Sentinel-3 SLSTR L1 and Marine L2 Products

Igor Tomažić[(1)], Anne O'Carroll[(1)], Prasanjit Dash[(1)], Francois Montagner[(1)], Vincenzo Santacesaria[(1)],
Tim Nightingale[(2)], , Jean-Francois Piollé[(3)], Craig Donlon[(4)] , Philippe Goryl [(5)]

(1)EUMETSAT, Eumetsat-Allee 1, 64295 Darmstadt, Germany,
(2)RAL Space, Rutherford Appleton Laboratory, Harwell Oxford
(3)IFREMER, 38 Rue du Port Blanc, 35800 Dinard, France
(4) ESA/ESTEC, Keplerlaan 1, 2201 AZ, Noordwijk, The Netherlands
(5) ESA/ESRIN, Frascati, Italy

Email: igor.tomazic@eumetsat.int

Introduction

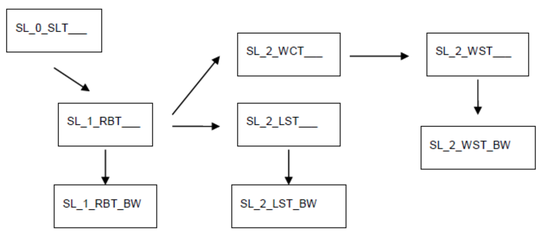
The European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) is contributing to the overall Sentinel-3 Mission Performance Activities in Commissioning (E1) and Routine Operations (E2) phases.

We are giving an overview of Sea and Land Surface Temperature Radiometer (SLSTR) L1 and marine L2 products. SLSTR set of products encompasses two user products, SLSTR L1B (SL_1_RBT___) and SLSTR L2P (SL_2_WST___) and one internal product (SL_2_WCT___) aimed for internal analysis and cal/val activities. The most complex of all Sentinel-3 products is SL_1_RBT____; that contains five different spatial resolution grids: 1 km, 500 m (A, B and TDI), tie point grid and two views: nadir, oblique and agnostic definition, spanning in total 111 files and almost 900 variables. On the opposite side is SL_2_WST___, the "simplest" sea surface temperature (SST) product, containing only two files and about 20 variables conforming to GHRSST (GDS2) specification. Finally, internal SLSTR L2 product (SL_2_WCT___) contains 20 files and about 100 variables associated with different SST algorithms and annotation data files.

SLSTR L1/L2 products

User Product Type	Number of Files	Number of MDFs	Number of ADFs	Number of variables
SL_1_RBT	111	34	76	~900
SL_2_WST	2	1	0	20
SL_2_WCT	20	5	15	~100
SL_2_LST	13	2	10	

•MDF – measurement data file
•ADF – annotation data file



SL_L1_RBT	SL_2_WCT	SL_2_WST
Measurement data files		
S[123]_radiance_an/ao S[456]_radiance_an/ao/bn/bo/cn/co S[789]_BT_in/io F[12]_BT_in/io	N2_SST_in, N3R_SST_in, N3_SST_in D2_SST_io, D3_SST_io	L2P
Annotation data files		
S1/S2/S3_quality_an/ao S4/S5/S6_quality_an/ao/bn/bo/cn/co S7/S8/S9/F1/F2_quality_in/io indices_an/ao/bn/bo/cn/co/in/io cartesian_an/ao/bn/bo/cn/co/in/io/tx	indices_in/io cartesian_in/io/tx	
flags_an/ao/bn/bo/cn/co/in/io	flags_in/io	
geodetics_an/ao/bn/bo/cn/co/in/io/tx	geodetic_in/io/tx	
time_an/bn/cn/in	time_in	
geometry_tn/to	geometry_tn/to	
met_tx Viscal	met_tx	
Total: 78 (112) = 22 (34) MDF + 54 (76) ADF + mfst+ report	Total: 22 = 5 MDF + 15 ADF + mfst+report	Total: 3= 1 MDF + mfst + report

Different grids: <g>
'i'=1 km Thermal IR;
'a'=500 m A stripe grid;
'b'=500 m B stripe grid;
'c'=500 m TDI grid;
't'=tie point grid (16 km)

