EUMETSAT

EUMETSAT Data Centre Archive netCDF Formats

Doc.No. : EUM/OPS/TEN/07/3317
Issue : v4
Date : 7 December 2009

EUMETSAT
EUMETSAT Allee 1, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555 Telex: 419 320 metsat d
http://www.eumetsat.int

Document Change Record

Issue / Revision	*Date*	*DCN. No*	*Changed Pages / Paragraphs*
1.0	9 December 2008		1st Version Beta.
2.0	30 September 2009		Updates to finalise version 1.0 of the EUMETSAT Data Centre Archive netCDF formats. Addition of MSGCLMK netCDF format.
4	7 Dec		Minor editorial updates resulting from the ORR.

Table of Contents

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to present the netCDF formats in the EUMETSAT Data Centre Archive. The document has been prepared based on recommendation from the EUMETSAT Format Advisory Group[1].

1.2 Document Structure

Section 1	Introduction
Section 2	Overview
Section 3	Formats
Section 4	Tools

1.3 Reference Documents

[1] netCDF Climate and Forecast Metadata Conventions version 1.3. 7 Nov, 2008.
[2] WMO Publication No. 386 (Manual on the Global Telecommunication System), pp A.II-15/31 to 36.
[3] OGC 06-122 Draft CF-netCDF Candidate Standard

[1] The main task of the Format Advisory Group is to collect the needs of the user community, international partners and organisations, and to ensure these are taken into account for all new native and delivery formats proposed. See EUM/OPS/MEM/08/3742 and EUM/OPS/MEM/08/4609 for the Terms of Reference.

2 OVERVIEW

This section provides an overview of the netCDF format, reasons for introducing netCDF as a delivery format in the EUMETSAT Data Centre Archive, the version and conventions used, and the current set of Archive data sets orderable in netCDF.

2.1 Introduction to netCDF

netCDF (Network Common Data Form) is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data[2]. Access to the data encapsulated within the format (stored as arrays) is done though Application Programming Interfaces (APIs) defined in various computer languages[3]. Using these APIs means data can be access without the need to understand the underlying structure of the data. Other advantages of netCDF are:

- *Self-Describing*. A netCDF file includes information about the data it contains.
- *Portable*. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- *Direct-access*. A small subset of a large dataset may be accessed efficiently, without first reading through all the preceding data.
- *Appendable*. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- *Sharable*. One writer and multiple readers may simultaneously access the same netCDF file.
- *Archivable*. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

NetCDF format provides APIs to create meta data information within its contents. Various user groups and organisations have proposed meta data conventions for the netCDF format to promote its use for data exchange as users and applications can relay on these conventions to work with and operate.

Many international organisations have adopted the netCDF format as a standard way to represent their scientific data; NOAA, NASA, ECMWF, EUMETSAT (Jason), etc.

NetCDF is developed and maintained by Unidata. Unidata is funded primarily by the National Science Foundation, is part of the University Corporation for Atmospheric Research (UCAR) Office of Programs (UOP). For 20 years, this organisation has been vested in the

[2] Completely free format means that no need to define bits/bytes offsets for locations of data. Data is stored in arrays access by software APIs.
[3] API are available in C/C++, Fortan, Java, Perl etc. See http://www.unidata.ucar.edu/software/netCDF/ for more details.

common goal of sharing atmospheric and climate data, and tools to access and visualising this data.

Presently there is also a standardisation in progress, were a draft CF-netCDF Candidate Standard as OGC pending document 09-122 is in progress. This document is based on the netCDF classic core (based on NASA ESDS SPG).

2.2 Introducing netCDF as a EUMETSAT Data Centre Archive Delivery Format

The EUMETSAT Data Centre Archive receives and permanently stores images and meteorological products (collectively referred to as Archive data sets) from EUMETSAT satellites 24 hours a day, for every day of the year. It provides a comprehensive data retrieval service for the Archive data sets in full, sub-sampled, sub-area form, and in a variety of formats.

To facilitate co-operation with EUMETSAT's international partners, the introduction of the netCDF format as an orderable Archive delivery format is necessary to utilise the advantages the format offers as well as come to a common format that is recognisable and used in the international community.

In addition to this, various freeware tools are available to query, extract and visualise data defined in this format.

2.2.1 EUMETSAT Data Centre Archive netCDF Version

There are two versions of the netCDF format. NetCDF 3 is the version established in the user community whilst netCDF 4 is the enhanced version addressing the long standing deficiencies in netCDF 3. NetCDF 4 offers new format types, complex data structures, data compression, no file size restrictions and data access performance enhancements such as chunking[4]. To utilise these feature, existing software have to be updated to use the new netCDF 4 APIs.

