BEAM/SMOS-BOX
Maintenance and Evolution

EE To NetCDF Format Conversion User Guide

Version 1.0
5th Aug 2014

Prepared by

Tom Block, Ralf Quast
Brockmann Consult GmbH

Changelog

Issue	Changes	Delivered
1.0	Initial version	05.08.2014

Contents

1 Introduction

1.1 Document purpose and scope

This User Guide covers all relevant information concerning the EEToNetCDF conversion tool. Subsections of the document explain the output file format, the supported SMOS product types, and the various possibilities to use the conversion tool. A concise description of the different interfaces and the parameters available is given.

1.2 Acronyms and abbreviations

BEAM	Basic Envisat AATSR and MERIS Toolbox
BC	Brockmann Consult GmbH
BT	Brightness Temperature
DDDB	Datafile Descriptor Data Base
EEF	Earth Explorer File
EO	Earth Observation
ESA	European Space Agency
ESRIN	European Space Research Institute
GPF	BEAM Graph Processing Framework
GPT	BEAM Graph Processing Tool
OGC	Open Geospatial Consortium
PDGS	Payload Data Ground Segment
ROI	Region Of Interest
RS	Remote Sensing
SMOS	Soil Moisture and Ocean Salinity
VISAT	(BEAM) Visualisation and Analysis Tool
WKT	Well Known Text

1.3 References

1.3.1 Applicable Documents

[AD.1]	SMOS Level 1 and Auxiliary Data Products Specifications, issue 5.26	SO-TN-IDR-GS-0005
[AD.2]	SMOS Level 2 and Auxiliary Data Products Specifications, issue 7.2	SO-TN-IDR-GS-0006
[AD.3]	BEAM + SMOS-Box Review, issue 1.1	IDEAS-SER-TOO-REP-1201
[AD.4]	SMOS NRT Product Format Specification, issue 3.8	SO-ID-DMS-GS-0002
[AD.5]	BEAM/SMOS-Box Maintenance and Evolution Requirements Baseline (RB)	

1.3.2 Reference documents

[RD.1]	NetCDF Java Library Documentation (version 4.3)	http://www.unidata.ucar.edu/software/

		thredds/current/netcdf-java/documentation.htm
[RD.2]	Unidata NetCDF best practices	http://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html
[RD.3]	NetCDF CF conventions (version 1.1)	http://cfconventions.org/
[RD.4]	Well Known Text format description	http://en.wikipedia.org/wiki/Well-known_text
[RD.5]	Beam GPT documentation	http://www.brockmann-consult.de/beam-wiki/display/BEAM/Bulk+Processing+with+GPT

2 Motivation

This SMOS Earth Explorer to NetCDF converter software shall enable a broader range of tools to make use of the SMOS data. Therefore, the widely supported NetCDF 4 file format has been chosen as target format.

The Earth Explorer format as being distributed by ESA is well suited for the SMOS data and for certain architectures of processing, especially cell-by-cell Level 3 operations. Nevertheless, in other situations it is more convenient to access the measurement variables directly than to be forced to iterate over a sequence of structures. The converter tool performs this re-mapping of the data by flattening the structures and mapping variables to data arrays ordered by grid-point or by snapshot.

3 Output file format

The converter output file format is NetCDF 4 with the option of writing the data in different compression levels.

The data structure as present in the original Earth Explorer binary data files is not suited to be directly transformed to NetCDF. The essential structure in the SMOS EEF product format is the grid point, which contains all measurements that were acquired for that grid point by different snapshots. This original data structure can roughly be described as a “list of structures that contains lists of structures”. Although, the format is perfectly suitable to represent the SMOS data, it needs to be modified to match the requirements of users.

Therefore, the NetCDF file contains a serialised version of the structured data. Each grid point or snapshot data variable is transformed into a NetCDF variable with an appropriate dimension. In the case of e.g. L1C Brightness Temperature (BT) measurements, the structure member is translated into a two-dimensional array, one dimension of this array is the number of grid-point measurements in the EE file, the other is the maximal number of snapshot measurements in all grid points of the product.