Different views: <v>
'n'=nadir
'o'=oblique
'x'=agnostic

SL_1_RBT___ →

SL_1_RBT____ SAFE directory content

```
cartesian_an.nc   flags_in.nc      S1_quality_ao.nc  S5_quality_an.nc  S6_radiance_co.nc
cartesian_ao.nc   flags_io.nc      S1_radiance_an.nc S5_quality_ao.nc  S7_BT_in.nc
cartesian_bn.nc   geodetic_an.nc   S1_radiance_ao.nc S5_quality_bn.nc  S7_BT_io.nc
cartesian_bo.nc   geodetic_ao.nc   S2_quality_an.nc  S5_quality_bo.nc  S7_quality_in.nc
cartesian_cn.nc   geodetic_bn.nc   S2_quality_ao.nc  S5_quality_cn.nc  S7_quality_io.nc
cartesian_co.nc   geodetic_bo.nc   S2_radiance_an.nc S5_quality_co.nc  S8_BT_in.nc
cartesian_in.nc   geodetic_cn.nc   S2_radiance_ao.nc S5_radiance_an.nc S8_BT_io.nc
cartesian_io.nc   geodetic_co.nc   S3_quality_an.nc  S5_radiance_ao.nc S8_quality_in.nc
cartesian_tx.nc   geodetic_in.nc   S3_quality_ao.nc  S5_radiance_bn.nc S8_quality_io.nc
F1_BT_in.nc       geodetic_io.nc   S3_radiance_an.nc S5_radiance_bo.nc S9_BT_in.nc
F1_BT_io.nc       geodetic_tx.nc   S3_radiance_ao.nc S5_radiance_cn.nc S9_BT_io.nc
F1_quality_in.nc  geometry_tn.nc   S4_quality_an.nc  S5_radiance_co.nc S9_quality_in.nc
F1_quality_io.nc  geometry_to.nc   S4_quality_ao.nc  S6_quality_an.nc  S9_quality_io.nc
F2_BT_in.nc       indices_an.nc    S4_quality_bn.nc  S6_quality_ao.nc  time_an.nc
F2_BT_io.nc       indices_ao.nc    S4_quality_bo.nc  S6_quality_bn.nc  time_bn.nc
F2_quality_in.nc  indices_bn.nc    S4_quality_cn.nc  S6_quality_cn.nc  time_cn.nc
F2_quality_io.nc  indices_bo.nc    S4_quality_co.nc  S6_quality_co.nc  time_in.nc
flags_an.nc       indices_cn.nc    S4_radiance_an.nc S6_quality_in.nc  viscal.nc
flags_ao.nc       indices_co.nc    S4_radiance_ao.nc S6_radiance_an.nc xfdumanifest.xml
flags_bn.nc       indices_in.nc    S4_radiance_bn.nc S6_radiance_ao.nc
flags_bo.nc       indices_io.nc    S4_radiance_bo.nc S6_radiance_bn.nc
flags_cn.nc       met_tx.nc        S4_radiance_cn.nc S6_radiance_bo.nc
flags_co.nc       S1_quality_an.nc S4_radiance_co.nc S6_radiance_cn.nc
```

SL_2_WCT____ SAFE directory content

```
cartesian_in.nc  flags_in.nc      geometry_tn.nc  N2_SST_in.nc
cartesian_io.nc  flags_io.nc      geometry_to.nc  N3R_SST_in.nc
cartesian_tx.nc  geodetic_in.nc   indices_in.nc    N3_SST_in.nc
D2_SST_io.nc     geodetic_io.nc   indices_io.nc   time_in.nc
D3_SST_io.nc     geodetic_tx.nc   met_tx.nc       xfdumanifest.xml
```

SL_2_WST____ SAFE directory content

```
L2P.nc  xfdumanifest.xml
```

Type:	Manifest	Time stamps file (rows) - contains both nadir + oblique and i, a, b, c Quality annotation (detectors, integrators, rows) (band, view, grid) VISCAL (integrators, swir_detectors, visible detectors, views)	Radiances/BTs (band, grid, view) Flags (cloud, pointing, confidence, bayes) Geodetic (lon, lat, elevation) Indices (detector, pixel, scan) Cartesian (x,y)	TP geodetic (lon, lat) TP cartesian (x,y) TP geometry (sataz, satzen, solaz, solzen, satpath, solpath) TP meteo data (cloud, wind, sst, tcwv, ...)
Format:	XML	NetCDF 4		
Resolution:		Mixed	Image grid (pixels + orphans)	Tie point grid

