Consiglio Nazionale delle Ricerche
Istituto di Fisica Applicata
"Nello Carrara"

ESA ESRIN Contract No.: 4000112093/14/I-LG
“Support to MIPAS Level 2 processor Verification and Validation – Phase F”

MIPAS L2 V8
Output Data Definition

Issue:2.0

Document number: IFAC_GA_2018_1_FB

Author: Flavio Barbara

Author: Piera Raspollini

Date: June 4, 2020

List of Abbreviations

ACDD Attribute Convention for Dataset Discovery

ACVE Atmospheric Composition Validation and Evolution

AK Averaging Kernel

CDL Common Data Language, text notation for netCDF objects and data

CF Climate and Forecast

CI Cloud index

CM Covariance Matrix

ECMWF European Centre for Medium-Range Weather Forecasts

ESA European Space Agency

L1b Level 1b

L2 Level 2

LOS Line Of Sight

MIPAS Michelson Interferometer for Passive Atmospheric Sounding

MTR Multi-target retrieval

MW Microwindow

NASA National Aeronautics and Space Administration

netCDF Network Common Data Form

OE Optimal Estimation

ORM Optimized Retrieval Model

pT pressure and temperature

SI International System of Units, from the French Système international (d'unités)

V8 Version 8

VMR Volume Mixing Ratio

Contents

1. Background

The ESA MIPAS Level 2 (L2) (Version 8 (V8)) data set is the final full mission reprocessed data set, collecting all improvements discussed in the frame of the MIPAS Quality Working Group, also taking advantage of the feedback of Validation teams. Among the improvements, a new output format for L2 outputs is adopted. This change answers to a special recommendation from the ACVE Workshop 2013 [4], i.e. to adopt for the L2 output files a more standard output format that can be more easily read by the whole atmospheric community. In particular, it was asked the

> *harmonisation of the data format for all specific instruments, possibly adopting an established, self-describing, easy readable format [Rec 30]. This could be achieved by providing all operational ESA atmospheric composition products in community accepted formats such as netCDF-4 with the netCDF Climate and Forecast (CF) Metadata Convention.*

The choice of the format of the output files was driven by the following needs:

- to select a file format that the users are already familiar with
- to use a base format that is already well supported in data-analysis packages
- to use a self-describing file format
- to build on experience from previous missions
- to adequate the format to the one that will be used by future missions, like the Sentinels.

The choice of both the content and the number of the provided output files was driven by the need to limit the complexity for accessing and using the files by the users, distinguishing between information that is needed by the users and information that is needed for making diagnostics of the products and for special analyses.

2. Purpose and structure of this document

The aim of this document is to describe the concepts and the guide lines followed during the definition of the format and structure of the new output files. Unlike previous versions of the L2 output file, whose manual was very large and fully comprehensive due to the difficulty of describing a custom binary file format, for V8 files we chose a standard self-describing format. As a consequence we do not have to explain where a variable can be found, since there are lots of tools which can be used to easily read the file, instead we have to describe in which way the information is stored inside variables.

In order to reach this goal, Sect. 3 and 4 of this document discuss respectively general features of the ORM V8 algorithm which impact the output files and the choices taken about format. Sect. 5 describes the conventions the names of the files have to be compliant with, Sect.6 lists and explains the global attributes. In Sect.7 all variables are partitioned into logical categories, whose characteristics are explained. The last two sections rely on the Common Data Language (CDL) expansion of the file content, showing in detail all dimensions, variables and attributes of the files.

3. General features of L2 retrieval algorithm impacting the outputs and changes in V8

MIPAS L2 processor, starting from the geolocated and calibrated spectra measured at the tangent altitudes of a scan, performs the inversion on selected spectral intervals, called microwindows (MWs) and derives temperature and VMR profiles in correspondence of the retrieved tangent pressures.

The Optimized Retrieval Model (ORM) V8 processor retrieves the different target species sequentially, after the simultaneous retrieval of temperature and tangent pressures. For each MW an altitude dependent continuum and an altitude independent offset, are fitted together with the profile of the target species. So the state vector of each retrieval, i.e. the array of the fitted parameters, includes either the temperature and pressure profile parameters or Volume Mixing Ratio (VMR) profile of the fitted target species, as well as continuum and offset parameters. The retrieval of continuum and offset improves the performances of the vertical profile retrieval but their outputs are not considered of scientific interest, the main outputs being only the temperature and VMR values retrieved in correspondence of the tangent pressures.

For some species, retrieval is now performed using Optimal Estimation (OE) method[12]. In these cases the output files contain the used a priori data.

Version 8 of the ESA processor can perform multi-target retrieval (MTR), namely the simultaneous analysis of more than one species, in case of species whose spectral features are mutually affecting in the selected micro-windows. In this case the state vector includes the profiles of the different retrieved species, as well as offsets and continuum profiles. Actually multi-target retrieval was not used to build V8 data set.

Furthermore, the L2 algorithm was modified to allow the retrieval of one or more species without the need of performing the whole retrieval sequence, taking advantage of the possibility of interfacing the processor with outputs provided by previous reprocessing (*slicing retrieval*). This feature was not used to build V8 data set, but it was used for making preparatory tests and for processing preceding diagnostic data sets.

In order to allow slicing retrievals, the retrieved profile was extended from 6 to 120 km and included in the output files.

4. Basic choices

4.1 Organisation of the content of the L2 output files

Temporal coverage Each output file contains information of all scans of a single orbit in order to maintain compatibility with the level 1b file structure. The chosen format allows to join many files in one, spanning a time frame as large as desired.

Two different types of output files The approach used to define the structure of the output data files of MIPAS is aimed to limit the complexity in accessing and using the files by the final users but at the same time to provide all information that can be needed for diagnostic and special analysis. To this purpose we released two different kinds of files: a standard file and an extended one.

> **The standard file:** It provides the information generally needed by data users (mainly geolocation information, altitude, pressure, temperature, retrieved profile and related Covariance Matrices (CMs) and Averaging Kernels (AKs), quality flags, ...). It covers a single orbit and a single species. When the species are obtained by a MTR, a number of standard files equal to the number of retrieved species (including temperature, if retrieved) are generated.
>
> **The extended file:** It is thought for expert users and for diagnostics. It completes the information obtained from the retrieval including the full state vector (retrieved profiles, atmospheric continuum and instrumental offset) with related CM and AK, and also with additional information about the retrieval process. This information is useful when making data fusion[5]. Due to the composition of the state vector, variables in the file are heterogeneous, i.e. their elements do not have the same units. The file covers a single orbit and a single retrieval. If the file is the result of a MTR, it includes information about several species.

We have to note that the number of provided output files is different in the case of pressure and temperature (pT), single target or multi-target retrieval (see table 7.5). In particular, for pT and single target VMR retrieval one standard and one extended file is provided. In case of multi-target retrieval, one extended file is provided, while the number of standard files is equal to the number of retrieved species of multi-target retrieval, or a sub-sample of it if some of those are considered to be not reliable.

4.2 Format choices

Data format The format of the output files of the MIPAS L2 V8 data set is the netCDF[10]. This format is used for the outputs of Sentinel missions, as well as of many other missions both at ESA and NASA. Using netCDF-4 allows data users to choose from a wide range of data-analysis packages (like IDL, Matlab, Mathematica) and programming languages (Python, C, C++, Java and Fortran) to access the data. Furthermore, netCDF-4 format allows a lot of advanced features (grouping, compression, user defined data types, ...).

Conventions and units Different choices were made for the two kind of files.

> **Standard file:** Climate and Forecast Metadata Conventions v1.6 [6] and Attribute Convention for Dataset Discovery (ACDD) v1.3 [1] were used in the standard file. Concerning the units, SI was used where possible, with a preference for standard units from "UDUNITS-2" units database [13].
>
> **Extended file:** Due to strong heterogeneity of the quantities stored in variables, no convention could be followed in these files.

Structure of the files The provided files contain a flat list of dimensions and variables that can be read using almost the same simple instructions used to read the well known netCDF-3 files.

The record variable is time and it is the only coordinate variable.

In standard file, dimensions are fixed in all the data set. It allows to easily merge subsets of standard files of the same species with standard netCDF tools.

On the contrary in extended file, dimensions are non fixed for all files, indeed for each retrieval and for each orbit the maximum number of parameters and the maximum number of MWs are estimated, and used to size the related dimensions, in order to limit the size of the extended data set.

