.5 NANRG Product Version	29
/Geolocation/	29
/Geolocation/Geolocation File Name	32
/Geolocation/Line of Sight North-South Speed	32
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/Geolocation/Nominal Satellite Longitude	36
/PSF Parameters	45
/Imager/	41
/Imager/Instrument Identifier	41
/Imager/Type	41
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Radiometry/	42
/Radiometry/Longwave Correction	43
/Radiometry/Resolution Enhancement Parameters	45
/Radiometry/Spectral Regression Parameters	46
/Radiometry/Thermal Flux	47
/Radiometry/Thermal Radiance	47
/Radiometry/A Values (per GERB detector cell)/	47
/Radiometry/C Values (per GERB detector cell)/	48
/Scene Identification/	48
/Scene Identification/Thermal Angular Dependency Model	51
/Scene Identification/Thermal Angular Dependency Models Set Version	51
/Times/	52
/Times/Time (per row)/	54

2.3.5 "<GERB_ID>_<IMAGER_ID>_L20S_H_EUROPE_<DATE>_<VERSION>.hdf"

This is a SHI product, defined in a window over Europe. This product contains solar radiances and fluxes, defined to be compatible with the SEVIRI radiance images. It can e.g. be used together with SEVIRI derived cloud products.

Table 6: Hierarchical structure of the product "L20S_H_EUROPE"

continued on next page

continued from previous page	
/Scene Identification/Cloud Cover	49
/Scene Identification/Cloud Phase	49
/Scene Identification/Solar Angular Dependency Model	50
/Scene Identification/Surface Type	50
/Scene Identification/Solar Angular Dependency Models Set Version	51
/Times/	52
/Times/Time (per row)/	54

2.3.6 "<GERB_ID>_<IMAGER_ID>_L20G_H_EUROPE_<DATE>_<VERSION>.hdf"

This product contains the geolocation (latitude, longitude) for the pixels of the "L20L_H_EUROPE" and "L20S_H_EUROPE" products.

Table 7: Hierarchical structure of the product "L20G_H_EUROPE"

Access Path Name	Page
/File Creation Time	25
/File Name	25
/GERB/	28
/GERB/Instrument Identifier	28
/Geolocation/	29
/Geolocation/Latitude	30
/Geolocation/Longitude	31
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Times/	52
/Times/First GERB Packet	52

2.3.7 "<GERB_ID>_<IMAGER_ID>_L20L_15M_50_<DATE>_<VERSION>.hdf"

This is a BARG product. This product contains thermal radiances and fluxes, defined for easy comparison with model output.

Table 8: Hierarchical structure of the product "L20L_15M_50"

Access Path Name	Page
/Summary Thermal Products Confidence	24
/Duplication Flag	24
/File Creation Time	25
continued on next page	

continued from previous page	
/File Name	25
/Radiation Type Identifier	25
/Angles/	25
/Angles/Viewing Zenith	27
/Angles/Relative Azimuth	26
/GERB/	28
/GERB/Instrument Identifier	28
/GGSPS/	28
/GGSPS/L1.5 NANRG File Name	29
/GGSPS/L1.5 NANRG Product Version	29
/Geolocation/	29
/Geolocation/Geolocation File Name	32
/Geolocation/Line of Sight North-South Speed	32
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/Geolocation/Nominal Satellite Longitude	36
/PSF Parameters	45
/Imager/	41
/Imager/Instrument Identifier	41
/Imager/Type	41
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Radiometry/	42
/Radiometry/Longwave Correction	43
/Radiometry/Resolution Enhancement Parameters	45
/Radiometry/Spectral Regression Parameters	46
/Radiometry/Thermal Radiance	47
/Radiometry/Thermal Flux	47
/Radiometry/A Values (per GERB detector cell)/	47
/Radiometry/C Values (per GERB detector cell)/	48
/Scene Identification/	48
/Scene Identification/Thermal Angular Dependency Models Set Version	51
/Times/	52
/Times/End of Integration	52
/Times/Start of Integration	53

2.3.8 "<GERB_ID>_<IMAGER_ID>_L20S_15M_50_<DATE>_<VERSION>.hdf"

This is a BARG product. This product contains solar fluxes, defined for easy comparison with model output.

Table 9: Hierarchical structure of the product "L20S_15M_50"

continued on next page

continued from previous page

/Scene Identification/Cloud Cover	49
/Scene Identification/Cloud Phase	49
/Scene Identification/Surface Type	50
/Scene Identification/Solar Angular Dependency Models Set Version	51
/Times/	52
/Times/Start of Integration	53
/Times/End of Integration	52

2.3.9 "<GERB_ID>_<IMAGER_ID>_L20G_15M_50_<DATE>_<VERSION>.hdf"

This product contains the geolocation (latitude, longitude) for the pixels of the "L20L_15M_50" and "L20S_15M_50" products.

Table 10: Hierarchical structure of the product "L20G_15M_50"

Access Path Name	Page
/File Creation Time	25
/File Name	25
/GERB/	28
/GERB/Instrument Identifier	28
/Geolocation/	29
/Geolocation/Latitude	30
/Geolocation/Longitude	31
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/RMIB/	41
/RMIB/Software Identifier	42
/RMIB/Product Version	42
/Times/	52
/Times/First GERB Packet	52

2.3.10 "<GERB_ID>_<IMAGER_ID>_L20A_H_<DATE>_<VERSION>.hdf"

This is a SHI product, defined over the full MSG disc.
It is not intended for routine distribution.

Table 11: Hierarchical structure of the product "L20A_H"

Access Path Name	Page
/Summary Thermal Products Confidence	24
/Summary Solar Products Confidence	24
/Duplication Flag	24

continued on next page

continued from previous page

continued on next page

continued from previous page	
/Radiometry/Longwave Correction	43
/Radiometry/Resolution Enhancement Parameters	45
/Radiometry/Shortwave Correction	44
/Radiometry/Solar Flux	45
/Radiometry/Solar Radiance	46
/Radiometry/Spectral Regression Parameters	46
/Radiometry/Thermal Flux	47
/Radiometry/Thermal Radiance	47
/Radiometry/A Values (per GERB detector cell)/	47
/Radiometry/C Values (per GERB detector cell)/	48
/Scene Identification/	48
/Scene Identification/Cloud Amount	49
/Scene Identification/Cloud Cover	49
/Scene Identification/Cloud Phase	49
/Scene Identification/Solar Angular Dependency Model	50
/Scene Identification/Surface Type	50
/Scene Identification/Thermal Angular Dependency Model	51
/Scene Identification/Solar Angular Dependency Models Set Version	51
/Scene Identification/Thermal Angular Dependency Models Set Version	51
/Times/	52
/Times/Time (per row)/	54

2.3.11 "<GERB_ID>_<IMAGER_ID>_L20G_H_<DATE>_<VERSION>.hdf"

This product contains the geolocation (latitude, longitude) for the pixels of the "L20A_H" product.

Table 12: Hierarchical structure of the product "L20G_H"

Access Path Name	Page
/File Creation Time	25
/File Name	25
/GERB/	28
/GERB/Instrument Identifier	28
/Geolocation/	29
/Geolocation/Latitude	30
/Geolocation/Longitude	31
/Geolocation/Rectified Grid/	32
/Geolocation/Rectified Grid/Grid Orientation	33
/Geolocation/Rectified Grid/Lap	33
/Geolocation/Rectified Grid/Lop	33
/Geolocation/Rectified Grid/Nr	33
/Geolocation/Rectified Grid/Nx	34
/Geolocation/Rectified Grid/Ny	34
/Geolocation/Rectified Grid/Xp	34
/Geolocation/Rectified Grid/Yp	35
/Geolocation/Rectified Grid/dx	35
/Geolocation/Rectified Grid/dy	35
/Geolocation/Rectified Grid/Resolution Flags/	35
/Geolocation/Rectified Grid/Resolution Flags/East West	36
/Geolocation/Rectified Grid/Resolution Flags/North South	36
/RMIB/	41
/RMIB/Software Identifier	42
continued on next page	

3 L2 RMIB GERB Products Contents

This section describes the contents of the fields from which the L2 products are composed. For each HDF field, the following information is provided:

- The field name.
- The HDF object type (see 4.4.1) is indicated between square brackets.
- A description of the field contents.
- The absolute name (see 4.4.2) of the field.
- If need be, some additional information.

3.1 Summary Thermal Products Confidence[Attribute]

Description

Product confidence summary value indicating overall usefulness of thermal products. This is the ratio of pixels that are valid. A valid pixel is one that is in the range defined by 3.60 and 3.61

HDF Path

"*/Summary Thermal Products Confidence*"

Additional Information

Data type: `H5T_IEEE_F64BE`

3.2 Summary Solar Products Confidence [Attribute]

Description

Product confidence summary value indicating overall usefulness of solar products. This is the ratio of pixels that are valid. A valid pixel is one that is in the range defined by 3.57 and 3.58

HDF Path

"*/Summary Solar Products Confidence*"

Additional Information

Data type: `H5T_IEEE_F64BE`.

3.3 Duplication Flag [Attribute]

Description

This flag is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. It indicates whether this NANRG product appears in two fifteen minute slots, and if so if it is the first or second duplication.

HDF Path

"*/Duplication Flag*"

Additional Information

Data type: `H5T_STD_I32BE`.
Allowed values: 0 = no duplication, 1 = nominal file, 2 = duplicated file
See level 1.5 NANRG documentation for comlplete information.

