The 7th Framework Programme
Theme SPACE
Project POPDAT

D 3.4 Metadata Description Report Version 1

Version: 1.0
Date: April 30, 2012

Project ref. no.	263240
Project title	POPDAT Problem-oriented Processing and Database Creation for Ionosphere Exploration

Deliverable status	**Final**
Contractual date of Delivery	Project Month 11
Actual date of delivery	Month 11
Deliverable number	D3.4
Deliverable title	Metadata Description
Type	Report
Status and version	1
Number of pages	27
WP contributing to the deliverable	WP3
WP/ Task responsible	SRC PAS
Author(s)	Project Consortium
EC Project Officer	Paola MOTA-ALVES
Key words	Ionospheric research data file format record structure metadata

POPDAT IONOSPHERE METADATA DESCRIPTION - DELIVERABLE D3.4

April 26, 2012

Contents

1 General introduction

Full description of the POPDAT project is described in the 'Problem-oriented Processing and Database Creation for Ionosphere Exploration - Annex I: Description of Work' document. Particular objectives of the project are:

- collection of datasets from different sources which concern the wave-like phenomena in the upper atmosphere and ionosphere to form an input data field for further processing;
- appropriate design of the flexible software tools for the search of the wavelike and/or any other anomalous behaviours detected in the satellite datasets;
- creation of the thematic catalogues of ionosphere perturbations, which have known, supposed or unknown origin;
- implementation of a Ionosphere Wave Service accessible to scientific communities and public users;
- Liaison with ULISSE information system;
- creation of Ionosphere Virtual Dynamic Observatory to assist educational activities;
- dissemination and promotion of the Ionosphere Wave Service at topical conferences and in scientific and public press.

1.1 Objectives

This report is related to the Task 3.3: *Service Architecture Design* from Work Package 3: *Ionosphere Waves Service Creation*. Its main goal is to provide general overview of the approach to POPDAT metadata description and structure.

The main objective of WP3 is the creation of an Ionosphere Waves Service for the scientific community. It corresponds to the development of database related to the problem oriented considerations for ionosphere wave research and the provision of useful tools to acces the database for the scientific community through a web portal.

Part of this task was to develop metadata description for the selected experiments. The detailed description can be obtained from document 'Problem-oriented Processing and Database Creation for Ionosphere Exploration - Annex I: Description of Work'.

1.2 Applicable Documents

In the text user will be often referred to the following documents:

- Problem-oriented Processing and Database Creation for Ionosphere Exploration - Annex I: Description of Work
- D1.3: Report on preliminary output of problem-oriented data processing.
- D2.2: Data File Common Part Record Structure
- D3.2: Ionosphere Waves Service Requirements Report
- D3.3: Ionosphere Waves Service Architecture Design Report Description

1.3 Overall goal and concept

Description of resource in the knowledge base with relevant metadata supports available scientific resources. Thus, the level of data description imposes the metadata structure:

- *descriptive metadata* - includes program, satellite and instrument information, as well as data details like observed phenomena, units of measures, time and spatial coverage, processing level;
- *structural metadata* - defines the details for data access and retrieval if selected by the user;
- in future, expand the notion of a data resource in our metadata definition to also include associated databases and computer programs (such as format conversion tools or data models) that generate requested data on demand.

1.4 Liaison to ULISSE

Structural metadata defines the structure of the file with full information on every field existing in the datafile (according to documents D2.2 'Data File Common Part Record Structure' that was agreed at Progress Meeting on November 2011 in Warsaw and the following report). Descriptive metadata refers to general information about the resource: origin of data, who is responsible for the data sharing, data policy applied to the datase etc.

Liaison with ULISSE: this is issue could be realized under circumstances if both projects were still in the operational and developement phase. Since the ULISSE project has finished on the 31 of December 2011, further actions concentrated on interaction between ULISSE and POPDAT can not be performed. Additionally, at the current level the metadata structure utilized in ULISSE can not be transformed to POPDAT. This is mainly due to the following reasons:

- ULISSE project was multidisciplinary project, where the datafiles belonged to wide variety of scientific disciplines. This imposed complexity of the files, so the common structure of the metadata was basicly limited to the descriptive type of metadata.

- After 6 months of the POPDAT project, the files and metafiles structure has been established, and this is the major and most suitable form for the type of data declared the POPDAT project.

2 Technical details

2.1 Data submission

In order to submit data each contributor must convert them to the NetCDF format and also prepare the events metadata in the XML file. Formats are described in Sections 2.2, 2.3, 2.4 and 2.5. Data submission will be done via POPDAT submission interfaces described in the D3.3 'IWS architecture design report' document. That requiers the data to be packed in a single zipfile.

2.2 General concept of the NetCDF files

The Wave Catalogue files will be stored in the NetCDF[1]standard. The abreviation NetCDF stands for Network Common Data Form, that is a format for the multidimensional scientific data. The main feautures of NetCDF are:

- *Self-Describing*: includes information about data it contains
- *Portable*: can be accessed by computers with different ways of storing integers, characters and floating point numbers
- *Scalable*: a small subset of a large dataset may be accessed efficiently
- *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.

Listed features satisfy requirements defined in the document 'D3.2 Ionosphere Waves Service Requirements Report'.

2.3 Creation of the NetCDF files

Each contributor must write a code that converts its data to the NetCDF format. The Fields the NetCDF file must contain are described in Sections: 2.4.1, 2.4.2, 2.4.3 and

[1]The NetCDF software was developed by Glenn Davis, Russ Rew, Ed Hartnett, John Caron, Steve Emmerson, and Harvey Davies at the Unidata Program Center in Boulder, Colorado, with contributions from many other NetCDF users. Read more: `http://www.unidata.ucar.edu/software/netcdf/`

2.4.5. However there exists software for generating NetCDF files that can be obtained from the UniData website. Below is a brief description, how to use that software.

1. At the initial stage each contributor must install the UniData software that is freeware and can be downloaded from the UniData website: `http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-install/index.html` and simply follow instraction therein.

