S-3 Toolbox/SMOS-BOX
Maintenance and Evolution

NetCDF Format Conversion User Guide

Version 5.4.0
1st Mar 2017

Prepared by

Tom Block
Brockmann Consult GmbH

Changelog

Issue	Changes	Delivered
1.0	Initial version	11.08.2014
1.1	Adapt to S-3 Toolbox	26.09.2014
1.2	Updated to latest release, added Java recommendation	22.01.2016
5.3.0	Updated to support schema v7.03 data files, changed versioning system	24.10.2016
5.4.0	Science file export bugfix	01.03.2017

Contents

1 Introduction

1.1 Document purpose and scope

This User Guide covers all relevant information concerning the *smos-ee-to-nc* 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

BC	Brockmann Consult GmbH
BT	Brightness Temperature
CDL	Network Common Data form Language
DDDB	Data file Descriptor Data Base
EEF	Earth Explorer File
EO	Earth Observation
ESA	European Space Agency
ESRIN	European Space Research Institute
GPF	SNAP Graph Processing Framework
GPT	SNAP Graph Processing Tool
OGC	Open Geospatial Consortium
PDGS	Payload Data Ground Segment
ROI	Region Of Interest
RS	Remote Sensing
S-3	Sentinel 3
SMOS	Soil Moisture and Ocean Salinity
WKT	Well Known Text

1.3 References

1.3.1 Applicable Documents

[AD.1]	SMOS Level 1 and Auxiliary Data Products Specifications, issue 6.2, 31.08.2016	SO-TN-IDR-GS-0005
[AD.2]	SMOS Level 2 and Auxiliary Data Products Specifications, issue 8.4, 30.08.2016	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)	
[AD.6]	SMOS Toolbox upgrade for new L1C and L2OS data format	IDEAS-SMOS-CP-026

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]	SNAP 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.

3.1 Data Format

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) translates 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.

3.2 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*".

3.3 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

3.4 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.

3.5 Flag coding

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.

3.6 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*".

3.7 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 by 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.

3.8 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 backwards 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 SNAP SMOS-Box Toolbox 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, 401
MIR_SCLD1C	200, 201, 300, 400, 401
MIR_SCLF1C	200, 201, 300, 400, 401
MIR_SCND1C	200
MIR_SCNF1C	200
MIR_SCSD1C	200, 201, 300, 400, 401
MIR_SCSF1C	200, 201, 300, 400, 401
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 format 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 informations 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 the S-3 Toolbox. The S-3 Toolbox 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_BWSD1C**	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_BWLF1C, **MIR_BWSF1C**	Flags, BT_Value_Imag, BT_Value_Real, 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_SCSD1C**	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_SCLF1C, **MIR_SCSF1C**	Software_Error_flag, Instrument_Error_flag, ADF_Error_flag, Calibration_Error_flag, Days, Seconds, Microseconds, Flags, BT_Value_Imag, BT_Value_Real, 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

Note: A minimal set of variables is required to be able to open the exported file in the SMOS-Box. These variables are listed in the Table below:

Table 6: Required variables for SMOS-Box import

Product Type	Required Variables
MIR_BWLD1C, **MIR_BWSD1C**	Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude
MIR_BWLF1C, **MIR_BWSF1C**	Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude
MIR_OSUDP2	Grid_Point_ID, Latitude, Longitude
MIR_SCLD1C, **MIR_SCSD1C**	Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude, Flags, Incidence_Angle
MIR_SCLF1C, **MIR_SCSF1C**	Grid_Point_ID, Grid_Point_Latitude, Grid_Point_Longitude, Flags, Incidence_Angle
MIR_SMUDP2	Grid_Point_ID, Latitude, Longitude

5.3 Compression

The target file format NetCDF and the associated software library for reading and writing the files implements a built-in compression mechanism to reduce the storage size of the data on the hard-drive.

Data compression allows the user to balance between file size and conversion time. The higher the compression ratio, the longer it takes to calculate the inflation algorithm for the data. The same – of course – is true for reading access to the data files written.

The compression factor is adjustable as user parameter which ranges from 0 to 9. A factor of 0 means that no compression at all is applied. A factor of 9 implies that the highest data reduction is achieved, at the cost of higher CPU load.

The increased CPU load for compression algorithm execution goes in conjunction with a decreased time required for storage, as lesser data has to be written to the hard-drive. For each hardware-setup there is a specific optimum setting that minimizes the conversion time.

6 Installations

The conversion tool is delivered in two separate installations. Firstly, it is implemented as an operator to be used from the Sentinel Toolbox Graph Processing Tool. This installation is an add-on to a regular SNAP Toolbox /SMOS-Box installation. When using the GPT operator, the conversion tasks can be embedded into larger processing chains using the Graph Description File XML interface.

The second installation is a self-contained command line executable. This installation is completely independent from SNAP Toolbox /SMOS-Box and offers a simple command line interface.

Both installation possibilities are described in detail in the following chapters.

6.1 S-3 Toolbox 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 S-3 Toolbox main documentation, chapter Graph Processing Framework and online in the S-3 Toolbox-Wiki: GPT bulk processing [RD.5]. This converter installation is automatically integrated into S-3 Toolbox when using the standard installer for the SMOS-Box software.

The hard- and software requirements for this installation follow the guidelines of SNAP Toolbox. Please refer to the SNAP Toolbox manual for details.

The GPT is invoked from the command-line using the syntax described in the corresponding sections of the SNAP Toolbox help. The conversion operator is invoked from GPT using the operator name “*SmosNetcdfExport*”.

The following table lists the operator parameters.

Table 7: 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.
geometry	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.
variableNames	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.

6.1.1 Examples

The following examples assume a Windows operating system. The same examples apply to Linux or MacOS, just replace the operating system paths with the corresponding system specific paths.

Example 1: Conversion of a single file, output file will be written to gpt working directory

```
gpt SmosNetcdfExport
C:/data/SMOS/MIR_BWLF1C\SM_OPER_MIR_BWLF1C_20111026T143206_20111026T152520_503_001_1.zip
```

GPT allows using various methods to pass parameter; the following examples only use the approach of passing in a graph.xml file for the definition of processing parameters. For other methods, please refer to the S-3 Toolbox documentation [RD.5].

Assuming the graph definition is written to a file named “smos-conversion.xml”, calls to GPT for the next examples always are:

```
gpt smos-conversion.xml
```

or passing in an absolute path to the graph definition file if it is not in the current working directory.

Example 2: Conversion of all compressed products in a source directory to a target directory, forcing to overwrite already existing files, using geographic sub-setting. The example “smos-conversion.xml” is shown below:

```
<graph id="SMOS test conversion">
  <version>1.0</version>
  <node id="smos-conversion">
    <operator>SmosNetcdfExport</operator>

    <parameters>
      <sourceProductPaths>C:/Data/SMOS/*.zip</sourceProductPaths>
       <targetDirectory>C:/Data/result</targetDirectory>
      <overwriteTarget>true</overwriteTarget>
      <geometry>POLYGON((-22 5,-22 7,-20 7,-20 5,-22 5))</ geometry >
    </parameters>
  </node>
</graph>
```

Example 3: Conversion of all L1C full polarization science products in a source directory to a target directory, forcing to overwrite already existing files, using maximum compression level, using variables sub-setting. The example “smos-conversion.xml” is shown below:

```
<graph id="SMOS test conversion">
  <version>1.0</version>
  <node id="smos-conversion">
```

```
      <operator>SmosGP2NetCDF</operator>

      <parameters>
        <sourceProductPaths>C:/Data/SMOS/*MIR_SC?F1C*</sourceProductPaths>
        <targetDirectory>C:/Data/result</targetDirectory>
        <overwriteTarget>true</overwriteTarget>
        <compressionLevel>9</compressionLevel>
        <variableNames>BT_Value_Imag,BT_Value_Real </variableNames>
      </parameters>
    </node>
</graph>
```

6.2 Stand-alone program

Additionally, the converter software is distributed as a self-contained zip archive that allows an installation independent from SNAP/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.

The hard- and software requirements for this installation are described in the following table.

Table 8: Hard- and Software Requirements

	Minimum	Recommended
CPU	Intel Core i5	Intel Core i7, 3 GHz or better
RAM	4 GB	8 GB
Java Runtime	Java 8 (32bit)	Java 8 (64bit)

Important notes on the use of 32 bit Java Virtual Machines

Note: It is strongly recommended to use a 64bit Java Virtual Machine as it has been observed that a 32bit VM quickly reaches memory limits when converting large (especially L1C) files. If you are experiencing Java memory heap allocation failures using a Java 32bit VM, try to adapt the maximum heap size allocated by editing the shell script files smos-ee-to-nc.bat/.sh and adapt the maximal heap size by changing the value "-Xmx2048M". The optimal value for 32bit VMs is depending on the system status, 1280 MB is a good starting point.

Note: The use of a 32bit Java Virtual Machine is only recommended for the conversion of L1C Browse and L2 products.