3.4 File Creation Time [Attribute]

Description

The UTC time when the file was created.

HDF Path

"*/File Creation Time*"

Additional Information

Data type: `H5T_C_S1`.

3.5 File Name [Attribute]

Description

The name of the file.

HDF Path

"*/File Name*"

Additional Information

Data type: `H5T_C_S1`.

3.6 Radiation Type Identifier [Attribute]

Description

This identifier indicates on of the two possible radiation types: emitted thermal radiation or reflected solar radiation.

HDF Path

"*/Radiation Type Identifier*"

Additional Information

Data type: `H5T_C_S1`.
Allowed values: "SOL" for solar, "TH" for thermal

3.7 Angles [Group]

Description

This group contains the angles that describe the viewing and solar geometry for each GERB observation.

HDF Path

"*/Angles/*"

Additional Information

The possible viewing and solar angles are illustrated in figure 1.

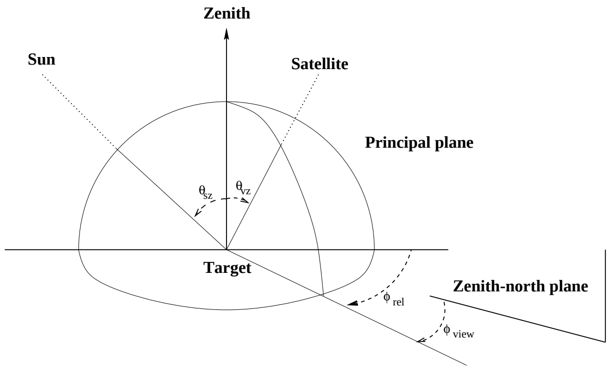

Figure 1: Sun-target-satellite geometry. Principal plane = plane which contains the sun and the zenith direction. Zenith-north plane = plane which contains the local northern direction and the zenith direction. θ_{sz} = solar zenith angle. θ_{vz} = viewing zenith angle. ϕ_{rel} = relative azimuth angle. ϕ_{view} = viewing azimuth angle.

3.8 Relative Azimuth [Dataset]

Description

See figure 1. The relative azimuth angle ϕ_{rel} is the angle of the satellite observation plane relative to the principal plane. The target is the observed scene on the surface of the earth, characterised by its geodetic geolocation parameters. The zenith direction is the local geodetic vertical direction. The satellite observation plane contains the zenith and the satellite. The principal plane contains the zenith and the sun.

HDF Path

"*/Angles/Relative Azimuth*"

Additional Information

Quantisation factor: 1 degree.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: (-180,180).

3.9 Viewing Azimuth [Dataset]

Description

See figure 1. The viewing azimuth angle ϕ_{view} is the angle of the satellite observation plane relative to the zenith-north plane. The target is the observed scene on the surface of the earth, characterised by its geodetic geolocation parameters. The zenith direction is the local geodetic vertical direction. The satellite observation plane contains the zenith and the satellite. The zenith plane contains the zenith and the local north direction.

HDF Path

"*/Angles/Viewing Azimuth*"

Additional Information

Quantisation factor: 1 degree.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: (0,360).

3.10 Solar Zenith [Dataset]

Description

See figure 1. The solar zenith angle θ_{sz} is the angle from the zenith direction towards the target-solar direction. The target is the observed scene on the surface of the earth, characterised by its geodetic geolocation parameters. The zenith direction is the local geodetic vertical direction.

HDF Path

"*/Angles/Solar Zenith*"

Additional Information

Quantisation factor: 1 degree.
Data type: `H5T_STD_I8BE`.
Error value: 255.
Range: (0,180) degree.

3.11 Viewing Zenith [Dataset]

Description

See figure 1. The viewing zenith angle θ_{vz} is the angle from the zenith direction towards the target-satellite direction. The target is the observed scene on the surface of the earth, characterised by its geodetic geolocation parameters. The zenith direction is the local geodetic vertical direction.

HDF Path

"*/Angles/Viewing Zenith*"

Additional Information

Quantisation factor: 1 degree.
Data type: `H5T_STD_I8BE`.
Error value: 255.
Range: (0,90) degree.

3.12 GERB [Group]

Description

This group contains information regarding the GERB instrument that was used.

HDF Path

"*/GERB/*"

Additional Information

None

3.13 Instrument Identifier [Attribute]

Description

This identifier indicates which GERB instrument is used.

HDF Path

"*/GERB/Instrument Identifier*"

Additional Information

Data type: `H5T_C_S1`.
Allowed values: "GERB1", "GERB2", "GERB3"

3.14 GGSPS [Group]

Description

This information in this group is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. It contains information about the GGSPS software that was used to produce this NANRG product.

HDF Path

"*/GGSPS/*"

Additional Information

See GGSPS documentation.

3.15 L1.5 NANRG File Name [Dataset]

Description

The name of the L1.5 NANRG files that have been used as input. The name of the main file is the first item.

HDF Path

"*/GGSPS/L1.5 NANRG File Name*"

Additional Information

Data type: `H5T_C_S1`.
See L1.5 NANRG documentation for more details.

3.16 L1.5 NANRG Product Version [Dataset]

Description

The product version of the L1.5 NANRG files that have been used as input. The version of the main file is the first item.

HDF Path

"*/GGSPS/L1.5 NANRG Product Version*"

Additional Information

Data type: `H5T_STD_I32BE`.
See L1.5 NANRG documentation for more details.

3.17 Geolocation [Group]

Description

This group contains the parameters that describe the location on earth for the GERB observations. The location is given in geodetic coordinates, relative to the earth reference ellipsoid.

HDF Path

"*/Geolocation/*"

Additional Information

The geolocation parameters for a geostationary satellite viewing the earth in an equi-angular projection are illustrated in figure 2. The earth reference ellipsoid is defined by a polar radius R_P and an equatorial radius R_E.

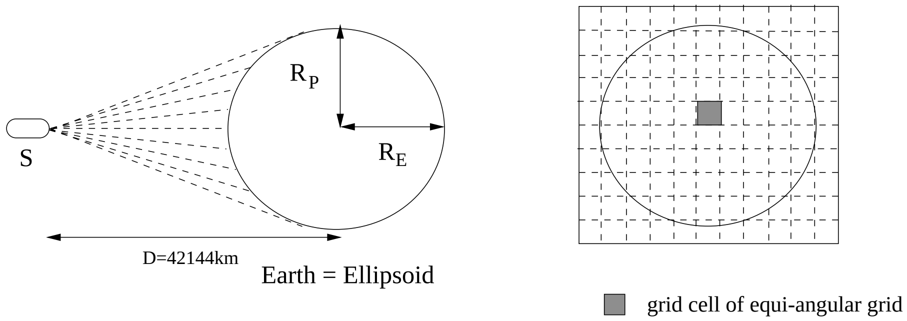

Figure 2: Geolocation parameters. Left = intersection of the lines of sight of a geostationary satellite with the earth reference ellipsiod. Right = the corresponding equi-angular grid projection of the earth as viewed from the satellite.

Consider the right hand axis system where the origin is the centre of the earth, the XY-plane is the equatorial plane, the YZ-plane is the Greenwhich meridian plane and the Z-axis points northwards (see figure 3). A point P on the earth reference ellipsoid with longitude ϕ and geocentric latitude θ_c has the cartesian coordinates

$$\begin{aligned} X &= R_E \cos(\phi)\cos(\theta_c) \\ Y &= R_E \sin(\phi)\cos(\theta_c) \\ Z &= R_P \sin(\theta_c) \end{aligned} \tag{1}$$

The geodetic vertical in the point P is defined as the outer normal to the earth reference ellipsoid in P. The geodetic latitude of the point P is the angle from the equatorial plane towards the geodetic vertical in P. The relationship between the geocentric latitude θ_c and the geodetic latitude θ_d is given by $\tan(\theta_d) = (R_E/R_P)^2 \tan(\theta_c)$.

3.18 Latitude [Dataset]

Description

See figures 2 and 3. The geodetic latitude is given for every GERB observation point. The GERB observation point is the intersection of the GERB line of sight and the earth reference ellipsoid. The geodetic latitude of the point P is the angle from the equatorial plane towards the geodetic vertical in P. The geodetic vertical in the point P is defined as the outer normal to the earth reference ellipsoid in P.

