EARLINET Data Quality Check Procedure – v3.0

Summary

V1.1

09 June 2017

1- Introduction

Data quality check procedures for EARLINET/ACTRIS database improved over the time. This document reports the automatic quality check procedures working on the EARLINET database since 25 March 2021.

When a product is submitted to the EARLINET database the following steps occur:

1. technical quality controls (BQC) are executed to ensure the product is compliant mainly from technical point of view with the defined standard.
2. advanced quality controls (AQC) are executed to assess the quality from a physical point of view of the product.
3. The 3 variables "quality_control_level", "basic_quality_control", and "advanced_quality_control" are computed by the ARES Data Center according to the results of the previous two steps and added into the product.
4. The product is renamed according to the filename conventions described earlier and definitively stored into the datacenter

Products not passing the step 1 are not accepted by the datacenter and a corresponding error message is shown to the data originator.
Products passing the step 1, but not passing the step 2 are accepted by the datacenter and labelled as Level 1 products.
Products passing both steps 1 and 2 are accepted by the datacenter and labelled as Level 2 products.
Both Level 1 and Level 2 products are made public as soon as they are accepted by the datacenter. This means that a product once submitted and accepted by the datacenter (independently of the assigned level) cannot be deleted anymore.
It is possible to submit a new version of an already submitted product only in specific time window communicated by the datacenter administrators.
A specific Record Version Control system has been developed to allow multiple versions of the same product. This is a primary and necessary tool both for data originators and end-users. Indeed, sometimes it can happen that data originators may realize that something is wrong or not optimized in the products already uploaded on database. Besides, if a new version of the retrieval algorithm is released, for example with a new SCC version release (Single Calculus Chain, the official analysis tool of the network), products need to be re-analyzed.
It must be remarked that if the submitted product is a new version of an already uploaded product it will be accepted ONLY IF the submission is done in a specific time slot decided by the datacenter management group. Submission of new product versions will be NOT possible outside these specific time slots.

In version 3.0, the Quality Control procedures are carried out exclusively *on-fly* during the uploading process.

There are two types of Quality Control procedures :

- *Basic quality control (BQC)*: technical quality control on the submitted product.
- *Advanced quality control (AQC)*: series of physical checks applied to the submitted product.

If one or more basic quality control fails the submitted product is *rejected* and a detailed feedback is shown to the data originator.
If the product overcomes all the basic quality controls it is sent in batch to the advanced quality controls.
If the input product fails one or more advanced quality control the product is labeled as *Level 1* otherwise *Level 2*.

The following table reports quantities useful to describe the quality controls implemented.

Field	**Description**
β	aerosol backscatter coefficient
$\Delta\beta$	error on aerosol backscatter coefficient
$\beta_{peak}(\lambda)$	aerosol backscatter peak depending on wavelength
β_{dect}	aerosol backscatter detection limit
β_{th}	aerosol backscatter threshold value
α	aerosol extinction coefficient
$\Delta\alpha$	error on aerosol extinction coefficient
$\alpha_{peak}(\lambda)$	aerosol extinction peak depending on wavelength
α_{dect}	aerosol extinction minimum aerosol layer detection limit
α_{th}	aerosol extinction threshold value
S	lidar ratio
ΔS	error on lidar ratio
dh	aerosol layer height : defined as the lowest layer that generally contains most of the aerosol except special elevated layers (like Saharan dust etc.)
IB	Integrated aerosol backscatter
AOD	Aerosol optical depth

In the following all the controls are described and new features included for the Version 3.0 of the quality control are reported in red.

2- Basic Quality Controls

BQC-00

This procedure checks that each file contains the mandatory products.
If the submitted product is a backscatter (b-file), the NetCDF file must mandatory contain the variables : β and Δβ (*backscatter* and *error_backscatter*). Moreover, these variables must not be *NaN, NULL* or *negative defined* (they must contain at least 1 valid value).
If the submitted product is an extinction (e-file), the NetCDF file must mandatory contain the variables α and Δα (*extinction* and *error_extinction*). Moreover, these variables must not be *NaN, NULL* or *negative defined* (they must contain at least 1 valid value).
The variables declaration in the NetCDF data file, for Backscatter, Error Backscatter, Extinction, Error Extinction, are the following:

double backscatter (wavelength, time, altitude) ;
backscatter:ancillary_variables = "error_backscatter vertical_resolution" ;
backscatter:coordinates = "longitude latitude" ;
backscatter:long_name = "aerosol backscatter coefficient" ;
backscatter:plausibility = "parameter passed the EARLINET quality assurance." ;
backscatter:units = "m-1*sr-1" ;
backscatter:_FillValue = 9.96920996838687e+36 ;

double error_backscatter(wavelength, time, altitude) ;
error_backscatter:coordinates = "longitude latitude" ;
error_backscatter:long_name = "statistical uncertainty of aerosol backscatter" ;
error_backscatter:plausibility = "parameter passed the EARLINET quality assurance." ;
error_backscatter:units = "m-1*sr-1" ;
error_backscatter:_FillValue = 9.96920996838687e+36 ;

double extinction(wavelength, time, altitude) ;
extinction:ancillary_variables = "error_extinction vertical_resolution" ;
extinction:coordinates = "longitude latitude" ;
extinction:long_name = "aerosol extinction coefficient" ;
extinction:plausibility = "parameter passed the EARLINET quality assurance." ;
extinction:units = "m-1" ;
extinction:_FillValue = 9.96920996838687e+36 ;

double error_extinction(wavelength, time, altitude) ;
error_extinction:coordinates = "longitude latitude" ;
error_extinction:long_name = "statistical uncertainty of aerosol extinction" ;
error_extinction:plausibility = "parameter passed the EARLINET quality assurance." ;
error_extinction:units = "m-1" ;

error_extinction:_FillValue = 9.96920996838687e+36 ;

If BQC-00 fails the interface will generate the following types of errors :
“empty variable.”
“variable has all NaN elements.”
“whole defined Negative Variable.”
"Missing [backscatter] Variable."
"Missing [error_backscatter] Variable."
"Missing [extinction] Variable."
"Missing [error_extinction] Variable."