2. After installation contributor must convert its data file in the CDL format. CDL format is described in Section 2.4.

3. CDL file can be directly converted to the NetCDF file by command **ncgen** typed in the Unix/Linux console/terminal.

4. CDL2NetCDF: **ncgen**

   ncgen generates either a NetCDF-3 (i.e. classic) binary .nc file, a NetCDF-4 (i.e. enhanced) binary .nc file or a file in some source language that when executed will construct the corresponding binary .nc file. For the purpose of POPDAT project the NetCDF-4 is used. The input to ncgen is a description of a NetCDF file in a small language known as CDL (network Common Data form Language). If no options are specified in invoking ncgen, it merely checks the syntax of the input CDL file, producing error messages for any violations of CDL syntax. Other options can be used, for example, to create the corresponding NetCDF file, or to generate a C program that uses the NetCDF C interface to create the NetCDF file[2].

5. Examples:

ncgen filename.cdl	Checks the syntax of the CDL file 'filename.cdl'
ncgen -o filename2.nc filename.cdl	From the CDL file 'filename.cdl', generates an equivalent binary NetCDF file named 'filename2.nc'
ncgen -b filename.cdl	From CDL file 'filename.cdl' generates an equivalent binary NetCDF file with name specified in 'filename.cdl' file (see Section 2.4)
ncgen -c -o filename2.nc filename.cdl	From the CDL file 'filename.cdl', generates a C program containing the NetCDF function invocations necessary to create an equivalent binary NetCDF file named 'filename2.nc'
ncdump filename2.nc > filename3.cdl	From the filename2.nc file generates filename3.cdl file

[2]To obtained more information about ncgen read: `http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/ncgen.html` or type **man ncgen** in command line (Unix/Linux).

2.4 CDL format description

CDL format (network Common Data form Language) is text notation for NetCDF objects and data[3]. The CDL file consists of three parts: dimensions, variables and data. Each CDL statement is terminated by semicolon. Spaces and tabs can be used freely for readability. Comments follow the double slash characters '//' an can be placed on any line.

A dimension is used to define the shape of one or more of the multidimensional variables described by the CDL description.

A variable represents a multidimensional array of values of the same type. A variable has a name, a data type, and a shape described by its list of dimensions. Each variable may also have associated attributes as well as data values. The name, data type, and shape of a variable are specified by its declaration in the variable section of a CDL description. A variable may have the same name as a dimension; by convention such a variable contains coordinates of the dimension it names.

An attribute contains information about a variable or about the whole NetCDF dataset or containing group. Attributes may be used to specify such properties as units, special values, maximum and minimum valid values, and packing parameters. Attribute information is represented by single values or one-dimensional arrays of values. An attribute has an associated variable, a name, a data type, a length, and a value. In contrast to variables that are intended for data, attributes are intended for ancillary data or metadata (data about data).

2.4.1 CDL header: common part

For CDL file preparation the convention described at UniData website: `http://www.unidata.ucar.edu/software/netcdf/docs/netcdf.html#Attribute-Conventions` is followed. The common part of the CDL file header was defined in the base of the document 'D2.2: Data File Common Part Record Structure'. In the mentioned document common part contains 15 fields. However taking into account differences between various experiments and measurements it appears that not all the fields each contrubiutor can provide, thus not all of them can be treated as mandatory. Detailed description of the common header is described in Section 2.4.1. Few modification to the defined structure were applied (the name written in bold letters refer to the name of field/variable defined in CDL header):

1. List of Mandatory fields-variables in NetCDF header:
   a) time (**time**)
   b) altitude (**alt**)
   c) geographic latitude (**glat**)
   d) geographic longitude (**glon**)
   e) local solar time (**lst**)

[3]http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/CDL-Syntax.html

2. Universal time units definition (**time:units**) the information about the reference time was added. For reference time International Standard Date Notation (YYYY-MM-DDThh:mm:ssZ) *ISO 8601* is used in its extended form. The 'Z' denotes the time zone, the Universal Time in paricular, named as well 'Zulu' time. In example: '**time:units="milliseconds since 2007-07-20T20:10:08Z";**'. Same time notation is used in global attribute that holds the information of processing date (**date_processing**).

3. According to the document 'D3.2: Ionosphere Waves Service Requirements Report' and after discussion on files metadata structure during Periodic Meeting in Budapest (16-17 April 2012) the attributes defining:

   - Title: title
   - Contributing organisation name: contributor_name
   - Contact email address: contributor_email
   - Name / version of processing algorithm: algorithm_version
   - Date processing: date_processing
   - Data use statement (Data Policy Document DTBD): data_usage_policy
   - Data manual (link to the document): data_manual
   - Contributing organisation http link: contributor_link
   - Project http link: project_link
   - Catalogue name: catalogue_id
   - NetCDF file description: description
   - Source data link: sourc_data_link

   were added as '**global attributes**'.

4. Report D2.2 defines also detection flags for each event subclass i.e. EMW: electromagnetic wavelike event can be WHF - whistler, or FWHF - fractional hop whistler, etc. During Periodic Meeting in Budapest (16-17 April 2012) it was decided that each event sublass will be pointed by seperate flag, that takes '1' for detection and '0' for no detection. Flag must be defined as boolean array.

There few very important things about header presented above:

- common fields are mandatory for the all NetCDF files in POPDAT catalogue, however the values filled are exemplary
- the variable names must be written in lower case
- the 'time' dimension is a mandatory one, however contributor may defined more dimensions if that's necessary
- attributes 'valid_max' and 'valid_min' defines ranges for particular file and not the possible ranges
- values of 'short_name' attribute should equals the variable's name