HDF Path

"*/Geolocation/Latitude*"

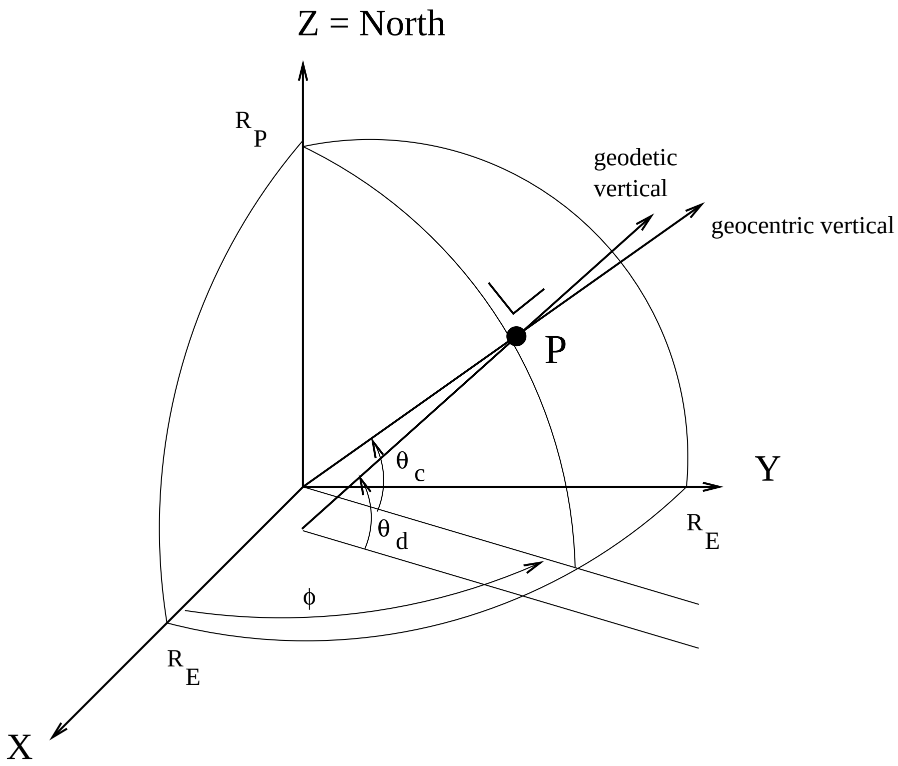

Figure 3: Geodetic versus geocentric geolocation. θ_c = geocentric latitude. θ_d = geodetic latitude. ϕ = (geodetic or geocentric) longitude.

Additional Information

Quantisation factor: $\frac{1}{128}$ degree.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: (-90,+90) degree.

3.19 Longitude [Dataset]

Description

See figures 2 and 3. The longitude is given for every GERB observation point. The GERB observation point is the intersection of the GERB line of sight and the earth reference ellipsoid. The longitude of the point P is the angle from the Greenwhich meridian plane towards P, positive in eastward direction.

HDF Path

"*/Geolocation/Longitude*"

Additional Information

Quantisation factor: $\frac{1}{128}$ degree.

Data type: H5T_STD_I16BE.
Error value: -32767.
Range: (-180,+180) degree.

3.20 Geolocation File Name [Attribute]

Description

Name of the geolocation file which contains the latitude and longitude for each pixel.

HDF Path

"*/Geolocation/Geolocation File Name*"

Additional Information

Data type: H5T_C_S1.

3.21 Line of Sight North-South Speed [Attribute]

Description

The angular speed in north-south direction of the GERB line of sight during the earth view.

HDF Path

"*/Geolocation/Line of Sight North-South Speed*"

Additional Information

Units: Degree per second.
Data type: H5T_IEEE_F64BE.

This line of sight speed influences the dynamic point spread function of the GERB instrument.

3.22 Rectified Grid [Group]

Description

This group contains the parameters that describe the rectified grid on which the GERB observation points are located. The rectified grid is the grid obtained by an equi-angular projection from a geostationary satellite as shown in figure 2.

HDF Path

"*/Geolocation/Rectified Grid/*"

Additional Information

None

3.23 Grid Orientation [Attribute]

Description

The angle between the increasing y axis and the meridian of the sub-satellite point in the direction of increasing latitude.

HDF Path

"*/Geolocation/Rectified Grid/Grid Orientation*"

Additional Information

Units: Millidegree.
Data type: `H5T_IEEE_F64BE`.

3.24 Lap [Attribute]

Description

Latitude of sub-satellite point.

HDF Path

"*/Geolocation/Rectified Grid/Lap*"

Additional Information

Units: Degree.
Data type: `H5T_IEEE_F64BE`.

3.25 Lop [Attribute]

Description

Longitude of sub-satellite point.

HDF Path

"*/Geolocation/Rectified Grid/Lop*"

Additional Information

Units: Degree.
Data type: `H5T_IEEE_F64BE`.

3.26 Nr [Attribute]

Description

The altitude of the imaging device from the earth's center, measured in units of the earth's (equatorial) radius.

HDF Path

"*/Geolocation/Rectified Grid/Nr*"

Additional Information

Units: Earth radius.
Data type: `H5T_IEEE_F64BE`.

3.27 Nx [Attribute]

Description

Number of points along x axis = number of columns = width image.

HDF Path

"*/Geolocation/Rectified Grid/Nx*"

Additional Information

Data type: `H5T_STD_I32BE`.

3.28 Ny [Attribute]

Description

Number of points along y axis = number of rows = height image.

HDF Path

"*/Geolocation/Rectified Grid/Ny*"

Additional Information

Data type: `H5T_STD_I32BE`.

3.29 Xp [Attribute]

Description

X-coordinate of sub satellite point.

HDF Path

"*/Geolocation/Rectified Grid/Xp*"

Additional Information

Units: Grid length.
Data type: `H5T_IEEE_F64BE`.

3.30 Yp [Attribute]

Description

Y-coordinate of sub-satellite point.

HDF Path

"*/Geolocation/Rectified Grid/Yp*"

Additional Information

Units: Grid length.
Data type: `H5T_IEEE_F64BE`.

3.31 dx [Attribute]

Description

Apparent diameter of earth in grid lengths, in x direction.

HDF Path

"*/Geolocation/Rectified Grid/dx*"

Additional Information

Units: Grid length.
Data type: `H5T_IEEE_F64BE`.

3.32 dy [Attribute]

Description

Apparent diameter of earth in grid lengths, in y direction.

HDF Path

"*/Geolocation/Rectified Grid/dy*"

Additional Information

Units: Grid length.
Data type: `H5T_IEEE_F64BE`.

3.33 Resolution Flags [Group]

Description

This group contains the north-south and the east-west resolution at nadir (subsatellite point) of the used rectified grid.

HDF Path

"*/Geolocation/Rectified Grid/Resolution Flags/*"

Additional Information

None

3.34 East West [Attribute]

Description

The east-west resolution at nadir (subsatellite point) of the used rectified grid.

HDF Path

"*/Geolocation/Rectified Grid/Resolution Flags/East West*"

Additional Information

Units: Degree.
Data type: `H5T_IEEE_F64BE`.

3.35 North South [Attribute]

Description

The north-south resolution at nadir (subsatellite point) of the used rectified grid.

HDF Path

"*/Geolocation/Rectified Grid/Resolution Flags/North South*"

Additional Information

Units: Degree.
Data type: `H5T_IEEE_F64BE`.

3.36 Nominal Satellite Longitude [Attribute]

Description

The nominal satellite longitude.

HDF Path

"*/Geolocation/Nominal Satellite Longitude*"

Additional Information

Units: Degree.
Data type: `H5T_IEEE_F64BE`.

3.37 Short Wave Image 1 [Group]

Description

This group contains geolocation information for the first of the three GERB short wave images that have been used as input.

HDF Path

"*/Geolocation/Short Wave Image 1/*"

Additional Information

None

3.38 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the first GERB short wave image.

HDF Path

"*/Geolocation/Short Wave Image 1/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.39 Short Wave Image 2 [Group]

Description

This group contains geolocation information for the second of the three GERB short wave images that have been used as input.

HDF Path

"*/Geolocation/Short Wave Image 2/*"

Additional Information

None

3.40 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the second GERB short wave image.

HDF Path

"*/Geolocation/Short Wave Image 2/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.41 Short Wave Image 3 [Group]

Description

This group contains geolocation information for the third of the three GERB short wave images that have been used as input.

HDF Path

"*/Geolocation/Short Wave Image 3/*"

Additional Information

None

3.42 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the third GERB short wave image.

HDF Path

"*/Geolocation/Short Wave Image 3/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.43 Total Image 1 [Group]

Description

This group contains geolocation information for the first of the three GERB total images that have been used as input.

HDF Path

"*/Geolocation/Total Image 1/*"

Additional Information

None

3.44 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the first GERB total image.

HDF Path

"*/Geolocation/Total Image 1/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.45 Total Image 2 [Group]

Description

This group contains geolocation information for the second of the three GERB total images that have been used as input.

HDF Path

"*/Geolocation/Total Image 2/*"

Additional Information

None

3.46 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the second GERB total image.

HDF Path

"*/Geolocation/Total Image 2/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.47 Total Image 3 [Group]

Description

This group contains geolocation information for the third of the three GERB total images that have been used as input.

HDF Path

"*/Geolocation/Total Image 3/*"

Additional Information

None

3.48 Histogram of Line of Sight East-West Positions [Dataset]

Description

This information is copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. This dataset contains the histogram of the measured east-west deviations of the GERB line of sight from its nominal position. One histogram is given for every image column of the third GERB total image.

HDF Path

"*/Geolocation/Total Image 3/Histogram of Line of Sight East-West Positions*"

Additional Information

Data type: `H5T_STD_U8BE`.

3.49 Imager [Group]

Description

This group contains information about the imager (SEVIRI or METEOSAT) that has been used as auxiliary data in the processing of GERB data.

HDF Path

"*/Imager/*"

Additional Information

None

3.50 Instrument Identifier [Attribute]

Description

This identifier indicates which imager of a particular type has been used.

HDF Path

"*/Imager/Instrument Identifier*"

Additional Information

Data type: `H5T_STD_I32BE`.

3.51 Type [Attribute]

Description

Indicates which one of two possible imager types has been used: SEVIRI or METEOSAT.