For example :
backscatter : empty variable.
backscatter : variable has all NaN elements.
backscatter : whole defined Negative Variable.

BQC-01

All *array variables* cannot be all undefined or negatives. This means that if the data file declares such a variable, this one cannot be empty or contain undefined or negative values.

If BQC-01 fails, the interface will generate the following types of errors :
“empty variable.”
“variable has all NaN elements.”
“whole defined Negative Variable.”
“value not allowed.”

e.g.:

cloud_mask : value not allowed. cloud_mask[2] = e

Or :

netcdf EARLINET_AerRemSen_pot_Lev02_e0355_201906131944_201906132129_v01_qc02 {
dimensions:
time = 1 ;
wavelength = 1 ;
altitude = 245 ;
nv = 2 ;
variables:
double time(time) ;
time:axis = "T" ;
time:bounds = "time_bounds" ;

time:calendar = "gregorian" ;
time:long_name = "time" ;
time:standard_name = "time" ;
time:units = "seconds since 1970-01-01T00:00:00Z" ;
…
double vertical_resolution(wavelength, time, altitude) ;
vertical_resolution:long_name = "Effective vertical resolution" ;
vertical_resolution:units = "m" ;
vertical_resolution:_FillValue = 9.96920996838687e+36 ;
…
// global attributes:
:Conventions = "CF-1.7" ;
…
data:
…
vertical_resolution = _, _, _, _, _, _,_, _, _,_, _, _ ;
//ERROR **vertical_resolution : variable has all NaN elements.**

…

}

BQC-02

If the MixingLayerHeight *mh* is present, the Aerosol Layer Height *dh* must also be present.
The variables declaration in the NetCDF data file, for MixingLayerHeight and Aerosol LayerHeight, are the following:

double aerosollayerheight(time) ;
aerosollayerheight:coordinates = "longitude latitude" ;
aerosollayerheight:long_name = "top of dust layer above sea level" ;
aerosollayerheight:plausibility = "parameter not quality assured by EARLINET." ;
aerosollayerheight:units = "m" ;
aerosollayerheight:_FillValue = 9.96920996838687e+36 ;
double mixinglayerheight(time) ;
mixinglayerheight:coordinates = "longitude latitude" ;
mixinglayerheight:long_name = "top of convective boundary layer above sea level" ;
mixinglayerheight:plausibility = "parameter not quality assured by EARLINET." ;
mixinglayerheight:units = "m" ;
mixinglayerheight:_FillValue = 9.96920996838687e+36 ;

If BQC-02 fails, the interface will generate the following types of errors :
"mixinglayerheight exists but aerosollayerheight is Missing."

This control is related to the definition of mixinglayerheight and aerosollayerheight (previously

named dust layer height) as reported at https://www.earlinet.org/index.php?id=125 .

BQC-03

The MixingLayerHeight, if present, must be lower than or equal to the Aerosol LayerHeight, that is the condition :

$$mh \leq dh$$

must always be preserved.

If BQC-03 fails, the interface will generate the following types of errors :
"mixinglayerheight higher then aerosollayerheight."

This control is related to the definition of *mixinglayerheight* and *aerosollayerheight* (previously named dust layer height) as reported at https://www.earlinet.org/index.php?id=125 .

BQC-04

The AerosolLayerHeight *dh* (if any) must be higher than the station altitude.

$$dh > station\_altitude$$

The MixingLayerHeight *mh* (if any) must be higher than the station altitude.

$$mh > station\_altitude$$

The variable *station_altitude* is declared in the NetCDF data file as follows:

float station_altitude ;
station_altitude:long_name = "station altitude above sea level" ;
station_altitude:units = "m" ; station_altitude:_FillValue = 9.96921e+36f ;

If BQC-04 fails, the interface will generate the following types of errors :
"aerosollayerheight is lower than station Altitude"
"mixinglayerheight is lower than station Altitude"

BQC-05

If the following variables are present in the data file a control must be done :

- If *volumedepolarization* is present, *error_volumedepolarization* has to be present as well

- If *particledepolarization* is present, *error_particledepolarization* has to be present as well
- If *watervapormixingratio* is present, *error_watervapor* has to be present as well

The variables declaration in the NetCDF data file for *volumedepolarization*,*error_volumedepolarization, particledepolarization*, *error_particledepolarization*,*watervapormixingratio*, *error_watervapor* are the following:

double volumedepolarization(wavelength, time, altitude) ;

volumedepolarization:ancillary_variables = "error_volumedepolarization" ;
volumedepolarization:coordinates = "longitude latitude" ;
volumedepolarization:long_name = "volume linear depolarization ratio" ;
volumedepolarization:plausibility = "parameter not quality assured by EARLINET." ;
volumedepolarization:units = "1" ;
volumedepolarization:_FillValue = 9.96920996838687e+36 ;

double error_volumedepolarization(wavelength, time, altitude) ;

error_volumedepolarization:coordinates = "longitude latitude" ;
error_volumedepolarization:long_name = "statistical uncertainty of volume linear depolarization ratio" ;
error_volumedepolarization:plausibility = "parameter not quality assured by EARLINET." ;
error_volumedepolarization:units = "1" ;
error_volumedepolarization:_FillValue = 9.96920996838687e+36 ;