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 smos-ee-to-nc.bat/.sh. The command line syntax is

```
smos-ee-to-nc [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 9: Stand-alone converter command line options

Option short name	Option long name	Argument	Default	Description
none	--compression-level	Integer	6	Target file compression level. 0 – no compression, 9 – highest compression

none	--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
none	--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}
none	--overwrite-target	none	false	If set, an eventually existing target product will be overwritten without warning
none	--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.
none	--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).
none	--target-directory	String	.	The directory where the target NetCDF files are written to.
-v	--version	none	none	Displays version information
none	--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.

6.2.1 Examples

The following examples assume a Linux operating system. The same examples apply to Windows, just exchange smos-ee-to-nc.sh with smos-ee-to-nc.bat.

Conversion of a single file with increased logging output:

```
        smos-ee-to-nc.sh -l ALL --target-directory /home/BC/data /usr/local/data/SMOS/
SM_OPER_MIR_SCSF1C_20120514T013734_20120514T023053_505_001_1.zip
```

Conversion of all Browse Products in a directory, recursive through all subdirectories:

```
        smos-ee-to-nc.sh  --target-directory /home/BC/data --source-product-paths /usr/local/data/**/*MIR_BW*
```

Conversion of a single product with variables sub-setting

```
        smos-ee-to-nc.sh  --target-directory /home/BC/data --variables
BT_Value,Grid_Point_Latitude,Grid_Point_Longitude
/usr/local/data/SMOS/SM_OPER_MIR_BWLF1C_20111026T143206_20111026T152520_503_001_1.HDR
```

Conversion of all L1C science products in a directory with geographic sub-setting

```
        smos-ee-to-nc.sh  --target-directory /home/BC/data --source-product-paths /usr/local/data/*SC*1C* --
region “POLYGON((-60 0, -60 10, -20 10,-20 0, -60 0))”
```

7 NetCDF file structure

This chapter lists the variables and variable attributes of the NetCDF products in detail. The data is based on converted Earth Explorer files of schema version V400. The file structure examples are written in NetCDF CDL.

7.1 MIR BWLF1C

```
dimensions:
    n_grid_points = 40821 ;
    n_bt_data = 4 ;
variables:
    ubyte Grid_Point_Mask(n_grid_points) ;
        Grid_Point_Mask:_Unsigned = "true" ;
        Grid_Point_Mask:_FillValue = NaNf ;
    float Grid_Point_Altitude(n_grid_points) ;
        Grid_Point_Altitude:_FillValue = NaNf ;
        Grid_Point_Altitude:units = "m" ;
    float Grid_Point_Latitude(n_grid_points) ;
        Grid_Point_Latitude:_FillValue = NaNf ;
        Grid_Point_Latitude:units = "deg" ;
    int Grid_Point_ID(n_grid_points) ;
        Grid_Point_ID:_Unsigned = "true" ;
        Grid_Point_ID:_FillValue = NaNf ;
    short Azimuth_Angle(n_grid_points, n_bt_data) ;
        Azimuth_Angle:_FillValue = 0.f ;
        Azimuth_Angle:scale_factor = 0.0054931640625 ;
        Azimuth_Angle:scale_offset = 0. ;
        Azimuth_Angle:_Unsigned = "true" ;
        Azimuth_Angle:units = "deg" ;
    float Grid_Point_Longitude(n_grid_points) ;
        Grid_Point_Longitude:_FillValue = NaNf ;
        Grid_Point_Longitude:units = "deg" ;
    short Flags(n_grid_points, n_bt_data) ;
        Flags:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
        Flags:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
        Flags:flag_meanings = "POL_FLAG_1 POL_FLAG_2 SUN_FOV SUN_GLINT_FOV MOON_GLINT_FOV
SINGLE_SNAPSHOT FTT SUN_POINT SUN_GLINT_AREA MOON_POINT AF_FOV EAF_FOV BORDER_FOV SUN_TAILS
RFI_1 RFI_
2" ;
        Flags:_Unsigned = "true" ;
        Flags:_FillValue = 0.f ;
    ubyte BT_Data_Counter(n_grid_points) ;
        BT_Data_Counter:_Unsigned = "true" ;
        BT_Data_Counter:_FillValue = NaNf ;
    short Footprint_Axis2(n_grid_points, n_bt_data) ;
        Footprint_Axis2:_FillValue = 0.f ;
        Footprint_Axis2:scale_factor = 1.52587890625e-005 ;
        Footprint_Axis2:scale_offset = 0. ;
        Footprint_Axis2:_Unsigned = "true" ;
        Footprint_Axis2:units = "km" ;
    short Footprint_Axis1(n_grid_points, n_bt_data) ;
        Footprint_Axis1:_FillValue = 0.f ;
```

```
            Footprint_Axis1:scale_factor = 1.52587890625e-005 ;
            Footprint_Axis1:scale_offset = 0. ;
            Footprint_Axis1:_Unsigned = "true" ;
            Footprint_Axis1:units = "km" ;
    float BT_Value(n_grid_points, n_bt_data) ;
            BT_Value:_FillValue = -999.f ;
            BT_Value:units = "K" ;
    short Radiometric_Accuracy_of_Pixel(n_grid_points, n_bt_data) ;
            Radiometric_Accuracy_of_Pixel:_FillValue = 0.f ;
            Radiometric_Accuracy_of_Pixel:scale_factor = 1.52587890625e-005 ;
            Radiometric_Accuracy_of_Pixel:scale_offset = 0. ;
            Radiometric_Accuracy_of_Pixel:_Unsigned = "true" ;
            Radiometric_Accuracy_of_Pixel:units = "K" ;
```

7.2 MIR BWSF1C

```
dimensions:
    n_grid_points = 116237 ;
    n_bt_data = 4 ;
variables:
    ubyte Grid_Point_Mask(n_grid_points) ;
            Grid_Point_Mask:_Unsigned = "true" ;
            Grid_Point_Mask:_FillValue = NaNf ;
    float Grid_Point_Altitude(n_grid_points) ;
            Grid_Point_Altitude:_FillValue = NaNf ;
            Grid_Point_Altitude:units = "m" ;
    float Grid_Point_Latitude(n_grid_points) ;
            Grid_Point_Latitude:_FillValue = NaNf ;
            Grid_Point_Latitude:units = "deg" ;
    int Grid_Point_ID(n_grid_points) ;
            Grid_Point_ID:_Unsigned = "true" ;
            Grid_Point_ID:_FillValue = NaNf ;
    short Azimuth_Angle(n_grid_points, n_bt_data) ;
            Azimuth_Angle:_FillValue = 0.f ;
            Azimuth_Angle:scale_factor = 0.0054931640625 ;
            Azimuth_Angle:scale_offset = 0. ;
            Azimuth_Angle:_Unsigned = "true" ;
            Azimuth_Angle:units = "deg" ;
    float Grid_Point_Longitude(n_grid_points) ;
            Grid_Point_Longitude:_FillValue = NaNf ;
            Grid_Point_Longitude:units = "deg" ;
    short Flags(n_grid_points, n_bt_data) ;
            Flags:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
            Flags:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
            Flags:flag_meanings = "POL_FLAG_1 POL_FLAG_2 SUN_FOV SUN_GLINT_FOV MOON_GLINT_FOV
SINGLE_SNAPSHOT FTT SUN_POINT SUN_GLINT_AREA MOON_POINT AF_FOV EAF_FOV BORDER_FOV SUN_TAILS
RFI_1 RFI_
2" ;
            Flags:_Unsigned = "true" ;
            Flags:_FillValue = 0.f ;
    ubyte BT_Data_Counter(n_grid_points) ;
            BT_Data_Counter:_Unsigned = "true" ;
            BT_Data_Counter:_FillValue = NaNf ;
    short Footprint_Axis2(n_grid_points, n_bt_data) ;
            Footprint_Axis2:_FillValue = 0.f ;
            Footprint_Axis2:scale_factor = 1.52587890625e-005 ;
            Footprint_Axis2:scale_offset = 0. ;
            Footprint_Axis2:_Unsigned = "true" ;
            Footprint_Axis2:units = "km" ;
    short Footprint_Axis1(n_grid_points, n_bt_data) ;
            Footprint_Axis1:_FillValue = 0.f ;
            Footprint_Axis1:scale_factor = 1.52587890625e-005 ;
            Footprint_Axis1:scale_offset = 0. ;
            Footprint_Axis1:_Unsigned = "true" ;
            Footprint_Axis1:units = "km" ;
    float BT_Value(n_grid_points, n_bt_data) ;
            BT_Value:_FillValue = -999.f ;
            BT_Value:units = "K" ;
    short Radiometric_Accuracy_of_Pixel(n_grid_points, n_bt_data) ;
            Radiometric_Accuracy_of_Pixel:_FillValue = 0.f ;
            Radiometric_Accuracy_of_Pixel:scale_factor = 1.52587890625e-005 ;
            Radiometric_Accuracy_of_Pixel:scale_offset = 0. ;
            Radiometric_Accuracy_of_Pixel:_Unsigned = "true" ;
            Radiometric_Accuracy_of_Pixel:units = "K" ;
```

7.3 MIR OSUDP2