HDF Path

"*/Imager/Type*"

Additional Information

Data type: `H5T_C_S1`.

3.52 RMIB [Group]

Description

This group contains information about the RMIB GERB processing software.

HDF Path

"*/RMIB/*"

Additional Information

None

3.53 Software Identifier [Attribute]

Description

Version number of the RMIB GERB processing software that has been used.

HDF Path

"*/RMIB/Software Identifier*"

Additional Information

Data type: `H5T_C_S1`.

3.54 Product Version [Attribute]

Description

Version number of the RMIB GERB processing products.

HDF Path

"*/RMIB/Product Version*"

Additional Information

Data type: `H5T_STD_I32BE`.

3.55 Radiometry [Group]

Description

This group contains information about the radiometric measurements of GERB.

HDF Path

"*/Radiometry/*"

Additional Information

None

3.56 Longwave Correction [Dataset]

Description

The longwave correction factor is the ratio between the GERB measured synthetic longwave filtered radiance, and the estimation of the same quantity from the imager data. If this ratio is outside a defined range, the limit of the range is applied (see 3.57 and 3.58).

HDF Path

"*/Radiometry/Longwave Correction*"

Additional Information

- The synthetic longwave filtered radiance is defined the total filtered radiance minus the factor 'A' times the short wave filtered radiance.

Quantisation factor: 0.005
Offset: 1.
Data type: `H5T_STD_I8BE`.
Range: (-1.635,1.635).

3.57 Longwave Minimum Correction Value[Attribute]

Description

If the longwave ratio between GERB and SEVIRI is lower than this value, this value has been applied as correction.

HDF Path

"*/Radiometry/Longwave Correction/Minimum Correction Value*"

Additional Information

Data type: `H5T_IEEE_F64BE`
If no lower limit has been used, this value is -1.

3.58 Longwave Maximum Correction Value[Attribute]

Description

If the longwave ratio between GERB and SEVIRI is higher than this value, this value has been applied as correction.

HDF Path

"*/Radiometry/Longwave Correction/Maximum Correction Value*"

Additional Information

Data type: `H5T_IEEE_F64BE`
If no upper limit has been used, this value is -1.

3.59 Shortwave Correction [Dataset]

Description

The shortwave correction factor is the ratio between the GERB measured shortwave filtered radiance, and the estimation of the same quantity from the imager data. If this ratio is outside a defined range, the limit of the range is applied (see 3.60 and 3.61).

HDF Path

"*/Radiometry/Shortwave Correction*"

Additional Information

Quantisation factor: 0.005
Offset: 1.
Data type: `H5T_STD_I8BE`.
Range: (-1.635,1.635).

3.60 Shortwave Minimum Correction Value[Attribute]

Description

If the shortwave ratio between GERB and SEVIRI is lower than this value, this value has been applied as correction.

HDF Path

"*/Radiometry/Shortwave Correction/Minimum Correction Value*"

Additional Information

Data type: `H5T_IEEE_F64BE`
If no lower limit has been used, this value is -1.

3.61 Shortwave Maximum Correction Value[Attribute]

Description

If the shortwave ratio between GERB and SEVIRI is higher than this value, this value has been applied as correction.

HDF Path

"*/Radiometry/Shortwave Correction/Maximum Correction Value*"

Additional Information

Data type: `H5T_IEEE_F64BE`
If no upper limit has been used, this value is -1.

3.62 PSF Parameters [Group]

Description

This group contains information about dynamic behaviour of GERB acquisition. The content is still TBD.

HDF Path

"*/PSF Parameters/*"

Additional Information

None

3.63 Resolution Enhancement Parameters [Dataset]

Description

This dataset contains the parameters that have been used for the resolution enhancement of GERB data by imager data. The imager can be the METEOSAT or the SEVIRI one. The content is still TBD.

HDF Path

"*/Radiometry/Resolution Enhancement Parameters*"

Additional Information

Data type: `H5T_IEEE_F64BE`.

3.64 Solar Flux [Dataset]

Description

This dataset contains the surface density of the radiative flux of reflected solar energy at the top of the atmosphere, referenced at the earth reference ellipsoid surface.

HDF Path

"*/Radiometry/Solar Flux*"

Additional Information

The earth surface can be calculated at different heights above the earth reference ellipsoid. Because the surface increases with height above the reference ellipsoid, the flux surface density decreases with height, by a factor $R^2/(R+h)^2$, with R the local radius of the reference ellipsoid and h the height above the reference ellipsoid. Quantisation factor: 0.25 W/m^2.

Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: [0,1500] W/m^2

3.65 Solar Radiance [Dataset]

Description

This dataset contains the GERB measured radiance of reflected solar energy at the top of the atmosphere.

HDF Path

"*/Radiometry/Solar Radiance*"

Additional Information

Quantisation factor: 0.05 $W/(m^2sr)$.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: [0,500] $W/(m^2sr)$

3.66 Spectral Regression Parameters [Dataset]

Description

This dataset contains the spectral regression parameters that have been used to estimate unfiltered radiances and GERB filtered radiances from the imager data.

HDF Path

"*/Radiometry/Spectral Regression Parameters*"

Additional Information

Data type: `H5T_IEEE_F64BE`.

3.67 Thermal Flux [Dataset]

Description

This dataset contains the surface density of the radiative flux of emitted thermal energy at the top of the atmosphere, referenced at the earth reference ellipsoid surface.

HDF Path

"*/Radiometry/Thermal Flux*"

Additional Information

The earth surface can be calculated at different heights above the earth reference ellipsoid. Because the surface increases with height above the reference ellipsoid, the flux surface density decreases with height, by a factor $R^2/(R+h)^2$, with R the local radius of the reference ellipsoid and h the height above the reference ellipsoid.

Quantisation factor: 0.25 W/m^2.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: [150, 450] W/m^2

3.68 Thermal Radiance [Dataset]

Description

This dataset contains the GERB measured radiance of emitted energy at the top of the atmosphere.

HDF Path

"*/Radiometry/Thermal Radiance*"

Additional Information

Quantisation factor: 0.05 W/m^2sr.
Data type: `H5T_STD_I16BE`.
Error value: -32767.
Range: [50, 150] W/m^2

3.69 A Values (per GERB detector cell) [Dataset]

Description

The 'A' value is used for the definition of the GERB synthetic longwave filtered radiance. 'A' is defined as the ratio between the GERB filtered total radiance and the GERB filtered short wave radiance for a 5800 K black body. The given 'A' values are copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG)

GERB radiance product that has been used as input file. They are measured with the sunlit onboard solar diffuser sphere. There are one value for each detector. If more than one NANRG file has been used, the used A values are the ones of the main file (TBC).

HDF Path

"*/Radiometry/A Values (per GERB detector cell)/*"

Additional Information

Data type: `H5T_IEEE_64BE`.
See GERB calibration documentation.

3.70 C Values (per GERB detector cell) [Dataset]

Description

The 'C' value is used to monitor the longwave leakage of the GERB quartz filter. 'C' is defined as the ratio between the GERB filtered shortwave radiance and the GERB filtered total radiance for a 300 K black body. The given 'C' values are copied from the level 1.5 Non Averaged Non Rectified Geolocated (NANRG) GERB radiance product that has been used as input file. They are measured with the onboard black body. If more than one NANRG file has been used, the used A values are the ones of the main file (TBC).

HDF Path

"*/Radiometry/C Values (per GERB detector cell)/*"

Additional Information

Data type: `H5T_IEEE_64BE`.
See GERB calibration documentation.

3.71 Scene Identification [Group]

Description

This group contains auxiliary scene identification given with the GERB measurements. Scene identification is only given during daytime. The given scene identification is the most accurate for the high resolution and for the binned 50 km resolution products. It is only indicative for the GERB footprint product.

HDF Path

"*/Scene Identification/*"

Additional Information

None

3.72 Cloud Amount [Dataset]

Description

The cloud amount is defined as $(L - L_{cs})/(L_{thick} - L_{cs})$, where L is the measured 0.6 micron SEVIRI radiance, L_{cs} is the corresponding clear sky radiance and L_{ov} is the theoretical radiance for an optically thick cloud. A cloud amount of 1 corresponds to an infinitely thick cloud, a cloud amount of 0 corresponds to an infinitely thin cloud.

HDF Path

"*/Scene Identification/Cloud Amount*"

Additional Information

Data type: `H5T_STD_U8BE`.
255 indicates an error.

3.73 Cloud Cover [Dataset]

Description

The cloud cover is the relative number of cloudy imager pixels in the GERB footprint or grid cell. A cloud cover of 0 corresponds to a clear sky footprint. A cloud cover of 100 corresponds to an overcast footprint.

HDF Path

"*/Scene Identification/Cloud Cover*"

Additional Information

Data type: `H5T_STD_U8BE`.
Error Value: 255.
Range: [1,100].

3.74 Cloud Phase [Dataset]

Description

This dataset contains the mean cloud phase (water or ice cloud) over the footprint. A cloud phase of 0 corresponds to a pure water cloud. A cloud phase of 100 corresponds to a pure ice cloud. And between this limits, mixed water-ice cloud.

HDF Path

"*/Scene Identification/Cloud Phase*"

Additional Information

Data type: `H5T_STD_U8BE`.
Error Value: 255.
Range: [0,100] (water=0, mixed, ice=100).