double particledepolarization(wavelength, time, altitude) ;

particledepolarization:ancillary_variables = "error_particledepolarization" ;
particledepolarization:coordinates = "longitude latitude" ;
particledepolarization:long_name = "aerosol linear depolarization ratio" ;
particledepolarization:plausibility = "parameter not quality assured by EARLINET." ;
particledepolarization:units = "1" ;
particledepolarization:_FillValue = 9.96920996838687e+36 ;

double error_particledepolarization(wavelength, time, altitude) ;

error_particledepolarization:coordinates = "longitude latitude" ;
error_particledepolarization:long_name = "statistical uncertainty of aerosol linear depolarization ratio" ;
error_particledepolarization:plausibility = "parameter not quality assured by EARLINET." ;
error_particledepolarization:units = "1" ;
error_particledepolarization:_FillValue = 9.96920996838687e+36 ;

double error_watervapor(wavelength, time, altitude) ;

error_watervapor:standard_name = "humidity_mixing_ratiostandard_error" ;
error_watervapor:coordinates = "longitude latitude" ;

error_watervapor:long_name = "statistical uncertainty of the water vapor mixing ratio" ;
error_watervapor:plausibility = "parameter not quality assured by EARLINET." ;
error_watervapor:units = "g/kg" ;
error_watervapor:_FillValue = 9.96920996838687e+36 ;

If BQC-05 fails, the interface will generate the following types of errors :
"volumedepolarization exists but error_volumedepolarization is Missing."
"error_volumedepolarization exists but volumedepolarization is Missing."
"volumedepolarization and error_volumedepolarization have differnt size."

"particledepolarization exists but error_particledepolarization is Missing."
"error_particledepolarization exists but particledepolarization is Missing."
"particledepolarization and error_particledepolarization have differnt size."

"watervapormixingratio exists but error_watervapor is Missing."
"error_watervapor exists but watervapormixingratio is Missing."
"watervapormixingratio and error_watervapor have differnt size."

BQC-06

The following *BYTE* and *FLOAT* variables are *MANDATORY* for the products whose attribute *measurement_start_datetime* is greater than 2019-06-24, date time of the release of the new database, with these specifications and constraints :

byte atmospheric_molecular_calculation_source ALWAYS MANDATORY
byte error_retrieval_method(wavelength) ALWAYS MANDATORY

if Backscatter is present :
a) **byte backscatter_evaluation_method(wavelength)** IS MANDATORY
if **backscatter_evaluation_method = 0** then
the variable **byte raman_backscatter_algorithm(wavelength)** IS MANDATORY
else if **backscatter_evaluation_method = 1** then the variable
byte elastic_backscatter_algorithm(wavelength) IS MANDATORY

b) **byte backscatter_calibration_range_search_algorithm(wavelength)** IS MANDATORY
float backscatter_calibration_value(wavelength) IS MANDATORY
float backscatter_calibration_search_range(wavelength, nv) IS MANDATORY
float backscatter_calibration_range(wavelength, nv) IS MANDATORY

if Extinction is present :
byte extinction_evaluation_algorithm(wavelength) IS MANDATORY

Here follows declaration in the NetCDF data file for all the cited variables:

byte atmospheric_molecular_calculation_source ;

atmospheric_molecular_calculation_source:long_name = "data source of the atmospheric molecular calculations" ;

atmospheric_molecular_calculation_source:_FillValue = -127b ;

atmospheric_molecular_calculation_source:flag_values = 0b, 1b, 2b, 3b, 4b ;

atmospheric_molecular_calculation_source:flag_meanings = "US_standard_atmosphereradiosoundingecmwf icon-iglo-12-13 gdas" ;

byte error_retrieval_method(wavelength) ;

error_retrieval_method:long_name = "method used for the retrieval of uncertainties" ;

error_retrieval_method:_FillValue = -127b ;

error_retrieval_method:flag_values = 0b, 1b ;

error_retrieval_method:flag_meanings = "monte_carloerror_propagation" ;

byte backscatter_evaluation_method(wavelength) ;

backscatter_evaluation_method:long_name = "method used for the backscatter retrieval" ;

backscatter_evaluation_method:_FillValue = -127b ;

backscatter_evaluation_method:flag_values = 0b, 1b ;

backscatter_evaluation_method:flag_meanings = "Raman elastic_backscatter" ;

byte raman_backscatter_algorithm(wavelength) ;

raman_backscatter_algorithm:long_name = "algorithm used for the retrieval of the Raman backscatter profile" ;

raman_backscatter_algorithm:_FillValue = -127b ;

raman_backscatter_algorithm:flag_values = 0b, 1b ;

raman_backscatter_algorithm:flag_meanings = "Ansmannvia_backscatter_ratio" ;

byte elastic_backscatter_algorithm(wavelength) ;

elastic_backscatter_algorithm:long_name = "0: Klett-Fernald, 1: iterative" ;

elastic_backscatter_algorithm:_FillValue = -127b ;

elastic_backscatter_algorithm:flag_values = 0b, 1b ;

elastic_backscatter_algorithm:flag_meanings = "Klett-Fernald iterative" ;

byte backscatter_calibration_range_search_algorithm(wavelength) ;

backscatter_calibration_range_search_algorithm:long_name = "algorithm used for the search of the calibration_range" ;

backscatter_calibration_range_search_algorithm:_FillValue = -127b ;

backscatter_calibration_range_search_algorithm:flag_values = 0b, 1b ;

backscatter_calibration_range_search_algorithm:flag_meanings = "minimum_of_signal_ratiominimum_of_elastic_signal" ;