All size reference variables translate into NetCDF dimensions; all structures are flattened. Array data with a variable dimension (like e.g. Brightness Temperature data for grid points) translate into NetCDF arrays with a fixed dimension (either set to the maximum value allowed by the data type or to the maximum value occurring in a file).

Variable attributes in NetCDF files like scaling, units, fill values, valid ranges, flag masks, and flag meanings are defined according to the product specifications.

3.1 Metadata

All metadata contained in the Earth Explorer file is transferred to the NetCDF file. In contrast to the XML-based metadata in the original file, NetCDF does not allow for structured global metadata elements. Therefore, the inherent structure is mapped to the metadata attribute names. Any metadata attribute originally contained in a structure will be converted to a NetCDF attribute whose name is preceded by the structure name, separated by a colon (“:”), nested structures are treated recursively, according to this rule.

Example:

The “*Validity_Start*” metadata-element contained in the “*Validity_Period*” structure nested within the “*Fixed_Header*” structure is stored in the NetCDF file as a global attribute, which is denoted "*Fixed_Header:Validity_Period:Validity_Start*".

3.2 Dimensions

A NetCDF file requires all dimensions being used for variables to be stored as global meta-information. The dimension names chosen for the output file reflect the entities stored from the Earth Explorer file and are self-explaining.

Table 1: Dimensions of NetCDF file

Dimension Name	Description
n_grid_points	Number of grid points stored. First dimension for all grid-point structure data members
n_bt_data	Number of brightness temperature measurements per grid point. Second dimension for all grid-point structure data members
n_snapshots	Number of snapshots. Dimension used for all snapshot related variables
n_radiometric_accuracy	Number of radiometric accuracy measurements

3.3 Invalid-pixel values

Not all values in a variable array contain valid measurement data; this is especially true for the grid-point brightness temperature measurements where a varying number of measurements are stored in an array of fixed dimensions.

Array fields not containing valid measurement data contain an invalid pixel value that is defined for each variable independently. A variable that has an invalid-pixel value defined owns an attribute named *"_FillValue"* that contains the invalid pixel value for this variable (RD.3). If no fill value is defined, invalid pixels contain a zero value.

3.4 Flag codings

Some of the variables in a SMOS product are flag variables. In addition to the raw flag data, these variables contain attributes that describe how the flag values are interpreted. The attribute naming follows the NetCDF CF conventions (RD.3).

Table 2: CF compliant variable attributes for flags

Attribute Name	Description
flag_masks	Comma separated list of binary masks. The boolean conditions are identified by performing bitwise AND of the variable value and the flag_masks. The data type of the mask must match the data type of the associated variable.
flag_meanings	Space-separated list of interpretations corresponding to each of the flag_values and/or flag_masks.
flag_values	Comma-separated list of map values. Flag_values maps each value in the variable to a value in the flag_meanings in order to interpret the meaning of the value in the array.

3.5 Data types

Wherever possible, the converted NetCDF variables will have the same data type as the EE file structure members originally defined. In some cases a type promotion has to be applied. This is the case for all unsigned integer data types, which lack support from the NetCDF Java API used.

Following the best practices document by Unidata (RD.2), these variables are stored using their signed data type counterparts and adding an attribute *"_Unsigned = true"*.

3.6 Variable scaling

When a variable value stored in the NetCDf file requires mathematical operations to be transformed to a value reflecting a physical unit, this is indicated by variable attributes.

The standard operation is the linear transformation. When this transformation is required, this is indicated be the two attributes "*scale_factor*" and "*scale_offset*". The transformation to be applied is in this case expressed as

$$value_{phys} = scale\_factor * value_{stored} + scale\_offset$$

Other operations are not required for the current version of the converter software.

3.7 Variable name conversion

Some of the variable names present in the Earth Explorer file structures cannot directly be mapped to the NetCDF variable names as the original names violate the NetCDF naming conventions. For these variables, the original variable name is converted to a compatible name, as close to the original as possible.