Sentinel-3 SAFE

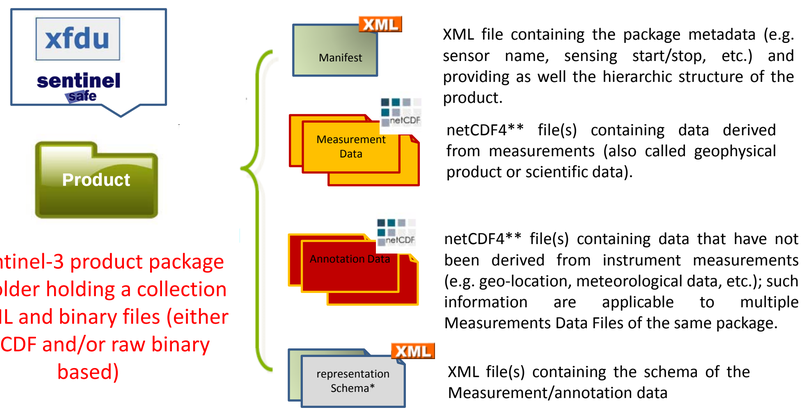
• SAFE specific to Sentinel-3 and to SLSTR/OLCI/SRAL

SAFE (Standard Archive Format for Europe)

•Designed to act as a common format for archiving and conveying data within Europe EO archiving capabilities
•Designed to be compliant with Open Archival Information System (OAIS)
• Instance of XML Formatted Data Units (XFDU)
• Restricts the XFDU specifications for specific utilization in the EO domain
•Use of XFDU (CCSDS 661.0-B-1, Blue Book, 09/2008)

XFDU (XML formatted data unit)

•Defined in **ISO 13527:2010:** Space data and information transfer systems
•XML formatted data unit (XFDU) structure and construction rules - defines how the packaging of data and metadata, including software, can be put into a single package (e.g. file or message) in order to facilitate information transfer and archiving.
•CCSDS (Consultative Committee for Space Data Systems) recommended XFDU as a standard for the packaging of data and metadata



A Sentinel-3 product package is a folder holding a collection of XML and binary files (either NetCDF and/or raw binary based)

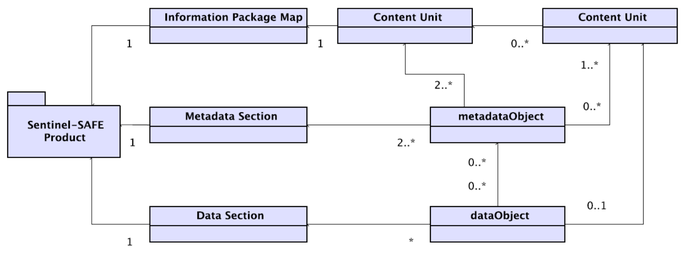
XML file containing the package metadata (e.g. sensor name, sensing start/stop, etc.) and providing as well the hierarchic structure of the product.

netCDF4** file(s) containing data derived from measurements (also called geophysical product or scientific data).

netCDF4** file(s) containing data that have not been derived from instrument measurements (e.g. geo-location, meteorological data, etc.); such information are applicable to multiple Measurements Data Files of the same package.