3.75 Solar Angular Dependency Model [Dataset]

Description

This dataset contains the identifier of the scene type dependent Angular Dependency Model that has been used to convert the reflected solar radiance in the reflected solar flux.

HDF Path

"*/Scene Identification/Solar Angular Dependency Model*"

Additional Information

Data type: `H5T_STD_U8BE`.
Error Value: -32367.
Range: 0,1024.

3.76 Surface Type [Dataset]

Description

This dataset contains the surface type within the GERB footprint or grid cell. The defined types are : 0=undefined, 1=ocean, dark vegetation, bright vegetation, dark desert, bright desert, snow.

HDF Path

"*/Scene Identification/Surface Type*"

Additional Information

Data type: `H5T_STD_U8BE`.
Allowed values:0=undefined, 1=ocean, 2=dark vegetation, 3=bright vegetation, 4=dark desert, 5=bright desert, 6=snow, 255=error.

3.77 Thermal Angular Dependency Model [Dataset]

Description

This dataset is reserved for future use. It contains the identifier of the scene type dependent Angular Dependency Model that has been used to convert the emitted thermal radiance in the emitted thermal flux.

HDF Path

"*/Scene Identification/Thermal Angular Dependency Model*"

Additional Information

Data type: `H5T_STD_U8BE`.
Valid values range from 0 to 254. 255 indicates an error.

3.78 Solar Angular Dependency Models Set Version [Attribute]

Description

Indicates which set of solar angular dependency models has been used to determine fluxes from radiances.

HDF Path

"*/Scene Identification/Solar Angular Dependency Models Set Version*"

Additional Information

None

3.79 Thermal Angular Dependency Models Set Version [Attribute]

Description

Indicates which set of thermal angular dependency models has been used to determine fluxes from radiances.

HDF Path

"*/Scene Identification/Thermal Angular Dependency Models Set Version*"

Additional Information

None

3.80 Times [Group]

Description

This group gives information about the sampling time of the GERB products. For the three types of products a different time sampling strategy is used, and hence different time parameters are given:

- The high resolution product is a 15(30) minute snapshot at SEVIRI(Meteosat7) acquistion times. A time per row is given.
- The GERB footprint product is an average of three consecutive GERB measurements. An integration start and end time per column is given.
- The binned 50 km is a fixed 15(30) minute average. An overall integration start and end time is given.

HDF Path

"*/Times/*"

Additional Information

None

3.81 End of Integration [Attribute]

Description

End time of integration for the complete image.

HDF Path

"*/Times/End of Integration*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS".

3.82 First GERB Packet [Attribute]

Description

In the radiance and flux files, this is the acquisition time of the first GERB packet that has been used.
In the geolocation files, the value of this field indicates the UTC time for which the data first applies.

HDF Path

"*/Times/First GERB Packet*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS".

3.83 Last GERB Packet [Attribute]

Description

This is the acquisition time of the last GERB packet that has been used.

HDF Path

"*/Times/Last GERB Packet*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS".

3.84 Start of Integration [Attribute]

Description

Start time of integration for the complete image.

HDF Path

"*/Times/Start of Integration*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS".

3.85 End of Integration (per column) [Dataset]

Description

The end of integration parameter gives the time of end of integration for every image column. The imager data is not considered: only time of GERB data used for column data generation is taken into account.

HDF Path

"*/Times/End of Integration (per column)/*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS.MMM".

3.86 Start of Integration (per column) [Dataset]

Description

The start of integration parameter gives the time of start of integration for every image column. The imager data is not considered: only time of GERB data used for column data generation is taken into account.

HDF Path

"*/Times/Start of Integration (per column)/*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS.MMM".

3.87 Time (per row) [Dataset]

Description

This dataset contains the acquisition time for every image row.

HDF Path

"*/Times/Time (per row)/*"

Additional Information

Data type: `H5T_C_S1`.
The format is "YYYYMMDD HH:MM:SS.MMM".

4 L2 RMIB GERB Products Data Access

4.1 Introduction

The L2 RMIB GERB flux products are stored using a file format designed by the National Center for Super-Computing Applications (NCSA): the Hierarchical Data Format version 5 (HDF5). This document describes how to retrieve the L2 RMIB GERB scientific data, from the perspective of a user who wants to use these data in his own programming applications.
We assume that the user's programs will be written in the C (or C++) programming language. Access to files in HDF5 format is through an application programming interface (API) written in C. This API is provided by the HDF library. A simplified version called API Lite is also available. This interface is sufficient to retrieve data from RMIB hdf files.

4.2 How to Obtain the L2 RMIB GERB Products

A user who wishes to have access to the L2 RMIB GERB Products must register at the RMIB GERB web site:

http://gerb.oma.be/

After the registration has been accepted, the user will be able to log on to the RMIB GERB ftp site:

ftp://gerb.oma.be

where the HDF files of the L2 RMIB GERB Products are stored.

4.3 How to Obtain the HDF5 Library

Pre-compiled binaries of the HDF5 libraries for various platforms are available for download at the following URL:

ftp://ftp.ncsa.uiuc.edu/HDF/HDF5/current/bin/

This guide is certainly not a complete HDF5 reference guide. For more advanced use of the HDF5 Library, please refer to the full documentation set, also available from the same site:

http://hdf.ncsa.uiuc.edu or ftp://ftp.ncsa.uiuc.edu/HDF/HDF5/docs/

4.4 Overview of HDF5

4.4.1 Objects

- *File* objects represent the HDF file.
- *Dataset* objects are used for storing multi-dimensional arrays of data.
- *Group* objects serve as containers for other HDF5 objects.
- *Attribute* objects are used for single-valued data that characterise another object (either a group or a dataset).

4.4.2 Hierarchical Layout

At the API level, the layout of an HDF5 file takes the form of a hierarchical structure. In fact, the data organisation closely resembles that of the UNIX file system. An object in an HDF5 file takes on the role of a file in the UNIX file system.
At creation time, an HDF5 object must be given a name (i.e. a string of characters). The name can be either *absolute* (i.e. starting with a '/' character) or *relative* (i.e. *not* starting with a '/' character). The '/' serves as a delimiter and indicates that the object whose name starts at the next character, stands one level deeper in the HDF5 hierarchy.
At the top of the hierarchy stands the *root group* object whose name is "/" (a single slash). The root group is automatically created at the file creation, even if the HDF file is otherwise empty (i.e. it does not contain any user data).
Each data structure within the HDF5 file is unequivocally identified by its absolute name. A *base name* is a relative name that does not contain any '/' character (except maybe as the last character of a group name).
For example, if

- "*First Group*" is the base name of a group object,
- "*My Dataset*" is the base name of a dataset object located in the group above,
- "*Some Attribute*" is the base name of an attribute object of the preceding dataset,

then the absolute name of "*Some Attribute*" is

"*/First Group/My Dataset/Some Attribute*".

4.4.3 API

The HDF5 library API is a set of C functions that manipulates the contents of HDF files. The creation, writing, reading and deletion of all HDF5 objects is done through this API.
The functions names are composed from the concatenation of the following parts:

1. The library identifier prefix: "`H5`",
2. A letter specifying which type of object the function is concerned with ("`F`" for a file object, "`G`" for a group, "`D`" for a dataset, "`A`" for an attribute, "`S`" for a data space object, and "`T`" for a data type object),
3. A string referring to type of action the function performs.

For example, `H5Dread` names a function that is concerned with the reading of data stored in a dataset object.
The library references an HDF5 object through a unique identifier the data type of which is `hid_t`. This identifier is supplied by an "open" function, specific to each type of HDF5 object, which is given the name of the object (see 4.4.2) as one of its arguments.

The necessary functions for retrieving data from an existing HDF file are explained in section 4.5. Note that we only present the subset of the API needed to extract the information stored in the RMIB GERB flux products files. For a complete coverage of the HDF5 library, please refer to the full NCSA documentation (see 4.3).

4.4.4 Lite API

The HDF5 Lite API consists of higher-level functions which do more operations per call than the basic HDF5 interface. The purpose is to wrap intuitive functions around certain sets of features in the existing APIs. This version of the API has two sets of functions: dataset and attribute related functions. These functions hide most of the API complexity.

4.5 How to Retrieve HDF Data

In order to retrieve the data stored in an HDF file, a program needs to gain access to the object containing these data. The basic usage scheme corresponds to the execution of the following steps:

1. Open the file.
2. Read the value(s) stored in the object
3. Close the file.

The functions that perform these operations are explained in the following sections.

4.5.1 File Access

To access the content of an existing file, the file must first be opened with a call to the function `H5Fopen()`:

```
hid_t H5Fopen(  const char *name,
                unsigned flag,
                hid_t access_properties  ) ;
```

where ***name*** is the name of the file to open and ***flag*** describes the type of access (`H5F_ACC_RDONLY` for read-only access). *access_properties* can be set to `H5P_DEFAULT` for the default access parameters.
The return value is an object ID for the open file. A negative return value indicates failure. This ID should be closed when the file is no longer needed, by calling `H5Fclose()`:

```
herr_t H5Fclose(  hid_t file_id  ) ;
```

The return value of `H5Fclose()` is zero for success and negative for failure.