Furthermore the reported variables change as a function of the species. In the standard "Temperature" file, resulting from pT retrieval, the temperature variable is not reported since the information about temperature is contained into the profile variable, see section 7.2. In addition cloud_index information is reported in this file only (see section 7.2.2)

When the retrieval is performed using OE, cost function and *a priori* input are also reported.

5. File name conventions

A convention was established to standardise the names of the output files. This names are 61 character length strings composed of fixed length sub-strings joined together by underscore character (_). The details of the convention are reported here and in Table 5.1.

```
MIPAS_2PS_XXXXXXX_20120101T001826_20021212T102113_51453_00.nc
1^^^^^^^2_3^^^^^^_4^^^^^^^^^^^^^^_5^^^^^^^^^^^^^^_6^^^^_7^.8^
```

1. MIPAS level2 file identifier;
2. kind of file identifier (S standard, E extended);
3. identifier for species or retrieval, (7 characters filled with "_" on the right, see table 5.1);
4. start time (date and time, ISO format);
5. stop time (date and time, ISO format);
6. orbit identifier;
7. retrieval identifier, 2 digit;
8. file extension, lower case (".nc").

6. Global attributes

To convey some general properties of a file, netCDF format makes available the global attributes feature.

Table 5.1: Strings used to identify the species

Formula	chemical species	identifier
	Temperature	TEMP (PT on extended file)
H_2O	Water Vapour	H2O
O_3	Ozone	O3
HNO_3	Nitric acid	HNO3
CH_4	Methane	CH4
N_2O	Nitrous oxide	N2O
NO_2	Nitrogen dioxide	NO2
CCl_3F	CFC-11	CCL3F
$ClONO_2$	Chlorine nitrate	CLONO2
N_2O_5	Dinitrogen pentaoxide	N2O5
CCl_2F_2	CFC-12	CCL2F2
COF_2	Carbonyl fluoride	COF2
CCl_4	Carbon tetrachloride	CCL4
HCN	Hydrogen cyanide	HCN
CF_4	CFC-14	CF4
$CHClF_2$	HCFC-22	CHCLF2
C_2H_2	Acetylene	C2H2
CH_3Cl	Chloromethane	CH3CL
$COCl_2$	Phosgene	COCL2
C_2H_6	Ethane	C2H6
OCS	Carbonyl sulfide	OCS
HDO	Deuterium hydrogen oxide	HDO

Most of the defined global attributes are self explaining and their contents deal with information about the creation of the file, references, instruments and platform identifiers and some others are needed to comply with the standard conventions. Some attributes provide information about the retrieval process and the structure of the file.

Here are more details about some remarkable global attributes:

title reports if the file is either a Standard or an Extended file;

species reports the name of the processed species. If the file is an extended file refering to pT or multi-target retrieval, the attribute contains a comma separated list of the retrieved species;

retrieval_type reports if the retrieval was performed using Levenberg-Marquardt or Optimal Estimation algorithm;

processor_version is the major release of the used processor;

auxdata_version is the major release of the auxiliary data.

To get information about minor releases of both software and auxiliary data, users have to read dedicated variables inside the file.

7. Variables

In the standard files as well as in the extended ones, all variables depend on time at least.

In each file all fields are fixed length variables, in order to maintain netCDF-3 compatibility.

Information is not completely duplicated in extended file: actually there are pieces of information that are in standard file but not in extended one and pieces of information present only in the extended file. For 5 variables, reported in both files with the same name (orange colored in table 7.1), the content may be different in presence of “holes” (see section 7.2) which are not reported in the extended file.

7.1 "Informational" variables

7.1.1 Geolocation

In the standard file only one geolocation is reported for each scan, namely the latitude and longitude corresponding to the Line Of Sight (LOS) tangent point closest to the scan mean time. In the extended file, instead, geographical coordinates relative to all tangent points contained in the retrieval grid are also provided.

In addition, the orbital coordinate is provided according to the technical note[11]: as for the geographical coordinates, only one value for each scan (corresponding to the tangent point closest to scan mean time) is reported in the standard file and also the whole profile in the extended one.

7.1.2 About the observations

Some variables are provided to quickly locate the current processed scan in the Level 1b (L1b) data set:

L1b_id is the name of the L1b file used as input by the ORM processor;

orbit_id is the sequential number of the orbit, uniquely defined for the ENVISAT mission;

scan_id is the sequential number of the scan inside the L1b file.

Some other variables are related to insolation condition during the measurement:

day_night is the variable, reported as provided from the L1b processor, indicating the scan insolation according to the following code numbering: -1 if all the sweeps are not lighted by the sun, 1 all the sweeps are lighted by the sun, 0 if only some sweeps are lighted;

solar_zenith_angle is the angle between the zenith and the direction of the solar rays in the LOS tangent point closest to scan mean time and its value is computed by ORM using time and geolocation.

Table 7.1: Comparison between the variables contained in a standard and an extended file. Variables in magenta are present only in standard file, those in cyan are only in extended file, orange variables are in both file with different content (see section.7)

standard file	extended file
double time(time)	double time(time)
char L1b_id(time, len_L1bid)	char L1b_id(time, len_L1bid)
int processor_patchlevel(time)	int processor_patchlevel(time)
byte auxdata_subversion(time)	byte auxdata_subversion(time)
int orbit_id(time)	int orbit_id(time)
int scan_id(time)	int scan_id(time)
byte obs_mode_flag(time)	byte obs_mode_flag(time)
float chi2(time)	float chi2(time)
float cost_function(time)	float cost_function(time)
	int gauss_iterations(time)
	int marquardt_iterations(time)
float lambda_marq(time)	float lambda_marq(time, targets)
byte day_night(time)	byte day_night(time)
float longitude(time)	float longitude(time)
float latitude(time)	float latitude(time)
float solar_zenith_angle(time)	float solar_zenith_angle(time)
float orbital_coordinate(time)	float orbital_coordinate(time)
float ECMWF_altitude_shift(time)	float ECMWF_altitude_shift(time)
byte conv_id(time)	byte conv_id(time)
byte quality_flag(time)	
byte post_quality_flag(time)	
	float longitude_profile(time,level)
	float latitude_profile(time,level)
	float orbital_coordinate_profile(time,level)
float pressure(time, level)	float pressure(time,level)
float pressure_error(time, level)	float pressure_error(time,level)
float height(time, level)	float height(time, level)
float height_error(time, level)	
float temperature(time, level)	float temperature(time, level)
float temperature_error(time, level)	float temperature_error(time, level)
float profile(time, level)	
float profile_error(time, level)	
float covariance_matrix(time, cmdim)	
float averaging_kernel(time, level, level)	
	int nparam_per_target(time, targets)
	byte param_units_flag(time, targets)
	int selected_occupation_matrixflag(time)
	byte effective_occupation_matrix(time, mwindows, level)
	byte retrieval_vectors(time, species, level)
	float state_vector(time, parameters)
	float full_covariance_matrix(time, cmdim)
	float full_averaging_kernel(time, parameters, parameters)
float extended_height(time, extended_level)	float extended_height(time, extended_level)
float extended_pressure(time, extended_level)	
float extended_profile(time, extended_level)	
float a_priori_profile(time, level)	
float a_priori_covariance(time, cmdim)	
float error_p_t_cm(time, cmdim)	
	float extended_left_gradient(time, n_gradients, extended_level)
	float extended_right_gradient(time, n_gradients, extended_level)
	float a_priori_state_vector(time, parameters)
	float a_priori_full_covariance(time, cmdim)

Table 7.2: Summary table of the observation modes

flag value	observation mode
-1	FR Nominal mode
0	RR17(Aug-Sep 2004)
1	OR Nominal mode
2	OR UTLS1
3	OR Middle Atmosphere
4	OR Upper Atmosphere
5	OR Noctilucent Clouds
6	OR Aircrafts Emissions
7	OR UTLS1 old
8	OR UTLS2

obs_mode_flag

This variable is provided for the first time in the ESA MIPAS L2 files. It directly gives the information about the processed observation mode. Table 7.2 shows the correspondence between the values reported in the variable and the observation modes. Details about observation modes can be found in appendix F of the ATBD[3].

The thresholds of χ^2, $\lambda_{Marquardt}$ and Maximum Error used to compute the quality flags are given as attributes of this variable, see section 7.1.5.