byte extinction_evaluation_algorithm(wavelength) ;

extinction_evaluation_algorithm:long_name = "algorithm used for the extinction retrieval";

extinction_evaluation_algorithm:_FillValue = -127b ;

extinction_evaluation_algorithm:flag_values = 0b, 1b ;

extinction_evaluation_algorithm:flag_meanings = "weighted_linear_fit non-weighted_linear_fit" ;

float backscatter_calibration_range(wavelength, nv) ;

backscatter_calibration_range:long_name = "altitude range where calibration was calculated" ;

backscatter_calibration_range:units = "m" ;

backscatter_calibration_range:_FillValue = 9.96921e+36f ;

float backscatter_calibration_search_range(wavelength, nv) ;

backscatter_calibration_search_range:long_name = "altitude range wherein calibration range is searched" ;

backscatter_calibration_search_range:units = "m" ;

backscatter_calibration_search_range:_FillValue = 9.96921e+36f ;

float backscatter_calibration_value(wavelength) ;

backscatter_calibration_value:long_name = "assumed backscatter-ratio value in calibration range" ;

backscatter_calibration_value:units = "m-1*sr-1" ;

backscatter_calibration_value:_FillValue = 9.96921e+36f ;

If BQC-06 fails, the interface will generate the following types of errors :

"Mandatory variable missing."

For example :

atmospheric_molecular_calculation_source : Mandatory variable missing.

BQC-07

For each declared BYTE variable, its *VALUES* are checked. Allowed values are retrieved from the variable *flag_values* attribute.

Here follows a list of all the declaration for the BYTE variables that can be contained in the NetCDF data file:

byte cirrus_contamination ;

cirrus_contamination:long_name = "do the profiles contain cirrus layers?" ;

cirrus_contamination:_FillValue = -127b ;

cirrus_contamination:valid_range = 0b, 3b ;

cirrus_contamination:flag_values = 0b, 1b, 2b ;

cirrus_contamination:flag_meanings = "not_available no_cirrus cirrus_detected" ;

byte cirrus_contamination_source ;

cirrus_contamination_source:long_name = "how was cirrus_contamination obtained?" ;
cirrus_contamination_source:_FillValue = -127b ;
cirrus_contamination_source:valid_range = 0b, 3b ;
cirrus_contamination_source:flag_values = 0b, 1b, 2b ;
cirrus_contamination_source:flag_meanings =
"not_availableuser_providedautomatic_calculated" ;

byte error_retrieval_method(wavelength) ;

error_retrieval_method:long_name = "method used for the retrieval of uncertainties" ;
error_retrieval_method:_FillValue = -127b ;
error_retrieval_method:flag_values = 0b, 1b ;
error_retrieval_method:flag_meanings = "monte_carloerror_propagation" ;

byte backscatter_evaluation_method(wavelength) ;

backscatter_evaluation_method:long_name = "method used for the backscatter retrieval" ;
backscatter_evaluation_method:_FillValue = -127b ;
backscatter_evaluation_method:flag_values = 0b, 1b ;
backscatter_evaluation_method:flag_meanings = "Raman elastic_backscatter" ;

byte elastic_backscatter_algorithm(wavelength) ;

elastic_backscatter_algorithm:long_name = "0: Klett-Fernald, 1: iterative" ;
elastic_backscatter_algorithm:_FillValue = -127b ;
elastic_backscatter_algorithm:flag_values = 0b, 1b ;
elastic_backscatter_algorithm:flag_meanings = "Klett-Fernald iterative" ;

byte backscatter_calibration_range_search_algorithm(wavelength) ;

backscatter_calibration_range_search_algorithm:long_name = "algorithm used for the
search of the calibration_range" ;
backscatter_calibration_range_search_algorithm:_FillValue = -127b ;
backscatter_calibration_range_search_algorithm:flag_values = 0b, 1b ;
backscatter_calibration_range_search_algorithm:flag_meanings =
"minimum_of_signal_ratiominimum_of_elastic_signal" ;

byte extinction_evaluation_algorithm(wavelength) ;

extinction_evaluation_algorithm:long_name = "algorithm used for the extinction retrieval" ;
extinction_evaluation_algorithm:_FillValue = -127b ;
extinction_evaluation_algorithm:flag_values = 0b, 1b ;
extinction_evaluation_algorithm:flag_meanings = "weighted_linear_fit non-

weighted_linear_fit" ;

byte raman_backscatter_algorithm(wavelength) ;

raman_backscatter_algorithm:long_name = "algorithm used for the retrieval of the Raman backscatter profile" ;

raman_backscatter_algorithm:_FillValue = -127b ;

raman_backscatter_algorithm:flag_values = 0b, 1b ;

raman_backscatter_algorithm:flag_meanings = "Ansmannvia_backscatter_ratio" ;

byte atmospheric_molecular_calculation_source ;

atmospheric_molecular_calculation_source:long_name = "data source of the atmospheric molecular calculations" ;

atmospheric_molecular_calculation_source:_FillValue = -127b ;

atmospheric_molecular_calculation_source:flag_values = 0b, 1b, 2b, 3b, 4b ;

atmospheric_molecular_calculation_source:flag_meanings = "US_standard_atmosphereradiosoundingecmwf icon-iglo-12-13 gdas" ;

byte cloud_mask(time, altitude) ;

cloud_mask:long_name = "cloud mask" ;

cloud_mask:_FillValue = -127b ;

cloud_mask:valid_range = 0b, 7b ;

cloud_mask:flag_masks = 1b, 2b, 4b ;

cloud_mask:flag_meanings = "unknown_cloudcirrus_cloudwater_cloud" ;

byte cloud_mask_type(time, altitude) ;

cloud_mask_type:long_name = "cloud mask type" ;

cloud_mask_type:_FillValue = -127b ;

cloud_mask_type:valid_range = 0b, 7b ;

cloud_mask_type:flag_masks = 1b, 2b, 4b ;

cloud_mask_type:flag_meanings = "no_cloudmask_availablemanual_cloudmaskautomatic_cloudmask" ;

If BQC-07 fails, the interface will generate the following types of errors :
“value not allowed.”