Table 3: Variable name conversions

Original Name	Converted Name
Tb_42.5H	Tb_42_5H
Sigma_Tb_42.5H	Sigma_Tb_42_5H
Tb_42.5V	Tb_42_5V
Sigma_Tb_42.5V	Sigma_Tb_42_5V
Tb_42.5X	Tb_42_5X
Sigma_Tb_42.5X	Sigma_Tb_42_5X
Tb_42.5Y	Tb_42_5Y
Sigma_Tb_42.5Y	Sigma_Tb_42_5Y

4 Supported SMOS product types

The converter software is designed to be backward compatible. Internal conversion parameters and the variables, types and dimensions to be written into the target product are read from the associated file format schema files. These are available either from the internal file format database (DDDB) or an external extension of it (please refer to the SMOS-Box online manual available from the VISAT help menu).

The conversion software supports conversion of SMOS L1C and L2 user product formats. The SMOS-data can be read either form Earth Explorer *.HRD/*.DBL file pairs or from zip-compressed products. A detailed list of the supported types and schema versions is given below.

Table 4: Supported product types and schema versions

Product type	BinX schema versions supported
MIR_BWLD1C	200, 201, 300, 400
MIR_BWLF1C	200, 201, 300, 400
MIR_BWND1C	200
MIR_BWNF1C	200
MIR_BWSD1C	200, 201, 300, 400
MIR_BWSF1C	200, 201, 300, 400
MIR_OSUDP2	200, 300, 400
MIR_SCLD1C	200, 201, 300, 400
MIR_SCLF1C	200, 201, 300, 400
MIR_SCND1C	200
MIR_SCNF1C	200
MIR_SCSD1C	200, 201, 300, 400
MIR_SCSF1C	200, 201, 300, 400
MIR_SMUDP2	200, 201, 202, 300, 400

5 Functionality

5.1 Geographic sub-setting

When desired, the converter can apply a geographic subset according to a user supplied Region of Interest (ROI). During the conversion process, the software compares each grid-point location with the ROI and writes only those contained in it to the target file. The current implementation supports ROIs consisting of either Polygons or Multi-Polygons. The polygons have to be passed as textual conversion arguments using the OGC defined Well Known Text (WKT) format. This is described in detail at [RD.4].

An example polygon:

POLYGON((lon1 lat1, lon2 lat2, ... , lon1 lat1))

For L1C science data, a geographic sub-setting is eventually followed by an associated sub-setting of the snapshot information stored. During the geographic processing, the converter keeps track of all snapshots that are covered by the grid-cell measurements written to the output file. A subsequent step reduces the list of all available snapshot information to keep only those that are referenced by measurement data exported.

5.2 Variable sub-setting

The NetCDF Converter software allows users to convert only a subset of the original variables contained in the Earth Explorer file. This is achieved by adding a comma-separated list of variable names desired to the command (please refer to 6 for details). Please note that the available variable names differ from the band names displayed in Visat. Visat displays the SMOS data as interpreted variable bands (e.g. polarisations applied) projected onto a rectangular longitude/latitude raster. The converter instead directly reads and writes the variable data as defined in the Earth Explorer BinX schema files. These may be obtained from ESA.

A list of variable names per product type can be found below; this table lists the variables as defined by the latest schema versions (version 400). Variable naming and availability may differ for older file versions.