Exemplary header with fields common for all catalogue types is presented below.

```
//THIS FILE DEFINES FIELDS COMMON FOR ALL EVENTS NetCDF FILES:

 netcdf EMW_07201_72549_72550 { //This line defines the name of the output NetCDF file
dimensions:
time = UNLIMITED ; // (1 currently)

variables:
float time(time) ;
time:long_name = "time" ;
time:short_name = "time" ;
time:standard_dame = "time";
time:description = "Universal time at the observat's point" ;
time:resolution = "up to 10 microsec - varies according to the experiments " ;
time:units = "milliseconds since 2007-07-20T20:10:08Z" ; // units are defined by the contributor
time:valid_min = 0.0 ;
time:valid_max = 172799999.8 ;

float alt(time) ;
alt:long_name = "Altitude" ;
alt:short_name = "alt" ;
alt:standard_name = "altitude";
alt:description = "The altitude of the spacecraft in km, or if other - defined by the contributor (ex f2peak)" ;
alt:resolution = 0.01 ;
alt:units = "km" ;
alt:valid_min =  ;
alt:valid_max =  ;

float glat(time) ;
glat:long_name = "Geographic latitude" ;
glat:short_name = "glat" ;
glat:standard_name = "latitude";
glat:description = "The geographic latitude of the observation's point" ;
glat:resolution = 0.001  ;
glat:units = "degrees_north" ;
glat:valid_min = -90.0 ;
glat:valid_max =  90.0 ;

float glon(time) ;
glon:long_name = "Geographic longitude" ;
glon:short_name = "glon" ;
glon:standard_name = "longitude";
glon:description = "The geographic longitude of observation's point" ;
glon:resolution = 0.001 ;
glon:units = "degrees_east" ;
glon:valid_min =  -180.0 ;
glon:valid_max =  180.0 ;// specific for the particular file and must be defined by the contributor

float lst(time) ;
lst:long_name = "Local Solar Time" ;
lst:short_name = "lst" ;
lst:description = "The local Solar time of the observation's point in decimal hours." ;
lst:resolution = 0.0001  ;
lst:units = "hours" ;
lst:valid_min =  0.0 ;
lst:valid_max = 24.0;



// HERE ARE THE FIELDS SPECIFIC FOR EACH EVENT TYPE



// global attributes:
:title: "EMW file #1"
```

```
:contributor_name = "Space Research Centre Polish Academy of Sciences, Warsaw" ;
:contributor_email = "contact@cbk.waw.pl" ;
:algorithm_version = "POPDAT_DB V.0.2" ;
:date_processing = "2012-03-26T12:00Z" ;
:data_usage_policy = "Data Policy- DTBD" ;
:data_manual = "Data Manual - DTBD";
:contributor_link = "http://www.cbk.waw.pl" ;
:project_link = "http://www.popdat.org";
:catalogue_id = "AGW"; // EMW, TID-TEC etc
:source_data_link = "";
:description = "description of the nc file";
```

Division into four seperate types of CDL files is a direct consequence of the different phenomena considered in the POPDAT project. Classification of three types is fully described in the document 'D1.3: Report on preliminary output of problem-oriented data processing'. During the Periodic Meeting in Budapest on 16-17 April 2012 one more catalogue has been added. Thus initial Ionosphere Waves Service will consists of following catalogues:

- EMW catalogue: Electromagnetic Wave-like Phenomena catalogue
- AGW catalogue: Atmospheric Gravity Waves catalogue
- TID - TEC catalogue: Travelling Ionospheric Disturbances (derived from TEC measurements) catalogue (added during Budapest Periodic Meeting)
- TID catalogue: Travelling Ionospheric Disturbances catalogue

The following subsections (2.4.2, 2.4.3, 2.4.4, 2.4.5) deal with the fields specific for each catalogue type. Catalogues discussed below will initially appear in IWS. However it will be possible in the future that contributors will define new catalogues.

There is an important NOTE according to the definition of the magnetic field ralated variables. In case of using magnetic coordinates such as Magnetic Latitude, Magnetic Longitude, each contributor must as well give the information of the model used for obtaining those values. This information should be located in the variable name: mlat_[model_used], mlon_[model_used], etc.

Each contributor must also define the boolean array with name 'detect_[name_of_event_type]' for each detected event type and for each event subclass.