```
dimensions:
    n_grid_points = 143890 ;
```

```
variables:
	float Sigma_SSS3(n_grid_points) ;
		Sigma_SSS3:_FillValue = -999.f ;
		Sigma_SSS3:units = "psu" ;
	float Sigma_SSS1(n_grid_points) ;
		Sigma_SSS1:_FillValue = -999.f ;
		Sigma_SSS1:units = "psu" ;
	short Dg_af_fov(n_grid_points) ;
		Dg_af_fov:_Unsigned = "true" ;
		Dg_af_fov:_FillValue = 0.f ;
	float Sigma_SSS2(n_grid_points) ;
		Sigma_SSS2:_FillValue = -999.f ;
		Sigma_SSS2:units = "psu" ;
	short Dg_border_fov(n_grid_points) ;
		Dg_border_fov:_Unsigned = "true" ;
		Dg_border_fov:_FillValue = 0.f ;
	int Grid_Point_ID(n_grid_points) ;
		Grid_Point_ID:_Unsigned = "true" ;
		Grid_Point_ID:_FillValue = 0.f ;
	short Dg_chi2_1(n_grid_points) ;
		Dg_chi2_1:_Unsigned = "true" ;
		Dg_chi2_1:_FillValue = 0.f ;
	short Dg_moonglint(n_grid_points) ;
		Dg_moonglint:_Unsigned = "true" ;
		Dg_moonglint:_FillValue = 0.f ;
	short Dg_chi2_2(n_grid_points) ;
		Dg_chi2_2:_Unsigned = "true" ;
		Dg_chi2_2:_FillValue = 0.f ;
	int Science_Flags_4(n_grid_points) ;
		Science_Flags_4:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
		Science_Flags_4:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
		Science_Flags_4:flag_meanings = "FG_SC_LAND_SEA_COAST1 FG_SC_LAND_SEA_COAST2
FG_SC_TEC_GRADIENT FG_SC_IN_CLIM_ICE FG_SC_ICE FG_SC_SUSPECT_ICE FG_SC_RAIN FG_SC_HIGH_WIND
FG_SC_LOW_WIND
FG_SC_HIGHT_SST FG_SC_LOW_SST FG_SC_HIGH_SSS FG_SC_LOW_SSS FG_SC_SEA_STATE_1
FG_SC_SEA_STATE_2 FG_SC_SEA_STATE_3 FG_SC_SEA_STATE_4 FG_SC_SEA_STATE_5 FG_SC_SEA_STATE_6
FG_SC_SST_FRONT FG_SC_SSS_FRONT F
G_SC_ICE_ACARD" ;
		Science_Flags_4:_Unsigned = "true" ;
		Science_Flags_4:_FillValue = 0.f ;
	short Dg_chi2_3(n_grid_points) ;
		Dg_chi2_3:_Unsigned = "true" ;
		Dg_chi2_3:_FillValue = 0.f ;
	int Science_Flags_3(n_grid_points) ;
		Science_Flags_3:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
		Science_Flags_3:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
		Science_Flags_3:flag_meanings = "FG_SC_LAND_SEA_COAST1 FG_SC_LAND_SEA_COAST2
FG_SC_TEC_GRADIENT FG_SC_IN_CLIM_ICE FG_SC_ICE FG_SC_SUSPECT_ICE FG_SC_RAIN FG_SC_HIGH_WIND
FG_SC_LOW_WIND
FG_SC_HIGHT_SST FG_SC_LOW_SST FG_SC_HIGH_SSS FG_SC_LOW_SSS FG_SC_SEA_STATE_1
FG_SC_SEA_STATE_2 FG_SC_SEA_STATE_3 FG_SC_SEA_STATE_4 FG_SC_SEA_STATE_5 FG_SC_SEA_STATE_6
FG_SC_SST_FRONT FG_SC_SSS_FRONT F
G_SC_ICE_ACARD" ;
		Science_Flags_3:_Unsigned = "true" ;
		Science_Flags_3:_FillValue = 0.f ;
	short Dg_chi2_Acard(n_grid_points) ;
		Dg_chi2_Acard:_Unsigned = "true" ;
		Dg_chi2_Acard:_FillValue = 0.f ;
	short Dg_Suspect_ice(n_grid_points) ;
		Dg_Suspect_ice:_Unsigned = "true" ;
		Dg_Suspect_ice:_FillValue = 0.f ;
	short Dg_chi2_P_Acard(n_grid_points) ;
		Dg_chi2_P_Acard:_Unsigned = "true" ;
		Dg_chi2_P_Acard:_FillValue = 0.f ;
	short Dg_RFI_probability(n_grid_points) ;
		Dg_RFI_probability:_FillValue = NaNf ;
		Dg_RFI_probability:_Unsigned = "true" ;
		Dg_RFI_probability:units = "%" ;
	int Science_Flags_1(n_grid_points) ;
		Science_Flags_1:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
		Science_Flags_1:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
```

```
        Science_Flags_1:flag_meanings = "FG_SC_LAND_SEA_COAST1 FG_SC_LAND_SEA_COAST2
FG_SC_TEC_GRADIENT FG_SC_IN_CLIM_ICE FG_SC_ICE FG_SC_SUSPECT_ICE FG_SC_RAIN FG_SC_HIGH_WIND
FG_SC_LOW_WIND
FG_SC_HIGHT_SST FG_SC_LOW_SST FG_SC_HIGH_SSS FG_SC_LOW_SSS FG_SC_SEA_STATE_1
FG_SC_SEA_STATE_2 FG_SC_SEA_STATE_3 FG_SC_SEA_STATE_4 FG_SC_SEA_STATE_5 FG_SC_SEA_STATE_6
FG_SC_SST_FRONT FG_SC_SSS_FRONT F
G_SC_ICE_ACARD" ;
        Science_Flags_1:_Unsigned = "true" ;
        Science_Flags_1:_FillValue = 0.f ;
    int Science_Flags_2(n_grid_points) ;
        Science_Flags_2:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
        Science_Flags_2:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s,
-32768s, 0s, 0s, 0s, 0s, 0s, 0s ;
        Science_Flags_2:flag_meanings = "FG_SC_LAND_SEA_COAST1 FG_SC_LAND_SEA_COAST2
FG_SC_TEC_GRADIENT FG_SC_IN_CLIM_ICE FG_SC_ICE FG_SC_SUSPECT_ICE FG_SC_RAIN FG_SC_HIGH_WIND
FG_SC_LOW_WIND
FG_SC_HIGHT_SST FG_SC_LOW_SST FG_SC_HIGH_SSS FG_SC_LOW_SSS FG_SC_SEA_STATE_1
FG_SC_SEA_STATE_2 FG_SC_SEA_STATE_3 FG_SC_SEA_STATE_4 FG_SC_SEA_STATE_5 FG_SC_SEA_STATE_6
FG_SC_SST_FRONT FG_SC_SSS_FRONT F
G_SC_ICE_ACARD" ;
        Science_Flags_2:_Unsigned = "true" ;
        Science_Flags_2:_FillValue = 0.f ;
    short Dg_num_meas_valid(n_grid_points) ;
        Dg_num_meas_valid:_Unsigned = "true" ;
        Dg_num_meas_valid:_FillValue = 0.f ;
    short Dg_RFI_L1(n_grid_points) ;
        Dg_RFI_L1:_Unsigned = "true" ;
        Dg_RFI_L1:_FillValue = NaNf ;
    short Dg_chi2_P_1(n_grid_points) ;
        Dg_chi2_P_1:_Unsigned = "true" ;
        Dg_chi2_P_1:_FillValue = 0.f ;
    float X_swath(n_grid_points) ;
        X_swath:_FillValue = -999.f ;
        X_swath:units = "m" ;
    short Dg_chi2_P_2(n_grid_points) ;
        Dg_chi2_P_2:_Unsigned = "true" ;
        Dg_chi2_P_2:_FillValue = 0.f ;
    short Dg_chi2_P_3(n_grid_points) ;
        Dg_chi2_P_3:_Unsigned = "true" ;
        Dg_chi2_P_3:_FillValue = 0.f ;
    short Dg_sun_tails(n_grid_points) ;
        Dg_sun_tails:_Unsigned = "true" ;
        Dg_sun_tails:_FillValue = 0.f ;
    short Dg_sun_glint_area(n_grid_points) ;
        Dg_sun_glint_area:_Unsigned = "true" ;
        Dg_sun_glint_area:_FillValue = 0.f ;
    ubyte Dg_num_iter_1(n_grid_points) ;
        Dg_num_iter_1:_Unsigned = "true" ;
        Dg_num_iter_1:_FillValue = 0.f ;
    float Longitude(n_grid_points) ;
        Longitude:_FillValue = -999.f ;
        Longitude:units = "deg" ;
    short Dg_galactic_Noise_Error(n_grid_points) ;
        Dg_galactic_Noise_Error:_Unsigned = "true" ;
        Dg_galactic_Noise_Error:_FillValue = 0.f ;
    float A_card(n_grid_points) ;
        A_card:_FillValue = -999.f ;
    ubyte Dg_num_iter_3(n_grid_points) ;
        Dg_num_iter_3:_Unsigned = "true" ;
        Dg_num_iter_3:_FillValue = 0.f ;
    short Dg_sky(n_grid_points) ;
        Dg_sky:_Unsigned = "true" ;
        Dg_sky:_FillValue = NaNf ;
    ubyte Dg_num_iter_2(n_grid_points) ;
        Dg_num_iter_2:_Unsigned = "true" ;
        Dg_num_iter_2:_FillValue = 0.f ;
    ubyte Dg_num_iter_4(n_grid_points) ;
        Dg_num_iter_4:_Unsigned = "true" ;
        Dg_num_iter_4:_FillValue = 0.f ;
    int Control_Flags_4(n_grid_points) ;
        Control_Flags_4:flag_masks = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
        Control_Flags_4:flag_values = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
        Control_Flags_4:flag_meanings = "FG_CTRL_RANGE FG_CTRL_SIGMA FG_CTRL_CHI2 FG_CTRL_CHI2_P
FG_CTRL_SUNGLINT FG_CTRL_MOONGLINT FG_CTRL_GAL_NOISE FG_CTRL_REACH_MAXITER
FG_CTRL_NUM_MEAS_MIN
```