Table 5: Variable names for products using schema version V400

Product Type	Variable Names
MIR_BWLD1C, MIR_BWLF1C, MIR_BWSD1C, MIR_BWSF1C	Flags, BT_Value, Radiometric_Accuracy_of_Pixel, Azimuth_Angle, Footprint_Axis1, Footprint_Axis2, Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude, Grid_Point_Altitude, Grid_Point_Mask, BT_Data_Counter
MIR_OSUDP2	Dg_chi2_1, Dg_chi2_2, Dg_chi2_3, Dg_chi2_Acard, Dg_chi2_P_1, Dg_chi2_P_2, Dg_chi2_P_3, Dg_chi2_P_Acard, Dg_quality_SSS_1, Dg_quality_SSS_2, Dg_quality_SSS_3, Dg_quality_Acard, Dg_num_iter_1, Dg_num_iter_2, Dg_num_iter_3, Dg_num_iter_4, Dg_num_meas_l1c, Dg_num_meas_valid, Dg_border_fov, Dg_af_fov, Dg_sun_tails, Dg_sun_glint_area, Dg_sun_glint_fov, Dg_sun_fov, Dg_sun_glint_L2, Dg_Suspect_ice, Dg_galactic_Noise_Error, Dg_sky, Dg_moonglint, Dg_RFI_L1, Dg_RFI_X, Dg_RFI_Y, Dg_RFI_probability, X_swath, Equiv_ftprt_diam, Mean_acq_time, SSS1, Sigma_SSS1, SSS2, Sigma_SSS2, SSS3, Sigma_SSS3, A_card, Sigma_Acard, WS, SST, Tb_42.5H, Sigma_Tb_42.5H, Tb_42.5V, Sigma_Tb_42.5V, Tb_42.5X, Sigma_Tb_42.5X, Tb_42.5Y, Sigma_Tb_42.5Y, Grid_Point_ID, Latitude, Longitude, Control_Flags_1, Control_Flags_2, Control_Flags_3, Control_Flags_4, Science_Flags_1, Science_Flags_1, Science_Flags_2, Science_Flags_3, Science_Flags_4
MIR_SCLD1C, MIR_SCLF1C, MIR_SCSD1C, MIR_SCSF1C	Software_Error_flag, Instrument_Error_flag, ADF_Error_flag, Calibration_Error_flag, Days, Seconds, Microseconds, Flags, BT_Value, Pixel_Radiometric_Accuracy, Incidence_Angle, Azimuth_Angle, Faraday_Rotation_Angle, Geometric_Rotation_Angle, Snapshot_ID_of_Pixel, Footprint_Axis1, Footprint_Axis2, Snapshot_ID, Snapshot_OBET, X_Position, Y_Position, Z_Position", X_Velocity, Y_Velocity, Z_Velocity, Vector_Source, Q0, Q1, Q2, Q3, TEC, Geomag_F, Geomag_D, Geomag_I, Sun_RA, Sun_DEC, Sun_BT, Accuracy, Radiometric_Accuracy, X-Band, Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude, Grid_Point_Altitude, Grid_Point_Mask, BT_Data_Counter
MIR_SMUDP2	Days, Seconds, Microseconds, DGG_Current_Flags, Tau_Cur_DQX, HR_Cur_DQX, N_RFI_X, N_RFI_Y, RFI_Prob, Processing_Flags, S_Tree_1, S_Tree_2, Science_Flags, N_Sky, Confidence_Flags, GQX, Chi_2, Chi_2_P, N_Wild, M_AVA0, M_AVA, AFP, N_AF_FOV, N_Sun_Tails, N_Sun_Glint_Area, N_Sun_FOV, N_RFI_Mitigations, N_Strong_RFI, N_Point_Source_RFI, N_Tails_Point_Source_RFI, N_Software_Error, N_Instrument_Error, N_ADF_Error, N_Calibration_Error, N_X_Band, Soil_Moisture, Soil_Moisture_DQX, Optical_Thickness_Nad, Optical_Thickness_Nad_DQX, Surface_Temperature, Surface_Temperature_DQX, TTH, TTH_DQX, RTT, RTT_DQX, Scattering_Albedo_H, Scattering_Albedo_H_DQX, DIFF_Albedos, DIFF_Albedos_DQX, Roughness_Param, Roughness_Param_DQX, Dielect_Const_MD_RE, Dielect_Const_MD_RE_DQX, Dielect_Const_MD_IM, Dielect_Const_MD_IM_DQX, Dielect_Const_Non_MD_RE, Dielect_Const_Non_MD_RE_DQX, Dielect_Const_Non_MD_IM, Dielect_Const_Non_MD_IM_DQX, TB_ASL_Theta_B_H, TB_ASL_Theta_B_H_DQX, TB_ASL_Theta_B_V, TB_ASL_Theta_B_V_DQX, TB_TOA_Theta_B_H, TB_TOA_Theta_B_H_DQX, TB_TOA_Theta_B_V, TB_TOA_Theta_B_V_DQX, Grid_Point_ID, Latitude, Longitude, Altitude, X_Swath