E.g.:
cirrus_contamination : value not allowed. cirrus_contamination = 8

BQC-08

The following *global attributes* are *MANDATORY* for the products whose attribute *measurement_start_datetime* is greater than 2019-06-24, date time of the release of the new database:

Name	Type
processor_name	string
PI	string
PI_affiliation	string
PI_email	string
Data_Originator	string
Data_Originator_affiliation	string
Data_Originator_email	string
hoi_system_ID	integer
hoi_configuration_ID	integer

Name	Type
Conventions	string
title	string
source	string
references	string
history	string
station_ID	string
location	string
system	string
institution	string
comment	string
measurement_start_datetime	string
measurement_stop_datetime	string

If BQC-08 fails the interface will generate the following types of errors :
“Mandatory global attribute missing.”

E.g.:
measurement_start_datetime : Mandatory global attribute missing.

BQC-09 [NEW]

Validation of *measurement_start_datetime* and *measurement_stop_datetime* global attributes. A control is performed in order to check if *measurement_start_datetime* and *measurement_stop_datetime* global attributes represent a valid date time and they are also

compared with the actual date time in order to avoid inconsistencies (e.g. a date time in the future). Moreover, a control is made on the variable *time* in order to ensure the consistency. The *time* variable cannot have values less than 1997-12-01 (which represents approximately the date on which the first file was uploaded to the network database) and greater than the actual date of upload.

The *time* variable in the NetCDF data file is declared as follows :

double time(time) ;
time:axis = "T" ;
time:bounds = "time_bounds" ;
time:calendar = "gregorian" ;
time:long_name = "time" ;
time:standard_name = "time" ;
time:units = "seconds since 1970-01-01T00:00:00Z" ;

If BQC-09 fails the interface will generate the following types of errors :
“Variable [time] value is NOT valid.”
"Global attribute [measurement_start_datetime] is NOT valid."
"Global attribute [measurement_stop_datetime] is NOT valid."
"[measurement_start_datetime] is greater than the [measurement_stop_datetime]"
"[measurement_start_datetime] is equal to [measurement_stop_datetime]"

For example:
netcdf EARLINET_AerRemSen_the_Lev01_b0355_201902181239_201902181310_v01_qc02 {
dimensions:
time = 1 ;
wavelength = 1 ;
altitude = 658 ;
nv = 2 ;

variables:
double time(time) ;
time:axis = "T" ;
time:bounds = "time_bounds" ;
time:calendar = "gregorian" ;
time:long_name = "time" ;
time:standard_name = "time" ;
time:units = "seconds since 1970-01-01T00:00:00Z" ;
double time_bounds(time, nv) ;

…

data:

time = 1 ;
//ERROR **Variable [time] value is NOT valid. : time[0] = 1 Value is less than 1997-12-01**

time_bounds =
1, _ ;

BQC-10

If the NetCDF data file contains the variable *__SkippedFraction* a control is performed in order to check if its value is between the range [0 , 1]

If BQC-10 fails, the interface will generate the following types of errors :
“SkippedFraction has a wrong value.”

BQC-11

This control is performed in order to check if the station coordinates (*latitude, longitude, altitude*) are correct within the errors :
latitude ±0.05°
longitude± 0.05°
altitude± 60*m*

The station coordinates (*latitude, longitude, altitude*) variables in a NetCDF file are declared as follows:

float longitude ;
longitude:long_name = "longitude of station";
longitude:standard_name = "longitude";
longitude:units = "degrees_east";

float latitude ;
latitude:long_name = "latitude of station";
latitude:standard_name = "latitude";
latitude:units = "degrees_north";

float station_altitude ;
station_altitude:long_name = "station altitude above see level";
station_altitude:units = "m";
station_altitude:_FillValue = 9.96921e+36f;

If BQC-11 fails, the interface will generate the following types of errors :
“Location [Latitude] is Wrong.”

“Location [Longitude] is Wrong.”
“Location [Altitude] is Wrong.”

BQC-12 [NEW]

This control checks that the *altitude* variable is not less than 0*m asl* and is not greater than a threshold value of *50km* (top of troposphere).

The variable *altitude* in a NetCDF data file is declared as follows:
double altitude (altitude);

altitude:axis = "Z";
altitude:long_name = "height above sea level";
altitude:positive = "up";
altitude:standard_name = "altitude";
altitude:units = "m";

If BQC-12 fails, the interface will generate the following types of errors :
“Altitude value out of limits”

For example, a possible output can be :
Altitude value out of limits : altitude[6] = -60
Altitude value out of limits : altitude[226] = 9.96921e+36