```
 FG_CTRL_NUM_MEAS_LOW FG_CTRL_MANY_OUTLIERS FG_CTRL_MARQ FG_CTRL_ROUGHNESS FG_CTRL_FOAM
FG_CTRL_ECMWF FG_CTRL_VALID FG_CTRL_NO_SURFACE FG_CTRL_RANGE_ACARD FG_CTRL_SIGMA_ACARD
FG_CTRL_QUALITY_ACARD FG_
CTRL_USED_FARATEC FG_CTRL_POOR_GEOPHYS FG_CTRL_POOR_RETRIEVAL FG_CTRL_SUSPECT_RFI
FG_CTRL_RFI_PRONE_X FG_CTRL_RFI_PRONE_Y FG_CTRL_ADJUSTED_RA FG_CTRL_RETRIEV_FAIL" ;
                Control_Flags_4:_Unsigned = "true" ;
                Control_Flags_4:_FillValue = 0.f ;
        int Control_Flags_2(n_grid_points) ;
                Control_Flags_2:flag_masks = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_2:flag_values = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_2:flag_meanings = "FG_CTRL_RANGE FG_CTRL_SIGMA FG_CTRL_CHI2 FG_CTRL_CHI2_P
FG_CTRL_SUNGLINT FG_CTRL_MOONGLINT FG_CTRL_GAL_NOISE FG_CTRL_REACH_MAXITER
FG_CTRL_NUM_MEAS_MIN
 FG_CTRL_NUM_MEAS_LOW FG_CTRL_MANY_OUTLIERS FG_CTRL_MARQ FG_CTRL_ROUGHNESS FG_CTRL_FOAM
FG_CTRL_ECMWF FG_CTRL_VALID FG_CTRL_NO_SURFACE FG_CTRL_RANGE_ACARD FG_CTRL_SIGMA_ACARD
FG_CTRL_QUALITY_ACARD FG_
CTRL_USED_FARATEC FG_CTRL_POOR_GEOPHYS FG_CTRL_POOR_RETRIEVAL FG_CTRL_SUSPECT_RFI
FG_CTRL_RFI_PRONE_X FG_CTRL_RFI_PRONE_Y FG_CTRL_ADJUSTED_RA FG_CTRL_RETRIEV_FAIL" ;
                Control_Flags_2:_Unsigned = "true" ;
                Control_Flags_2:_FillValue = 0.f ;
        float Sigma_Tb_42_5H(n_grid_points) ;
                Sigma_Tb_42_5H:_FillValue = -999.f ;
                Sigma_Tb_42_5H:units = "K" ;
        int Control_Flags_3(n_grid_points) ;
                Control_Flags_3:flag_masks = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_3:flag_values = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_3:flag_meanings = "FG_CTRL_RANGE FG_CTRL_SIGMA FG_CTRL_CHI2 FG_CTRL_CHI2_P
FG_CTRL_SUNGLINT FG_CTRL_MOONGLINT FG_CTRL_GAL_NOISE FG_CTRL_REACH_MAXITER
FG_CTRL_NUM_MEAS_MIN
 FG_CTRL_NUM_MEAS_LOW FG_CTRL_MANY_OUTLIERS FG_CTRL_MARQ FG_CTRL_ROUGHNESS FG_CTRL_FOAM
FG_CTRL_ECMWF FG_CTRL_VALID FG_CTRL_NO_SURFACE FG_CTRL_RANGE_ACARD FG_CTRL_SIGMA_ACARD
FG_CTRL_QUALITY_ACARD FG_
CTRL_USED_FARATEC FG_CTRL_POOR_GEOPHYS FG_CTRL_POOR_RETRIEVAL FG_CTRL_SUSPECT_RFI
FG_CTRL_RFI_PRONE_X FG_CTRL_RFI_PRONE_Y FG_CTRL_ADJUSTED_RA FG_CTRL_RETRIEV_FAIL" ;
                Control_Flags_3:_Unsigned = "true" ;
                Control_Flags_3:_FillValue = 0.f ;
        int Control_Flags_1(n_grid_points) ;
                Control_Flags_1:flag_masks = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_1:flag_values = 2s, 4s, 8s, 16s, 64s, 128s, 256s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Control_Flags_1:flag_meanings = "FG_CTRL_RANGE FG_CTRL_SIGMA FG_CTRL_CHI2 FG_CTRL_CHI2_P
FG_CTRL_SUNGLINT FG_CTRL_MOONGLINT FG_CTRL_GAL_NOISE FG_CTRL_REACH_MAXITER
FG_CTRL_NUM_MEAS_MIN
 FG_CTRL_NUM_MEAS_LOW FG_CTRL_MANY_OUTLIERS FG_CTRL_MARQ FG_CTRL_ROUGHNESS FG_CTRL_FOAM
FG_CTRL_ECMWF FG_CTRL_VALID FG_CTRL_NO_SURFACE FG_CTRL_RANGE_ACARD FG_CTRL_SIGMA_ACARD
FG_CTRL_QUALITY_ACARD FG_
CTRL_USED_FARATEC FG_CTRL_POOR_GEOPHYS FG_CTRL_POOR_RETRIEVAL FG_CTRL_SUSPECT_RFI
FG_CTRL_RFI_PRONE_X FG_CTRL_RFI_PRONE_Y FG_CTRL_ADJUSTED_RA FG_CTRL_RETRIEV_FAIL" ;
                Control_Flags_1:_Unsigned = "true" ;
                Control_Flags_1:_FillValue = 0.f ;
        short Dg_sun_glint_fov(n_grid_points) ;
                Dg_sun_glint_fov:_Unsigned = "true" ;
                Dg_sun_glint_fov:_FillValue = 0.f ;
        float Sigma_Acard(n_grid_points) ;
                Sigma_Acard:_FillValue = -999.f ;
        float Latitude(n_grid_points) ;
                Latitude:_FillValue = -999.f ;
                Latitude:units = "deg" ;
        short Dg_quality_Acard(n_grid_points) ;
                Dg_quality_Acard:_Unsigned = "true" ;
                Dg_quality_Acard:_FillValue = 0.f ;
        float SSS3(n_grid_points) ;
                SSS3:_FillValue = -999.f ;
                SSS3:units = "psu" ;
        short Dg_num_meas_l1c(n_grid_points) ;
                Dg_num_meas_l1c:_Unsigned = "true" ;
                Dg_num_meas_l1c:_FillValue = 0.f ;
        float SSS2(n_grid_points) ;
                SSS2:_FillValue = -999.f ;
                SSS2:units = "psu" ;
        float SSS1(n_grid_points) ;
                SSS1:_FillValue = -999.f ;
                SSS1:units = "psu" ;
        float Tb_42_5Y(n_grid_points) ;
```