XML file(s) containing the schema of the Measurement/annotation data

Manifest file



The *Information Package Map contains the logical view of the package.*

The *Metadata Section records all of the metadata for all items in the package.*
Primary (common) and Secondary (specific to instrument/level) metadata

The *Data Object Section contains all the physical information needed to get the location of each file composing the package.*

Sentinel-3 filename convention

MMM_SS_L_TTTTTT_ <DATA_START>_<DATA_STOP>_<CREATION_TIME>_<instance_ID>_GGG_<classID>.<ext>

MMM – mission ID: **S3A** = Sentinel- 3A, **S3B** = Sentinel-3B, **S3_** = both Sentinel 3A and 3B

SS - data source: **OL** = OLCI, **SL** = SLSTR, **SR** = SRAL, **DO** = DORIS, **MW** = MWR, **GN** = GNSS, **SY** = Instruments Synergy, **TM** = telemetry data (e.g. HKTM, navigation, attitude, time), AX = for multi instrument auxiliary data

L - Processing level: "**0**" for Level-0, "**1**" for Level-1, "**2**" for Level-2

TTTTTT - Data Type ID: (EFR___, SLT___, RBT___, WST___, WCT___, ...) : suffix "AX " in the last 2 digits indicates an auxiliary data, suffix "BW" indicates a browse product.

Data Start time, stop time and creation time: **YYYYMMDDTHHMMSS**
Instance_ID: 17 chars: STRIPE or FRAME or TILE
STRIPE: **DDDD_CCC_LLL_____**
FRAME: **DDDD_CCC_LLL_FFFF**
TILE:
tile covering the whole globe: "GLOBAL_____"
tile cut according to specific geographical criteria: **ttttttttttttttt**

GGG - Product Generating Centre: '**LN1**', '**LN2**', '**MAR**', ...

<ClassID>: **P_XX_NNN** where:

P – platform: **O** for operational, **F** for reference, **D** for development, **R** for reprocessing or underscore "_" if not relevant.

XX - timeliness: **NR** for NRT, **ST** for STC, **NT** for NTC, ...

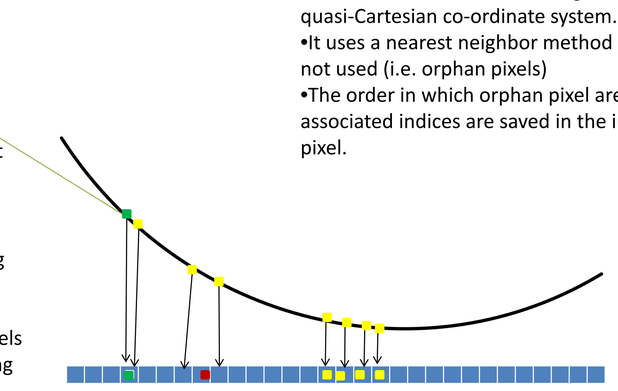
NNN – baseline collection

<ext>: extension: **SEN3**

Duration
"***DDDD" = 4 digits; orbit duration:***Sensing data time interval in seconds.
Cycle
"CCC"= 3 digits; cycle number at the start sensing time of the product
Frame along track coordinate
"**FFFF"= four digits; elapsed time in** seconds from the ascending node indicating the frame start time.
Tile identifier
"ttttttttttttttt"= 17 characters, either letters or digits or underscores "_" or any combination of them. It identifies the geographical area covered by the tile. There are two cases:
1) tile covers a pre-defined area of interest. (e.g. AFRICA_____)
2) tile covers an area according to a regular meshed predefined global grid (e.g. TILE_ID_001)

S3A_SL_1_RBT____20160523T094210_20160523T094510_20160523T113538_0179_004_250_2340_MAR_O_NR_001.SEN3
S3A_SL_1_RBT____20160531T225635_20160531T225935_20160601T010401_0179_004_372_1980_MAR_F_NR_001.SEN3
S3A_SL_1_RBT____20160523T094210_20160523T094510_20160525T022953_0179_004_250_2340_LN2_O_NT_001.SEN3
S3A_SL_2_WST____20160422T032431_20160422T032931_20160423T141910_0299_003_189_____MAR_O_NR_001.SEN3
S3A_SL_2_WCT____20160520T082344_20160520T082644_20160520T100158_0179_004_206_____MAR_O_NR_001.SEN3

Instrument → image grid



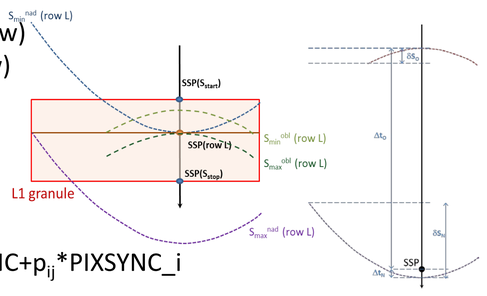
- Duplicate pixel - During the regriding process, if the pixel is already filled, set image pixel as well as orphan one to duplicate
- Cosmetic pixels - Pixels filled with cosmetic value, where they are missing (either from the re-gridding process or from missing or invalid data in the L0 product). It uses primarily adjacent pixels in the along track direction, or if missing in the across track one.