2.4.2 CDL: EMW files

The fields presented below are the specific fields defined for EMW events catalogue. Here only the variables that are different than those in the common part are presented. The variables 'freq' and 'spec' have other dimension than 'time'. Their additional dimensions: 'n_freq' and 'n_spec' must be defined in the 'dimension' field of the header.

```
// specific attributes corresponding to the EMW catalogue records

float mlat_model(time) ;
mlat_model:long_name = "Geomagnetic latitude" ;
mlat_model:short_name = "mlat_model" ;
mlat_model:description = "The geomagnetic latitude of the spacecraft orbital position. IGRF or specified lat
```

```
mlat_model:resolution = 0.01  ;
mlat_model:units = "degrees_north" ;
mlat_model:valid_min = -90.0 ;
mlat_model:valid_max =  90.0 ;

float mlon_model(time) ;
mlon_model:long_name = "Geomagnetic longitude" ;
mlon_model:short_name = "mlon_model" ;
mlon_model:description = "The geomagnetic longitude of the spacecraft orbital position. IGRF or specified later" ;
mlon_model:resolution = 0.01  ;
mlon_model:units = "degrees_east" ;
mlon_model:valid_min =  -180.0 ;
mlon_model:valid_max =  180.0 ;// specific for the particular file and must be defined by the contributor

float lval(time) ;
lval:long_name = "McIlwain parameter - L" ;
lval:short_name = "lval" ;
lval:description = "L-shell value (radial altitude of geomagnetic field apex in Earth's radii)." ;
lval:resolution = 0.01 ;
lval:units = "Normalized Magn. Apex Altitude" ;
lval:valid_min = " 1.0  (at geomagnetic equator and on Earth's surface)" ;
lval:valid_max = "L around geomagnetic poles" ;

float mlt(time) ;
mlt:long_name = "Magnetic Local Time" ;
mlt:short_name = "mlt" ;
mlt:description = "The longitude angle between the Zero Magnetic meridian (~ -69o) and the sub-solar point in hours
mlt:resolution = 0.01 ;
mlt:units = "hours" ;
mlt:valid_min = 0.0 ;
mlt:valid_max =  24.0 ;

byte detect_whf(time);
detect_whf:detect_detection=1;
detect_whf:long_name = "Detection Flag for whistler" ;
detect_whf:short_name = "detect_whf" ;
detect_whf:description = "Whistler detected" ;
detect_whf:resolution = " " ;
detect_whf:units = "" ;
detect_whf:detected_value_max="";
detect_whf:definite_dispersion="";
detect_whf:despersion_range="";
detect_whf:valid_min = 0 ;
detect_whf:valid_max = 1 ;

byte detect_fwhf(time);
detect_fwhf:detect_detection=1;
detect_fwhf:long_name = "Detection Flag for fractional hop whistler" ;
detect_fwhf:short_name = "detect_fwhf" ;
detect_fwhf:description = "Fractional Hop whistler detected" ;
detect_fwhf:resolution = " " ;
detect_fwhf:units = "" ;
detect_fwhf:sensor_sensitivity= "";
detect_fwhf:detected_value_max="";
detect_fwhf:definite_dispersion="";
detect_fwhf:despersion_range="";
detect_fwhf:valid_min = 0 ;
detect_fwhf:valid_max = 1 ;

byte detect_iwhf(time);
detect_iwhf:detect_detection=1;
detect_iwhf:long_name = "Detection Flag for ionwhistler" ;
detect_iwhf:short_name = "detect_iwhf" ;
detect_iwhf:description = "Ionwhistler detected" ;
detect_iwhf:resolution = " " ;
detect_iwhf:units = "" ;
detect_iwhf:detected_value_max="";
```

```
detect_iwhf:definite_dispersion="";
detect_iwhf:despersion_range="";
detect_iwhf:valid_min = 0 ;
detect_iwhf:valid_max = 1 ;

byte detect_whpf(time);
detect_whpf:detect_detection=1;
detect_whpf:long_name = "Detection Flag for whistler pairs" ;
detect_whpf:short_name = "detect_whpf" ;
detect_whpf:description = "Whistler pairs detected" ;
detect_whpf:resolution = " " ;
detect_whpf:units = "" ;
detect_whpf:detected_value_max="";
detect_whpf:definite_dispersion="";
detect_whpf:despersion_range="";
detect_whpf:valid_min = 0 ;
detect_whpf:valid_max = 1 ;

byte detect_spwhf(time);
detect_spwhf:detect_detection=1;
detect_spwhf:long_name = "Detection Flag for SpW" ;
detect_spwhf:short_name = "detect_spwhf" ;
detect_spwhf:description = "SpW detected" ;
detect_spwhf:resolution = " " ;
detect_spwhf:units = "" ;
detect_spwhf:detected_value_max="";
detect_spwhf:definite_dispersion="";
detect_spwhf:despersion_range="";
detect_spwhf:valid_min = 0 ;
detect_spwhf:valid_max = 1 ;

byte detect_stwf(time);
detect_stwf:detect_detection=1;
detect_stwf:long_name = "Detection Flag for STW" ;
detect_stwf:short_name = "detect_stwf" ;
detect_stwf:description = "STWF detected" ;
detect_stwf:resolution = " " ;
detect_stwf:units = "" ;
detect_stwf:detected_value_max="";
detect_stwf:definite_dispersion="";
detect_stwf:despersion_range="";
detect_stwf:valid_min = 0 ;
detect_stwf:valid_max = 1 ;

byte detect_xwhf(time);
detect_xwhf:detect_detection=1;
detect_xwhf:long_name = "Detection Flag for XWHF" ;
detect_xwhf:short_name = "detect_xwhf" ;
detect_xwhf:description = "crossed trace whistlers detected" ;
detect_xwhf:resolution = " " ;
detect_xwhf:units = "" ;
detect_xwhf:detected_value_max="";
detect_xwhf:definite_dispersion="";
detect_xwhf:despersion_range="";
detect_xwhf:valid_min = 0 ;
detect_xwhf:valid_max = 1 ;

byte detect_tlewhf(time);
detect_tlewhf:detect_detection=1;
detect_tlewhf:long_name = "Detection Flag for TLEWHF" ;
detect_tlewhf:short_name = "detect_tlewhf" ;
detect_tlewhf:description = "TLE associated whistlers detected" ;
detect_tlewhf:resolution = " " ;
detect_tlewhf:units = "" ;
detect_tlewhf:detected_value_max="";
detect_tlewhf:definite_dispersion="";
detect_tlewhf:despersion_range="";
```

```
detect_tlewhf:valid_min = 0 ;
detect_tlewhf:valid_max = 1 ;

byte detect_chrf(time);
detect_chrf:detect_detection=1;
detect_chrf:long_name = "Detection Flag for CHRF" ;
detect_chrf:short_name = "detect_chrf" ;
detect_chrf:description = "Chorus identified" ;
detect_chrf:resolution = " " ;
detect_chrf:units = "" ;
detect_chrf:detected_value_max="";
detect_chrf:definite_dispersion="";
detect_chrf:despersion_range="";
detect_chrf:valid_min = 0 ;
detect_chrf:valid_max = 1 ;

byte detect_hissf(time);
detect_hissf:detect_detection=1;
detect_hissf:long_name = "Detection Flag for HISSF" ;
detect_hissf:short_name = "detect_hissf" ;
detect_hissf:description = "Hiss  identified" ;
detect_hissf:resolution = " " ;
detect_hissf:units = "" ;
detect_hissf:detected_value_max="";
detect_hissf:definite_dispersion="";
detect_hissf:despersion_range="";
detect_hissf:valid_min = 0 ;
detect_hissf:valid_max = 1 ;

byte detect_trigf(time);
detect_trigf:detect_detection=1;
detect_trigf:long_name = "Detection Flag for TRIGF" ;
detect_trigf:short_name = "detect_trigf" ;
detect_trigf:description = "Triggered emission detected" ;
detect_trigf:resolution = " " ;
detect_trigf:units = "" ;
detect_trigf:detected_value_max="";
detect_trigf:definite_dispersion="";
detect_trigf:despersion_range="";
detect_trigf:valid_min = 0 ;
detect_trigf:valid_max = 1 ;

byte detect_seismf(time);
detect_seismf:detect_detection=1;
detect_seismf:long_name = "Detection Flag for SEISMF" ;
detect_seismf:short_name = "detect_seismf" ;
detect_seismf:description = "Seismic related signals found" ;
detect_seismf:resolution = " " ;
detect_seismf:units = "" ;
detect_seismf:detected_value_max="";
detect_seismf:definite_dispersion="";
detect_seismf:despersion_range="";
detect_seismf:valid_min = 0 ;
detect_seismf:valid_max = 1 ;

byte detect_anmf(time);
detect_anmf:detect_detection=1;
detect_anmf:long_name = "Detection Flag for ANMF" ;
detect_anmf:short_name = "detect_anmf" ;
detect_anmf:description = "Anomalistic wave detection perforemd" ;
detect_anmf:resolution = " " ;
detect_anmf:units = "" ;
detect_anmf:detected_value_max="";
detect_anmf:definite_dispersion="";
detect_anmf:despersion_range="";
detect_anmf:valid_min = 0 ;
detect_anmf:valid_max = 1 ;
```

```
byte detect_bdnoise(time);
detect_bdnoise:detect_detection=1;
detect_bdnoise:long_name = "Detection Flag for BDNOISE" ;
detect_bdnoise:short_name = "detect_bdnoise" ;
detect_bdnoise:description = "Broadband nosie detection" ;
detect_bdnoise:resolution = " " ;
detect_bdnoise:units = "" ;
detect_bdnoise:detected_value_max="";
detect_bdnoise:definite_dispersion="";
detect_bdnoise:despersion_range="";
detect_bdnoise:valid_min = 0 ;
detect_bdnoise:valid_max = 1 ;

float freq(n_frequency) ;
freq:long_name = "FFT frequency" ;
freq:short_name = "freq" ;
freq:description = "FFT frequency " ;
freq:resolution = 115;
freq:units = "Hz" ;
freq:valid_min = 0.0  ;
freq:valid_max = 19685.0 ;

float spec(time, n_spectrum, n_frequency) ;
spec:long_name = " " ;
spec:short_name = "spec" ;
spec:description = "One-sided autospectral density function"
spec:resolution = "0.00315 s " ;
spec:units = "(mikroV/m)**2/Hz" ;
spec:valid_min = -3.2899 ;
spec:valid_max = 7.0407 ;
```