4.5.2 Dataset Access

In order to retrieve the data, the contents of the dataset must be read into memory. This is achieved by a call to the function `H5LTread_dataset_[type]()`.

```
herr_t H5LTread_dataset_[type]( hid_t loc_id,
                                 hid_t dset_name,
                                 [type] *buffer    ) ;
```

where *[type]* can take one of the following values {char, short, int, long, float, double}, *loc_id* is the file ID, *dset_name* is the name of the dataset, *buffer* is the location in memory where the data values will be written to. A negative return value indicates failure.
When calling `H5LTread_dataset_[type]()`, it is assumed that *buffer* is a memory location big enough to contain all the data to be read. Otherwise, a memory fault will likely result. To be safe, we have to query the dataset for the number of data it contains and afterwards dynamically allocate enough memory with the standard function `malloc()` (see 4.5.5).
Two functions are needed to determine the number of data points stored in a dataset, `H5LTget_dataset_ndims()` and `H5LTget_dataset_info()` .
The first returns the dimensionality of a dataset.

```
herr_t H5LTget_dataset_ndims ( hid_t loc_id,
                               const char dset_name,
                               int *rank              ) ;
```

where *loc_id* is the file ID, *dset_name* is the datset name, *rank* is filled with the dimensionality of the dataset. A negative return value indicates failure.
The second returns the size of each dimension, the data class and the data size.

```
herr_t H5LTget_dataset_info ( hid_t loc_id,
                              const char dset_name,
                              hsize_t *dims,
                              H5T_class_t *class_id,
                              size_t *type_size      ) ;
```

where *loc_id* is the file ID, *dset_name* is the dataset name, *dims* is filled with the size of each dimension, *class_id* is filled with the data class identifier, *type_size* is filled with the size of the datatype in bytes. A negative return value indicates failure. *dims* must point to previously allocated memory of sufficient size (sizeof(size_t)*rank) where rank has been determined by a call to `H5LTget_dataset_ndims()` .

4.5.3 Attribute Access

An attribute is primarily used to provide single-valued metadata information to another HDF5 object (group or dataset). The value of the attribute is retrieved by a call to the function `H5LTget_attribute_[type]()`:

```
herr_t H5LTget_attribute_[type]( hid_t loc_id,
                                  const char *obj_name,
                                  const char *attr_name,
                                  [type] *data           ) ;
```

where ***[type]*** can take one of the following values {string,char, short, int, long, float, double}, ***loc_id*** is the file ID, ***obj_name*** is the name of the object that the attribute is attached to, ***dset_name*** is the name of the attribute, ***buffer*** is the location in memory where the attribute value will be written to. A negative return value indicates failure.

4.5.4 Data Types

One of the aims of HDF is portability, meaning that files can be created on one computer, and read on another. This is not obvious for binary files, if the memory layout of multi-byte data differ in the two machines (little-endian vs big-endian format). But the HDF library takes care of the conversion. Nevertheless, when reading values from an HDF file, the memory data type must be supplied as one of the arguments to `H5Dread()` or `H5Aread()`. This is so because the memory data type must be compatible with the data type stored in the HDF file for the library to be able to perform the conversion. For example, the library will automatically convert from signed big-endian two-byte integer (as stored in the HDF file) to a signed four-byte little-endian memory type; but, trying to retrieve float data (as stored in the HDF file) into an integer memory data type will generate an error.

Number Data The following table indicates memory data type ID (the value of argument ***mem_type_id*** in the "read" functions) compatible with the given stored data type; the third column indicates the corresponding C types (the type of argument ***buffer***).

Data Type Stored in HDF File	Memory Data Type	C Type
H5T_IEEE_F64BE	H5T_NATIVE_DOUBLE	double
H5T_STD_I16BE	H5T_NATIVE_SHORT	short
H5T_STD_U8BE	H5T_NATIVE_UCHAR	unsigned char
H5T_STD_I8BE	H5T_NATIVE_CHAR	char

Character String Attributes The C language uses the term char to represent one-byte numeric data and does not make character strings a first-class datatype. HDF5 makes a distinction between integer and character data and maps the C signed char (`H5T_NATIVE_CHAR`) and unsigned char (`H5T_NATIVE_UCHAR`) datatypes to the HDF5 integer type class.

When the value of an attribute is a character string, to be able to allocate memory to fit the whole string, we must know its length. Actually the string length is stored by the library as a part of the type information. The size of a string can be retrieve using the following function `H5LTget_attribute_info()`:

```
herr_t H5LTget_attribute_info(  hid_t loc_id,
                                const char *obj_name,
                                const char *attr_name,
                                hsize_t *dims,
                                H5T_class_t *type_class,
                                size_t *type_size        ) ;
```

where ***loc_id*** is the file ID, ***obj_name*** is the name of the object that the attribute is attached to, ***attr_name*** is the name of the attribute, ***dims*** is `NULL`, ***type_class*** is the location in memory where the class identifier will be written to (should be `H5T_STRING` for a string), ***type_size*** is the location in memory where the size of the string will be written to.

Character String Dataset The character string dataset are not completely handle by the Lite API. This paragraph introduce briefly some concepts to understand how to retrieve these datasets. The information of a character string dataset can be retrieve using the same functions as for datasets `H5LTget_dataset_ndims()` and `H5LTget_dataset_info()` (see 4.5.2). The size of one data is the number of characters including the terminating 0 of each string of the dataset. A dataset is just like a string array in C: the strings have the same length and are null terminated. To retrieve a character string dataset, the size of each string must be given as argument. This is done trough the definition of a type. This type should belong to the class of null terminated strings (which is `H5T_C_S1`) and have the adequate size. This is done through the follwing functions:

```
hid_t H5Tcopy (  hid_t type_id  ) ;
```

where ***type_id*** is the datatype to copy (`H5T_C_S1` for null terminated strings). A negative return value indicates failure.

```
herr_t H5Tset_size(  hid_t type_id,
                     size_t size     ) ;
```

where ***type_id*** is the identifier of datatype to change size , ***size*** size in bytes to modify datatype. A negative return value indicates failure.
The datatype identifier returned should be released with `H5Tclose` or resource leaks will occur.

```
herr_t H5Tclose(  hid_t type_id  ) ;
```

where ***type_id*** is the identifier of datatype to release. A negative return value indicates failure.

4.5.5 Example

This section presents the listing of an example C program that will read the HDF file *"demo.hdf"* provided with this guide. The code shows how to use the functions of the HDF5 API discussed in the preceding sections. Namely, it demonstrates how an application can open an HDF file, and access its groups, datasets and attributes. The file *"read_demo_file.c"* contains the source code of the example. To compile it, you should have the HDF5 Library installed (see section 4.3). Please refer also to appendix A for additional examples. There are two versions of the same data extraction in this file: one using dynamic allocation of memory and the other using static arrays. Using only static arrays can lead to memory availability problems.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


   /* HDF5 Library header file. */
#include <hdf5.h>
#include <H5LT.h>

   /* The 'assert' macro is used to exit from the program in case
      something went wrong. */
#include <assert.h>


#define DIM_0  20
#define DIM_1  10

int main( void )
{
      /* Name of the HDF file. */
   char *file_name = "demo.hdf" ;

      /* IDs for accessing HDF5 objects. */
   hid_t f_id ;

      /* Will contain information about the dataset (number of dimensions,
         number of data points, dimensions sizes). */
   int num_dims ;
   hsize_t num_points, *dims;

      /* Will contain the size in bytes of one datasat data */
   size_t type_size;

      /* Will contain the data from ''/First Group/My Dataset''. */
   int *data = NULL, data_stat[DIM_0][DIM_1] ;
      /* Will contain the class identifier of the dataset
               Outside the scope of this simple demo*/
      H5T_class_t class_id;

      /* Will contain the value of attribute
         ''/First Group/My Dataset/Some Attribute''. */
   double attr_double ;

      /* Will contain number of characters in attribute
         ''/First Group/My Dataset/Some Other Attribute''. */
   size_t attr_string_size ;

      /* Will contain the value of attribute
         ''/First Group/My Dataset/Some Other Attribute''. */
   char *attr_string_dyn = NULL, attr_string_stat[256] ;

      /* Error reporting. */
   herr_t error ;

     /* Will contain the data type of attribute
         ''/First Group/My Dataset/Some Other Attribute''. */
   H5T_class_t type_class;

   int n, i, j ;

      /* Open the file. */
   f_id = H5Fopen( file_name, H5F_ACC_RDONLY, H5P_DEFAULT ) ;
   assert( f_id > 0 ) ;

      /*** Reading Dataset ***/
      /* dynamic allocation */
      /*------------------*/
```

```
        /* Retrieve the number of dimensions. */
    error = H5LTget_dataset_ndims(f_id,"/First Group/My Dataset",&num_dims);
    assert( !(error < 0) ) ;
    assert( num_dims==2) ;

    /* Allocate array to contain the size of each dimension. */
dims = (hsize_t *) malloc( num_dims * sizeof(hsize_t) ) ;

    /* Retrieve the dimension sizes. */
error = H5LTget_dataset_info(f_id,"/First Group/My Dataset", dims, &class_id, &type_size);
    assert( !(error < 0) ) ;

        /* Check that the dataset is of type integer 4 bytes. */
    assert(class_id==H5T_INTEGER);
    assert(type_size==sizeof(int));

        /* Compute number of data points stored in the dataset. */
    num_points = dims[0]*dims[1];

