 S-3 Toolbox/SMOS-BOX

Maintenance and Evolution

NetCDF Format Conversion User Guide

Version 5.9.0 17th May 2022

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 |
| 5.6.0 | Version update | 16.07.2019 |
| 5.8.0 | Added support for v724 data | 31.08.2020 |
| 5.9.0 | Increased only the version | 17.05.2022 |

Contents

[1 Introduction 4](#_Toc45678)

[1.1 Document purpose and scope 4](#_Toc45679)

[1.2 Acronyms and abbreviations 4](#_Toc45680)

[1.3 References 4](#_Toc45681)

[1.3.1 Applicable Documents 4](#_Toc45682)

[1.3.2 Reference documents 5](#_Toc45683)

[2 Motivation 6](#_Toc45684)

[3 Output file format 6](#_Toc45685)

[3.1 Data Format 6](#_Toc45686)

[3.2 Metadata 6](#_Toc45687)

[3.3 Dimensions 6](#_Toc45688)

[3.4 Invalid-pixel values 7](#_Toc45689)

[3.5 Flag coding 7](#_Toc45690)

[3.6 Data types 7](#_Toc45691)

[3.7 Variable scaling 8](#_Toc45692)

[3.8 Variable name conversion 8](#_Toc45693)

[4 Supported SMOS product types 8](#_Toc45694)

[5 Functionality 9](#_Toc45695)

[5.1 Geographic sub-setting 9](#_Toc45696)

[5.2 Variable sub-setting 9](#_Toc45697)

[5.3 Compression 11](#_Toc45698)

[6 Installations 12](#_Toc45699)

[6.1 S-3 Toolbox Graph Processing Tool operator 12](#_Toc45700)

[6.1.1 Examples 13](#_Toc45701)

[6.2 Stand-alone program 14](#_Toc45702)

[6.2.1 Examples 15](#_Toc45703)

[7 NetCDF file structure 16](#_Toc45704)

[7.1 MIR BWLF1C 16](#_Toc45705)

[7.2 MIR BWSF1C 17](#_Toc45706)

[7.3 MIR OSUDP2 18](#_Toc45707)

[7.4 MIR SCLF1C 21](#_Toc45708)

[7.5 MIR SCSF1C 23](#_Toc45709)

[7.6 MIR SMUDP2 26](#_Toc45710)

# Introduction

## 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.

## 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 |

## References

### Applicable Documents

|  |  |  |
| --- | --- | --- |
| [AD.1] | SMOS Level 1 and Auxiliary Data Products  Specifications, issue 6.4, 25.05.2018 | SO-TN-IDR-GS-0005 |
| [AD.2] | SMOS Level 2 and Auxiliary Data Products  Specifications, issue 8.6, 31.01.2020 | 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 |

### Reference documents

|  |  |  |
| --- | --- | --- |
| [RD.1] | NetCDF Java Library Documentation (version 4.3) | http://www.unidata.ucar.edu/software/ thredds/current/netcdfjava/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/Wellknown\_text |
| [RD.5] | SNAP GPT documentation | http://www.brockmannconsult.de/beam-  wiki/display/BEAM/Bulk+Processing+wit h+GPT |

# 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.

# Output file format

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

## 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.

## 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".

## 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 |

## 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.

## 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. |

## 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”.

## 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

valuephys = scale\_factor \* valuestored + scale\_offset

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

## 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 |

# 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 |

# Functionality

## 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.

## 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 |

## 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.

# 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.

## 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. |

### 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\_00

1\_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>

## 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-knowntext (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.

### 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))”

# 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.

## 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" ;

## 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" ;

## 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" ;

## 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 ;

## 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 ;

## 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" ;