2.4.3 CDL: AGW files

The exemplary fields specific for AGW files are presented below in this section. NOTE: there are additional global attributes deifined for AGW event types.

```
//THIS FILE DEFINES SPECIFIC FIELDS FOR AGW EVENTS NetCDF FILES:

float ilat(time) ;
ilat:long_name = "Invariant Latitude (IGRF model)" ;
ilat:short_name = "ilat";
ilat:description = "Geographic Latitude of the satellite position corresponding to it L-shell value." ;
ilat:sampling_frequency = 1  ;
ilat:units = "degrees" ;
ilat:valid_min = 0.0 ;
ilat:valid_max = 90.0 ;

float lval(time) ;
lval:long_name = "McIlwain parameter - L" ;
lval:short_name = "lval";
lval:description = "L-shell value (radial altitude of geomagnetic field apex in Earth's radii)." ;
lval:sampling_frequency = 1 ;
lval:units = "Normalized Magn. Apex Altitude" ;
lval:valid_min = 1 ;
lval:valid_max = 5 ;

float mlt(time) ;
mlt:long_name = "Magnetic Local Time" ;
mlt:short_name = "mlt";
mlt:description = "The longitude angle between the Zero Magnetic meridian (~ -69o) and the sub-solar point i
mlt:sampling_frequency = 0.01 ;
mlt:units = "hours" ;
mlt:valid_min =  0.0 ;
```

```
mlt:valid_max = 24.0 ;

float orbit(time) ;
orbit:long_name = "Orbit number" ;
orbit:short_name = "orbit";
orbit:description = "The number of the spacecraft full rotations around the Earth since the beginning of the DEMETE
orbit:sampling_frequency =  ;
orbit:units = "orbit" ;
orbit:valid_min = 0 ;
orbit:valid_max = 8201;

  double relative_total_density(time) ;
relative_total_density:long_name = "Relative density of atmosphere components - Atmosphere Gravity Waves" ;
relative_total_density:short_name = "relative_total_density";
relative_total_density:description = "AGW" ;
relative_total_density:instrument = "NACS" ;
relative_total_density:instrument_long_name = "Neutral Atmosphere Composition Spectrometer" ;
relative_total_density:units = "percent" ;
relative_total_density:sampling_frequency = 1 ;
relative_total_density:sampling_frequency_units = "Hz" ;
relative_total_density:valid_min=;
relative_total_density:valid_max=;

double trend_total_density(time) ;
trend_total_density:long_name = "Background trend" ;
trend_total_density:short_name = "trend_total_density";
trend_total_density:description = "Background trend in density" ;
trend_total_density:instrument = "NACS" ;
trend_total_density:instrument_long_name = "Neutral Atmosphere Composition Spectrometer" ;
trend_total_density:units = "g/cm^3" ;
trend_total_density:sampling_frequency = 1 ;
trend_total_density:sampling_frequency_units = "Hz" ;
trend_total_density:valid_min=;
trend_total_density:valid_max=;

byte detect_pagw(time);
detect_pagw:long_name = "Polar Atmosphere Gravity Waves distribution" ;
detect_pagw:short_name = "detect_pagw";
detect_pagw:description = "Polar Atmosphere Gravity Waves distribution on whole orbit 1-exist 0-notexist" ;
detect_pagw:sampling_frequency = 1 ;
detect_pagw:sampling_frequency_units = "Hz" ;
detect_pagw:valid_min=;
detect_pagw:valid_max=;

byte detect_lagw(time) ;
detect_lagw:long_name = "Local Wave Packets of Atmosphere Gravity Waves distribution" ;
detect_lagw:short_name = "detect_lagw";
detect_lagw:description = "Local Wave Packets of Atmosphere Gravity Waves distribution on whole orbit 1-exist 0-not
detect_lagw:sampling_frequency = 1 ;
detect_lagw:sampling_frequency_units = "Hz" ;
detect_lagw:valid_min=;
detect_lagw:valid_max=;

byte detect_bagw(time) ;
detect_bagw:long_name = "Background of Atmosphere Gravity Waves distribution" ;
detect_bagw:short_name = "detect_bagw";
detect_bagw:description = "Background of Atmosphere Gravity Waves distribution on whole orbit 1-exist 0-notexist" ;
detect_bagw:sampling_frequency = 1 ;
detect_bagw:sampling_frequency_units = "Hz" ;
detect_bagw:valid_min=;
detect_bagw:valid_max=;

//specific global attributs for AGW netCDF file

:filter_upper_frequency =
:filter_lower_frequency =
:satellite_name=
```

```
:flag_pagw= 1       //0 | 1. 0 means no detection in whole file
:flag_bagw= 1       //0 | 1. 0 means no detection in whole file
:flag_lagw= 0       //0 | 1. 0 means no detection in whole file
```