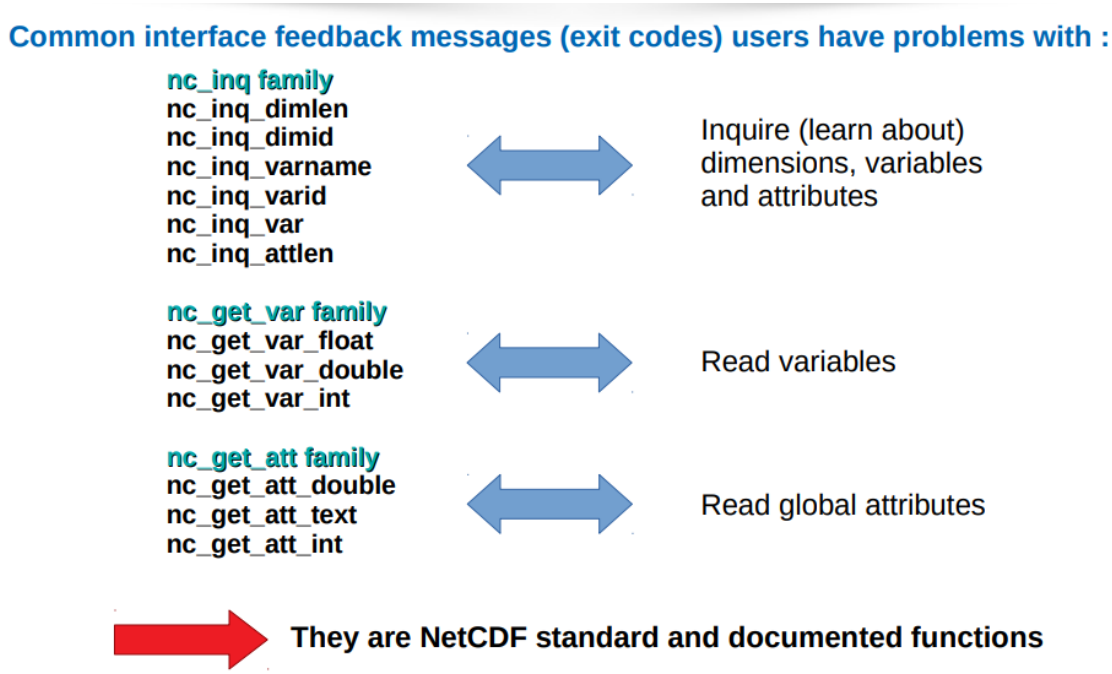
FURTHER TECHNICAL CONSIDERATIONS

Since every product in the network is a NetCDF file, when a product is uploaded its integrity is verified. This means that a data file must be compliant with the NetCDF data format.
(https://www.unidata.ucar.edu/software/netcdf/docs/index.html)
If a product is not compliant with the NetCDF data format, the interface can generate a series of errors all related to data format. The explenation of each of these errors can be found at:

https://www.unidata.ucar.edu/software/netcdf/docs/modules.html

The following types of errors, reported by the upload interface, are the most common faced by the users and they are all well documented in the official website above cited.
"nc_open File Failed. Likely, the file you submitted is not a NetCDF file."
"nc_create File Failed."
"nc_inq Failed."

"nc_inq_dimlen Failed."
"nc_inq_dimid Failed."
"nc_inq_varname Failed."
"nc_inq_varid Failed."
"nc_inq_var Failed."
"nc_get_var_float Failed."
"nc_get_var_double Failed."
"nc_get_var_int Failed."
"nc_inq_attlen Failed."
"nc_get_att_double Failed."
"nc_get_att_text Failed."
"nc_get_att_int Failed."
"nc_def_dim Failed."
"nc_def_var Failed."
"nc_enddef Failed."
"nc_put_var_float Failed."
"nc_put_var_int Failed."
"nc_put_att_text Failed."
"nc_put_att_float Failed."
"nc_put_att_int Failed."



3- Advanced Quality Controls

AQC-00

Profile scanning and checking that error on the optical properties is positive for all defined values of the corresponding optical property.
A product file does not pass this control if, for defined optical property value, at least for 1 point, the error is negative, zero or not defined.

Depending on the kind of product, this control is applied to the couples of variables:

backscatter - error_backscatter

extinction - error_extinction

If AQC-00 fails the interface will generate the following types of errors :
"error_backscatter variable is not positive for all defined value of the backscatter"
"error_extinction variable is not positive for all defined value of the extinction"

AQC-01

This control checks for negative peaks. It performs a profile scanning and checks that the aerosol optical properties are positive within 3σ, that there are not negative peaks and that very extreme values are present only in cirrus cloud cases.

Checks on backscatter if backscatter is negative:

$\beta+\beta_{th} \geq 0$

OR $|\beta| < 3\,\Delta\beta$

$\beta < \beta_{peak}(\lambda)$ if it does not belong to cirrus category

Checks on extinction:

$\alpha+\alpha_{th} \geq 0$

OR $|\alpha| < 3\,\Delta\alpha$

$\alpha < \alpha_{peak}(\lambda)$ if it does not belong to cirrus category

For a first screening of unrealistic negative peak value, a value representative of an aerosol layer is used. In particular $\beta_{th} = 5 \cdot 10\text{-}7$ *m-1 sr-1* and $\alpha_{th} = 2.5 \cdot 10\text{-}5$ *m-1*.
These threshold values are set as 50 times the calibration value for the 1064*nm*.
Wavelength and *IB* are scaled in the most conservative way, and threshold is kept constant over the different wavelengths.
Peak values had been set up studying the *pdf* of optical properties values for cirrus cases:
even for cirrus cases the probability of having $\alpha > 0.005$ *m-1* is less than 5‰. Correspondingly a $\beta_{peak}=1.7 \cdot 10\text{-}$

4 *m-1 sr-1* had set up assuming a lidar ratio of 30 *sr* (most conservative value)
has been assumed.