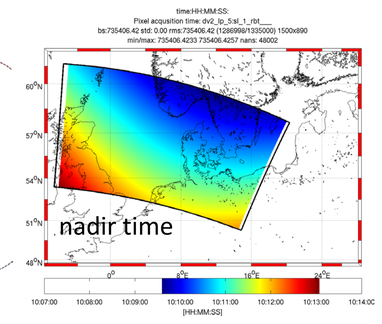
• Procedure remaps the measured nadir and along-track instrument pixels from their positions on the curved instrument scans to a uniform grid of points in the common quasi-Cartesian co-ordinate system.
•It uses a nearest neighbor method and retains pixels that are not used (i.e. orphan pixels)
•The order in which orphan pixel are stored does not matter - associated indices are saved in the image grid of each orphan pixel.

•measurements from the detectors are interleaved in a quasi-random fashion, determined by effect of the satellite orbit and the surface topology on the L1b regridding process.

SLSTR calculating time of every pixel

T_{init} = Nadir/Oblique_Minimal_ts_i(row)
S_{init} = Nadir/Oblique_First_scan_i(row)
SCANSYNC = Scan period (300 ms)
PIXSYNC_i = Pixel period (80 µs)
SSP = sub satellite point
s_{ij} = indices scan (row, col)
p_{ij} = indices pixel (row, col)
$time(row, col)=T_{init}+(s_{ij}-S_{init})*SCANSYNC+p_{ij}*PIXSYNC\_i$

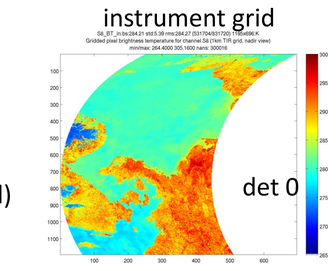


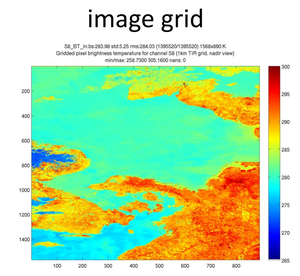


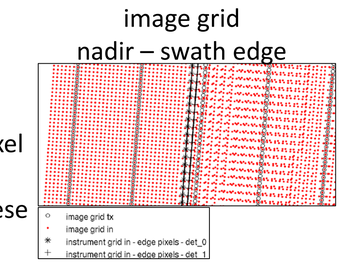
SLSTR image & instrument grid

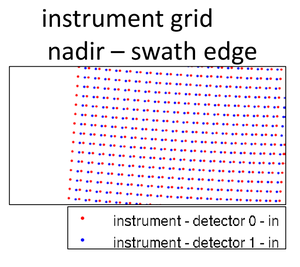
- all parameters indexed on image grid – continuity requirement
- remapping from instrument curved scans to uniform image grid in quasi-Cartesian system done using nearest neighbor method with retaining pixels that are not used (i.e. orphans)
-remapping keeps original pixel positions (nearest method) therefore image grid does not look so regular for higher satza (oblique view and nadir swath edge – see Figure)
-using image and orphan pixels, and information about scans, pixels, detectors and cosmetic fill pixels
it is possible to retrieve instrument grid:
a) for each detector and for every scan line retrieve corresponding true (not cosmetic) image and orphan pixel indices
b) get corresponding scan/pixel and detector values (these become indices in instrument grid)
c) use retrieved image and orphan indices to convert parameter on image grid to instrument grid