```
                Tb_42_5Y:_FillValue = -999.f ;
                Tb_42_5Y:units = "K" ;
        float Tb_42_5X(n_grid_points) ;
                Tb_42_5X:_FillValue = -999.f ;
                Tb_42_5X:units = "K" ;
        float WS(n_grid_points) ;
                WS:_FillValue = -999.f ;
                WS:units = "m s-1" ;
        short Dg_quality_SSS_1(n_grid_points) ;
                Dg_quality_SSS_1:_Unsigned = "true" ;
                Dg_quality_SSS_1:_FillValue = 0.f ;
        float Tb_42_5V(n_grid_points) ;
                Tb_42_5V:_FillValue = -999.f ;
                Tb_42_5V:units = "K" ;
        short Dg_quality_SSS_3(n_grid_points) ;
                Dg_quality_SSS_3:_Unsigned = "true" ;
                Dg_quality_SSS_3:_FillValue = 0.f ;
        short Dg_quality_SSS_2(n_grid_points) ;
                Dg_quality_SSS_2:_Unsigned = "true" ;
                Dg_quality_SSS_2:_FillValue = 0.f ;
        float Sigma_Tb_42_5V(n_grid_points) ;
                Sigma_Tb_42_5V:_FillValue = -999.f ;
                Sigma_Tb_42_5V:units = "K" ;
        float Sigma_Tb_42_5X(n_grid_points) ;
                Sigma_Tb_42_5X:_FillValue = -999.f ;
                Sigma_Tb_42_5X:units = "K" ;
        float Tb_42_5H(n_grid_points) ;
                Tb_42_5H:_FillValue = -999.f ;
                Tb_42_5H:units = "K" ;
        float Sigma_Tb_42_5Y(n_grid_points) ;
                Sigma_Tb_42_5Y:_FillValue = -999.f ;
                Sigma_Tb_42_5Y:units = "K" ;
        short Dg_sun_glint_L2(n_grid_points) ;
                Dg_sun_glint_L2:_Unsigned = "true" ;
                Dg_sun_glint_L2:_FillValue = 0.f ;
        short Dg_RFI_Y(n_grid_points) ;
                Dg_RFI_Y:_Unsigned = "true" ;
                Dg_RFI_Y:_FillValue = NaNf ;
        short Dg_RFI_X(n_grid_points) ;
                Dg_RFI_X:_Unsigned = "true" ;
                Dg_RFI_X:_FillValue = NaNf ;
        float SST(n_grid_points) ;
                SST:_FillValue = -999.f ;
                SST:units = "°C" ;
```

7.4 MIR SCLF1C

```
dimensions:
        n_grid_points = 68595 ;
        n_bt_data = 300 ;
        n_radiometric_accuracy = 2 ;
        n_snapshots = 1746 ;
variables:
        double Geomag_F(n_snapshots) ;
                Geomag_F:_FillValue = 0.f ;
                Geomag_F:units = "nT" ;
        float Grid_Point_Altitude(n_grid_points) ;
                Grid_Point_Altitude:_FillValue = -999.f ;
                Grid_Point_Altitude:units = "m" ;
        double Geomag_I(n_snapshots) ;
                Geomag_I:_FillValue = 0.f ;
                Geomag_I:units = "deg" ;
        double Y_Position(n_snapshots) ;
                Y_Position:_FillValue = 0.f ;
                Y_Position:units = "m" ;
        short Azimuth_Angle(n_grid_points, n_bt_data) ;
                Azimuth_Angle:_FillValue = 0.f ;
                Azimuth_Angle:scale_factor = 0.0054931640625 ;
                Azimuth_Angle:scale_offset = 0. ;
                Azimuth_Angle:_Unsigned = "true" ;
                Azimuth_Angle:units = "deg" ;
        float Sun_RA(n_snapshots) ;
                Sun_RA:_FillValue = 0.f ;
                Sun_RA:units = "deg" ;
        float Accuracy(n_snapshots) ;
                Accuracy:_FillValue = 0.f ;
                Accuracy:units = "K" ;
        double Y_Velocity(n_snapshots) ;
                Y_Velocity:_FillValue = 0.f ;
                Y_Velocity:units = "m/s" ;
        short Incidence_Angle(n_grid_points, n_bt_data) ;
                Incidence_Angle:_FillValue = 0.f ;
                Incidence_Angle:scale_factor = 0.001373291015625 ;
                Incidence_Angle:scale_offset = 0. ;
                Incidence_Angle:_Unsigned = "true" ;
                Incidence_Angle:units = "deg" ;
        ubyte Instrument_Error_flag(n_snapshots) ;
                Instrument_Error_flag:_Unsigned = "true" ;
                Instrument_Error_flag:_FillValue = 0.f ;
        double Geomag_D(n_snapshots) ;
                Geomag_D:_FillValue = 0.f ;
                Geomag_D:units = "deg" ;
        ubyte Grid_Point_Mask(n_grid_points) ;
                Grid_Point_Mask:_Unsigned = "true" ;
                Grid_Point_Mask:_FillValue = NaNf ;
        int Snapshot_ID(n_snapshots) ;
                Snapshot_ID:_Unsigned = "true" ;
                Snapshot_ID:_FillValue = 0.f ;
        int Grid_Point_ID(n_grid_points) ;
                Grid_Point_ID:_Unsigned = "true" ;
                Grid_Point_ID:_FillValue = 0.f ;
        float Radiometric_Accuracy(n_snapshots, n_radiometric_accuracy) ;
                Radiometric_Accuracy:_FillValue = 0.f ;
                Radiometric_Accuracy:scale_factor = 48. ;
                Radiometric_Accuracy:scale_offset = 0. ;
                Radiometric_Accuracy:units = "K" ;
        float Sun_BT(n_snapshots) ;
                Sun_BT:_FillValue = 0.f ;
                Sun_BT:units = "K" ;
        double Z_Position(n_snapshots) ;
                Z_Position:_FillValue = 0.f ;
                Z_Position:units = "m" ;
        int Microseconds(n_snapshots) ;
                Microseconds:_Unsigned = "true" ;
                Microseconds:_FillValue = 0.f ;
        int Seconds(n_snapshots) ;
                Seconds:_Unsigned = "true" ;
                Seconds:_FillValue = 0.f ;
        int Days(n_snapshots) ;
                Days:_FillValue = 0.f ;
        int Snapshot_ID_of_Pixel(n_grid_points, n_bt_data) ;
                Snapshot_ID_of_Pixel:_Unsigned = "true" ;
                Snapshot_ID_of_Pixel:_FillValue = 0.f ;
        ubyte Calibration_Error_flag(n_snapshots) ;
                Calibration_Error_flag:_Unsigned = "true" ;
                Calibration_Error_flag:_FillValue = 0.f ;
        ubyte ADF_Error_flag(n_snapshots) ;
                ADF_Error_flag:_Unsigned = "true" ;
                ADF_Error_flag:_FillValue = 0.f ;
        short Faraday_Rotation_Angle(n_grid_points, n_bt_data) ;
                Faraday_Rotation_Angle:_FillValue = 0.f ;
                Faraday_Rotation_Angle:scale_factor = 0.0054931640625 ;
                Faraday_Rotation_Angle:scale_offset = 0. ;
                Faraday_Rotation_Angle:_Unsigned = "true" ;
                Faraday_Rotation_Angle:units = "deg" ;
        double X_Position(n_snapshots) ;
                X_Position:_FillValue = 0.f ;
                X_Position:units = "m" ;
        double TEC(n_snapshots) ;
                TEC:_FillValue = 0.f ;
                TEC:units = "TECU" ;
        short Geometric_Rotation_Angle(n_grid_points, n_bt_data) ;
                Geometric_Rotation_Angle:_FillValue = 0.f ;
                Geometric_Rotation_Angle:scale_factor = 0.0054931640625 ;
                Geometric_Rotation_Angle:scale_offset = 0. ;
                Geometric_Rotation_Angle:_Unsigned = "true" ;
                Geometric_Rotation_Angle:units = "deg" ;
        float Sun_DEC(n_snapshots) ;
                Sun_DEC:_FillValue = 0.f ;
                Sun_DEC:units = "deg" ;
        short Footprint_Axis2(n_grid_points, n_bt_data) ;
                Footprint_Axis2:_FillValue = 0.f ;
                Footprint_Axis2:scale_factor = 0.0007476806640625 ;
                Footprint_Axis2:scale_offset = 0. ;
                Footprint_Axis2:_Unsigned = "true" ;
                Footprint_Axis2:units = "km" ;
        short Footprint_Axis1(n_grid_points, n_bt_data) ;
                Footprint_Axis1:_FillValue = 0.f ;
                Footprint_Axis1:scale_factor = 0.0007476806640625 ;
                Footprint_Axis1:scale_offset = 0. ;
                Footprint_Axis1:_Unsigned = "true" ;
                Footprint_Axis1:units = "km" ;
        double Z_Velocity(n_snapshots) ;
                Z_Velocity:_FillValue = 0.f ;
                Z_Velocity:units = "m/s" ;
        float BT_Value_Real(n_grid_points, n_bt_data) ;
                BT_Value_Real:_FillValue = -999.f ;
                BT_Value_Real:units = "K" ;
        int64 Snapshot_OBET(n_snapshots) ;
                Snapshot_OBET:_Unsigned = "true" ;
                Snapshot_OBET:_FillValue = 0.f ;
        float BT_Value_Imag(n_grid_points, n_bt_data) ;
                BT_Value_Imag:_FillValue = -999.f ;
                BT_Value_Imag:units = "K" ;
        double Q0(n_snapshots) ;
                Q0:_FillValue = 0.f ;
        float Grid_Point_Latitude(n_grid_points) ;
                Grid_Point_Latitude:_FillValue = -999.f ;
                Grid_Point_Latitude:units = "deg" ;
        double Q1(n_snapshots) ;
                Q1:_FillValue = 0.f ;
        double Q2(n_snapshots) ;
                Q2:_FillValue = 0.f ;
        double Q3(n_snapshots) ;
                Q3:_FillValue = 0.f ;
        ubyte Software_Error_flag(n_snapshots) ;
                Software_Error_flag:_Unsigned = "true" ;
                Software_Error_flag:_FillValue = 0.f ;
        float Grid_Point_Longitude(n_grid_points) ;
                Grid_Point_Longitude:_FillValue = -999.f ;
                Grid_Point_Longitude:units = "deg" ;
        short Flags(n_grid_points, n_bt_data) ;
                Flags:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
                Flags:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -32768s ;
                Flags:flag_meanings = "POL_FLAG_1 POL_FLAG_2 SUN_FOV SUN_GLINT_FOV MOON_GLINT_FOV
SINGLE_SNAPSHOT FTT SUN_POINT SUN_GLINT_AREA MOON_POINT AF_FOV EAF_FOV BORDER_FOV SUN_TAILS
RFI_1 RFI_
2" ;
                Flags:_Unsigned = "true" ;
                Flags:_FillValue = 0.f ;
        short BT_Data_Counter(n_grid_points) ;
                BT_Data_Counter:_Unsigned = "true" ;
                BT_Data_Counter:_FillValue = NaNf ;
        ubyte Vector_Source(n_snapshots) ;
                Vector_Source:_Unsigned = "true" ;
                Vector_Source:_FillValue = 0.f ;
        double X_Velocity(n_snapshots) ;
                X_Velocity:_FillValue = 0.f ;
                X_Velocity:units = "m/s" ;
        short Radiometric_Accuracy_of_Pixel(n_grid_points, n_bt_data) ;
                Radiometric_Accuracy_of_Pixel:_FillValue = 0.f ;
                Radiometric_Accuracy_of_Pixel:scale_factor = 0.000732421875 ;
                Radiometric_Accuracy_of_Pixel:scale_offset = 0. ;
                Radiometric_Accuracy_of_Pixel:_Unsigned = "true" ;
                Radiometric_Accuracy_of_Pixel:units = "K" ;
        ubyte X-Band(n_snapshots) ;
                X-Band:_Unsigned = "true" ;
                X-Band:_FillValue = 0.f ;
```

7.5 MIR SCSF1C