7.1.3 Processor and auxiliary data

While the major version of the processor (ORM_V8.xx) and auxiliary data (9.xx) is reported as global attribute, the minor release versions are contained in variables, in order to have a data format that allows to merge some standard files in one without losing this information.

Since versioning of the processor and auxiliary data are uniquely defined, complete versions are enough to have all details about the used processor and auxiliary data.

The filename of the used occupation matrix is needed to complete information about inputs of the processing.

Details about used version of the ORM processor can be found in the detailed processing model document [7] and in the user manual [8].

Details about auxiliary data can be found in auxiliary data description document[2].

The provided variables are reported below:

processor_patchlevel is the processor minor version;

auxdata_subversion is the auxiliary data minor version;

selected_occupation_matrix_flag is an integer defining the used selected matrix, according to the filename convention for occupation matrix file[2]. In particular, the name of the selected occupation matrix file can be obtained concatenating the variable attribute "`occupation_matrix_label_prefix`" with the variable itself, as a 3 digit zero padded integer.

Table 7.3: Output status of the retrieval procedure and related values of the "conv_id" flag

conv_id	Status
0	convergence reached
1	maximum number of macro-iterations exceeded
2	maximum number of micro-iterations exceeded
5	convergence reached, but final matrix was singular
6	maximum number of macro-iterations exceeded, and final matrix was singular
7	maximum number of micro-iterations exceeded, and final matrix was singular

7.1.4 Quality control variables

Details about the retrieval process and its quality is stored in the following variables:

chi2 is the value of the reduced χ^2 at the last iteration: its expected value is 1, but, due to the systematic errors, it can be significantly larger;

cost_function is the value of the reduced cost function at the last iteration (see ATBD [3]), reported only in case of OE retrieval;

lambda_marq is the value of the Marquardt dumping parameter at the last iteration;

conv_id is the convergence flag: it has values related to the output status of the retrieval process (see Table 7.3); good values are 0 and 5;

gauss_iterations is the number of Gauss iterations performed in the retrieval;

marquardt_iterations is the number of Marquardt iterations performed in the retrieval.

7.1.5 Filtering criteria and quality flag

In order to provide an easy way to filter out bad quality profiles, two quality flags are provided, in standard file only, as follows:

quality_flag: it checks whether the following quality control variables:

- χ^2,
- Marquardt parameter, and
- maximum random error in the scan (for retrievals performed with Levenberg-Marquardt method)

are not greater than their related thresholds. The thresholds used for the checks are reported as variable attribute of the variable "obs_mode_flag"; if all quantities are lower than the thresholds, its value is set to 0, otherwise is set to 1;

post_quality_flag: its value is set 0 if both the quality_flag and the conv_id have good values (0 for quality_flag, 1 or 5 for conv_id), otherwise is set to 1.

We recommend to consider as good only the profiles whose "`post_quality_flag`" is equal to 0

7.2 "Profile type" variable

A "profile type" variable is a float variable with a fixed dimension of 27, regardless of the observation mode. The levels are reported in descending order from the top of the atmosphere, following the way in which the measurement was acquired.

The first point always corresponds to the highest altitude of the measurement grid retrieved by ORM. If some levels at the top of the profile are damaged, for example as a result of a corrupted spectrum, or excluded by the occupation matrix, they are not reported. So the first point of a profile is always a valid value. The only exception is the cloud index profile variable, which has a smaller vertical range than the retrieved temperature profile, being defined only up to 44 km: above this altitude, the cloud index is set to the "`missing_value`".

In the standard file the "`missing_value`" is used to indicate either a cloud contaminated or a corrupted sweep, or also a level for which the VMR value is not retrieved, according to the retrieval vector (see 7.3 p.18). The "`_FillValue`" indicates a point outside the retrieval range for the considered mode.

The adoption of two different values, indicating the not available levels, together with information provided by the "informational" variables, provide information on how many levels are expected for the considered observation mode and how many levels are either not included in the retrieval grid or filtered out for cloud contamination (the highest "`missing_value`" indicating the cloud top height) or even skipped because of observation band corruption. This information was not reported in the previous L2 file. A schematic example is reported in figures 7.4a and 7.4b.

In extended file "`missing_value`" is not used, so the "holes", when present, are stripped out and the number of "`_FillValue`" at the end of the file is incremented of the number of the "`missing_value`" in the standard file.

The "profile type" variables are: "`pressure`", "`pressure_error`", "`height`", "`height_error`", "`temperature`", "`temperature_error`", "`profile`", "`profile_error`", "`longitude_profile`", "`latitude_profile`", "`orbital_coordinate_profile`", "`a_priori_profile`".

7.2.1 Height profile and its error

ORM V8 retrieves temperature and pressure values corresponding to the engineering tangent points, while the altitudes are recomputed exploiting hydrostatic equilibrium. This method allows to evaluate the difference between two altitude levels, while in order to obtain the absolute value of the altitude grid, one altitude has to be anchored to a known variable[3, sec:4.9]. Any error in this anchor point will lead to an artificial shift of the entire altitude grid. The total error associated to each altitude has to be computed taking into account the contribution of the random error resulting from the propagation of

Table 7.4: Schematics of the "holes", example: orbit 20716, scan_id 10, OR utls1 mode

(a) Temperature standard file

Pressure	Temperature	Cloud index	
0.442885	266.742340	-88888.8	
0.774718	273.240051	-88888.8	Altitudes out of the range of
1.366089	261.389984	-88888.8	evaluation of the cloud index
2.457789	259.354980	23.987223	
-88888.8	-88888.8	-88888.8	Band A corruption
6.883483	230.588974	12.773521	
10.646873	230.341232	10.305495	
14.277794	228.814789	12.598475	
19.193096	228.487030	10.451555	
25.795712	221.981491	9.883818	
32.372517	213.516647	8.543941	
41.071716	204.194397	7.985986	
52.525150	198.129150	8.819544	
67.723114	194.675980	8.646605	
87.893967	193.293488	7.429865	
-88888.8	-88888.8	3.772150	
-88888.8	-88888.8	2.347401	Filtered out by the cloud index
-88888.8	-88888.8	2.632897	
-88888.8	-88888.8	2.277902	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	Values out of the measurement range:
-99999.9	-99999.9	-99999.9	utls1 scans have only 19 heights
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	

(b) Methane standard file

Pressure	Temperature	CH_4 VMR	
0.442885	266.742340	0.228466	
0.774718	273.240051	0.357467	
1.366089	261.389984	0.729090	
2.457789	259.354980	1.183151	
-88888.8	-88888.8	-88888.8	Band A corruption
6.883483	230.588974	1.326362	
10.646873	230.341232	1.454992	
14.277794	228.814789	-88888.8	
19.193096	228.487030	1.637790	
25.795712	221.981491	-88888.8	Retrieval not performet for this heights:
32.372517	213.516647	1.665591	removed by the occupation matrix
41.071716	204.194397	-88888.8	
52.525150	198.129150	1.671525	
67.723114	194.675980	-88888.8	
87.893967	193.293488	1.674452	
-88888.8	-88888.8	-88888.8	
-88888.8	-88888.8	-88888.8	Filtered out by the cloud index
-88888.8	-88888.8	-88888.8	
-88888.8	-88888.8	-88888.8	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	Values out of the measurement range:
-99999.9	-99999.9	-99999.9	utls1 scans have only 19 heights
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	
-99999.9	-99999.9	-99999.9	

the error on the retrieved pressure and temperature and the uncertainty associated to the altitude used as anchor for rebuilding the altitude grid.

In this release of data the anchor altitude, corresponding to the lowest level of the retrieval range, is corrected using information from the co-located ECMWF ERA interim reanalysis z-p profile, if available. This brings an uncertainty on the lowest point of about 150 m[9].

When a coincident ECMWF profile is not available, the lowest point is anchored to the corresponding engineering altitude as given in the L1b file. This choice brings an uncertainty on the lowest point of about 600 m[9].

The difference between the ECMWF corrected height and the engineering one is reported in "`ECMWF_altitude_shift`"

ECMWF_altitude_shift

This variable reports the correction applied to the lowest engineering altitude (among the ones included in the retrieval grid), which is used to rebuild the height grid from pressure and temperature profiles using hydrostatic equilibrium. The correction is computed exploiting ECMWF pressure and height profiles, when available. If this variable is set to "`_FillValue`" it means that the ECMWF correction was not applied and "`height`" was rebuilt using the lowest engineering altitude as anchor altitude. More details can be found in section 7.2.1.