Users have been slow to move to netCDF 4 because:

- Resources needed to port existing operational systems to use new APIs especially if the new features are not utilised.
- Some of the new features have not been implemented or are stable enough for operations.
- Users of the Java version of the API loses some portability as the APIs require HDF5 libraries and these are only available from the HDF Group[5] in C requiring compilation for the specific development platform.

[4] Chunking is the access data with the format structure in blocks rather than individual items.
[5] See http://www.hdfgroup.org/index.html.

The EUMETSAT Data Centre Archive has created netCDF data sets as a delivery format using the netCDF 3 libraries[6]. These are compatible for use by new applications using the netCDF 4 APIs. Porting to netCDF 4 is planned but the approach taken by the Archive will be to use the netCDF 4 libraries to create netCDF data sets that following the classic[7] data model (following OGC 06-122). This approach may be revised in the future based on requirements from the user community and future EUMETSAT Programmes.

2.2.2 EUMETSAT Data Centre Archive netCDF Conventions

While netCDF is intended for "self-documenting data", it is necessary for data writers and readers to agree upon attribute conventions and representations for discipline-specific data structures. Unidata recommends several netCDF conventions[8] and the one adopted by the EUMETSAT Data Centre Archive is the conventions for Climate and Forecast[9] (CF [1]).

The CF conventions are increasingly gaining acceptance and have been adopted by a number projects and organisations as the primary standard for netCDF meta-data definition. The conventions define metadata that provide a definitive description of what the data in each variable represents, and the spatial and temporal properties of the data. This enables users of data from different sources to decide which quantities are comparable, and facilitates building applications with powerful extraction, re-gridding, and display capabilities. It is also the convention recommended by Unidata for meta-data stored in the netCDF format.

netCDF data sets defined in this document shall following the CF conventions as closely as possible. There are some deficiencies with the convention for satellite data sets but these will be indicated if needed.

In addition to meta-data conventions, all Archive netCDF filenames defined for Archive data sets shall adopt the WMO filename convention[10] [2]. This convention provides the important file attributes in the filename without the need to open the file.

2.2.3 Deliverable Archive netCDF Data Sets

The following list the current orderable Archive data sets in netCDF. The Format Advisory Group has recommended that netCDF 4[11] as the format for delivery of images and products from the EUMETSAT Data Centre. This recommendation will be implemented based on:

- Requests from the user community.
- Native data sets are simply too large and contain too much information thus creating netCDF files isolating the frequently used data fields would improve data retrieval services.

[6] Implementation of the EUMETSAT Data Centre netCDF formats using the Java APIs began in March 2008. A beta version of the netCDF 4 Java APIs were released by Unidata in the 1st quarter of 2009.
[7] See http://www.unidata.ucar.edu/software/netCDF/workshops/2009/datamodel/index.html for this model.
[8] Unidata recommended netCDF conventions: http://www.unidata.ucar.edu/software/netCDF/conventions.html.
[9] CF conventions can be found at http://cf-pcmdi.llnl.gov/.
[10] WMO Publication No. 386 (Manual on the Global Telecommunication System), pp A.II-15/31 to 36.
[11] See minutes of the Minutes of the Format Advisory Group Meeting 05 (EUM/OPS/MIN/09/2671).

- Creating netCDFs targeted for used by existing tools to improve the usability of the data as well as allow visualisation of the product's content where applicable in a geo-located manner[12].

Meteosat First Generation Satellite (MFG):

- MTP15 netCDF data set shall hold:
  - Full scan or a sub-area.
  - Calibration slopes and offsets
  - Pixel counts for at least 1 of the 3 channels available.

Meteosat Second Generation Satellite (MSG):

- MSG15 netCDF data set shall hold:
  - Full scan or a sub-area.
  - Calibration slopes and offsets
  - Channels pixel counts (except HRV).

- CLMK netCDF data set shall hold:
  - Geo co-ordinates for the grid represented.
  - Byte grid containing flags indicating if the grid point contains cloud or otherwise.

EUMETSAT Polar System (EPS):

- IASI 1C netCDF data set shall hold:
  - Full orbit or a section of the orbit.
  - Onboard UTC times.
  - Geo-location variables and Spectrum values.

2.3 netCDF Common Data Form Language (CDL)

The EUMETSAT Data Centre Archive netCDF formats are defined using the Common Dataform Language (CDL). CDL is the ASCII representation of the contents of a netCDF file. Tools are available to dump the CDL for a netCDF file and to generate a netCDF file from CDL. CDL files (ASCII) end with extension ".cdl". NetCDF files (binary) end with extension ".nc".

[12] Provided the product contains geo-location information.

3 ARCHIVE NETCDF FORMATS

3.1 MFG::MTP15 netCDF

The purpose of this product is to deliver MTP rectified data in the netCDF format. Individual channels and sub-area of the MTP15 native data set can be defined in this format.