2.4.4 CDL: TID - TEC files

This section presents fields specific for TID - TEC events. NOTE: there are additional global attributes defined for TID - TEC events.

```
// HERE ARE THE FIELDS SPECIFIC FOR TID_TEC CATALOGUE TYPE

byte detect_mstid(time) ;
detect_mstid:detection_flag= 1;//1-detected, 0-not detected
detect_mstid:long_name = "Flag Medium Scale TID" ;
detect_mstid:short_name = "detect_mstid" ;
detect_mstid:description = "Detection flag for Medium Scale Travelling Ionosphere Disturbance" ;
detect_mstid:resolution = "" ;
detect_mstid:units = "" ;
detect_mstid:valid_min = "" ;
detect_mstid:valid_max = " ;

byte detect_lstid(time) ;
detect_lstid:detection_flag= 1;
detect_lstid:long_name = "Flag Large Scale TID " ;
detect_lstid:short_name = "detect_lstid" ;
detect_lstid:description = "Detection flag for Large Scale Travelling Ionosphere Disturbance" ;
detect_lstid:resolution = "" ;
detect_lstid:units = "" ;
detect_lstid:valid_min = "" ;
detect_lstid:valid_max = " ;

float elangle(time) ;
elangle:long_name = "Elevation Angle" ;
elangle:short_name = "elangle" ;
elangle:description = " Elevation angle of the svn-station link at the observation\'s point" ;
elangle:resolution = "" ;
elangle:units = "degrees" ;
elangle:valid_min = "" ;
elangle:valid_max = " ;

float fvtec_mstid(time) ;
fvtec_mstid:long_name = "Filtered Vertical TEC Medium Scale TID" ;
fvtec_mstid:short_name = "fvtec_mstid" ;
fvtec_mstid:description = "Vertical Total Electron Content filtered in the frequency band defined to observe
fvtec_mstid:resolution = "" ;
fvtec_mstid:units = "TEC units" ;
fvtec_mstid:valid_min = "" ;
fvtec_mstid:valid_max = " ;

float fvtec_lstid(time) ;
fvtec_lstid:long_name = "Filtered Vertical TEC Medium Scale TID" ;
fvtec_lstid:short_name = "fvtec_lstid" ;
fvtec_lstid:description = "Vertical Total Electron Content filtered in the frequency band defined to observe
fvtec_lstid:resolution = "" ;
fvtec_lstid:units = "TEC units" ;
fvtec_lstid:valid_min = "" ;
fvtec_lstid:valid_max = " ;


// global attributes specific for TID-TEC NetCDF data files
:station_id = "" ;
:svn_id = "" ;
:valid_range_mstid = "" ;
:valid_range_lstid = "" ;
```

```
:filter_band_mstid = "" ;
:filter_band_lstid = "" ;
```

