Customer	**:**	**ESRIN**	**Document Ref**	**:**	**BAQ-TEC-TEN-SER-025**
Contract No	**:**	**4000126749/19/I-NS**	**Issue Date**	**:**	**19 March 2020**
WP No	**:**	**4100**	**Issue**	**:**	**1.0**

Lidar analysis chain files description

Abstract : Technical note describing the files produced by the LIDAR acquisition and analysis system

Author : Gabriele Mevi
Serco

Approval : Anna Maria Iannarelli
Serco

Distribution : ESA/ESRIN EOP-GMQ
BAQUNIN Leadership Team

Serco Italia SpA
Sede Operativa: Via Sciadonna, 24-26 - Frascati (Roma)
Tel: +39 06 98354400 Fax:
www.serco.com

Issue 1.0

CHANGE HISTORY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

ISSUE	DATE	REASON
1.0	19 Mar 2020	First version

Issue 1.0

INTRODUCTION

This document is a description of the files produced by the BAQUNIN project LIDAR. The LIDAR system is situated at the Atmospheric Physics Laboratory (APL), University Sapienza of Rome; its acquisition and analysis system has been developed by the BAQUNIN team and produces several types of files.

The LIDAR analysis software (LAS) is described in the BAQ-MGT-TEM-SER-018 technical note; the reader should refer to this for a detailed description of LAS methods and algorithms.

APPLICABLE DOCUMENTS

The following is a list of documents with a direct bearing on the content of this report. Where referenced in the text, these are identified as RD.n, where 'n' is the number in the list below:

[RD.1] BAQ-MGT-TEM-SER-018, latest applicable issue

ACRONYMS

Acronym	Definition
APL	Atmospheric Physics Laboratory (at Sapienza)
BAQUNIN	Boundary-layer Air Quality-analysis Using Network of INstruments
EVDC	ESA Atmospheric Validation Data Centre
LAS	LIDAR Analysis Software
ACS	Acquisition Software

Issue 1.0

1. OVERVIEW

The analysis chain for LIDAR measurements is composed by six steps, corresponding to different operations on the acquired data. Each one of these operations produces a specific file that is used as input file for the next analysis step. The operations and associated files are described by the scheme in Figure 1.