The following example CDL defines the format consisting of identification information, calibration, location, sub-setting information and pixel counts. The size of the product varies depending on what channels are defined and dimensions of the image sub-area. N.B. the variable ch1, ch2 and ch3 only exists for the channels requested. If radiance is required, the user can calculate this using the following equation using the channel pixel counts with its associated calibration coefficients.

PixelRadiance = Channel_Scale_Factor * (pixelCount - Channel_Add_Offset)

```
netCDF W_XX-EUMETSAT-Darmstadt\,VIS+IR+IMAGERY\,MET07+MVIRI_C_EUMS_20080801053000 {
dimensions:
        lat = 2500 ;
        lon = 2500 ;
        southMostLine = 1 ;
        eastMostPixel = 1 ;
        northMostLine = 2500 ;
        westMostPixel = 2500 ;
        numberOfChannels = 3 ;
        totalNumberOfChannels = 3 ;
variables:
        byte ch1(lat, lon) ;
                ch1:standard_name = "satellite_geo_meteosat_vis_pixel_count" ;
                ch1:long_name = "MVIRI visible pixel counts" ;
                ch1:scale_factor = 1.f ;
                ch1:add_offset = 0.f ;
                ch1:units = "1" ;
                ch1:valid_min = "0b" ;
                ch1:valid_max = "255b" ;
                comment = “Radiance in W m-2 sr-1 = scale_factor * ( pixelCount – add_offset )”;
        byte ch2(lat, lon) ;
                ch2:standard_name = "satellite_geo_meteosat_ir_pixel_count" ;
                ch2:long_name = "MVIRI infra-red pixel counts" ;
                ch2: scale_factor = 0.10334f ;
                ch2: add_offset = 5.f ;
                ch2:units = "1" ;
                ch2:valid_min = "0b" ;
                ch2:valid_max = "255b" ;
                comment = “Radiance in W m-2 sr-1 = scale_factor * ( pixelCount – add_offset )”;
        byte ch3(lat, lon) ;
                ch3:standard_name = "satellite_geo_meteosat_wv_pixel_count" ;
                ch3:long_name = "MVIRI water vapour pixel counts" ;
                ch3:scale_factor = 0.01205f ;
                ch3:add_offset = 6.f ;
                ch3:units = "1" ;
                ch3:valid_min = "0b" ;
                ch3:valid_max = "255b" ;
                comment = “Radiance in W m-2 sr-1 = scale_factor * ( pixelCount – add_offset )”;
        byte channelAvailableFlags(totalNumberOfChannels) ;
                channelAvailableFlags:standard_name = "satellite_geo_channels_available_flags" ;
                channelAvailableFlags:flag_meaning = "channel_not_available_in_file channel_available_in_file" ;
                channelAvailableFlags:flag_masks = "0b, 1b" ;
                channelAvailableFlags:valid_max = "1b" ;
                channelAvailableFlags:long_name = "Channel Available in File Flags" ;
                channelAvailableFlags:valid_min = "0b" ;
                channelAvailableFlags:units = "1" ;

        float lat(lat, lon) ;
```

```
                lat:standard_name = "latitude" ;
                lat:_CoordinateAxisType = "Lat" ;
                lat:valid_max = "90.0f" ;
                lat:long_name = "Latitudes for each pixel count" ;
                lat:valid_min = "-90.0f" ;
                lat:units = "degrees_north" ;
        float lon(lat, lon) ;
                lon:standard_name = "longitude" ;
                lon:_CoordinateAxisType = "Lon" ;
                lon:valid_max = "90.0f" ;
                lon:long_name = "Longitude for each pixel count" ;
                lon:valid_min = "-90.0f" ;
                lon:units = "degrees_east" ;

// global attributes:
                :Conventions = "CF-1.4, _Coordinates" ;
                :Metadata_Conventions = "Unidata Dataset Discovery v1.0" ;
                :title = "MTP15 channel data in netCDF." ;
                :summary = "MTP15 channel pixel counts with calibration coefficients and geo-location values." ;
                :keywords = "EUMETSAT, GSICS, ARCHIVE, UNIDATA, MVIRI, CALIBRATION COEFFICIENT - SLOPE-
OFFSET, netCDF" ;
                :history = "V1.0 - EUMETSAT COPYRIGHT 2009" ;
                :comment = "OPERATIONAL VERSION" ;
                :creator_name = "EUMETSAT Archive" ;
                :creator_url = "http://archive.eumetsat.int" ;
                :creator_email = "ops@eumetsat.int" ;
                :institution = "EUMETSAT" ;
                :time_converage_start = "2008-08-01T05:00:00Z" ;
                :time_converage_end = "2008-08-01T05:30:00Z" ;
                :license = "CopyRight EUMETSAT 2009" ;
                :references = "Unidata netCDF, Climate Format Conventions, EUMETSAT MTP1.5 Native Format Guide" ;
                :format_authors = "EUMETSAT" ;
                :format_version = "1.0" ;
                :source_formats = "METEOSAT7-MVIRI-MTP15-NA-NA-20080801053000.000000000Z-917918" ;
                :format_name = "MTP15, RECT2LP" ;
                :satellite_identifier = "M7" ;
                :instrument = "MVIRI" ;
                :disposition_mode = "OPE" ;
                :quality = "OK" ;
                :wmo_filename = "W_XX-EUMETSAT-
Darmstadt,VIS+IR+IMAGERY,MET07+MVIRI_C_EUMS_20080801053000.nc" ;
                :subSatellitePoint = "57" ;
                :originalSubSatellitePoint = "57" ;
                :scanMode = "ALTHRV" ;
                :pixelScanTimeFormula = "StartScanTime+(LineNumber/NumberOfLines)*NorthSouthScanTime" ;
                :althrv_northsouthscantime = "0.6*numberOfPixels" ;
                :redscn_northsouthscantime = "0.6*numberOfPixels" ;
                :channel_available_flag_1 = "VIS" ;
                :channel_available_flag_2 = "IR" ;
                :channel_available_flag_3 = "WV" ;

data: // Examples

        ch1 = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

        ch2 = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

        ch3 = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

        channelAvailableFlags = 1, 1, 1;

        lat = -999.0, …, -90.0, -90.0, …, -999.0;
        lon = -999.0, …, -90.0, -90.0, …, -999.0;
}
```