7.2.2 Cloud index profile

A special case of "profile type" variable is the "`cloud_index`". This field is reported only in the standard "Temperature" file. Its length is the same as the related profile variable, so it contains the same number of "`_FillValue`" in the last part of the vector.

Conversely the first part of the array is filled with "`missing_value`" since CI is evaluated only below 44 Km, see table 7.4a

7.2.3 Extended profiles

These profiles, contained in "`extended_level`" sized variables, may be used by ORM as initial guess or assumed profile and they are needed as input profiles for a slicing retrieval, see section 3.

Extended profiles cover the whole atmosphere from 120 km down to about 2 km below the lowest retrieved level. They are filled with the retrieved values, in which the "holes" are substituted by interpolated values, and above the highest retrieved value and below the lowest one with the scaled IG2 profile[2] .

In each extended file, right and left gradients are reported for pressure, temperature and retrieved species, on the same grid of the extended profiles.

Extended profile variables are:

"`extended_height`", "`extended_pressure`", "`extended_profile`", "`extended_left_gradient`", "`extended_right_gradient`".

Table 7.5: Relationship between type of retrieval, number of retrieved profiles (p,T, VMR), total number of targets, number of provided standard files, *n_gas* is the number of trace species which the retrieval derives.

type of retrieval	number of retrieved profiles (p, T, VMR)	number of targets	number of provided standard files
pT retrieval	2	4	1
single target	1	3	1
multi-target(only gases)	*n_gas*	*n_gas* + 2	*n_gas*
multi-target(with pT)	*n_gas* + 2	*n_gas* + 4	*n_gas* + 1

7.3 State vector and its ancillary variables

The complete state vector is reported in the extended file. This array includes all the retrieved quantities, that can be grouped in three parts:

1. the retrieved profiles, which, according to the type of retrieval, are temperature and pressure profiles (if pT retrieval is performed, considered 2 targets) and/or VMR profile of n_gas gases (considered n_gas targets)

2. the continuum profile for each MW (considered 1 target)

3. the offset value for each MW (considered 1 target)

The complete state vector is contained in a variable of "`parameters`" elements, named "`state_vector`". In order to decode the content of the state vector some ancillary variables are provided:

nparam_per_target is a variable, "`targets`" sized, which contains the number of the elements of the state vector relative to each target, see in table 7.5 the relation between the type of retrieval and the number of targets, as well as the number of provided standard files; the sum of the number of the values of the state vector relative to each target is the effective number of retrieved parameters (we will call them *effective parameters*)

$$effective\_parameters = \sum_{1}^{\texttt{targets}} \texttt{nparam\_per\_target}(itarget) \leq \texttt{parameters}$$

The last two parameters are the number of retrieved atmospheric continua, and the number of retrieved instrument offsets, equal to the number of used MWs. Values of the state vector exceeding this number are filled with "`_FillValue`".

param_units_flag is the flag indicating the units of each type of target, according to table 7.6;

effective_occupation_matrix is the effective used occupation matrix, obtained filtering the nominal occupation matrix, as contained in the auxiliary data, according to cloud coverage and band corruption;

retrieval_vectors are the effective retrieval vectors for the considered scan indicating the correspondences between measured sweeps and retrieval grid for the considered trace species; the effective retrieval vectors are obtained filtering the nominal retrieval vectors, provided in the occupation matrix file, taking into cloud coverage and band corruptions of the scan. Only one array is provided in case of either pT or single target retrieval, *n_gas* arrays for multi-target retrieval, *n_gas* + 1 arrays for multi-target retrieval including pressure and temperature.

The pressure grids of each retrieval of the considered scan, i.e. the pressure levels in correspondence of which the values of profiles are retrieved, are obtained by filtering the "`pressure`" variable with the corresponding retrieval vectors. The same grids can be obtained by stripping out from the variable "`pressure`", contained in the standard files, the values filled with "`_FillValue`" or "`missing_value`"; different targets relative to the same scan can have different numbers of retrieval grid points.

The number of pressure levels of this grid, for the considered trace species, is indicated as igrid in the subsequent parts of the chapter.

7.4 Matrices

Two kind of square matrices are reported in the products: covariance matrices and averaging kernel matrices.

In standard files, the matrices relative to the retrieved single species profile are reported, while in the extended one, those relative to the entire state vector. According to section 7.3 the former has rank equal to *igrid*, the latter equal to *effective_parameters*.

Since, as said at the beginning of this chapter, all variables are fixed in dimension, the matrices are contained in variables whose dimension is generally overestimated. Not used values are filled with "`_FillValue`".

In extended file, the variables containing these matrices are compressed with a "per variable" deflate compression algorithm.

Table 7.6: Meaning of the flag values reporting units

type of target	flag value	units
pressure	0	hPa
temperature	1	K
VMR	2	10^{-6}
continuum trasmission	3	1
offset	4	$\frac{\text{nW}}{(\text{cm}^2\ \text{sr}\ \text{cm}^{-1})}$

7.4.1 Covariance matrices

A covariance matrix is square and symmetric. This allows to store it in a packed format, providing the lower triangular matrix of the effective matrix in a simple array. The array is built queuing the first i values of the i-th row, and spanning all the rows of the matrix.

This array contains $cmdim = irank(irank + 1)/2$ elements different from "_FillValue", while its dimension is $dim(dim + 1)/2$, with dim and $irank$ defined as follows:

standard file

$$dim = \texttt{level} = 27$$
$$irank = igrid(itarget) \leq 27$$

where $itarget$ is the sequential number of the target relative to the retrieved species.

extended file

$$dim = \texttt{parameters}$$
$$irank = effective\_parameters \leq \texttt{parameters}$$

where "parameters" is set to the maximum number of parameters in the file, see section 9.

Covariance matrices in standard file are: "covariance_matrix", "a_priori_covariance" and "error_p_t_cm". The "error_p_t_cm" gives the propagation of pressure and temperature CM in VMR retrieval.

Covariance matrices in extended file are: "full_covariance_matrix" and "a_priori_full_covariance".

A priori CM are present only in files reporting results of optimal estimation retrievals, both in standard and in extended files.

7.4.2 Averaging kernel matrices

The variables containing Averaging Kernels are square matrices. Using the same definitions of section 7.4.1, these are (dim, dim) shaped variables, filled with the effective AK matrices, $(irank, irank)$ shaped, at the top left side.

8. Structure of standard file

The so-called standard file contains, for each species, the time of the measurement (expressed in seconds since 2000-01-01T00:00:00 UTC), all the parameters that allow to locate the measurement both in the L1b file structure and in space, pressure, altitude, cloud index (only in temperature file), temperature profiles with their errors, retrieved profile (temperature or VMRs) with its errors and covariance matrix and averaging kernel matrix. Finally the pressure, temperature and VMR profiles on a extended grid are reported, and also the *a priori* profile with its covariance matrix, if Optimal Estimation is used.

8.1 CDL header template

Here is the template that the standard files follow, any field in the following legend can be found in template as <field>.

Legend of the template values:

StandardFileName Name of the file as defined in section 5;

nscans number of scans currently contained in a file;

species_stdname standard name of the profile of retrieved species, as defined in CF Convention[6];

retrieved_species Common name of the retrieved chemical species;

species_units units of the retrieved species;

retrieval_type Levenberg-Marquardt or Optimal Estimation;

orbit_id orbit number, given as string;

creation_time file creation date and time;