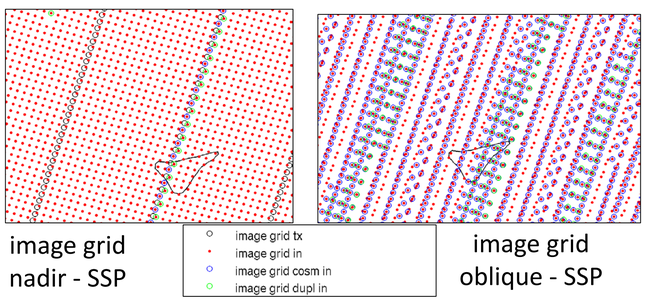
Note: Due to the current processing implementation following flags cannot be accurately remapped to instrument grid:
• bayes flags: all
• confidence flags: unfilled, day, twilight, snow
• cloud flags: histogram tests, spatial coherence, view difference tests
Default value for this flags is set to False (zero)!











SLSTR SAFE readers

Python

Cerbere: "free and open source python modules for the reading, interpretation, and writing of (primarly ocean) geophysical data."
•https://git.cersat.fr/cerbere/cerbere
•Felyx core component

```
from cerbere.mapper.safeslfile import SAFESLFile
fname ='S3A_SL_2_WCT___....SEN3'
fd = SAFESLFile(url=fname)
swath.load(fd)
lats = swath.get_lat()
lons = swath.get_lon()
times = swath.get_times()
```

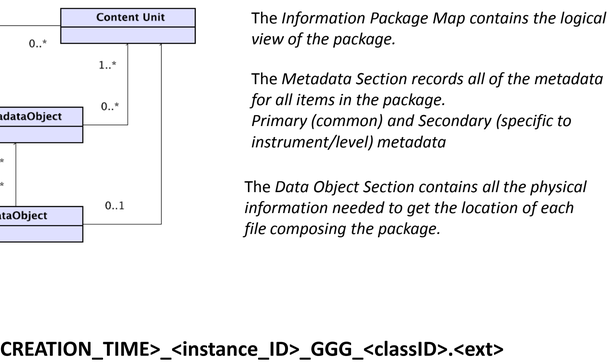
Matlab/Octave

- SL_1/2 and OL_1/2 product reader (available on demand)
- selective reading based on band, data type, grid, view
- unpacking flags
- parsing S3 filename, manifest
- output in structure (M/O)
- available on demand (internal development)

s3=s3Read(fn_s3_safe, bands, datas, grids, views,
'flags', { {FILENAME1, {{VARNAME1, {FLAGNAME1, FLAGNAME2, ..}}}}}
'band_flags', {EXCEPTION_FLAGS})

e.g.
s3=s3Read('S3_SL_1_RBT___... ', {'S7', 'S8', 'S9'}, {'BT', 'geodetic', 'flags'}, {'i'}, {'n', 'o'});
s3=s3Read('S3_SL_2_WST____...', {}, {'L2P'},{},{});
s3=s3Read('S3_SL_2_WCT_____...', {},{},{},{},{}) – read all from product

Sentinel Application Platform (SNAP)



http://step.esa.int/main/toolboxes/snap/

ODA downloader: ./s3GetFilesFromODA_pub.sh [download files from ODA using bash/wget]
MANDATORY:
-d <ODA DATASET>: e.g. PDGS_SL_1_RBT____NR, PDGS_OL_1_EFR____NR, PDGS_SL_2_WST____NR,....
-f <FILTER BY S3 SAFE NAME>: e.g data start stop (e.g. 20160523T102110_20160523T102410); orbit number (e.g. _194_), ...
OPTIONAL:
-c <FILTER BY S3 SAFE CONTENT>: comma separated list of filename patterns enclosed by quotes e.g. "*.xml"; "*.xml,*_in.nc"; .... or predefined FILTERS: (currently F_SL_1_RBT_IR; F_OL_1_EFR_RGB, ...)
-u <oda username>; -p <oda password>; -s <output directory>
-t :TEST don't perform download only display full wget command line

References
SLSTR Level 1 & Level 2 Instrument Products Data Format Specification , S3IPF.PDS.005 , 1.11, 2015

Acknowledgements
We would like to acknowledge The European Commission Copernicus Programme; The European Space Agency; Scientists and Industry throughout Europe.
Special thank to Claire Henocq and to full EUMETSAT S3-OPIT team.