3.2 MSG::MSG15 netCDF

The purpose of this product is to deliver MSG rectified data in the netCDF format. Individual channels and sub-area of the MSG15 native data set can be defined in this format.

The following example CDL defines the format consisting of identification information, calibration, location, sub-setting information and pixel counts. The size of the product varies depending on what channels are defined and dimensions of the image sub-area. N.B. the variable ch1, ch2, ch3, ch4, ch5, ch6, ch7, ch8, ch9, ch10 and ch11 only exists for the channels requested. If radiance is required, the user can calculate this using the following equation on the channel pixel counts with its associated calibration coefficients.

PixelRadiance = Channel_Add_Offset + (Channel_Scale_Factor * pixelCount)

```
netCDF W_XX-EUMETSAT-Darmstadt,VIS+IR+IMAGERY,MET09+SEVIRI_C_EUMG_20081106122740.nc {
dimensions:
        lat = 2061 ;
        lon = 2076 ;
        southMostLine = 827 ;
        eastMostPixel = 820 ;
        northMostLine = 2887 ;
        westMostPixel = 2892 ;
        numberOfChannels = 11 ;
        totalNumberOfSeviriChannels = 12 ;
variables:
        short ch1(lat, lon) ;
                ch1:standard_name = "satellite_geo_meteosat_vis_0.6_pixel_count" ;
                ch1:long_name = "SEVIRI visible 0.6 Âµm pixel counts" ;
                ch1:scale_factor = 0.02013549953699112;
                ch1:add_offset = -1.026910476386547";
                ch1:units = "1" ;
                ch1:valid_min = "0b" ;
                ch1:valid_max = "1023b" ;
                ch1:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
        short ch2(lat, lon) ;
                ch2:standard_name = "satellite_geo_meteosat_vis_0.8_pixel_count" ;
                ch2:long_name = "SEVIRI visible 0.8 Âµm pixel counts" ;
                ch2:scale_factor = 0.025922000408172607;
                ch2:add_offset = -1.322022020816803;
                ch2:units = "1" ;
                ch2:valid_min = "0b" ;
                ch2:valid_max = "1023b" ;
                ch2:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
        short ch3(lat, lon) ;
                ch3:standard_name = "satellite_geo_meteosat_ir_1.6_pixel_count" ;
                ch3:long_name = "SEVIRI infra-red 1.6 Âµm pixel counts" ;
                ch3:scale_factor = 0.022258499637246132;
                ch3:add_offset = -1.1351834814995527;
                ch3:units = "1" ;
                ch3:valid_min = "0b" ;
                ch3:valid_max = "1023b" ;
                ch3:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
        short ch4(lat, lon) ;
                ch4:standard_name = "satellite_geo_meteosat_ir_3.9_pixel_count" ;
                ch4:long_name = "SEVIRI infra-red 3.9 Âµm pixel counts" ;
                ch4:scale_factor = 0.0036586668696007222;
                ch4:add_offset = -0.18659201034963685;
                ch4:units = "1" ;
                ch4:valid_min = "0b" ;
                ch4:valid_max = "1023b" ;
                ch4:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
        short ch5(lat, lon) ;
                ch5:standard_name = "satellite_geo_meteosat_wv_6.2_pixel_count" ;
                ch5:long_name = "SEVIRI water vapour 6.2 pixel counts" ;
```

```
        ch5:scale_factor = 0.008318111189855896;
        ch5:add_offset = -0.42422367068265077;
        ch5:units = "1" ;
        ch5:valid_min = "0b" ;
        ch5:valid_max = "1023b" ;
        ch5:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch6(lat, lon) ;
        ch6:standard_name = "satellite_geo_meteosat_wv_7.3_pixel_count" ;
        ch6:long_name = "SEVIRI water vapour 7.3 pixel counts" ;
        ch6:scale_factor = 0.03862196817549823;
        ch6:add_offset = -1.9697203769504095;
        ch6:units = "1" ;
        ch6:valid_min = "0b" ;
        ch6:valid_max = "1023b" ;
        ch6:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch7(lat, lon) ;
        ch7:standard_name = "satellite_geo_meteosat_ir_8.7_pixel_count" ;
        ch7:long_name = "SEVIRI infra-red 8.7 Âµm pixel counts" ;
        ch7:scale_factor = 0.1267443186859149;
        ch7:add_offset = -6.46396025298166;
        ch7:units = "1" ;
        ch7:valid_min = "0b" ;
        ch7:valid_max = "1023b" ;
        ch7:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch8(lat, lon) ;
        ch8:standard_name = "satellite_geo_meteosat_ir_9.7_pixel_count" ;
        ch8:long_name = "SEVIRI infra-red 9.7 Âµm pixel counts" ;
        ch8:scale_factor = 0.10396091069757771;
        ch8:add_offset = -5.302006445576463;
        ch8:units = "1" ;
        ch8:valid_min = "0b" ;
        ch8:valid_max = "1023b" ;
        ch8:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch9(lat, lon) ;
        ch9:standard_name = "satellite_geo_meteosat_ir_10.8_pixel_count" ;
        ch9:long_name = "SEVIRI infra-red 10.8 Âµm pixel counts" ;
        ch9:scale_factor = 0.2050356762076601;
        ch9:add_offset = -10.456819486590666;
        ch9:units = "1" ;
        ch9:valid_min = "0b" ;
        ch9:valid_max = "1023b" ;
        ch9:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch10(lat, lon) ;
        ch10:standard_name = "satellite_geo_meteosat_ir_12.0_pixel_count" ;