sensing_start_time Date and time when observation starts

sensing_stop_time Date and time when observation stops

minimum_latitude Minimum latitude covered in a file

maximum_latitude Maximum latitude covered in a file

minimum_longitude Minimum longitude covered in a file

maximum_longitude Maximum longitude covered in a file

```
netcdf <StandardFileName> {
dimensions:
    len_L1b_id = 62 ;
    extended_level = 121 ;
    cmdim = 378 ;
    level = 27 ;
    time = UNLIMITED ; // (<nscans> currently)
variables:
    double time(time) ;
            time:long_name = "Scan time (ZPD time of sweep closest to
                              scans mean time)" ;
            time:units = "seconds since 2000-01-01 00:00:00 UTC" ;
            time:standard_name = "time" ;
    char L1b_id(time, len_L1b_id) ;
            L1b_id:long_name = "L1b file used for the retrieval" ;
    int processor_patchlevel(time) ;
            processor_patchlevel:long_name = "Patch level of the processor" ;
    byte auxdata_subversion(time) ;
            auxdata_subversion:long_name = "Subversion of auxiliary files" ;
    int orbit_id(time) ;
            orbit_id:long_name = "Orbit sequential number" ;
    int scan_id(time) ;
            scan_id:long_name = "Scan counter in the L1b file" ;
    byte obs_mode_flag(time) ;
            obs_mode_flag:long_name = "Observation mode flag" ;
            obs_mode_flag:flag_values = -1b, 0b, 1b,
                                         2b, 3b, 4b,
                                         5b, 6b, 7b, 8b ;
            obs_mode_flag:flag_meanings = "fr_nominal rr17 or_nominal
                                  or_utls1 or_middle_atm or_upper_atm
                                  or_nlc or_ae or_utls1_old or_utls2" ;
            obs_mode_flag:chi2_Threshold = "3.6   4.0   3.5
                                            3.2   3.9   3.4
                                            3.6   6.9  20.0  20.0" ;
            obs_mode_flag:lambda_Threshold = "10.0  10.0  10.0
                                              10.0  10.0  10.0
                                              10.0  10.0  10.0  10.0" ;
            obs_mode_flag:max_err_Threshold = "1.3E+00 2.7E+00 8.6E+00
                                               2.2E+00 1.2E+01 1.1E+01
                                            1.2E+01 3.0E+00 1.0E+03 1.0E+03" ;
    float chi2(time) ;
            chi2:long_name = "Final normalized chi square" ;
            chi2:valid_min = 0.f ;
            chi2:_FillValue = -99999.9f ;
    float lambda_marq(time) ;
            lambda_marq:long_name = "Final Marquardt dumping parameter" ;
            lambda_marq:valid_min = 0.f ;
            lambda_marq:_FillValue = -99999.9f ;
    byte day_night(time) ;
            day_night:long_name = "Three level Day/Night flag" ;
            day_night:flag_values = -1b, 0b, 1b ;
            day_night:flag_meanings = "nighttime_scan undef daytime_scan" ;
    float longitude(time) ;
            longitude:long_name = "Longitude of LOS tangent point closest to
                                   scans mean time" ;
            longitude:units = "degrees_east" ;
            longitude:valid_range = -180.f, 180.f ;
            longitude:_FillValue = -99999.9f ;
    float latitude(time) ;
            latitude:long_name = "Latitude of LOS tangent point closest to
                                  scans mean time" ;
            latitude:units = "degrees_north" ;
            latitude:valid_range = -90.f, 90.f ;
            latitude:_FillValue = -99999.9f ;
    float solar_zenith_angle(time) ;
            solar_zenith_angle:long_name = "Solar zenith angle of LOS tangent
                                           point closest to scans mean time" ;
```

```
        solar_zenith_angle:units = "degrees" ;
        solar_zenith_angle:valid_range = 0.f, 180.f ;
        solar_zenith_angle:_FillValue = -99999.9f ;
float orbital_coordinate(time) ;
        orbital_coordinate:long_name = "Orbital coordinate of LOS tangent
                                   point closest to scans mean time" ;
        orbital_coordinate:units = "degree from equator (ascending)" ;
        orbital_coordinate:valid_range = 0.f, 365.f ;
        orbital_coordinate:reference ="TN_UNIBO-IAC_MR-LS_2016_ORMHG_v1.0
                                    - Tech note: implementation of the
                                    horizontal gradients in ORM V8" ;
        orbital_coordinate:_FillValue = -99999.9f ;
float ECMWF_altitude_shift(time) ;
        ECMWF_altitude_shift:long_name = "ECMWF altitude shift wrt
                                          retrieved altitude" ;
        ECMWF_altitude_shift:units = "m" ;
        ECMWF_altitude_shift:_FillValue = -99999.9f ;
        ECMWF_altitude_shift:comment = "Z_ecmwfcorrected - Z_retrieved,
                              (a)if this variable is to a valid value,
                              ECMWF pressure profile was used to place
                           the lowermost altitude used in pT retrieval,
                              (b)if this variable is set to _FillValue,
                            the lowermost altitude used in pT retrieval
                                 was set to its engineering estimate" ;
byte quality_flag(time) ;
        quality_flag:long_name = "Quality flag for the retrieval" ;
        quality_flag:standard_name = "air_temperature status_flag" ;
        quality_flag:flag_values = 0b, 1b ;
        quality_flag:flag_meanings = "reliable_data  unreliable_data" ;
byte conv_id(time) ;
        conv_id:long_name = "Convergency flag" ;
        conv_id:flag_values = 0b, 1b,
                              2b, 5b,
                              6b, 7b ;
        conv_id:flag_meanings = "convergence_reached max_macro_exceed
                    max_micro_exceed convergence_reached_sing_mat
            max_macro_exceed_sing_mat max_micro_exceed_sing_mat" ;
byte post_quality_flag(time) ;
        post_quality_flag:long_name = "A posteriori quality flag" ;
        post_quality_flag:flag_values = 0b, 1b ;
        post_quality_flag:flag_meanings = "reliable_data
                                           unreliable_data" ;
float pressure(time, level) ;
        pressure:long_name = "Pressure" ;
        pressure:units = "hPa" ;
        pressure:standard_name = "air_pressure" ;
        pressure:valid_range = 0.f, 10000.f ;
        pressure:missing_value = -88888.8f ;
        pressure:_FillValue = -99999.9f ;
float pressure_error(time, level) ;
        pressure_error:long_name = "Pressure error" ;
        pressure_error:units = "hPa" ;
        pressure_error:standard_name = "air_pressure standard_error" ;
        pressure_error:missing_value = -88888.8f ;
        pressure_error:_FillValue = -99999.9f ;
float height(time, level) ;
        height:long_name = "Height" ;
        height:units = "km" ;
        height:standard_name = "height_above_reference_ellipsoid" ;
        height:valid_range = 0.f, 200.f ;
        height:missing_value = -88888.8f ;
        height:_FillValue = -99999.9f ;
        height:comment = "(a)If the variable ECMWF_altitude_shift
                         is set to a valid value this variable
                         reports the tangent heights of the scan,
                         anchored to the lowermost altitude used in pT
                         retrieval as obtained from ECMWF pressure profile;
```

```
                        (b)if the variable ECMWF_altitude_shift is
                        set to the _FillValue it reports the tangent
                        heights  of the scan anchored to the engeneering
                     estimate the lowermost altitude used in pT retrieval" ;
        height:reference = "IFAC_GA_2007_12_SC issue:7 Rev:0 -
                          High level algorithm definition and physical and
                            mathematical optimisations (MIPAS Level 2
                            Algorithm Theoretical Baseline Document)" ;
    float height_error(time, level) ;
        height_error:long_name = "Height error" ;
        height_error:units = "km" ;
        height_error:standard_name = "height_above_reference_ellipsoid
                                                 standard_error" ;
        height_error:missing_value = -88888.8f ;
        height_error:_FillValue = -99999.9f ;
        height_error:comment = "It is the error of the retrieved tangent
             heights,assuming the lowest tangent altitude as known and hence
             with 0 error. To get the total error of heights one should add to
                   this error component the error of the lowermost height:
             (a)If the variable ECMWF_altitude_shift is set to a valid value,
             the error obtained from the ECMWF ERA Interim pressure profile;
             (b)if the variable ECMWF_altitude_shift is set to the _FillValue,
                      the engineering pointing error.
             According to Multi-TASTE validation, this uncertainty generally
                      consists of: (a)150m; (b)600m.
               Reference: https://earth.esa.int/documents/700255/2621625/
                  TN-BIRA-IASB-MultiTASTE-Phase-F-MIPAS-ML2PP7-Iss1-RevB/
                         34f7f395-75ef-46c4-855e-a0f9d225e7c2" ;
    float temperature(time, level) ;
        temperature:long_name = "Temperature" ;
        temperature:units = "K" ;
        temperature:standard_name = "air_temperature" ;
        temperature:valid_range = 0.f, 350.f ;
        temperature:missing_value = -88888.8f ;
        temperature:_FillValue = -99999.9f ;
    float temperature_error(time, level) ;
        temperature_error:long_name = "Temperature error" ;
        temperature_error:units = "K" ;
        temperature_error:standard_name = "air_temperature
                                               standard_error" ;
        temperature_error:missing_value = -88888.8f ;
        temperature_error:_FillValue = -99999.9f ;
    float cloud_index(time, level) ;
        cloud_index:long_name = "Cloud detection index" ;
        cloud_index:units = "1" ;
        cloud_index:missing_value = -88888.8f ;
        cloud_index:_FillValue = -99999.9f ;
        cloud_index:references = "IFAC_GA_2007_12_SC issue:7 Rev:0 -
                      High level algorithm definition and physical
                      and mathematical optimisations (MIPAS Level 2
                      Algorithm Theoretical Baseline Document)" ;
    float profile(time, level) ;
        profile:long_name = "Vertical profile of <retrieved_species>" ;
        profile:units = "<species_units>" ;
        profile:standard_name = "<species_stdname>" ;
        profile:valid_range = 0.f, 400.f ;
        profile:missing_value = -88888.8f ;
        profile:_FillValue = -99999.9f ;
    float profile_error(time, level) ;
        profile_error:long_name = "Standard error of
                                Vertical profile of <retrieved_species>" ;
        profile_error:units = "<species_units>" ;
        profile_error:standard_name = "<species_stdname> standard_error";
        profile_error:missing_value = -88888.8f ;
        profile_error:_FillValue = -99999.9f ;
    float covariance_matrix(time, cmdim) ;
        covariance_matrix:long_name = "Covariance matrix of
```

```
                                    Vertical profile of <retrieved_species>" ;
            covariance_matrix:units = "<species_units>**2" ;
            covariance_matrix:comment = "First i elements of row #i" ;
            covariance_matrix:_FillValue = -99999.9f ;
    float averaging_kernel(time, level, level) ;
            averaging_kernel:long_name = "Averaging kernel of
                                   Vertical profile of  <retrieved_species>" ;
            averaging_kernel:units = "1" ;
            averaging_kernel:comment = "the rows of the matrix are the
                                          averaging kernels" ;
            averaging_kernel:_FillValue = -99999.9f ;
    float extended_height(time, extended_level) ;
            extended_height:long_name = "Extended height" ;
            extended_height:units = "km" ;
            extended_height:valid_range = 0.f, 200.f ;
            extended_height:missing_value = -88888.8f ;
            extended_height:_FillValue = -99999.9f ;
    float extended_pressure(time, extended_level) ;
            extended_pressure:long_name = "Extended pressure" ;
            extended_pressure:units = "hPa" ;
            extended_pressure:valid_range = 0.f, 10000.f ;
            extended_pressure:missing_value = -88888.8f ;
            extended_pressure:_FillValue = -99999.9f ;
    float extended_profile(time, extended_level) ;
            extended_profile:long_name = "Extended Vertical
                                          profile of <retrieved_species>" ;
            extended_profile:units = "<species_units>" ;
            extended_profile:valid_range = 0.f, 400.f ;
            extended_profile:missing_value = -88888.8f ;
            extended_profile:_FillValue = -99999.9f ;
    float a_priori_profile(time, level) ;
            a_priori_profile:long_name = "A priori VMR vertical profile of
                                          <retrieved_species>" ;
            a_priori_profile:units = "<species_units>" ;
            a_priori_profile:missing_value = -88888.8f ;
            a_priori_profile:_FillValue = -99999.9f ;
    float a_priori_covariance(time, cmdim) ;
            a_priori_covariance:long_name = "A priori covariance matrix of
                              VMR vertical profile of <retrieved_species>" ;
            a_priori_covariance:units = "<species_units>**2" ;
            a_priori_covariance:comment = "First i elements of row #i" ;
            a_priori_covariance:_FillValue = -99999.9f ;
    float error_p_t_cm(time, cmdim) ;
           error_p_t_cm:long_name = "Error pT covariance VMR vertical profile
                                      of <retrieved_species>" ;
            error_p_t_cm:units = "<species_units>**2" ;
            error_p_t_cm:comment = "First i elements of row #i" ;
            error_p_t_cm:_FillValue = -99999.9f ;

// global attributes:
            :title = "Level 2 MIPAS products - simplified file" ;
            :Conventions = "CF1.6 ACDD-1.3" ;
            :species = "<retrieved_species>" ;
            :retrieval_type = "<retrieval_type>" ;
            :platform = "ENVISAT" ;
            :sat_id = "EN" ;
            :sensor = "MIPAS" ;
            :source = "MIPAS satellite observations" ;
            :level = "L2" ;
            :temporal = "orbital" ;
            :product_type = " MIPAS_2PS_" ;
            :orbit = "<orbit_id>" ;
            :comment = "MIPAS reprocessing 2018" ;
            :id =  ;
            :naming_authority = "ESA" ;
            :institution = "European Space Agency (ESA)" ;
            :date_created = "<creation_time>" ;
```

```
:creator_name = "MIPAS Quality Working Group (QWG)" ;
:creator_url =
       "https://earth.esa.int/web/guest/missions/esa-operational-
         eo-missions/envisat/instruments/mipas" ;
:creator_email = "eohelp@esa.int" ;
:processor_version = "ORM_V8.xx" ;
:auxdata_version = "9.xx" ;
:references = "https://earth.esa.int/web/sppa/mission-performance/
               esa-missions/envisat/mipas/products-and-algorithms/
               products-information" ;
:reference_document = "IFAC_GA_2018_1_FB issue:2.0 --
                       MIPAS L2 V8 output data definition" ;
:time_coverage_start = "<sensing_start_time>" ;
:time_coverage_end = "<sensing_stop_time>" ;
:geospatial_lat_min = <minimum_latitude> ;
:geospatial_lat_max = <maximum_latitude> ;
:geospatial_lon_min = <minimum_longitude> ;
:geospatial_lon_max = <maximum_longitude> ;
```