```
dimensions:
        n_grid_points = 116041 ;
        n_bt_data = 300 ;
        n_radiometric_accuracy = 2 ;
        n_snapshots = 2507 ;
variables:
        double Geomag_F(n_snapshots) ;
                Geomag_F:_FillValue = 0.f ;
                Geomag_F:units = "nT" ;
        float Grid_Point_Altitude(n_grid_points) ;
                Grid_Point_Altitude:_FillValue = -999.f ;
                Grid_Point_Altitude:units = "m" ;
        double Geomag_I(n_snapshots) ;
                Geomag_I:_FillValue = 0.f ;
                Geomag_I:units = "deg" ;
```

```
    double Y_Position(n_snapshots) ;
        Y_Position:_FillValue = 0.f ;
        Y_Position:units = "m" ;
    short Azimuth_Angle(n_grid_points, n_bt_data) ;
        Azimuth_Angle:_FillValue = 0.f ;
        Azimuth_Angle:scale_factor = 0.0054931640625 ;
        Azimuth_Angle:scale_offset = 0. ;
        Azimuth_Angle:_Unsigned = "true" ;
        Azimuth_Angle:units = "deg" ;
    float Sun_RA(n_snapshots) ;
        Sun_RA:_FillValue = 0.f ;
        Sun_RA:units = "deg" ;
    float Accuracy(n_snapshots) ;
        Accuracy:_FillValue = 0.f ;
        Accuracy:units = "K" ;
    double Y_Velocity(n_snapshots) ;
        Y_Velocity:_FillValue = 0.f ;
        Y_Velocity:units = "m/s" ;
    short Incidence_Angle(n_grid_points, n_bt_data) ;
        Incidence_Angle:_FillValue = 0.f ;
        Incidence_Angle:scale_factor = 0.001373291015625 ;
        Incidence_Angle:scale_offset = 0. ;
        Incidence_Angle:_Unsigned = "true" ;
        Incidence_Angle:units = "deg" ;
    ubyte Instrument_Error_flag(n_snapshots) ;
        Instrument_Error_flag:_Unsigned = "true" ;
        Instrument_Error_flag:_FillValue = 0.f ;
    double Geomag_D(n_snapshots) ;
        Geomag_D:_FillValue = 0.f ;
        Geomag_D:units = "deg" ;
    ubyte Grid_Point_Mask(n_grid_points) ;
        Grid_Point_Mask:_Unsigned = "true" ;
        Grid_Point_Mask:_FillValue = NaNf ;
    int Snapshot_ID(n_snapshots) ;
        Snapshot_ID:_Unsigned = "true" ;
        Snapshot_ID:_FillValue = 0.f ;
    int Grid_Point_ID(n_grid_points) ;
        Grid_Point_ID:_Unsigned = "true" ;
        Grid_Point_ID:_FillValue = 0.f ;
    float Radiometric_Accuracy(n_snapshots, n_radiometric_accuracy) ;
        Radiometric_Accuracy:_FillValue = 0.f ;
        Radiometric_Accuracy:scale_factor = 48. ;
        Radiometric_Accuracy:scale_offset = 0. ;
        Radiometric_Accuracy:units = "K" ;
    float Sun_BT(n_snapshots) ;
        Sun_BT:_FillValue = 0.f ;
        Sun_BT:units = "K" ;
    double Z_Position(n_snapshots) ;
        Z_Position:_FillValue = 0.f ;
        Z_Position:units = "m" ;
    int Microseconds(n_snapshots) ;
        Microseconds:_Unsigned = "true" ;
        Microseconds:_FillValue = 0.f ;
    int Seconds(n_snapshots) ;
        Seconds:_Unsigned = "true" ;
        Seconds:_FillValue = 0.f ;
    int Days(n_snapshots) ;
        Days:_FillValue = 0.f ;
    int Snapshot_ID_of_Pixel(n_grid_points, n_bt_data) ;
        Snapshot_ID_of_Pixel:_Unsigned = "true" ;
        Snapshot_ID_of_Pixel:_FillValue = 0.f ;
    ubyte Calibration_Error_flag(n_snapshots) ;
        Calibration_Error_flag:_Unsigned = "true" ;
        Calibration_Error_flag:_FillValue = 0.f ;
    ubyte ADF_Error_flag(n_snapshots) ;
        ADF_Error_flag:_Unsigned = "true" ;
        ADF_Error_flag:_FillValue = 0.f ;
    short Faraday_Rotation_Angle(n_grid_points, n_bt_data) ;
        Faraday_Rotation_Angle:_FillValue = 0.f ;
        Faraday_Rotation_Angle:scale_factor = 0.0054931640625 ;
        Faraday_Rotation_Angle:scale_offset = 0. ;
        Faraday_Rotation_Angle:_Unsigned = "true" ;
        Faraday_Rotation_Angle:units = "deg" ;
    double X_Position(n_snapshots) ;
        X_Position:_FillValue = 0.f ;
        X_Position:units = "m" ;
    double TEC(n_snapshots) ;
        TEC:_FillValue = 0.f ;
        TEC:units = "TECU" ;
    short Geometric_Rotation_Angle(n_grid_points, n_bt_data) ;
        Geometric_Rotation_Angle:_FillValue = 0.f ;
        Geometric_Rotation_Angle:scale_factor = 0.0054931640625 ;
        Geometric_Rotation_Angle:scale_offset = 0. ;
        Geometric_Rotation_Angle:_Unsigned = "true" ;
        Geometric_Rotation_Angle:units = "deg" ;
    float Sun_DEC(n_snapshots) ;
        Sun_DEC:_FillValue = 0.f ;
        Sun_DEC:units = "deg" ;
    short Footprint_Axis2(n_grid_points, n_bt_data) ;
        Footprint_Axis2:_FillValue = 0.f ;
        Footprint_Axis2:scale_factor = 0.0007476806640625 ;
        Footprint_Axis2:scale_offset = 0. ;
        Footprint_Axis2:_Unsigned = "true" ;
        Footprint_Axis2:units = "km" ;
    short Footprint_Axis1(n_grid_points, n_bt_data) ;
        Footprint_Axis1:_FillValue = 0.f ;
        Footprint_Axis1:scale_factor = 0.0007476806640625 ;
        Footprint_Axis1:scale_offset = 0. ;
        Footprint_Axis1:_Unsigned = "true" ;
        Footprint_Axis1:units = "km" ;
    double Z_Velocity(n_snapshots) ;
        Z_Velocity:_FillValue = 0.f ;
        Z_Velocity:units = "m/s" ;
    float BT_Value_Real(n_grid_points, n_bt_data) ;
        BT_Value_Real:_FillValue = -999.f ;
        BT_Value_Real:units = "K" ;
    int64 Snapshot_OBET(n_snapshots) ;
        Snapshot_OBET:_Unsigned = "true" ;
        Snapshot_OBET:_FillValue = 0.f ;
    float BT_Value_Imag(n_grid_points, n_bt_data) ;
        BT_Value_Imag:_FillValue = -999.f ;
        BT_Value_Imag:units = "K" ;
    double Q0(n_snapshots) ;
        Q0:_FillValue = 0.f ;
    float Grid_Point_Latitude(n_grid_points) ;
        Grid_Point_Latitude:_FillValue = -999.f ;
        Grid_Point_Latitude:units = "deg" ;
    double Q1(n_snapshots) ;
        Q1:_FillValue = 0.f ;
    double Q2(n_snapshots) ;
        Q2:_FillValue = 0.f ;
    double Q3(n_snapshots) ;
        Q3:_FillValue = 0.f ;
    ubyte Software_Error_flag(n_snapshots) ;
        Software_Error_flag:_Unsigned = "true" ;
        Software_Error_flag:_FillValue = 0.f ;
    float Grid_Point_Longitude(n_grid_points) ;
        Grid_Point_Longitude:_FillValue = -999.f ;
        Grid_Point_Longitude:units = "deg" ;
    short Flags(n_grid_points, n_bt_data) ;
        Flags:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 128s, 256s, 512s, 1024s, 4096s, 8192s, 64s, 16384s, -32768s, 2048s ;
        Flags:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 128s, 256s, 512s, 1024s, 4096s, 8192s, 64s, 16384s, -32768s, 2048s ;
        Flags:flag_meanings = "POL_FLAG_1 POL_FLAG_2 SUN_FOV SUN_GLINT_FOV MOON_FOV
SINGLE_SNAPSHOT SUN_POINT SUN_GLINT_AREA MOON_POINT AF_FOV BORDER_FOV SUN_TAILS RFI_H_POL
RFI_V_POL RFI_2 RF
I_3" ;
        Flags:_Unsigned = "true" ;
        Flags:_FillValue = 0.f ;
    short BT_Data_Counter(n_grid_points) ;
        BT_Data_Counter:_Unsigned = "true" ;
        BT_Data_Counter:_FillValue = NaNf ;
    ubyte Vector_Source(n_snapshots) ;
        Vector_Source:_Unsigned = "true" ;
        Vector_Source:_FillValue = 0.f ;
    double X_Velocity(n_snapshots) ;
        X_Velocity:_FillValue = 0.f ;
        X_Velocity:units = "m/s" ;
    short Radiometric_Accuracy_of_Pixel(n_grid_points, n_bt_data) ;
        Radiometric_Accuracy_of_Pixel:_FillValue = 0.f ;
        Radiometric_Accuracy_of_Pixel:scale_factor = 0.000732421875 ;
        Radiometric_Accuracy_of_Pixel:scale_offset = 0. ;
        Radiometric_Accuracy_of_Pixel:_Unsigned = "true" ;
        Radiometric_Accuracy_of_Pixel:units = "K" ;
    ubyte X-Band(n_snapshots) ;
        X-Band:_Unsigned = "true" ;
        X-Band:_FillValue = 0.f ;
```

7.6 MIR SMUDP2