        ch10:long_name = "SEVIRI infra-red 12.0 Âµm pixel counts" ;
        ch10:scale_factor = 0.2223111466808946;
        ch10:add_offset = -11.337868480725625;
        ch10:units = "1" ;
        ch10:valid_min = "0b" ;
        ch10:valid_max = "1023b" ;
        ch10:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
short ch11(lat, lon) ;
        ch11:standard_name = "satellite_geo_meteosat_ir_13.4_pixel_count" ;
        ch11:long_name = "SEVIRI infra-red 13.4 Âµm pixel counts" ;
        ch11:scale_factor = 0.15760689687780738;
        ch11:add_offset = -8.037951740768175;
        ch11:units = "1" ;
        ch11:valid_min = "0b" ;
        ch11:valid_max = "1023b" ;
        ch11:comment = "Radiance in mW m-2 sr-1(cm-1)-1 = add_offset + ( pixelCount * scale_factor )" ;
byte channelAvailableFlags(totalNumberOfSeviriChannels) ;
        channelAvailableFlags:standard_name = "satellite_geo_channels_available_flags" ;
        channelAvailableFlags:flag_meaning = "channel_not_available_in_file channel_available_in_file" ;
        channelAvailableFlags:flag_masks = "0b, 1b" ;
        channelAvailableFlags:valid_max = "1b" ;
        channelAvailableFlags:long_name = "Channel Available in File Flags" ;
        channelAvailableFlags:valid_min = "0b" ;
        channelAvailableFlags:units = "1" ;
float lat(lat, lon) ;
        lat:standard_name = "latitude" ;
        lat:_CoordinateAxisType = "Lat" ;
        lat:valid_max = "90.0f" ;
        lat:long_name = "Latitudes for each pixel count" ;
```

```
                lat:valid_min = "-90.0f" ;
                lat:units = "degrees_north" ;
        float lon(lat, lon) ;
                lon:standard_name = "longitude" ;
                lon:_CoordinateAxisType = "Lon" ;
                lon:valid_max = "90.0f" ;
                lon:long_name = "Longitude for each pixel count" ;
                lon:valid_min = "-90.0f" ;
                lon:units = "degrees_east" ;

// global attributes:
                :Conventions = "CF-1.4" ;
                :Metadata_Conventions = "Unidata Dataset Discovery v1.0" ;
                :title = "MSG15 channel data in netCDF." ;
                :summary = "MSG15 channel pixel counts with calibration coefficients and geo-location values." ;
                :keywords = "EUMETSAT, GSICS, ARCHIVE, UNIDATA, SEVIRI, CALIBRATION COEFFICIENT - SLOPE-
OFFSET, netCDF" ;
                :history = "V1.0 - EUMETSAT COPYRIGHT 2009" ;
                :comment = "OPERATIONAL VERSION" ;
                :creator_name = "EUMETSAT Archive" ;
                :creator_url = "http://archive.eumetsat.int" ;
                :creator_email = "ops@eumetsat.int" ;
                :institution = "EUMETSAT" ;
                :time_converage_start = "2008-11-06T12:15:10Z" ;
                :time_converage_end = "2008-11-06T12:27:40Z" ;
                :license = "CopyRight EUMETSAT 2009" ;
                :references = "Unidata netCDF, Climate Format Conventions, EUMETSAT MSG1.5 Native Format Guide" ;
                :format_authors = "EUMETSAT" ;
                :format_version = "1.0" ;
                :source_formats = "MSG2-SEVI-MSG15-0100-NA-20081106122740.845000000Z-923596.nat" ;
                :format_name = "SEVIRI_CALIBRATION_COEFFICIENT" ;
                :satellite_identifier = "MSG2" ;
                :instrument = "SEVI" ;
                :disposition_mode = "OPE" ;
                :quality = "OK" ;
                :wmo_filename = "W_XX-EUMETSAT-
Darmstadt,VIS+IR+IMAGERY,MET09+SEVIRI_C_EUMG_20081106122740.nc" ;
                :wmo_data_category = "101" ;
                :wmo_international_data_sub_category = "000" ;
                :subSatellitePoint = "000.0000" ;
                :originalSubSatellitePoint = "0.0" ;
                :scanMode = "ALTHRV" ;
                :fullScanTime = "742.4_seconds" ;
                :rsScanTime = "246_seconds" ;
                :pixelScanTimeFormula = "StartScanTime+(LineNumber/NumberOfLines)*NorthSouthScanTime" ;
                :channel_available_flag_1 = "VIS06" ;
                :channel_available_flag_2 = "VIS08" ;
                :channel_available_flag_3 = "IR16" ;
                :channel_available_flag_4 = "IR39" ;
                :channel_available_flag_5 = "WV62" ;
                :channel_available_flag_6 = "WV73" ;
                :channel_available_flag_7 = "IR87" ;
                :channel_available_flag_8 = "IR97" ;
                :channel_available_flag_9 = "IR108" ;
                :channel_available_flag_10 = "IR120" ;
                :channel_available_flag_11 = "IR134" ;
                :channel_available_flag_12 = "NotUsedAlwaysSetToZero" ;
data: // Example

        ch1 =   0, 0, 0, 96, 97, 150, 202, 192, 191, 210, 259, 277, 219, 156, 111, 144,
                 204, 152, 120, 170, 210, 156, 112, 123, 155, 245, 276, 200, 117, 117, …
        :
        :
        ch11 =  0, 0, 0, 250, 276, 198, 188, 174, 153, 170, 221, 222, 186, 201, 207, 172, 199,
                260, 269, 251, 292, 330, 303, 248, 224, 248, 258, 218, 193, 233, 271,  …

        channelAvailableFlags = 1, 1, 1;

        lat = -999.0, …, -90.0, -90.0, …, -999.0;
        lon = -999.0, …, -90.0, -90.0, …, -999.0;
}
```