If AQC-01 fails the interface will generate, for example, error(s) like the following :

OVER PEAK : bck = 0.000237872 err_bck = 1.17592e-05
OVER PEAK : bck = 0.00495033 err_bck = 14.322
OVER PEAK : ext = 0.01091 err_ext = 0.00215
OVER PEAK : ext = 0.00737 err_ext = 0.00247

bck = -7.35e-07 err_bck = 1.15e-07 - [over 3*Sigma OR over threshold]
bck = -6.14e-07 err_bck = 1.43e-07 - [over 3*Sigma OR over threshold]

where *bck* and *err_bck* stand for *backscatter* and *error_backscatter,* and *ext* and *err_ext* stand for *extinction* and *error_ extinction.*

AQC-02

Integrated quantities of the aerosol properties should be positive and not exceed very extreme values not realistic for aerosol layer cases. Such very high values could occur even in very specific and intense aerosol layers, however such cases should be treated in a careful way dealing with multiple scattering. For these reasons, cases corresponding to very extreme integrated values are not labelled as Level 2 data highlighting to data users that care should be taken handling such data.
The *aerosol optical depth AOD* (without any assumption in the lowest troposphere, so evaluated only in the portion of atmosphere covered by the provided profile) should remain under an established threshold for data not belonging to the cirrus category.

$AOD > 0$

$AOD < AOD_{th}$ if it does not belong to cirrus category

The threshold values AOD_{th} = 1.5 had been set up studying the *pdf* of AOD for cirrus cases available on the EARLINET database until July 2016: even for cirrus cases the probability of having AOD > 1.5 is less than 5‰. The scaling with the wavelength had been set up in the most conservative way: considering that on average over Europe AERONET report 1.1-1.5 as typical Angstrom exponent, the threshold value was scaled with wavelength considering an Angstrom value of 0, so no wavelength dependence at all.

If AQC-02 fails, the interface will generate, for example, error(s) like the following :

AOD greater than Threshold value : 75.0141
AOD greater than Threshold value : 7.00155e+39

AOD NEGATIVE : -3.3e+08

AOD UNDEFINED

AQC-03

As for *AOD*, integrated quantities of the aerosol backscatter should be positive and not exceed very extreme values not realistic for aerosol layer cases. Such very high values could occur even in very specific and intense aerosol layers, however such cases should be treated in a careful way dealing with multiple scattering. For these reasons, cases corresponding to very extreme integrated values are not labelled as Level 2 data highlighting to data users that care should be taken handling such data.
The aerosol *integrated backscatter IB* (without any assumption in the lowest troposphere, so evaluated only in the portion of atmosphere covered by the provided profile) should remain under an established threshold for data not belonging to the cirrus category.

$IB > 0$

$IB < IB_{th}$ if it does not belong to cirrus category

Starting from the AOD_{th} discussed above, the IB_{th}= 0.05 sr-1 is defined assuming a low (30*sr*) lidar ratio value (most conservative choice) for converting extinction into backscatter and an Angstrom of 0 for scaling with the wavelength.

If AQC-03 fails the interface will generate, for example, error(s) like the following:

IB greater than Threshold value : 1.3629e+250

IB NEGATIVE : -0.0285286
IB NEGATIVE : -0.0278059

IB UNDEFINED

AQC-04

This control performs a check on the *Lidar Ratio* values when Extinction and Backscatter are provided in the same product file (i.e. are provided at the same vertical resolution).
Lidarratio is defined as positive value and values are typically between 10 and 120 *sr*.
A wider window is defined for excluding not realistic values without constraining the *pdf* of the *lidarratio* variable values.

$S \in [0, 200]$ *sr* within 3 ΔS

However, lidar ratio is an intensive property so it is defined only where aerosols are significantly present. Therefore the limit on *S* should be valid only where an aerosol layer is present, so where extinction and/or backscatter exceed a certain value that can be treated as a minimum aerosol layer detection limit and with a limited statistical uncertainty.

In particular, the following check is performed:

If $\alpha > \alpha_{dect}(\lambda)$

AND ($\Delta\alpha/\alpha$) < 50%
AND If $\beta > \beta_{dect}(\lambda)$

AND ($\Delta\beta / \beta$) < 50%

S must be between [0, 200]*sr* within 3 Δ*S* with:

β_{dect} = 5 ·10-7m-1sr-1 and

α_{dect} = 2.5 ·10-5m-1

where the uncertainty on *extinction* and *backscatter* is lower than 50%

Lidar ratio and its error variables are declared in the NetCDF data file as follows:

double lidarratio(wavelength, time, altitude);
lidarratio:ancillary_variables = "error_lidarratio";
lidarratio:coordinates = "longitude latitude";
lidarratio:long_name = "aerosol extinction-to-backscatter ratio";
lidarratio:plausibility = "parameter calculated from backscatter and extinction.";
lidarratio:units = "sr" ;
lidarratio:_FillValue = 9.96920996838687e+36;

double error_lidarratio(wavelength, time, altitude);
error_lidarratio:coordinates = "longitude latitude";
error_lidarratio:long_name = "statistical uncertainty of lidar ratio";
error_lidarratio:plausibility = "parameter calculated from error_backscatteranderror_extinction.";
error_lidarratio:units = "sr";
error_lidarratio:_FillValue = 9.96920996838687e+36;

If AQC-04 fails, the interface will generate the following types of errors :
"Lidar Ratio value NOT allowable"
"Lidar Ratio + (3*errLR) is Negative"

AQC-05

This control is performed on the *volume depolarization* variable and its error (*error_volumedepolarization*) in order to check that the following conditions are preserved :

|*volumedepolarization*|< 3 *error_volumedepolarization*
volumedepolarization must be between [0, 1] within its error

If AQC-05 fails, the interface will generate, for example, error(s) like the following :

volumedepolarization = 1.19425 error_volumedepolarization = 0.0472857 - [over 3*Sigma OR over threshold]

volumedepolarization = 1.25597 error_volumedepolarization = 0.040426 - [over 3*Sigma OR over threshold]