    /* Allocate enough memory to contain all the data in the dataset. */
data = (int *) malloc( num_points * sizeof(int) ) ;

        /* Read the data from the dataset. */
    error = H5LTread_dataset_int ( f_id, "/First Group/My Dataset", data );
    assert( !(error < 0) ) ;

    /*** Reading Dataset ***/
    /* static allocation */
    /*————————*/

error = H5LTread_dataset_int ( f_id, "/First Group/My Dataset", &(data_stat[0][0]) );
    assert( !(error < 0) ) ;

    /*** Reading Attribute "Some Attribute" ***/
    /*——————————————-*/

        /* Read contents of the attribute "Some Attribute". */
    error = H5LTget_attribute_double( f_id, "/First Group/My Dataset", "Some Attribute" , &attr_double);
    assert( !(error < 0));

    /*** Reading Attribute "Some Other Attribute" ***/
    /* dynamic allocation                          */
    /*————————————————-*/

    /* Check that it is a string and obtain the number of characters in the attribute value. */
error = H5LTget_attribute_info( f_id, "/First Group/My Dataset", "Some Other Attribute" , NULL, &type_class, &attr_string_siz
assert( !(error < 0) && (type_class==H5T_STRING) ) ;

   /* Allocate enough memory to contain the string. */
    attr_string_dyn = (char *) malloc( attr_string_size * sizeof(char) ) ;


   /* Obtain the value of the "Some Other Attribute" attribute. */
    error = H5LTget_attribute_string( f_id, "/First Group/My Dataset", "Some Other Attribute" , attr_string_dyn );
    assert( !(error < 0));

    /*** Reading Attribute "Some Other Attribute" ***/
    /* static allocation                           */
    /*————————————————-*/

    /* Obtain the value of the "Some Other Attribute" attribute. */
    error = H5LTget_attribute_string( f_id, "/First Group/My Dataset", "Some Other Attribute" , attr_string_stat );
    assert( !(error < 0));


    /* Do whatever with the data retrieved from the HDF file . . .
```

```
        Here we just print it. */
    printf("--- dynamic version ---\n");

    for ( n = 0 ; n < num_dims ; n++ )
        printf( "Size of dimension %d is %d\n", n, (int)dims[n] ) ;

    for ( i = 0 ; i < dims[0] ; i++ )
    {
        for ( j = 0 ; j < dims[1] ; j++ )
            printf( "%d\t", data[ i * dims[1] + j ] ) ;
        printf( "\n" ) ;
    }

    printf( "%.16f\n", attr_double ) ;
    printf( "%s\n", attr_string_dyn) ;

    printf("--- static version ---\n");

     for ( i = 0 ; i < dims[0] ; i++ )
    {
        for ( j = 0 ; j < dims[1] ; j++ )
            printf( "%d\t", data_stat[ i ][ j ] ) ;
        printf( "\n" ) ;
    }

    printf( "%.16f\n", attr_double ) ;
    printf( "%s\n", attr_string_stat ) ;

        /* Clean up. */
    free( data ) ;
    free( dims ) ;
    free( attr_string_dyn ) ;

    return 0 ;
}
```

When this program has been compiled, it should produce the output given in figure 4.

4.6 HDF Tools Provided by NCSA

Several utility programs are bundled with the HDF5 library. Among them, "`h5ls`" and "`h5dump`" are particularly useful for examining the contents of an HDF5 file. The transcript in figures 5 and 6 shows the output of these two programs on the HDF file *"demo.hdf"* provided with this document.

```
--- dynamic version ---
Size of dimension 0 is 20
Size of dimension 1 is 10
1       2       3       4       5       6       7       8       9       10
2       4       6       8       10      12      14      16      18      20
3       6       9       12      15      18      21      24      27      30
4       8       12      16      20      24      28      32      36      40
5       10      15      20      25      30      35      40      45      50
6       12      18      24      30      36      42      48      54      60
7       14      21      28      35      42      49      56      63      70
8       16      24      32      40      48      56      64      72      80
9       18      27      36      45      54      63      72      81      90
10      20      30      40      50      60      70      80      90      100
11      22      33      44      55      66      77      88      99      110
12      24      36      48      60      72      84      96      108     120
13      26      39      52      65      78      91      104     117     130
14      28      42      56      70      84      98      112     126     140
15      30      45      60      75      90      105     120     135     150
16      32      48      64      80      96      112     128     144     160
17      34      51      68      85      102     119     136     153     170
18      36      54      72      90      108     126     144     162     180
19      38      57      76      95      114     133     152     171     190
20      40      60      80      100     120     140     160     180     200
3.1415926535897931
This string is the value of the attribute.
--- static version ---
1       2       3       4       5       6       7       8       9       10
2       4       6       8       10      12      14      16      18      20
3       6       9       12      15      18      21      24      27      30
4       8       12      16      20      24      28      32      36      40
5       10      15      20      25      30      35      40      45      50
6       12      18      24      30      36      42      48      54      60
7       14      21      28      35      42      49      56      63      70
8       16      24      32      40      48      56      64      72      80
9       18      27      36      45      54      63      72      81      90
10      20      30      40      50      60      70      80      90      100
11      22      33      44      55      66      77      88      99      110
12      24      36      48      60      72      84      96      108     120
13      26      39      52      65      78      91      104     117     130
14      28      42      56      70      84      98      112     126     140
15      30      45      60      75      90      105     120     135     150
16      32      48      64      80      96      112     128     144     160
17      34      51      68      85      102     119     136     153     170
18      36      54      72      90      108     126     144     162     180
19      38      57      76      95      114     133     152     171     190
20      40      60      80      100     120     140     160     180     200
3.1415926535897931
This string is the value of the attribute.
```

Figure 4: Output of command "read_demo_file"

```
/demo.hdf/First\ Group   Group
/demo.hdf/First\ Group/My\ Dataset Dataset {20, 10}
```

Figure 5: Output of command "h5ls -r demo.hdf"

```
HDF5 "demo.hdf" {
GROUP "/" {
   GROUP "First Group" {
      DATASET "My Dataset" {
         DATATYPE  H5T_STD_I32BE
         DATASPACE  SIMPLE { ( 20, 10 ) / ( 20, 10 ) }
         DATA {
            1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
            2, 4, 6, 8, 10, 12, 14, 16, 18, 20,
            3, 6, 9, 12, 15, 18, 21, 24, 27, 30,
            4, 8, 12, 16, 20, 24, 28, 32, 36, 40,
            5, 10, 15, 20, 25, 30, 35, 40, 45, 50,
            6, 12, 18, 24, 30, 36, 42, 48, 54, 60,
            7, 14, 21, 28, 35, 42, 49, 56, 63, 70,
            8, 16, 24, 32, 40, 48, 56, 64, 72, 80,
            9, 18, 27, 36, 45, 54, 63, 72, 81, 90,
            10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
            11, 22, 33, 44, 55, 66, 77, 88, 99, 110,
            12, 24, 36, 48, 60, 72, 84, 96, 108, 120,
            13, 26, 39, 52, 65, 78, 91, 104, 117, 130,
            14, 28, 42, 56, 70, 84, 98, 112, 126, 140,
            15, 30, 45, 60, 75, 90, 105, 120, 135, 150,
            16, 32, 48, 64, 80, 96, 112, 128, 144, 160,
            17, 34, 51, 68, 85, 102, 119, 136, 153, 170,
            18, 36, 54, 72, 90, 108, 126, 144, 162, 180,
            19, 38, 57, 76, 95, 114, 133, 152, 171, 190,
            20, 40, 60, 80, 100, 120, 140, 160, 180, 200
         }
         ATTRIBUTE "Some Attribute" {
            DATATYPE  H5T_IEEE_F64BE
            DATASPACE  SCALAR
            DATA {
               3.14159
            }
         }
         ATTRIBUTE "Some Other Attribute" {
            DATATYPE  H5T_STRING {
                  STRSIZE 43;
                  STRPAD H5T_STR_NULLTERM;
                  CSET H5T_CSET_ASCII;
                  CTYPE H5T_C_S1;
               }
            DATASPACE  SCALAR
            DATA {
               "This string is the value of the attribute."
            }
         }
      }
   }
}
}
```

Figure 6: Output of command “h5dump demo.hdf”

A Sample Code

Before being able to use the scientific data produced at RMIB, the user must retrieve them from their HDF file, and format them according to his needs. This appendix presents sample code to help the user of the L2 RMIB GERB Products develop his own reading applications. Each example is self-contained (except for the HDF5 Library, which must be separately installed) and its compilation produces an executable program. Each code snippet is meant to illustrate a simple way to access a given HDF data structure and retrieve its data. It should be fairly straightforward for the user to adapt the examples to his own purposes.

Note: The programs produced out of these source codes must be given a single command line argument which is the name of an L2 RMIB GERB Product HDF file. This file must obviously contain the HDF structure that the program wants to read (see comments embedded in the code) or an error will occur.

A.1 Float Image

The following code illustrates how to extract a float image from quantised data stored in a 2-dimensional dataset. The quantisation factor is stored as a dataset attribute of type `double`. This is the factor by which each element from the dataset must be multiplied to obtain a meaningful value. In the example below, this means that, to obtain the physical value of the solar flux (in watts per square meter), one must multiply the dataset data (count value) by the value of the quantisation factor. The source code file name is *"read_rgp_float_dataset.c"*.