3.3 EPS::IASI1c netCDF

The purpose of this product is to deliver spectrum information from EPS IASI 1c data set. A full or section of the orbit can be defined in this format.

The following example CDL defines the format consisting of identification information, onboard UTC time, geo-location information, scaling factors, wave numbers and the spectrum values.

```
netCDF
W_XX-EUMETSAT-Darmstadt,HYPERSPECT+SOUNDING,METOPA+IASI_C_EUMP_20080709210322_8938_eps_o_l1.nc
{
dimensions:

        time = 97 ;             // Ulimited, but can be calculated by software in the IASI 1C
        along_track = 97 ;      // Number of MDRs
        across_track = 120 ;    // Number of EFOVs x Number of Pixels
        spectral = 8700 ;       // Fixed constant of 8700 measurements.

variables:

        float time(time) ;
        time:standard_name = "time" ;
        time:gsics_reference_time = "2008-07-09T00:00:00+00:00" ;
        time:units = "seconds since 2008-07-09" ;
        time:comment = "Onboard UTC index is time( along_track )" ;
        time:long_name = "Onboard UTC in seconds after the gsics_reference_time for each along_track measurement (Number of MDR) averaged from EFOV[15] and EFOV[16]" ;

        float lat(along_track, across_track) ;
        lat:standard_name = "latitude" ;
        lat:_CoordinateAxisType = "Lat" ;
        lat:valid_max = "90.0f" ;
        lat:long_name = "Location of pixel center in geodatic lat coordinates for each sounder pixel" ;
        lat:valid_min = "-90.0f" ;
        lat:units = "degree_north" ;
        lat:comment = "Latitude indices are latitude( along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)) )" ;

        float lon(along_track, across_track) ;
        lon:standard_name = "longitude" ;
        lon:_CoordinateAxisType = "Lon" ;
        lon:valid_max = "180.0f" ;
        lon:long_name = "Location of pixel center in geodatic lon coordinates for each sounder pixel" ;
        lon:valid_min = "-180.0f" ;
        lon:units = "degree_east" ;
        lon:comment = "Longitude indices are longitude( along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)) )" ;
        float satellite_Zenith_Angle(along_track, across_track) ;
        satellite_Zenith_Angle:standard_name = "satellite_zenth_angle" ;
        satellite_Zenith_Angle:valid_max = "180.0f" ;
        satellite_Zenith_Angle:long_name = "Satellite zenth angle for each sounder pixel" ;
        satellite_Zenith_Angle:units = "degree" ;
        satellite_Zenith_Angle:valid_min = "0.0f" ;
        satellite_Zenith_Angle:comment = "Satellite zenth angle indices are satellite_zenth_angle( along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)) )" ;

        float satellite_Azimuth_Angle(along_track, across_track) ;
        satellite_Azimuth_Angle:standard_name = "satellite_azimuth_angle" ;
        satellite_Azimuth_Angle:valid_max = "180.0f" ;
        satellite_Azimuth_Angle:long_name = "Satellite azimuth angle for each sounder pixel" ;
        satellite_Azimuth_Angle:units = "degree" ;
        satellite_Azimuth_Angle:valid_min = "-180.0f" ;
        satellite_Azimuth_Angle:comment = "Satellite azimuth angle indices are satellite_azimuth_angle( along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)) )" ;