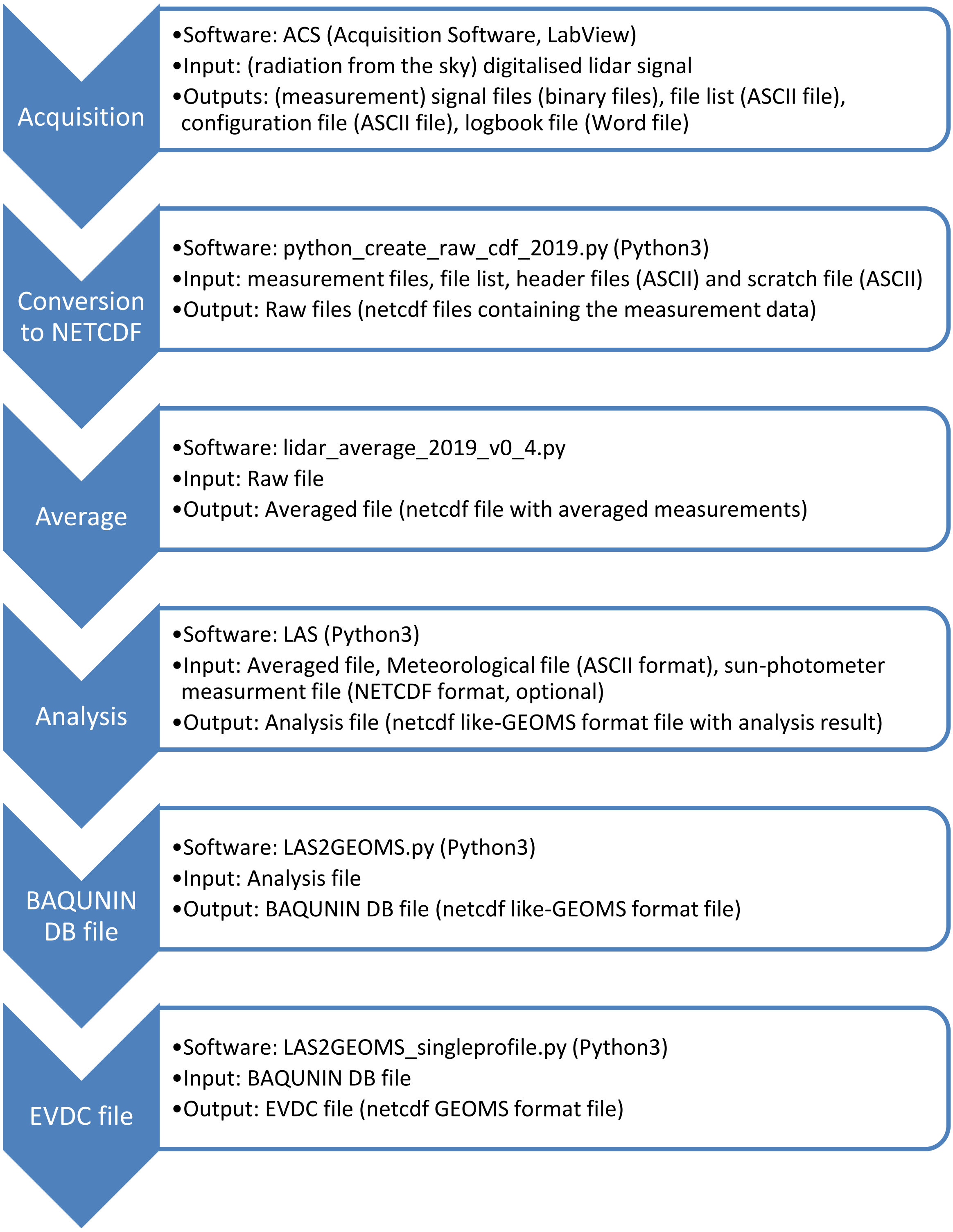

Figure 1: scheme of the analysis chain operations.

Issue 1.0

The scheme shows the passages used to produce the files that the BAQUNIN team disseminates on the BAQUNIN site (www.baqunin.eu) and on the EVDC portal (https://evdc.esa.int/). The codes employed are written in LabView for the ACQ system, (https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/labview_documentation_resources/), and Python3 languages for the other software (https://docs.python.org/3/reference). In the following paragraphs the files produced by each step of this process will be described in detail.

The files produced by the first four steps are not public available but are stored in the BAQUNIN database. The BAQUNIN DB file and EVDC file are disseminated on the BAQUNIN website and EVDC web-archive respectively. The main difference between BAQUNIN DB and EVDC files is that the first ones contain the results from the entire measurement session, while the EVDC files contain results from a single profile.

The Python3 library used for read and write netcdf files is NetCDF4 (documentation at https://unidata.github.io/netcdf4-python/netCDF4); the format used for file production is NETCDF3-CLASSIC.

Issue 1.0

2. ACQUISITION

The lidar signals are acquired by a LICEL Transient recorder system (www.licel.com) and elaborated by the ACS that produces one binary file for each **measurement**.

In this document a measurement is defined as the set of signals collected during a single integration time, a quantity that can be selected on the ACS control panel. The typical integration time is 10 seconds.

The Transient recorder is composed by six acquisition modules, referred as “licel” in this document; each licel can receive an analogic and a digital signal simultaneously, resulting in a total of 12 acquisition **channels**. The analogic channels measure the output current produced by the LIDAR phototubes while the digital channels the number of photons received. The channels are identified by a progressive number from 0 to 11, while the licels by a progressive number from 0 to 5: channel 0 is the analogic channel of the licel 0, channel 1 the digital channel of licel 0, channel 2 the analogic channel of licel 1 and so on. Analogic channels are marked by even numbers, while digital channels by odds.

The binary file contains the signal profiles acquired by all the 12 acquisition channels of the Transient recorder system. Note that the binary file does **not contain** any information regarding the wavelength of the radiation collected by each channel; this information is stored in the LIDAR **logbook file.** The logbook file is a Word file produced by the LIDAR operator describing the radiation measured by each channel and other ancillary information regarding a specific measurement session, such as the start and stop time or annotation on meteorological conditions.

The binary file name is composed by the date and time of the measurement in the following format: *yydddHHMM.0SS.* The first two characters represent the year (abbreviated form), followed by the day number (example: 1st Feb day number is 32), hour (24h format), minutes, the string “.0” and the seconds.

The ACS produces also a **configuration file,** an ASCII file that reports all the settings used by the ACS for that measurement session. The configuration file name is in the format *yydddHHMM.txt.config.log*. The information stored in this file is not used in the analysis chain.

Another output of the ACS is the **file list**. The file list is an ASCII file containing the list of the binary files created during a measurement session, named *yydddHHMM.txt*.