2.4.5 CDL: TID files

This section describes fields specific for TID events.

```
// HERE ARE THE FIELDS SPECIFIC FOR TID CATALOGUE TYPE

float ilat(time) ;
ilat:long_name = "invariant latitude" ;
ilat:short_name = "ilat" ;
ilat:description = "Geographic Latitude of the satellite position corresponding to it L-shell value." ;
ilat:resolution = 0.01 ;
ilat:units = "degrees" ;
ilat:valid_min =  0.0 ;
ilat:valid_max = 90.0 ;
ilat:idiosyncracies = "Undefined closely around geomagnetic field poles. Always positive." ;

float lval(time) ;
lval:long_name = "mcilwain parameter - l" ;
lval:short_name = "lval" ;
lval:description = "L-shell value (radial altitude of geomagnetic field apex in Earth's radii)." ;
lval:resolution =  0.01 ;
lval:units = "Normalized Magn. Apex Altitude" ;
lval:valid_min = 1.0 ; // (at geomagnetic equator and on Earth's surface)
lval:valid_max = 50.0 ;
lval:idiosyncracies = "Undefined closely around geomagnetic field poles" ;

float mlt(time) ;
mlt:long_name = "magnetic local time" ;
mlt:short_name = "mlt" ;
mlt:description = "The longitude angle between the Zero Magnetic meridian (~ -69o) and the sub-solar point in hours
mlt:resolution =  0.01 ;
mlt:units = "hours" ;
mlt:valid_min =   0.0 ;
mlt:valid_max =  24.0 ;

float incl(time) ;
incl:long_name = "magnetic inclination" ;
incl:short_name = "incl" ;
incl:description = "The magnetic inclination angle at the spacecraft position in degrees" ;
incl:resolution = 0.01 ;
incl:units = "degrees" ;
incl:valid_min = -90.0 ;
incl:valid_max = +90.0 ;
incl:idiosyncracies = "Positive in the north hemisphere" ;

float decl(time) ;
decl:long_name = "magnetic declination" ;
decl:short_name = "decl" ;
decl:description = "The magnetic declination angle at the spacecraft position in degrees" ;
decl:resolution = 0.01 ;
decl:units = "degrees" ;
decl:valid_min =  -90.0 ;
decl:valid_max =  +90.0 ;
decl:idiosyncracies = "Positive eastward" ;

float dlat(time) ;
dlat:long_name = "dipole latitude" ;
dlat:short_name = "dlat" ;
dlat:description = "The magnetic dip latitude at the spacecraft position in degrees" ;
dlat:resolution = 0.01 ;
dlat:units = "degrees" ;
dlat:valid_min = -90.0 ;
```

```
dlat:valid_max = +90.0 ;
dlat:idiosyncracies = "Positive in the north hemisphere" ;

byte detect_ptid(time) ;//short
detect_ptid:long_name = "Polar TID type" ;
detect_ptid:short_name = "detect_ptid";
detect_ptid:description = "Polar TID detected if detect_ptid = 1" ;
detect_ptid:sampling_frequency = 2.0 ;
detect_ptid:sampling_frequency_units = "Hz" ;
detect_ptid:valid_min= 0 ;
detect_ptid:valid_max= 1 ;

byte detect_ltid(time) ;//short
detect_ltid:long_name = "Local TID type" ;
detect_ltid:short_name = "detect_ltid";
detect_ltid:description = "Local TID detectedif detect_ltid = 1" ;
detect_ltid:sampling_frequency = 2.0 ;
detect_ltid:sampling_frequency_units = "Hz" ;
detect_ltid:valid_min= 0 ;
detect_ltid:valid_max= 1 ;

byte detect_btid(time) ;//short
detect_btid:long_name = "Background TID type" ;
detect_btid:short_name = "detect_btid";
detect_btid:description = "Background TID detected if detect_btid = 1" ;
detect_btid:sampling_frequency = 2.0 ;
detect_btid:sampling_frequency_units = "Hz" ;
detect_btid:valid_min= 0 ;
detect_btid:valid_max= 1 ;

float ni_tot(time) ;
ni_tot:long_name = "ion density" ;
ni_tot:short_name = "ni_tot" ;
ni_tot:description = "Total Ion density" ;
ni_tot:resolution = 1.0 ;
ni_tot:units = "ions per cc" ;
ni_tot:valid_min =  1.0e2 ;
ni_tot:valid_max =  5.0e6 ;

float ni_std(time) ;
ni_std:long_name = "normalized ion density standard deviations ni_std/ni_tot" ;
ni_std:short_name = "ni_std" ;
ni_std:description = "Normalized Total Ion density Standard Deviation" ;
ni_std:resolution =  0.01 ;
ni_std:units = "N/a" ;
ni_std:valid_min = 0.0 ;
ni_std:valid_max = 1.0 ;

float oi_den(time) ;
oi_den:long_name = "o+ density" ;
oi_den:short_name = "oi_den" ;
oi_den:description = "Oxigen Ion density" ;
oi_den:resolution =  1. ;
oi_den:units = "ions per cc" ;
oi_den:valid_min =  1.0e2 ;
oi_den:valid_max =  5.0e6 ;

float he_den(time) ;
he_den:long_name = "he+ density" ;
he_den:short_name = "he_den" ;
he_den:description = "Helium Ion density" ;
he_den:resolution = 1. ;
he_den:units = "ions per cc" ;
he_den:valid_min =  1.0e2 ;
he_den:valid_max =  5.0e6 ;

float hi_den(time) ;
```

```
hi_den:long_name = "h+ density" ;
hi_den:short_name = "hi_den" ;
hi_den:description = "Hydrogen Ion density" ;
hi_den:resolution =  1. ;
hi_den:units = "ions per cc" ;
hi_den:valid_min = 1.0e2 ;
hi_den:valid_max = 5.0e6 ;

float ti_tmp(time) ;
ti_tmp:long_name = "ti ion temperature" ;
ti_tmp:short_name = "ti_tmp" ;
ti_tmp:description = "Ion Temperature" ;
ti_tmp:resolution = 0.1 ;
ti_tmp:units = "Degree Kelvin" ;
ti_tmp:valid_min =  300.0 ;
ti_tmp:valid_max = 10000.0 ;

float ti_std(time) ;
ti_std:long_name = "normalized ion temperature standard deviations ti_std/ti_tmp" ;
ti_std:short_name = "ti_std" ;
ti_std:description = "Normalized Ion Temperature Standard Deviation" ;
ti_std:resolution = 0.01 ;
ti_std:units = "N/a" ;
ti_std:valid_min = 0.0 ;
ti_std:valid_max = 1.0 ;

float vi_ram(time) ;
vi_ram:long_name = "vi ram ion velocity" ;
vi_ram:short_name = "vi_ram" ;
vi_ram:description = "Ram Ion Velocity" ;
vi_ram:resolution = 0.1 ;
vi_ram:units = " m/s " ;
vi_ram:valid_min = -4500.0 ;
vi_ram:valid_max =  4500.0 ;

float vi_ram_std(time) ;
vi_ram_std:long_name = "normalized ram ion velocity standard deviations vi_ram_std/vi_ram" ;
vi_ram_std:short_name = "vi_ram_std" ;
vi_ram_std:description = "Normalized Ram Ion Velocity Standard Deviation" ;
vi_ram_std:resolution = 0.01 ;
vi_ram_std:units = "N/a" ;
vi_ram_std:valid_min = 0.0 ;
vi_ram_std:valid_max = 1.0 ;

float vi_ver(time) ;
vi_ver:long_name = "vi vertical ion velocity" ;
vi_ver:short_name = "vi_ver" ;
vi_ver:description = "Vertical Ion Velocity" ;
vi_ver:resolution = 0.1 ;
vi_ver:units = " m/s " ;
vi_ver:valid_min = -4500.0 ;
vi_ver:valid_max =  4500.0 ;

float vi_ver_std(time) ;
vi_ver_std:long_name = "normalized vertical ion velocity standard deviations vi_ver_std/vi_ver" ;
vi_ver_std:short_name = "vi_ver_std" ;
vi_ver_std:description = "Normalized Vertical Ion Velocity Standard Deviation" ;
vi_ver_std:resolution = 0.01 ;
vi_ver_std:units = "N/a" ;
vi_ver_std:valid_min = 0.0 ;
vi_ver_std:valid_max = 1.0 ;

float vi_hor(time) ;
vi_hor:long_name = "vi horizontal ion velocity" ;
vi_hor:short_name = "vi_hor" ;
vi_hor:description = "Horizontal Ion Velocity" ;
vi_hor:resolution = 0.1 ;
```

```
vi_hor:units = " m/s " ;
vi_hor:valid_min = -4500.0 ;
vi_hor:valid_max =  4500.0 ;

float vi_hor_std(time) ;
vi_hor_std:long_name = "normalized horizontal ion velocity standard deviations vi_hor_std/vi_hor" ;
vi_hor_std:short_name = "vi_hor_std" ;
vi_hor_std:description = "Normalized Horizontal Ion Velocity Standard Deviation" ;
vi_hor_std:resolution = 0.01 ;
vi_hor_std:units = "N/a" ;
vi_hor_std:valid_min = 0.0 ;
vi_hor_std:valid_max = 1.0 ;
```