```

float solar_Zenith_Angle(along_track, across_track) ;
solar_Zenith_Angle:standard_name = "solar_zenth_angle" ;
solar_Zenith_Angle:valid_max = "180.0f" ;
solar_Zenith_Angle:long_name = "Solar zenth angle at the surface of each sounder pixel" ;
solar_Zenith_Angle:units = "degree" ;
solar_Zenith_Angle:valid_min = "0.0f" ;
solar_Zenith_Angle:comment = "Solar zenth angle indices are solar_zenth_angle(along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)))" ;

float solar_Azimuth_Angle(along_track, across_track) ;
solar_Azimuth_Angle:standard_name = "solar_azimuth_angle" ;
solar_Azimuth_Angle:valid_max = "180.0f" ;
solar_Azimuth_Angle:long_name = "Solar azimuth angle at the surface of each sounder pixel" ;
solar_Azimuth_Angle:units = "degree" ;
solar_Azimuth_Angle:valid_min = "-180.0f" ;
solar_Azimuth_Angle:comment = "Solar azimuth angle indices are solar_azimuth_angle(along_track, across_track (Number of EFOV * 4 (4 pixels per measurement)))" ;

float scale_factor(spectral) ;
scale_factor:standard_name = "spectral_radiance_scale_factor" ;
scale_factor:long_name = "Scale factor for each spectral radiance" ;
scale_factor:units = "1" ;
scale_factor:comment = "Scale factor index is spectral_radiance_scale_factor(number_of_spectral_measurements)" ;

float wavenumber(spectral) ;
wavenumber:standard_name = "radiation_wavenumber" ;
wavenumber:long_name = "Radiation wave number for each spectral radiance" ;
wavenumber:units = "m-1" ;
wavenumber:comment = "Radiation wave number index is radiation_wavenumber(number_of_spectral_measurements)" ;

short spectral_radiance(along_track, across_track, spectral) ;
spectral_radiance:standard_name = "spectral_radiance_per_wavenumber" ;
spectral_radiance:long_name = "Spectral radiance" ;
spectral_radiance:units = "W/m2/sr/m-1" ;
spectral_radiance:comment = "Spectral radiance per wave number indices are spectral_radiance_per_wavenumber(along_track, across_track, number_of_spectral_measurements)" ;

// global attributes:

:Conventions = "CF-1.4" ;
:Metadata_Conventions = "Unidata Dataset Discovery v1.0" ;
:title = "IASI 1C spectra data in netCDF" ;
:keywords = "EUMETSAT, GSICS, ARCHIVE, UNIDATA, IASI 1C, SPECTRUM, netCDF, SPECTRA DATA" ;
:history = "V1.0 - EUMETSAT COPYRIGHT 2009" ;
:comment = "OPERATIONAL VERSION" ;
:creator_name = "EUMETSAT Archive" ;
:creator_url = "http://archive.eumetsat.int" ;
:creator_email = "ops@eumetsat.int" ;
:institution = "EUMETSAT" ;
:time_converage_start = "2008-07-09T21:03:22Z" ;
:time_converage_end = "2008-07-09T21:16:18Z" ;
:license = "CopyRight EUMETSAT 2009" ;
:references = "Unidata netCDF, Climate Format Conventions, EUMETSAT EPS IASI 1C Format Guide" ;
:format_authors = "EUMETSAT" ;
:format_version = "1.0" ;
:source_files = "IASI_xxx_1C_M02_20080709195354Z_20080709213554Z_N_O_20080709214017Z.nat" ;
:source_formats = "IASI1C" ;
:format_name = "IASI_SPECTRA" ;
:satellite_identifier = "M02" ;
:instrument = "IASI" ;
:wmo_filename = "W_XX-EUMETSAT-Darmstadt,HYPERSPECT+SOUNDING,METOPA+IASI_C_EUMP_20080709210322_8938_eps_o_l1.nc" ;
:orbit_start = "8938" ;
:orbit_end = "8939" ;