```
#include <stdio.h>
#include <stdlib.h>

   /* HDF5 Library header file. */
#include <hdf5.h>
#include <H5LT.h>

   /* The 'assert' macro is used to detect any error and exit from the program
      in case something went wrong. */
#include <assert.h>

#define XDIM 256
#define YDIM 256

int main( int argc, char *argv[] )
{
   char *file_name ;
   hid_t file_id;
   herr_t error ;
   int i, j ;
   int solar_flux_int16[YDIM][XDIM] ;
   float solar_flux[YDIM][XDIM] ;
   double quantisation_factor ;

      /* The argument on the command line is the name of an HDF file.
         The file must contain a "/Radiometry/Solar Flux" dataset of
         size 256 x 256. The dataset values are integers. Each value
         results from the quantisation of float values representing
         solar fluxes (in watts per meter square). The quantisation
         factor is stored as an attribute of the dataset. */
```

```
    assert( argc == 2 ) ;
    file_name = argv[1] ;

    /*** Open the HDF file. ***/

    file_id = H5Fopen( file_name, H5F_ACC_RDONLY, H5P_DEFAULT ) ;
    assert( !(file_id < 0) ) ;

    /*** Obtain the data stored in the "/Radiometry/Solar Flux" dataset. ***/

        error = H5LTread_dataset_int ( file_id, "/Radiometry/Solar Flux", solar_flux_int16[0]);
        assert( !(error < 0) ) ;

         /*** Obtain the value of the "Quantisation Factor" attribute inside the dataset. ***/

    error = H5LTget_attribute_double( file_id, "/Radiometry/Solar Flux", "Quantisation Factor", &quantisation_factor );
    assert( !(error < 0) ) ;


    /* Close the HDF file. */
    error = H5Fclose( file_id ) ;
    assert( !(error < 0) ) ;

    /*** Compute the float values of the solar flux and print them to standard output. ***/

    for ( i = 0 ; i < YDIM ; i++ )
    {
        for ( j = 0 ; j < XDIM ; j++ )
        {
            solar_flux[i][j] = quantisation_factor * solar_flux_int16[i][j] ;
            printf( "%.2f ", solar_flux[i][j] ) ;
        }

        printf( "\n" ) ;
    }

    return 0 ;
}
```

A.2 Integer Image

This code also extracts an image but does not perform integer to float conversion. The source code file name is *"read_rgp_integer_dataset.c"*.

```
#include <stdio.h>
#include <stdlib.h>

    /* HDF5 Library header file. */
#include <hdf5.h>
#include <H5LT.h>

    /* The 'assert' macro is used to detect any error and exit from the program
       in case something went wrong. */
#include <assert.h>

#define XDIM 256
#define YDIM 256

herr_t H5LTread_dataset_uchar( hid_t loc_id,
                               const char *dset_name,
                               char *data );

int main( int argc, char *argv[] )
{
    char *file_name ;
```

```
    hid_t file_id;
    herr_t error ;
    int i, j ;
    unsigned char surface_type[YDIM][XDIM] ;

        /* The argument on the command line is the name of an HDF file which
           must contain a "/Scene Identification/Surface Type" dataset of size
           256 x 256. The dataset values are integers. Each value is an index
           representing a surface type {0=undefined, 1=ocean, 2=dark vegetation,
                   3=bright vegetation, 4=dark desert, 5=bright desert, 6=snow, 255=error}. */
    assert( argc == 2 ) ;
    file_name = argv[1] ;

    /*** Open the HDF file. ***/

    file_id = H5Fopen( file_name, H5F_ACC_RDONLY, H5P_DEFAULT ) ;
    assert( file_id > 0 ) ;

    /*** Obtain the data stored in the
         "/Scene Identification/Surface Type" dataset. ***/

        error = H5LTread_dataset_uchar ( file_id, "/Scene Identification/Surface Type", surface_type[0]);
        assert( !(error < 0) ) ;


    /*** Print the surface type indices to standard output. ***/

    for ( i = 0 ; i < YDIM ; i++ )
    {
        for ( j = 0 ; j < XDIM ; j++ )
        {
            printf( "%d ", surface_type[i][j] ) ;
        }

        printf( "\n" ) ;
    }

    return 0 ;
}

herr_t H5LTread_dataset_uchar( hid_t loc_id,
                               const char *dset_name,
                               char *data )
{
 hid_t   dataset_id;

 /* Open the dataset. */
 if ( (dataset_id = H5Dopen( loc_id, dset_name )) < 0 )
  return -1;

 /* Read */
 if ( H5Dread( dataset_id, H5T_NATIVE_UCHAR, H5S_ALL, H5S_ALL, H5P_DEFAULT, data ) < 0 )
  goto out;

 /* End access to the dataset and release resources used by it. */
 if ( H5Dclose( dataset_id ) )
  return -1;

 return 0;

out:
 H5Dclose( dataset_id );
 return -1;

}
```

A.3 String Data

The following code shows how to retrieve the value of a string attribute. The source code file name is *"read_rgp_string.c"*.

```
#include <stdio.h>
#include <stdlib.h>

   /* HDF5 Library header file. */
#include <hdf5.h>
#include <H5LT.h>

   /* The 'assert' macro is used to detect any error and exit from the program
         in case something went wrong. */
#include <assert.h>


int main( int argc, char *argv[] )
{
    char        *file_name ;
    hid_t       file_id ;
    herr_t      error ;
    char        *attr_string_value = NULL ;
    size_t      attr_string_size ;
    H5T_class_t type_class;

        /* The argument on the command line is the name of an HDF file which
           must contain a "/File Creation Time" string attribute. */
    assert( argc == 2 ) ;
    file_name = argv[1] ;

    /* Open the HDF file. */
    file_id = H5Fopen( file_name, H5F_ACC_RDONLY, H5P_DEFAULT ) ;
    assert( file_id > 0 ) ;

    /* Check that it is a string and obtain the number of characters in the attribute value. */
    error = H5LTget_attribute_info( file_id, "/", "File Creation Time", NULL, &type_class, &attr_string_size );
    assert( !(error < 0) && (type_class==H5T_STRING) ) ;

    /* Allocate enough memory to contain the string. */
        attr_string_value = (char *) malloc( attr_string_size * sizeof(char) ) ;

    /* Obtain the value of the "/File Creation Time" attribute. */
        error = H5LTget_attribute_string( file_id, "/", "File Creation Time", attr_string_value );
        assert( !(error < 0));

    /* Close the HDF file. */
    error = H5Fclose( file_id ) ;
    assert( error == 0 ) ;

    /* Print the attribute string to standard output. */
    printf( "The file '%s' was created on --%s-- UTC time.\n",
            file_name, attr_string_value ) ;

    /* Release memory. */
    free( attr_string_value ) ;


    return 0 ;
}
```

A.4 String List

The following code retrieves a series of string attributes whose names are strings representing numbers. The source code file name is *"read_rgp_string_list.c"*.

```
# include <stdio.h>
# include <stdlib.h>

   /* HDF5 Library header file. */
# include <hdf5.h>
# include <H5LT.h>

   /* The 'assert' macro is used to detect any error and exit from the program
      in case something went wrong. */
# include <assert.h>

# define NUMCOLUMNS 256

int main( int argc, char *argv[] )
{
   char          *file_name ;
   hid_t         file_id ;
   herr_t        error ;
      hsize_t       dim, i;
   hid_t         atype;
   int           rank;
   char          *buffer=NULL;
   size_t        type_size;
   H5T_class_t class_id;

      /* The argument on the command line is the name of an HDF file which
         must contain a "/Times/Start of Integration (per column)" group
         which itself contains a list of 256 string attributes. The names of
         these attributes are the string representation (in base 10) of the
         numbers between 0 and 255 inclusive. */
   assert( argc == 2 ) ;
   file_name = argv[1] ;

   /*** Open the HDF file. ***/

   file_id = H5Fopen( file_name, H5F_ACC_RDONLY, H5P_DEFAULT ) ;
   assert( file_id > 0 ) ;


   /* obtain the dimensions of the dataset */
   error = H5LTget_dataset_ndims ( file_id, "/Times/Start of Integration (per column)", &rank );
      assert( !(error < 0) && (rank==1) ) ;

   /* obtain information about the dataset */
   error = H5LTget_dataset_info ( file_id, "/Times/Start of Integration (per column)", &dim, &class_id, &type_size );
   assert( !(error < 0) && (class_id==H5T_STRING) ) ;

   /* allocate the buffer */
   buffer = (char *) malloc(sizeof(char)*dim*type_size);

   /* create a data type according to the dataset type: */
      /* strings null terminated of size type_size */

      atype = H5Tcopy(H5T_C_S1);
      assert( !(atype < 0) ) ;

   error = H5Tset_size(atype, type_size);
      assert( !(error < 0) ) ;

      /* read the dataset */
   error = H5LTread_dataset ( file_id, "/Times/Start of Integration (per column)", atype, buffer );
```

```
        assert( !(error < 0));

    printf( "Start of Integration (UTC Time):\n" ) ;
    for ( i = 0 ; i < dim ; i++ )
    {
        printf( "%d:\t--%s--\n", (int)i,
                buffer+(i*type_size));
    }

    /*** Release memory. ***/

    free( buffer) ;
    H5Tclose(atype);

    return 0 ;
}
```

Index