9. Structure of extended file

In order to limit disk occupation of the extended files set, some variables are sized on "per file" basis, forgoing the possibility of merging the files.

For reaching this goal the "`parameters`" dimension, that applies to the state vector, but also to the covariance and the AK matrices, is set to the maximum number of parameters found in state vectors for all scans contained in the file. The "`mwindows`" dimension is set to the maximum number of MWs used in the orbit. The "`species`" dimension is set to the number of trace species of which the retrieval is performed, while the "`target`" dimension is set to "`species` + 2" (continuum and offset are considered as targets); the relationship between "`target`" and "`species`" is more clearly explained in table 7.5.

9.1 How to get information from state vector and its related matrices

As said in section 7.3, the state vector is the union of "`targets`" subvectors. Each of these is made of "`nparam_per_target`" elements.

In order to estract information from the state vector (but also from the CMs and AKs), user has to use the content of the ancillary variables: "`nparam_per_target`" to get the number of parameters relative to the target of interest; "`param_units_flag`", with its "`flag_meanings`" attribute, to get units; "`pressure`" (or "`height`"), filtered with "`retrieval_vectors`", to get pressure (altitude) grid.

To take information from "`full_covariance_matrix`" and "`full_averaging_kernel`", the number of elements from "`nparam_per_target`" has to be considered to extract the related submatrices.

9.2 CDL header template

Here is the template that the extended files follow, any field in the following legend can be found in template as <field>.