```
dimensions:
        n_grid_points = 81759 ;
variables:
        short N_Sun_FOV(n_grid_points) ;
                N_Sun_FOV:_Unsigned = "true" ;
                N_Sun_FOV:_FillValue = 0.f ;
        float TB_ASL_Theta_B_H(n_grid_points) ;
                TB_ASL_Theta_B_H:_FillValue = -999.f ;
                TB_ASL_Theta_B_H:units = "K" ;
        float TB_ASL_Theta_B_H_DQX(n_grid_points) ;
                TB_ASL_Theta_B_H_DQX:_FillValue = -999.f ;
                TB_ASL_Theta_B_H_DQX:units = "K" ;
        short N_Wild(n_grid_points) ;
                N_Wild:_Unsigned = "true" ;
                N_Wild:_FillValue = 0.f ;
        float TB_TOA_Theta_B_V(n_grid_points) ;
                TB_TOA_Theta_B_V:_FillValue = -999.f ;
                TB_TOA_Theta_B_V:units = "K" ;
        ubyte DGG_Current_Flags(n_grid_points) ;
                DGG_Current_Flags:flag_masks = 1s, 2s, 4s, 8s, 16s ;
                DGG_Current_Flags:flag_values = 1s, 2s, 4s, 8s, 16s ;
                DGG_Current_Flags:flag_meanings = "FL_CURRENT_TAU_NADIR_LV FL_CURRENT_TAU_NADIR_FO
FL_CURRENT_HR FL_CURRENT_RFI FL_CURRENT_FLOOD" ;
                DGG_Current_Flags:_Unsigned = "true" ;
                DGG_Current_Flags:_FillValue = 0.f ;
        float TTH(n_grid_points) ;
                TTH:_FillValue = -999.f ;
        float Scattering_Albedo_H(n_grid_points) ;
                Scattering_Albedo_H:_FillValue = -999.f ;
        short M_AVA(n_grid_points) ;
                M_AVA:_Unsigned = "true" ;
                M_AVA:_FillValue = 0.f ;
        float TB_TOA_Theta_B_H(n_grid_points) ;
                TB_TOA_Theta_B_H:_FillValue = -999.f ;
                TB_TOA_Theta_B_H:units = "K" ;
        float Scattering_Albedo_H_DQX(n_grid_points) ;
                Scattering_Albedo_H_DQX:_FillValue = -999.f ;
        float Dielect_Const_Non_MD_IM_DQX(n_grid_points) ;
                Dielect_Const_Non_MD_IM_DQX:_FillValue = -999.f ;
                Dielect_Const_Non_MD_IM_DQX:units = "Fm-1" ;
        float Surface_Temperature_DQX(n_grid_points) ;
                Surface_Temperature_DQX:_FillValue = -999.f ;
                Surface_Temperature_DQX:units = "K" ;
        float Altitude(n_grid_points) ;
                Altitude:_FillValue = -99999.f ;
                Altitude:units = "m" ;
        int Grid_Point_ID(n_grid_points) ;
                Grid_Point_ID:_Unsigned = "true" ;
                Grid_Point_ID:_FillValue = 0.f ;
        float TB_TOA_Theta_B_H_DQX(n_grid_points) ;
                TB_TOA_Theta_B_H_DQX:_FillValue = -999.f ;
                TB_TOA_Theta_B_H_DQX:units = "K" ;
        short N_AF_FOV(n_grid_points) ;
                N_AF_FOV:_Unsigned = "true" ;
                N_AF_FOV:_FillValue = 0.f ;
        int Seconds(n_grid_points) ;
                Seconds:_FillValue = NaNf ;
                Seconds:_Unsigned = "true" ;
                Seconds:units = "s" ;
        short N_Sun_Glint_Area(n_grid_points) ;
                N_Sun_Glint_Area:_Unsigned = "true" ;
                N_Sun_Glint_Area:_FillValue = 0.f ;
        float Dielect_Const_MD_IM_DQX(n_grid_points) ;
                Dielect_Const_MD_IM_DQX:_FillValue = -999.f ;
                Dielect_Const_MD_IM_DQX:units = "Fm-1" ;
        short Confidence_Flags(n_grid_points) ;
                Confidence_Flags:flag_masks = 2s, 4s, 16s, 32s, 64s, 128s, 256s ;
                Confidence_Flags:flag_values = 2s, 4s, 16s, 32s, 64s, 128s, 256s ;
                Confidence_Flags:flag_meanings = "FL_RFI_PRONE_H FL_RFI_PRONE_V FL_NO_PROD FL_RANGE FL_DQX
FL_CHI2_P FL_FARADAY_ROTATION_ANGLE" ;
                Confidence_Flags:_Unsigned = "true" ;
                Confidence_Flags:_FillValue = 0.f ;
        short N_Software_Error(n_grid_points) ;
                N_Software_Error:_Unsigned = "true" ;
                N_Software_Error:_FillValue = 0.f ;
        short N_Sun_Tails(n_grid_points) ;
                N_Sun_Tails:_Unsigned = "true" ;
                N_Sun_Tails:_FillValue = 0.f ;
        float Soil_Moisture(n_grid_points) ;
                Soil_Moisture:_FillValue = -999.f ;
                Soil_Moisture:units = "m3 m-3" ;
        float Soil_Moisture_DQX(n_grid_points) ;
                Soil_Moisture_DQX:_FillValue = -999.f ;
                Soil_Moisture_DQX:units = "m3 m-3" ;
        short N_RFI_Mitigations(n_grid_points) ;
                N_RFI_Mitigations:_Unsigned = "true" ;
                N_RFI_Mitigations:_FillValue = 0.f ;
        int Science_Flags(n_grid_points) ;
                Science_Flags:flag_masks = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -
32768s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Science_Flags:flag_values = 1s, 2s, 4s, 8s, 16s, 32s, 64s, 128s, 256s, 512s, 1024s, 2048s, 4096s, 8192s, 16384s, -
32768s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s ;
                Science_Flags:flag_meanings = "FL_NON_NOM FL_SCENE_T FL_BARREN FL_TOPO_S FL_TOPO_M FL_OW
FL_SNOW_MIX FL_SNOW_WET FL_SNOW_DRY FL_FOREST FL_NOMINAL FL_FROST FL_ICE FL_WETLANDS
FL_FLOOD_P
ROB FL_URBAN_LOW FL_URBAN_HIGH FL_SAND FL_SEA_ICE FL_COAST FL_OCCUR_T FL_LITTER FL_PR
FL_INTERCEP FL_EXTERNAL FL_RAIN FL_TEC FL_TAU_FO FL_WINTER_FOREST FL_DUAL_RETR_FNO_FFO" ;
                Science_Flags:_Unsigned = "true" ;
                Science_Flags:_FillValue = 0.f ;
        float Dielect_Const_MD_RE_DQX(n_grid_points) ;
                Dielect_Const_MD_RE_DQX:_FillValue = -999.f ;
                Dielect_Const_MD_RE_DQX:units = "Fm-1" ;
        short N_Sky(n_grid_points) ;
                N_Sky:_Unsigned = "true" ;
                N_Sky:_FillValue = 0.f ;
        float DIFF_Albedos(n_grid_points) ;
                DIFF_Albedos:_FillValue = -999.f ;
        ubyte Chi_2(n_grid_points) ;
                Chi_2:scale_factor = 0.207843149546534 ;
                Chi_2:scale_offset = 0. ;
                Chi_2:_Unsigned = "true" ;
                Chi_2:_FillValue = 0.f ;
        float TB_ASL_Theta_B_V(n_grid_points) ;
                TB_ASL_Theta_B_V:_FillValue = -999.f ;
                TB_ASL_Theta_B_V:units = "K" ;
        float Longitude(n_grid_points) ;
                Longitude:_FillValue = -999.f ;
                Longitude:units = "deg" ;
        short N_Point_Source_RFI(n_grid_points) ;
                N_Point_Source_RFI:_Unsigned = "true" ;
                N_Point_Source_RFI:_FillValue = 0.f ;
        short Processing_Flags(n_grid_points) ;
                Processing_Flags:flag_masks = 1s, 2s, 4s, 8s ;
                Processing_Flags:flag_values = 1s, 2s, 4s, 8s ;
                Processing_Flags:flag_meanings = "FL_R4 FL_R3 FL_R2 FL_MD_A" ;
                Processing_Flags:_Unsigned = "true" ;
                Processing_Flags:_FillValue = 0.f ;
        float Optical_Thickness_Nad(n_grid_points) ;
                Optical_Thickness_Nad:_FillValue = -999.f ;
                Optical_Thickness_Nad:units = "Np" ;
        short N_Tails_Point_Source_RFI(n_grid_points) ;
                N_Tails_Point_Source_RFI:_Unsigned = "true" ;
                N_Tails_Point_Source_RFI:_FillValue = 0.f ;
        short N_X_Band(n_grid_points) ;
                N_X_Band:_Unsigned = "true" ;
                N_X_Band:_FillValue = 0.f ;
        float Surface_Temperature(n_grid_points) ;
                Surface_Temperature:_FillValue = -999.f ;
                Surface_Temperature:units = "K" ;
        short N_RFI_Y(n_grid_points) ;
                N_RFI_Y:_Unsigned = "true" ;
                N_RFI_Y:_FillValue = 0.f ;
        float Optical_Thickness_Nad_DQX(n_grid_points) ;
                Optical_Thickness_Nad_DQX:_FillValue = -999.f ;
                Optical_Thickness_Nad_DQX:units = "Np" ;
        short N_RFI_X(n_grid_points) ;
                N_RFI_X:_Unsigned = "true" ;
                N_RFI_X:_FillValue = 0.f ;
        ubyte Chi_2_P(n_grid_points) ;
                Chi_2_P:scale_factor = 0.00392156885936856 ;
                Chi_2_P:scale_offset = 0. ;
                Chi_2_P:_Unsigned = "true" ;
                Chi_2_P:_FillValue = 0.f ;
        float DIFF_Albedos_DQX(n_grid_points) ;
                DIFF_Albedos_DQX:_FillValue = -999.f ;
        short N_Calibration_Error(n_grid_points) ;
```