2.5 Creation of XML files

The XML as well contain common fields and fields specific for different event types. The common part depends on the fact if the actual NetCDF file can be submitted to the database or not due to Data Policy restrictions. In the first case the common part should contain fields:

- description
- contributor_name
- contributor_mail
- date_processing (written in Standard Date Notation ISO 8601: YYYY-MM-DDThh:mmZ)
- algorithm_version
- event_type
- catalogue_id
- processed_file
- index_start
- index_stop

If data cannot be submitted in the NetCDF form, because of the Data Policy restrictions instead of fields 'index_start' and 'index_stop' the file must have defined the following fields:

- source_data_link
- date_start
- date_stop
- glon_start
- glon_stop
- glat_start

- glat_stop
- alt_start
- alt_stop
- alt_min
- alt_max
- lst_start
- lst_stop

Below is presented common structure of the XML file.

```
<popdat version="1">
   <event>
      <description>TID computed from GPS data</description>
      <contributor_name>NOVELTIS</contributor_name>
      <contributor_email>contact@noveltis.fr</contributor_email>
      <date_processing>13-Apr-2012</date_processing>
      <algorithm_version>prototype-v0.2</algorithm_version>
      <event_type>MSTID</event_type>
      <catalogue_id>TIDsTEC</catalogue_id>
      <!--if the data can be submitted-->
      <processed_file>file.nc</processed_file>
      <index_start>18240</index_start><!--the common variable in nc will be assessed-->
      <index_stop>21180</index_stop>
      <!--if the data cannot be submitted-->
      <source_data_link>url</source_data_link>
      <date_start>time</date_start>
      <date_stop>time</date_stop>
      <glon_start>glon</glon_start>
      <glon_stop>glon</glon_stop>
      <glat_start>glat</glat_start>
      <glat_stop>glat</glat_stop>
      <alt_start>altitude</alt_start>
      <alt_stop>altitude</alt_stop>
      <alt_min>altmin</alt_min>
      <alt_max>altmax</alt_max>
      <lst_start>lst</lst_start>
      <lst_stop>lst</lst_stop>

      <!--fields specific for different events-->

   </event>
 </popdat>
```

2.5.1 XML for EMW events

This subsection presents XML fields specific for the EMW events.

```
<popdat version="1">
   <event>

      <!--common xml fields-->

      <!--fields specific for EMW event type-->
      <mlat_model_min>36.6351</mlat_model_min>
```

```
      <mlat_model_max>36.6351</mlat_model_max>
      <mlon_model_min>103.974</mlon_model_min>
      <mlon_model_max>103.974</mlon_model_max>
      <lval_min>1.57768</lval_min>
      <lval_max>1.57768</lval_max>
      <disp_min>1.57768</disp_min>
      <disp_max>1.57768</disp_max>
      <frequency_min>0.0</frequency_min>
      <frequency_max>19685.0</frequency_max>
      <spectrogram_time_resolution>0.00315</spectrogram_time_resolution>
      <detect_significance_value_min>7.0407</detect_significance_value_min>
      <detect_significance_value_max>7.0407</detect_significance_value_max>
    </event>
  </popdat>
```

2.5.2 XML for AGW events

This subsection presents XML fields specific for the AGW events.

```
<popdat version="1">
 <event>


        <!--common xml fields-->

        <!-- special parameters for AGW events -->
        <ilat_start> </ilat_start>
        <ilat_stop> </ilat_stop>
        <lval_start> </lval_start>
        <lval_stop> </lval_stop>
        <mlt_start> </mlt_start>
        <mlt_stop> </mlt_stop>
        <max_amplitude_relative_density> </max_amplitude_relative_density>
        <standart_deviation_relative_density> </standart_deviation_relative_density>
        <max_frequency> </max_frequency>
        <min_frequency> </min_frequency>

   </event>
</popdat>
```

2.5.3 XML for TID - TEC evetns

This subsection presents XML fields specific for the TID - TEC events.

```
<popdat version="1">
   <event>

      <!--common xml fields-->

      <station_id>ACOR</station_id>
      <svn_id>08</svn_id>
      <amplitude_fvtec_abs>  3.1769</amplitude_fvtec_abs>
      <amplitude_fvtec_rel> 72.0306</amplitude_fvtec_rel>
      <period_min> 90.0000</period_min>
      <period_max>120.0000</period_max>
      <elangle_min> 18.7411</elangle_min>
      <elangle_max> 71.1345</elangle_max>
   </event>
</popdat>
```

2.5.4 XML for TID events

This subsection presents XML fields specific for the TID events.

```
<popdat version="1">
<event>

    <!--common xml fields-->

    <!-- special parameters for TID events -->
    <ilat_start>ilat</ilat_start>
    <ilat_stop>ilat</ilat_stop>
    <lval_start>lval</lval_start>
    <lval_stop>lval</lval_stop>
    <mlt_start>mlt</mlt_start>
    <mlt_stop>mlt</mlt_stop>
    <relative_max_wave_amplitude>amplitude</relative_max_wave_amplitude>
    <min_frequency>frequency</min_frequency>
    <max_frequency>frequency</max_frequency>
</event>
</popdat>
```