Issue 1.0

3. NETCDF CONVERSION

The binary files are converted in netcdf files by the Python3 script **python_create_raw_cdf_2019.py,** that reads the binary files recorded in the file list created by the acquisition software.

In order to correctly execute the conversion, the script needs as inputs the **header files,** ASCII files containing the information about the configurations and properties of the different signals that can be received by the LIDAR. An example of header file is shown in Box 3.1. The comments are preceded by the character ";", because the header files were originally designed for an IDL script; they are marked in a different color for clarity.

Box 3.1: example of header file

```
UNIROMA1 LIDAR                  ;System
Rome-Italy                      ;Location
41.9                            ;Longitude[°E]
12.5                            ;Latitude[°N]
75.                             ;Altitude[mASL]
Rayleigh-Raman Monostatic LIDAR ;System Description
Quanta Ray                      ;Laser
30                              ;Repetition Rate [Hz]
NA                              ;Energy per pulse at 1064 nm [mJ]
1064-532-355                    ;Emission wavelength[nm]
1064                            ;Detected wavelength[nm]
0                               ;Zenith angle[°]
LICEL-TR20-160                  ;Acquisition system
1064 analog                     ;channel description
100mm Cassegrain                ;Receiver
0.3 nm BW filters               ;filters
TEST 2018                       ;Comments
```

The headers' file names are in the format *cdf_header_rome_[signal name].txt;* the **signal name** is a string that identifies a specific signal. Table 1 defines the signal names used in the major part of BAQUNIN measurements.

Table 1: signal names

Signal name	Notes
1064t	1064 nm signal, analogic, total polarization
532Hitan	532 nm signal, analogic, total polarization, measured with high range optical receiver
532Hipan	532 nm signal, analogic, parallel polarization, measured with high range optical receiver
532tan	532 nm signal, analogic, total polarization, measured with low range optical receiver
532pan	532 nm signal, analogic, parallel polarization, measured with low range optical receiver
355an	355 nm signal, analogic, total polarization
355ph	355 nm signal, digital, total polarization.
N2d	386 nm N2 RAMAN signal, digital, total polarization
N2a	386 nm N2 RAMAN signal, analogic, total polarization
H2Od	407 nm H2O RAMAN signal, digital, total polarization
H2Oa	407 nm H2O RAMAN signal, analogic, total polarization
ch[num]	Channel not used in the acquisition

The script needs as inputs the **scratch file**: this is an ASCII file containing the input and output files folders and the information on the type of signal collected by the LICEL channels. Below an example of scratch file content is shown.

Box 3.2: example of scratch file

```
;the line below contains the input data path
C:/…
;the line below contains the header files path and first part of their names
C:/..
;the line below contains the output path and the initial part of the output file name
C:/..
6                    ;number of licel used (each licel is composed by two channels, analogic and digital)
12                   ;channels number
1064t,N2d,532Hitan,355ph,532Hipan,ch05,532tan,ch07,532pan,ch09,H2Oa,H2Od,        ;signal associated
;line below contains the delay file complete path and name
C:/.. /delay_roma_201906.txt
```

An important part of the scratch file is the ninth line that relates the channels with their signal: the first element of the line is the signal measured by the channel 0, the second element is the signal measured by channel 1 and so on. Channels not used in the measurement session are marked by the string “ch” followed by their number. The correct associations channel/signal can be obtained from the logbook file.

The last line of the file is the **delay file** name and path. The front end electronic can introduce a delay between the production of the laser pulse and the start of the signal acquisition by the licel. This can lead to an incorrect association between the lidar profile bins and their altitude. The delay file is an ASCII file containing the number of bins the signals must been shifted in order to be synchronized with the pulse emission. An example of delay file is reported in Box 3.3.

Box 3.3: the delay file

```
; list of licel trigger delays observed during the delay test of June 2019
1                    ;channel0 (licel0, analog)
-1                   ;channel1 (licel0 photon)
1                    ;channel2 (licel1, analog)
-1                   ;channel3 (licel1 photon)
1                    ;channel4 (licel2, analog)
-1                   ;channel5 (licel2 photon)
7                    ;channel6 (licel3, analog)
-1                   ;channel7 (licel3 photon)
7                    ;channel8 (licel4, analog)
-1                   ;channel9 (licel4 photon)
8                    ;channel10 (licel5, analog)
-1                   ;channel11 (licel5 photon)
```

The conversion output is a netcdf file, defined **raw file,** for each signal described in the ninth line of the scratch file. The raw files contain all the information of the header files as global attributes and three variables: “*ch*”, “*nsht*” and “*time*”.