6 Installations

6.1 BEAM Graph Processing Tool operator

The NetCDF conversion tool is implemented as a GPT operator. This allows using the converter in batch mode using the Graph Processing command line tool. Information about the GPT can be found in the VISAT main documentation, chapter Graph Processing Framework and online in the BEAM-Wiki: GPT bulk processing [RD.5].

The GPT is invoked from the command-line using the syntax described in the corresponding sections of the BEAM help. The conversion operator is invoked from GPT using the operator name "*SmosGP2NetCDF*".

The following table lists the operator parameters.

Table 6: GPT operator parameters

Name	Default Value	Description
sourceProducts	None	The source products to be converted. If not given, the parameter 'sourceProductPaths' must be provided.
sourceProductPaths	None	Comma-separated list of file paths specifying the source products. Each path may contain the wildcards '**' (matches recursively any directory), '*' (matches any character sequence in path names) and '?' (matches any single character).
targetDirectory	.	The target directory for the converted data. If not existing, directory will be created.
overwriteTarget	False	Set true to overwrite already existing target files.
region	None	Target geographical region as a geometry in well-known text format (WKT). The output product will be tailored according to the region.
institution	None	Set institution field for file metadata. If left empty, no institution metadata is written to output file.
contact	None	Set contact field for file metadata. If left empty, no contact information is written to output file.
outputBandNames	None	Comma separated list of band names to export. If left empty, no band sub-setting is applied.
compressionLevel	6	Output file compression level. 0 - no compression, 9 - highest compression.

6.2 Stand-alone program

Additionally, the converter software is distributed as a self-contained zip archive that allows an installation independent from BEAM/VISAT/SMOS-Box. This distribution comprises a command-line interface, allowing the tool to be integrated into various scripts or to be invoked from other third-party software.

To install the tool simply extract the content of the zip archive into a folder of your choice.

The conversion tool is invoked using a shell script file named smosEEToNetCDF.bat/.sh. The command line syntax is *smosEEToNetCDF [options] file ...*

When invoked without command line parameters, the conversion tool prints its usage to the console window. The possible options are listed in the table below.

Table 7: Stand-alone converter command line options

Option short name	Option long name	Argument	Default	Description
-	--compression-level	Integer	6	Target file compression level. 0 – no compression, 9 – highest compression
-	--contact	String	None	Contact information to be included in the global attributes of the target file.
-e	--errors	None	None	Produce execution error messages when program ends with an exit code different from 0.
-h	--help	None	None	Display help information
-	--institution	String	None	Institution information to be included in the global attributes of the target file.
-l	--log-level	String	INFO	Set the log-level, where the level must be one of {ALL, INFO, CONFIG, WARNING, SEVERE, OFF}
-	--overwrite-target	None	false	If set, an eventually existing target product will be overwritten without warning
-	--region	String	None	A region of interest (ROI) specified in geographic coordinates using well-known-text (WKT) format. The target product will only contain grid-cells data that is contained in the ROI.
-	--source-product-paths	String	None	Comma-separated list of file paths specifying the source products. Each path may contain the wildcards '**' (matches recursively any directory), '*' (matches any character sequence in path names) and '?' (matches any single character).
-	--target-directory	String	.	The directory where the target NetCDF files are written to.
-v	--version	None	None	Displays version information
-	--variables	String	None	A comma separated list of variables to be included into the target product. If left empty, all variables are converted.

Note: Command line parameters that require an argument composed of comma separated list of strings, like *"—variables*" or *"--source-product-paths*" should be entered either without blank characters between the comma and a value or the argument should be quoted.