```
                N_Calibration_Error:_Unsigned = "true" ;
                N_Calibration_Error:_FillValue = 0.f ;
        float Dielect_Const_Non_MD_IM(n_grid_points) ;
                Dielect_Const_Non_MD_IM:_FillValue = -999.f ;
                Dielect_Const_Non_MD_IM:units = "Fm-1" ;
        float Dielect_Const_Non_MD_RE(n_grid_points) ;
                Dielect_Const_Non_MD_RE:_FillValue = -999.f ;
                Dielect_Const_Non_MD_RE:units = "Fm-1" ;
        float RTT_DQX(n_grid_points) ;
                RTT_DQX:_FillValue = -999.f ;
        float TTH_DQX(n_grid_points) ;
                TTH_DQX:_FillValue = -999.f ;
        int Microseconds(n_grid_points) ;
                Microseconds:_FillValue = NaNf ;
                Microseconds:_Unsigned = "true" ;
                Microseconds:units = "?s" ;
        float Dielect_Const_MD_IM(n_grid_points) ;
                Dielect_Const_MD_IM:_FillValue = -999.f ;
                Dielect_Const_MD_IM:units = "Fm-1" ;
        ubyte S_Tree_1(n_grid_points) ;
                S_Tree_1:_Unsigned = "true" ;
                S_Tree_1:_FillValue = 0.f ;
        int Days(n_grid_points) ;
                Days:_FillValue = NaNf ;
                Days:units = "days" ;
        ubyte S_Tree_2(n_grid_points) ;
                S_Tree_2:_Unsigned = "true" ;
                S_Tree_2:_FillValue = 0.f ;
        float RTT(n_grid_points) ;
                RTT:_FillValue = -999.f ;
        float Latitude(n_grid_points) ;
                Latitude:_FillValue = -999.f ;
                Latitude:units = "deg" ;
        short X_Swath(n_grid_points) ;
                X_Swath:_FillValue = 0.f ;
                X_Swath:scale_factor = 0.0320444367825985 ;
                X_Swath:scale_offset = 0. ;
                X_Swath:units = "km" ;
        float Roughness_Param_DQX(n_grid_points) ;
                Roughness_Param_DQX:_FillValue = -999.f ;
                Roughness_Param_DQX:units = "K" ;
        float Roughness_Param(n_grid_points) ;
                Roughness_Param:_FillValue = -999.f ;
                Roughness_Param:units = "K" ;
        ubyte GQX(n_grid_points) ;
                GQX:_Unsigned = "true" ;
                GQX:_FillValue = 0.f ;
        ubyte RFI_Prob(n_grid_points) ;
                RFI_Prob:scale_factor = 0.00499999988824129 ;
                RFI_Prob:scale_offset = 0. ;
                RFI_Prob:_Unsigned = "true" ;
                RFI_Prob:_FillValue = 0.f ;
        float HR_Cur_DQX(n_grid_points) ;
                HR_Cur_DQX:_FillValue = -999.f ;
        short N_ADF_Error(n_grid_points) ;
                N_ADF_Error:_Unsigned = "true" ;
                N_ADF_Error:_FillValue = 0.f ;
        short N_Strong_RFI(n_grid_points) ;
                N_Strong_RFI:_Unsigned = "true" ;
                N_Strong_RFI:_FillValue = 0.f ;
        short N_Instrument_Error(n_grid_points) ;
                N_Instrument_Error:_Unsigned = "true" ;
                N_Instrument_Error:_FillValue = 0.f ;
        float TB_TOA_Theta_B_V_DQX(n_grid_points) ;
                TB_TOA_Theta_B_V_DQX:_FillValue = -999.f ;
                TB_TOA_Theta_B_V_DQX:units = "K" ;
        float AFP(n_grid_points) ;
                AFP:_FillValue = -999.f ;
                AFP:units = "km" ;
        short M_AVA0(n_grid_points) ;
                M_AVA0:_Unsigned = "true" ;
                M_AVA0:_FillValue = 0.f ;
        float Dielect_Const_MD_RE(n_grid_points) ;
                Dielect_Const_MD_RE:_FillValue = -999.f ;
                Dielect_Const_MD_RE:units = "Fm-1" ;
        float TB_ASL_Theta_B_V_DQX(n_grid_points) ;
                TB_ASL_Theta_B_V_DQX:_FillValue = -999.f ;
                TB_ASL_Theta_B_V_DQX:units = "K" ;
        float Tau_Cur_DQX(n_grid_points) ;
        Tau_Cur_DQX:_FillValue = -999.f ;
    float Dielect_Const_Non_MD_RE_DQX(n_grid_points) ;
        Dielect_Const_Non_MD_RE_DQX:_FillValue = -999.f ;
        Dielect_Const_Non_MD_RE_DQX:units = "Fm-1" ;
```