The signal is stored in the variable *ch,* a matrix *npnt* x *nrec*, where *npnt* is the number of altitude bins composing each profile, equal to 3000 in a standard acquisition, and *nrec* is the number of profiles acquired. A **profile** is produced by the sum of the signals acquired during the integration time by a channel. The variable attributes contain also information about the wavelength measured, the vertical resolution, the channel that collects the signal and the polarization.

Issue 1.0

The variable *nsht* is a vector of length equal to *nrec* which contains the information about the number of lidar profiles accumulated during the integration time for each measurement.

The variable *time* contains the information about the date and time of the profiles in MJD2K (modified julian day, number of days elapsed since 01/01/2000 at 00:00 plus fraction of day).

The raw file names is in the format *rome_raw_[signal name]_[date time of start measurement].nc*, with the date and time in format yyyymmddHHMMSS; a short summary describing a raw file is in

```
netcdf file {
 dimensions:
  npnt = 3000;
  nrec = UNLIMITED;   // (2002 currently)
 variables:
  double time(nrec=2002);
   :LongName = "Time";
   :Units = "MJD2K";
   :NOTES = "universal time in days since 2000-01-01";

  int nsht(nrec=2002);
   :LongName = "LaserShots";
   :Units = " ";

  float ch(nrec=2002, npnt=3000);
   :LongName = "RawSignal";
   :Units = "a.u.";
   :VPMT_V = 400; // int
   :Wavelength_nm = 1064; // int
   :Polarization = 0; // int
   :Vertical_resolution_m = 7.5; // double
   :bin_number = 3000; // int
   :Range_Discriminator = 0; // int
   :Licel_channel = 0; // int
   :Trigger_delay = 1.0; // double

 // global attributes:
 :System = "UNIROMA1 LIDAR          ";
 :Location = "Rome-Italy              ";
 :Longitude = "41.9                   ";
 :Latitude = "12.5                    ";
 :Altitude = "75.                     ";
 :SystemDescription = "Rayleigh-Raman Monostatic LIDAR       ";
 :Laser = "Quanta Ray          ";
 :RepetitionRate = "30                      ";
 :Energyperpulse = "NA                      ";
 :Emissionwavelength = "1064-532-355                   ";
 :Detectedwavelength = "1064\t\t             ";
 :Zenithangle = "0                       ";
 :Acquisitionsystem = "LICEL-TR20-160                ";
 :chdescription = "1064 analog    ";
 :telescope = "100mm Cassegrain                  ";
 :filter = "0.3 nm BW filters             ";
 :Comments = "TEST 2018     }
```

.

Box 3.4: summary of a raw file

```
netcdf file {
 dimensions:
  npnt = 3000;
  nrec = UNLIMITED;   // (2002 currently)
```

Issue 1.0

```
variables:
  double time(nrec=2002);
    :LongName = "Time";
    :Units = "MJD2K";
    :NOTES = "universal time in days since 2000-01-01";

  int nsht(nrec=2002);
    :LongName = "LaserShots";
    :Units = " ";

  float ch(nrec=2002, npnt=3000);
    :LongName = "RawSignal";
    :Units = "a.u.";
    :VPMT_V = 400; // int
    :Wavelength_nm = 1064; // int
    :Polarization = 0; // int
    :Vertical_resolution_m = 7.5; // double
    :bin_number = 3000; // int
    :Range_Discriminator = 0; // int
    :Licel_channel = 0; // int
    :Trigger_delay = 1.0; // double

// global attributes:
:System = "UNIROMA1 LIDAR         ";
:Location = "Rome-Italy           ";
:Longitude = "41.9                ";
:Latitude = "12.5                 ";
:Altitude = "75.                  ";
:SystemDescription = "Rayleigh-Raman Monostatic LIDAR      ";
:Laser = "Quanta Ray          ";
:RepetitionRate = "30                 ";
:Energyperpulse = "NA                 ";
:Emissionwavelength = "1064-532-355               ";
:Detectedwavelength = "1064\t\t           ";
:Zenithangle = "0                     ";
:Acquisitionsystem = "LICEL-TR20-160              ";
:chdescription = "1064 analog    ";
:telescope = "100mm Cassegrain                ";
:filter = "0.3 nm BW filters            ";
:Comments = "TEST 2018     }
```

Issue 1.0

4. AVERAGE DATA

The raw file data needs to be averaged in order to estimate the signal noise and to obtain an acceptable signal/noise ratio. This operation is executed by the Python3 script **lidar_average_2019_v0_4.py.** This script reads as input the raw files and produces for each one an **averaged file.** The averaged files contain all the global attributes of the raw files and the variable *ch, nsht, time, starttime, stoptime* and *err*.