data:

time = …

lat = …

lon = …

satellite_Zenith_Angle = …

satellite_Azimuth_Angle = …

solar_Zenith_Angle = …

solar_Azimuth_Angle = …

scale_factor = …

wavenumber = …

spectral_radiance = …

}

3.4 MSG::MSGCLMK netCDF

The purpose of this product is to deliver MSG Cloud Mask Product in the netCDF format.

```
netCDF CLMK
{
dimensions:
        lat = numLines;    // Use section 3, bytes 35-38 (numLines).
        lon = numPixels;  // Use section 3, bytes 31-34 (numPixels).
variables:

    // Geo latitude and longitude dimension where the cloud mask is applicable to.
    float lat( lat, lon ) ;
        lat:standard_name = "latitude" ;
        lat:units = "degrees_north" ;
        lat:_fillValue = -999.0;

    float lon( lat, lon ) ;
        lon:standard_name = "longitude" ;
        lon:units = "degrees_east" ;
        lon:_fillValue = -999.0;

    // Mask index by the lat and lon dimensions.
    byte cloudMask( lat, lon ) ;
        cloudMask:standard_name = “cloud_mask”;
        cloudMask:valid_min = 0b ;
        cloudMask:valid_max = 3b ;
        cloudMask:flag_values = 0b, 1b, 2b, 3b ;
        cloudMask:_fillValue = 255b ;
        cloudMask:flag_values = “clear_sky_over_water
                                clear_sky_over_land
                                cloud
                                no_data (space)”;
data:
    // The lat and lon arrays are calculated based on the values in GRIB section 3,
    // bytes 77-80 (startLine), 73-76 (startPixel), 35-38 (numLines), 31-34 (numPixels),
    lat = … ;
    lon = … ;
    // The cloud mask is extracted from GRIB section 7, byte 6 to 3444741.  Each pixel count two bits is copied to each byte.
    cloudMask = …;

// Global Meta data

:Conventions = "CF-1.4"
:Metadata_Conventions = "Unidata Dataset Discovery v1.0";
:title = "EUMETSAT-ARCHIVE-CLOUD-MASK-netCDF";
:institution = "EUMETSAT";
:history= "2009-06-30T12:00:00Z Archive Cloud Mask netCDF Version 1.0";
:filename = “W_XX-EUMETSAT-Darmstadt,SING+LEV+SAT,MET09+CLM_C_EUMG_yyyymmddhhmiss.nc”;
:creator_url = "http://archive.eumetsat.int";
:creator_email = "ops@eumetsat.int";

:source = “MSG2-CLMK-2009….”;
:comment: "Demonstration Version";
:license: "Copyright EUMETSAT 2009";

// The following is calculated from Section 1, bytes 13-19.
:time_converage_start = "2009-06-03T17:30:23Z";
:time_converage_end = "2009-06-03T17:50:27Z";

:keywords ="EUMETSAT, ARCHIVE, netCDF, CLOUD MASK";
```

netCDF Software

3.5 Tools

The netCDF format can be visualised using the following tools.

- Unidata IDV[13].
- Unidata ToolsUI[14].
- ncgen[15] and ncdump[16].
- ncBrowse[17].

3.6 netCDF APIs

Unidata has provided APIs for various languages to simply the creation of the netCDF format file. These libraries can be down loaded from the Unidata website[18].

The steps for creating a netCDF file are:

1. Decide what data you wish to store.

2. Split these data into:
   a. Variables
   b. Dimensions
   c. Attributes.

3. Create a CDL version of the format.

4. Use this as a template to create the code using the libraries downloaded from Unidata.

See section 2.3 for additional information.

[13] http://www.unidata.ucar.edu/software/idv/
[14] http://www.unidata.ucar.edu/software/netCDF-java/
[15] http://www.unidata.ucar.edu/software/netCDF/docs/ncgen-man-1.html
[16] http://www.unidata.ucar.edu/software/netCDF/docs/ncdump-man-1.html
[17] http://www.epic.noaa.gov/java/ncBrowse/
[18] The Unidata netCDF page is http://www.unidata.ucar.edu/downloads/netCDF/index.jsp