Legend of the template values:

ExtendedFileName name of the file as defined in section 5;

nmaxparams maximum number of parameters in the file;

nmaxmws maximum number of MWs in the file;

ntargets number of targets;

nspecies number of gases of which the retrieval is performed;

nscans number of scans currently contained in a file;

retrieved_targets common names of the retrieved chemical species, comma separated;

retrieval_type Levenberg-Marquardt or Optimal Estimation;

orbit_id orbit number, given as string;

creation_time file creation date and time;

sensing_start_time date and time when observation start;

sensing_stop_time date and time when observation stop.

```
netcdf <ExtendedFileName> {
dimensions:
    len_L1b_id = 62 ;
    extended_level = 121 ;
    n_gradients = 2 ;
    cmdim = <nmaxparams>*(<nmaxparams>+1)/2 ;
    parameters = <nmaxparams> ;
    mwindows = <nmaxmws> ;
    targets = <ntargets> ;
    species = <nspecies> ;
    level = 27 ;
    time = UNLIMITED ; // (<nscans> currently)
variables:
    double time(time) ;
            time:long_name = "Scan time (ZPD time of sweep closest to
                              scans mean time)" ;
            time:units = "seconds since 2000-01-01 00:00:00 UTC" ;
            time:standard_name = "time" ;
    char L1b_id(time, len_L1b_id) ;
            L1b_id:long_name = "L1b file used for the retrieval" ;
    int processor_patchlevel(time) ;
            processor_patchlevel:long_name = "Patch level of the processor" ;
    byte auxdata_subversion(time) ;
            auxdata_subversion:long_name = "Subversion of auxiliary files" ;
    int orbit_id(time) ;
            orbit_id:long_name = "Orbit sequential number" ;
    int scan_id(time) ;
            scan_id:long_name = "Scan counter in the L1b file" ;
    byte obs_mode_flag(time) ;
            obs_mode_flag:long_name = "Observation mode flag" ;
            obs_mode_flag:flag_values = -1b, 0b, 1b,
                                        2b, 3b, 4b,
                                        5b, 6b, 7b, 8b ;
            obs_mode_flag:flag_meanings = "fr_nominal rr17 or_nominal
                                  or_utls1 or_middle_atm or_upper_atm
                                  or_nlc or_ae or_utls1_old or_utls2" ;
            obs_mode_flag:chi2_Threshold = "3.6   4.0   3.5
                                           3.2   3.9   3.4
                                           3.6   6.9  20.0  20.0" ;
            obs_mode_flag:lambda_Threshold = "10.0  10.0  10.0
                                             10.0  10.0  10.0
                                             10.0  10.0  10.0  10.0" ;
            obs_mode_flag:max_err_Threshold = "1.3E+00 2.7E+00 8.6E+00
                                               2.2E+00 1.2E+01 1.1E+01
                                          1.2E+01 3.0E+00 1.0E+03 1.0E+03" ;
    float chi2(time) ;
            chi2:long_name = "Final normalized chi square" ;
            chi2:valid_min = 0.f ;
            chi2:_FillValue = -99999.9f ;
    int gauss_iterations(time) ;
            gauss_iterations:long_name = "Number of Gauss iterations
                                                              performed" ;
    int marquardt_iterations(time) ;
            marquardt_iterations:long_name = "Number of Marquardt iterations
                                                              performed" ;
```

```
    float lambda_marq(time, targets) ;
            lambda_marq:long_name = "Final Marquardt dumping parameters" ;
            lambda_marq:valid_min = 0.f ;
            lambda_marq:_FillValue = -99999.9f ;
    byte day_night(time) ;
            day_night:long_name = "Three level Day/Night flag" ;
            day_night:flag_values = -1b, 0b, 1b ;
            day_night:flag_meanings = "nighttime_scan undef daytime_scan" ;
    float longitude(time) ;
            longitude:long_name = "Longitude of LOS tangent point closest to
                                   scans mean time" ;
            longitude:units = "degrees_east" ;
            longitude:valid_range = -180.f, 180.f ;
            longitude:_FillValue = -99999.9f ;
    float latitude(time) ;
            latitude:long_name = "Latitude of LOS tangent point closest to
                                  scans mean time" ;
            latitude:units = "degrees_north" ;
            latitude:valid_range = -90.f, 90.f ;
            latitude:_FillValue = -99999.9f ;
    float solar_zenith_angle(time) ;
            solar_zenith_angle:long_name = "Solar zenith angle of LOS tangent
                                           point closest to scans mean time" ;
            solar_zenith_angle:units = "degrees" ;
            solar_zenith_angle:valid_range = 0.f, 180.f ;
            solar_zenith_angle:_FillValue = -99999.9f ;
    float orbital_coordinate(time) ;
            orbital_coordinate:long_name = "Orbital coordinate of LOS tangent
                                           point closest to scans mean time" ;
            orbital_coordinate:units = "degree from equator (ascending)" ;
            orbital_coordinate:valid_range = 0.f, 365.f ;
            orbital_coordinate:reference ="TN_UNIBO-IAC_MR-LS_2016_ORMHG_v1.0
                                           - Tech note: implementation of the
                                            horizontal gradients in ORM V8" ;
            orbital_coordinate:_FillValue = -99999.9f ;
    float ECMWF_altitude_shift(time) ;
            ECMWF_altitude_shift:long_name = "ECMWF altitude shift wrt
                                               retrieved altitude" ;
            ECMWF_altitude_shift:units = "m" ;
            ECMWF_altitude_shift:_FillValue = -99999.9f ;
            ECMWF_altitude_shift:comment = "Z_ecmwfcorrected - Z_retrieved,
                                       (a)if this variable is to a valid value,
                                        ECMWF pressure profile was used to place
                                   the lowermost altitude used in pT retrieval,
                                      (b)if this variable is set to _FillValue,
                                    the lowermost altitude used in pT retrieval
                                          was set to its engineering estimate" ;
    byte conv_id(time) ;
            conv_id:long_name = "Convergency flag" ;
            conv_id:flag_values = 0b, 1b,
                                  2b, 5b,
                                  6b, 7b ;
            conv_id:flag_meanings = "convergence_reached max_macro_exceed
                                  max_micro_exceed convergence_reached_sing_mat
                         max_macro_exceed_sing_mat max_micro_exceed_sing_mat" ;
    float longitude_profile(time, level) ;
            longitude_profile:long_name = "Longitude of LOS tangent points" ;
            longitude_profile:units = "degree_east" ;
            longitude_profile:standard_name = "longitude" ;
            longitude_profile:_FillValue = -99999.9f ;
    float latitude_profile(time, level) ;
            latitude_profile:long_name = "Latitude of LOS tangent points" ;
            latitude_profile:units = "degree_north" ;
            latitude_profile:standard_name = "latitude" ;
            latitude_profile:_FillValue = -99999.9f ;
    float orbital_coordinate_profile(time, level) ;
            orbital_coordinate_profile:long_name = "Orbital coordinate of LOS
```

```
                                                        tangent points" ;
            orbital_coordinate_profile:units = "degree from equator
                                                          (ascending)" ;
            orbital_coordinate_profile:valid_range = 0.f, 365.f ;
            orbital_coordinate_profile:reference =
                                       "TN_UNIBO-IAC_MR-LS_2016_ORMHG_v1.0
                                        - Tech note: implementation of the
                                        horizontal gradients in ORM V8" ;
            orbital_coordinate_profile:_FillValue = -99999.9f ;
    float height(time, level) ;
            height:long_name = "Height" ;
            height:units = "km" ;
            height:standard_name = "height_above_reference_ellipsoid" ;
            height:valid_range = 0.f, 200.f ;
            height:_FillValue = -99999.9f ;
            height:comment = "(a)If the variable ECMWF_altitude_shift
                               is set to a valid value this variable
                               reports the tangent heights of the scan,
                               anchored to the lowermost altitude used in pT
                               retrieval as obtained from ECMWF pressure profile;
                               (b)if the variable ECMWF_altitude_shift is
                               set to the _FillValue it reports the tangent
                               heights  of the scan anchored to the engeneering
                           estimate the lowermost altitude used in pT retrieval" ;
            height:reference = "IFAC_GA_2007_12_SC issue:7 Rev:0 -
                                  High level algorithm definition and physical and
                                    mathematical optimisations (MIPAS Level 2
                                    Algorithm Theoretical Baseline Document)" ;
    float pressure(time, level) ;
            pressure:long_name = "Pressure" ;
            pressure:units = "hPa" ;
            pressure:standard_name = "air_pressure" ;
            pressure:valid_range = 0.f, 10000.f ;
            pressure:_FillValue = -99999.9f ;
    float pressure_error(time, level) ;
            pressure_error:long_name = "Pressure error" ;
            pressure_error:units = "hPa" ;
            pressure_error:standard_name = "air_pressure standard_error" ;
            pressure_error:_FillValue = -99999.9f ;
    float temperature(time, level) ;
            temperature:long_name = "Temperature" ;
            temperature:units = "K" ;
            temperature:standard_name = "air_temperature" ;
            temperature:valid_range = 0.f, 350.f ;
            temperature:_FillValue = -99999.9f ;
    float temperature_error(time, level) ;
            temperature_error:long_name = "Temperature error" ;
            temperature_error:units = "K" ;
          temperature_error:standard_name = "air_temperature standard_error" ;
            temperature_error:missing_value = -88888.8f ;
            temperature_error:_FillValue = -99999.9f ;
    int nparam_per_target(time, targets) ;
            nparam_per_target:long_name = "Number of parameter for
                                                         each target" ;
            nparam_per_target:only_one_continuum_profile = "F" ;
            nparam_per_target:_FillValue = -99999 ;
    byte param_units_flag(time, targets) ;
            param_units_flag:long_name = "Flag indicating the
                                       measurement units of each target" ;
            param_units_flag:flag_values = 0b, 1b, 2b, 3b, 4b ;
            param_units_flag:flag_meanings = "hPa K 1E-6 1
                                                   nW/(cm^2 sr cm^{-1})" ;
            param_units_flag:_FillValue = -99b ;
    int selected_occupation_matrix_flag(time) ;
            selected_occupation_matrix_flag:occupation_matrix_label_prefix =
                                                                 "OM_PT__" ;
            selected_occupation_matrix_flag:flag_meanings = "1:nominal
```

```
                                                            501:utls1
                                                            701:ma
                                                            801:ua
                                                            601:ae
                                                            901:nlc
                                                            401:utls1_old
                                                            301:utls2" ;
            selected_occupation_matrix_flag:comment = "The name of the
                                selected occupation matrix can be obtained
                              concatenating the occupation_matrix_label_prefix
                                with the selected_occupation_matrix_flag,
                                as a 3 digit zero padded integer" ;
            selected_occupation_matrix_flag:_FillValue = -99999 ;
    byte effective_occupation_matrix(time, mwindows, level) ;
            effective_occupation_matrix:_FillValue = -99b ;
    byte retrieval_vectors(time, species, level) ;
            retrieval_vectors:_FillValue = -99b ;
    float state_vector(time, parameters) ;
            state_vector:long_name = "Array of all the retrieved parameters";
            state_vector:_FillValue = -99999.9f ;
    float full_covariance_matrix(time, cmdim) ;
            full_covariance_matrix:long_name = "Covariance matrix of all
                                                    retrieved parameters" ;
            full_covariance_matrix:comment = "First i elements of row #i" ;
            full_covariance_matrix:_FillValue = -99999.9f ;
    float full_averaging_kernel(time, parameters, parameters) ;
            full_averaging_kernel:long_name = "Averaging kernel of all
                                                    retrieved parameters" ;
            full_averaging_kernel:comment = "the rows of the matrix are
                                                   the averaging kernels" ;
            full_averaging_kernel:_FillValue = -99999.9f ;
    float extended_height(time, extended_level) ;
            extended_height:long_name = "Extended height" ;
            extended_height:units = "km" ;
            extended_height:valid_range = 0.f, 200.f ;
            extended_height:_FillValue = -99999.9f ;
    float extended_left_gradient(time, n_gradients, extended_level) ;
            extended_left_gradient:long_name = "Left gradient
                                                    on the extended grid" ;
            extended_left_gradient:units = "km" ;
            extended_left_gradient:_FillValue = -99999.9f ;
    float extended_right_gradient(time, n_gradients, extended_level) ;
            extended_right_gradient:long_name = "Right gradient
                                                    on the extended grid" ;
            extended_right_gradient:units = "km" ;
            extended_right_gradient:_FillValue = -99999.9f ;
    float a_priori_state_vector(time, parameters) ;
            a_priori_state_vector:long_name = "A priori array of
                                            all the retrieved parameters" ;
            a_priori_state_vector:_FillValue = -99999.9f ;
    float a_priori_full_covariance(time, cmdim) ;
           a_priori_full_covariance:long_name = "A priori covariance matrix of
                                            all the retrieved parameters" ;
            a_priori_full_covariance:comment = "First i elements of row #i" ;
            a_priori_full_covariance:_FillValue = -99999.9f ;

// global attributes:
            :title = "Level 2 MIPAS products - extended file" ;
            :species = "<retrieved_targets>" ;
            :retrieval_type = "<retrieval_type>" ;
            :platform = "ENVISAT" ;
            :sat_id = "EN" ;
            :sensor = "MIPAS" ;
            :source = "MIPAS satellite observations" ;
            :level = "L2" ;
            :temporal = "orbital" ;
            :product_type = " MIPAS_2PE_" ;
```

```
:orbit = "<orbit_id>" ;
:comment = "MIPAS reprocessing 2018" ;
:id = "<ExtendedFileName>" ;
:naming_authority = "ESA" ;
:institution = "European Space Agency (ESA)" ;
:date_created = "<creation_time>" ;
:creator_name = "MIPAS Quality Working Group (QWG)" ;
:creator_url =
       "https://earth.esa.int/web/guest/missions/esa-operational-
         eo-missions/envisat/instruments/mipas" ;
:creator_email = "eohelp@esa.int" ;
:processor_version = "ORM_V8.xx" ;
:auxdata_version = "9.xx" ;
:references = "https://earth.esa.int/web/sppa/mission-performance/
              esa-missions/envisat/mipas/products-and-algorithms/
              products-information" ;
:reference_document = "IFAC_GA_2018_1_FB issue:2.0 --
                       MIPAS L2 V8 output data definition" ;
:time_coverage_start = "<sensing_start_time>" ;
:time_coverage_end = "<sensing_stop_time>" ; }
```