AQC-06

This control is performed on the *particle depolarization* variable and its error (*error_particledepolarization*) in order to check that the following conditions are preserved :

|*particledepolarization*|< 3 *error_particledepolarization*
particledepolarization must be between [0, 1] within its error

If AQC-06 fails, the interface will generate, for example, error(s) like the following :

particledepolarization = -8.0734 error_particledepolarization = 1.91339 - [over 3*Sigma OR over threshold]
particledepolarization = -7.47232 error_particledepolarization = 1.34767 - [over 3*Sigma OR over threshold]

AQC-07

This control is performed on the *water vapor mixing ratio* variable and its error (*error_watervapor*) in order to check that the following conditions are preserved :

|*watervapormixingratio*|< 3 *error_watervapor*
watervapormixingratio must be between [0, 100 *g/Kg*] within its error

If AQC-07 fails, the interface will generate errors similar to the ones reported for AQC-05 an AQC-06.

AQC-08 [New]

The use of standard atmosphere profiles is source of potentially high error in the optical property profiles. Therefore data products obtained using such molecular profiles are not considered as not high quality and will eb labelled as Level 1 data product.

This control is performed on the variable *atmospheric_molecular_calculation_source*. If its value is 0 (*zero*) the product is labelled as *Level 1.* This control will be applied on products whose start time is higher than *202103250000***.**

The variable *atmospheric_molecular_calculation_source* in a NetCDF data file is declared as follows:

byte atmospheric_molecular_calculation_source;
atmospheric_molecular_calculation_source:long_name = "data source of the atmospheric molecular calculations"; atmospheric_molecular_calculation_source:_FillValue = -127b;
atmospheric_molecular_calculation_source:flag_values = 0b, 1b, 2b, 3b, 4b;
atmospheric_molecular_calculation_source:flag_meanings = "US_standard_atmosphere radiosounding ecmwf icon-iglo-12-13 gdas";

AQC-09 [New]

Data products uploaded before the new release of the database (2019-06-24), labeled as cirrus, and not reporting the cloud mask as vertical information are labeled as *Level 1*. This control is implemented as a message to data users of handling with care such data, because a cirrus cloud is present but the location in the vertical dimension of such cloud is not precisely reported into the data product.

If AQC-09 fails, the interface will generate the following types of errors :
"Product is labelled as cirrus but cloud_mask variable is missing"

AQC-10 [New]

Only data products obtained using the approved configuration into the Single Calculus Chain (i.e. Operational product) can be fully quality controlled products. Experimental products are still useful and interesting products btu the data users should be aware that they are not fully quality controlled in the workflow from preforming measurements till the optical data product provision.
This control is performed on the variable *scc_product_type.* It will be applied on products whose start time is higher than *202103250000***.**

If :
scc_product_type = 1 it is an *experimental product* and the product is labelled as *Level 1*
scc_product_type = 2 it is an *operational product* and the product is labelled as *Level 2*

The control will be performed from a certain datetime on.

The variable *scc_product_type* in a NetCDF data file is declared as follows:

byte scc_product_type;
scc_product_type:_FillValue = -127b ;
scc_product_type:long_name = "SCC product type" ;
scc_product_type:valid_range = 1b, 2b ;
scc_product_type:flag_values = 1b, 2b ;
scc_product_type:flag_meanings = "experimental operational" ;

If AQC-10 fails, the interface will generate the following types of errors :
"scc_product_type = 1 the product is experimental"
In case not allowed value is reported (for example x) the following will appear
"scc_product_type = x value not allowed"

Variables added by the Database Processor

For the sake of completeness, let's remark that after all the Quality Control have been applied to a product data file the Database Processor adds a series of variables to the uploaded product. These variables keep track of quality control level, the Basic and Advanced quality controls the product has passed.

Here are listed the declarations of the *integer* variables added by the system at the end of the uploading phase:

int quality_control_level;

quality_control_level:long_name = "Quality Control Level";
quality_control_level:flag_values = 0, 1, 2;
quality_control_level:flag_meanings =
"File_does_not_overcome_one_or_more_on_fly_quality_control
File_does_overcome_all_on_fly_quality_control_but_fails_one_or_more_technical_quality_control
File_does_overcome_all_technical_quality_control_and_physical_quality_control ";
quality_control_level:version = "3.0";
quality_control_level:references = " https://www.earlinet.org/index.php?id=293 ";

int basic_quality_control;

basic_quality_control:long_name = "Basic Quality Control";
basic_quality_control:valid_range = 0, 7;
basic_quality_control:flag_masks = 1, 2, 4;
basic_quality_control:flag_meanings = "Check_if_file_contains_data
Check_for_Undefined_Variables_and_Global_Attributes Check_Coordinates_Consistency" ;
basic_quality_control:references = " https://www.earlinet.org/index.php?id=293 ";

int advanced_quality_control;

advanced_quality_control:long_name = "Advanced Quality Control" ;
advanced_quality_control:valid_range = 0, 2039 ;
advanced_quality_control:flag_masks = 1, 2, 4, 16, 32, 64, 128, 256, 512, 1024 ;
advanced_quality_control:flag_meanings = "Checks_for_Negative_Errors Negative_peaks
Check_on_AOD Check_on_LidarRatio Check_on_Volumedepolarization Check_on_Particledepolarization
Check_on_Watervapormixingratio Check_on_atmospheric_molecular_calculation_source
Check_on_old_cirrus_product Check_on_SCC_product_type" ;
advanced_quality_control:references = "https://www.earlinet.org/index.php?id=293" ;