Reference Documents

[1] Earth Science Information Partners (ESIP). *Attribute Convention for Data Discovery v1.3.* 2015. URL: http://wiki.esipfed.org/index.php/Attribute_Convention_for_Data_Discovery_1-3.

[2] Massimo Bianchini, Piera Raspollini, and Flavio Barbara. *Format of Auxiliary Data ADF2_V9.06 for ORM V8.* IFAC_GA_2019_1_pr. Version 1. CNR – IFAC. ESA, 2019.

[3] B. Carli et al. *High level algorithm definition and physical and mathema tical optimisations (MIPAS Level 2 Algorithm Theoretical Baseline Document).* IFAC_GA_2007_12_SC. Version issue:7 Rev:0. 2019. URL: https://earth.esa.int/documents/700255/2621625/Algorithm+Theoretical+Baseline+Document++MIPAS_L2_ATBD_Issue_7.0.pdf.

[4] Stefano Casadio et al. "The Atmospheric Composition Validation and Evolution Workshop (ACVE2013) - Recommendations". In: *Annals of Geophysics* 56.0 (2014). ISSN: 2037-416X. URL: http://www.annalsofgeophysics.eu/index.php/annals/article/view/6535.

[5] Simone Ceccherini, Bruno Carli, and Piera Raspollini. "Equivalence of data fusion and simultaneous retrieval". In: *Opt. Express* 23.7 (Apr. 2015), pp. 8476–8488. DOI: 10.1364/OE.23.008476. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-7-8476.

[6] Brian Eaton et al. *NetCDF Climate and Forecast (CF) Metadata Conventions - Version 1.6.* Accessed: 2015-10-01. Dec. 2011. URL: http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html.

[7] Marco Gai, Flavio Barbara, and Piera Raspollini. *MIPAS L2 V8 Detailed Processing Model.* IFAC_GA_2019_4_MG. Version 1. CNR – IFAC, 2019. in preparation.

[8] Marco Gai, Piera Raspollini, and Flavio Barbara. *ORM V8 User Manual.* IFAC_GA_2017_3_mg. Software Manual. Version 2.2. 2018.

[9] D. Hubert et al. *Multi-TASTE Phase F Validation Report / Comparison of MIPAS ML2PP 7.03 products to sonde and lidar.* Validation Report TN-BIRA-IASB-MultiTASTE-Phase-F-MIPAS-ML2PP7-Iss1-RevB. Version Issue 1 Rev. B. BIRA-IASB,

[10] Russ Rew et al. *The NetCDF User's Guide.* Accessed: 2019-10-18. 2019. URL: https://www.unidata.ucar.edu/software/netcdf/docs/user_guide.html.

[11] Marco Ridolfi and Luca Sgheri. *Tech note: implementation of the horizontal gradients in the ORM V8.* TN_UNIBO-IAC_MR-LS_2016_ORMHG. Version 1. CNR – IAC. ESA, 2016.

[12] C.D. Rodgers. *Inverse Methods for Atmospheric Sounding: Theory and Practice.* Series on atmospheric, oceanic and planetary physics. World Scientific, 2000. ISBN: 9789810227401.

[13] University Corporation for Atmospheric Research (UCAR). *UDUNITS 2.2.19 Manual*. Accessed: 2015-10-05. 2014. URL: `http://www.unidata.ucar.edu/software/udunits/udunits-2.2.19/doc/udunits/udunits2.html%5C#Database`.