The variable *ch* stores the accumulated profiles; an accumulated profile is the signal accumulated during the averaging time obtained summing together the raw profiles. The variable *ch* is a matrix *npnt* x *nrec*, where *npnt* is the number of altitude bins composing each profile and *nrec* is the number of profiles in the file.

The variable *nsht* is a vector of length equal to *nrec* which contains the total number of summed lidar profiles producing each record

The variables *starttime, stoptime* and *time* contain the start, stop and average date and time of the accumulated profiles respectively, expressed in MJD2K.

The variable *err* contains the statistical uncertainty accumulated profile calculated during the averaging process.

The averaged file name is in the format *[location]_[average time in minutes]_[signal name]_[date time of start measurement].nc*, with the date and time in format yyyymmddHHMMSS; a short summary describing an averaged file is in Box 4.1.

Box 4.1: summary of an averaged file

```
rome_010min_532Hitan_20200310095800.nc {
 dimensions:
  npnt = 3000;
  nrec = UNLIMITED;   // (29 currently)
 variables:
  double time(nrec=29);
   :Units = "MJD2K";
   :LongName = "Time";

  double startime(nrec=29);
   :LongName = "StartTime";
   :Units = "MJD2K";

  double endtime(nrec=29);
   :LongName = "EndTime";
   :Units = "MJD2K";

  int nsht(nrec=29);
   :Units = " ";
   :LongName = "LaserShots";

  float ch(nrec=29, npnt=3000);
   :Units = "a.u.";
   :LongName = "AveragedSignal";
   :VPMT_V = 1000; // int
   :Wavelength_nm = 532; // int
   :Polarisation = 0; // int
   :bin_number = 3000; // int
   :Vertical_resolution_m = 7.5; // double
   :Range_Discriminator = 1; // int
   :Licel_channel = 2; // int
   :Trigger_delay = 1.0; // double
```

Issue 1.0

```
   float err(nrec=29, npnt=3000);
     :LongName = "SignalStandardDeviation";
     :Units = "a.u.";

 // global attributes:
 :System = "UNIROMA1 LIDAR         ";
 :Location = "Rome-Italy              ";
 :Longitude = "41.9                    ";
 :Latitude = "12.5                     ";
 :Altitude = "75.                      ";
 :SystemDescription = "Rayleigh-Raman Monostatic LIDAR      ";
 :Laser = "Quanta Ray          ";
 :RepetitionRate = "30                         ";
 :Energyperpulse = "NA                          ";
 :Emissionwavelength = "1064-532-355                      ";
 :Detectedwavelength = "532\t\t              ";
 :Zenithangle = "0                         ";
 :Acquisitionsystem = "LICEL-TR20-160                ";
 :chdescription = "532 total high range analog    ";
 :telescope = "100mm Cassegrain                    ";
 :filter = "0.3 nm BW filters               ";
 :Comments = "TEST 2018                  ";
 :_CoordSysBuilder = "ucar.nc2.dataset.conv.DefaultConvention";
}
```

5. LAS FILE

The averaged files are analysed by LAS (LIDAR Analysis Software). This software, its methods and algorithms and the additional input files needed to correctly operate have been extensively described in the technical note BAQ-MGT-TEM-SER-018, this paragraph will focus on the file produced by the analysis.

LAS needs as inputs a **meteorological file**, an ASCII file containing the pressure and temperature profiles. This file is the output produced by Pratica di Mare radiosounding station; the meteorological file is in a standard format and it can be easily downloaded at the following website: http://weather.uwyo.edu/upperair/sounding.html.

An additional file that can be used to produce higher quality data is the **sun-photometer file**. This file is a netcdf file containing the results of CIMEL or PREDE sun-photometers measurements. These two instrument are installed on the APL roof (see https://www.baqunin.eu for a description of the BAQUNIN supersite instruments); LAS employs the aerosol optical depth measured by them to produce an estimation of the LIDAR Ratio value of the aerosol layers. If no sun-photometers measurement is used in retrieval, the results are defined as level 1.0 analysis, otherwise the analysis level will be 1.5.

LAS output file is a netcdf file in a like-GEOMS format (http://evdc.esa.int/documentation/geoms/). It contains several flags and variables that are not expected by the GEOMS catalogue, but they are recorded trying to follows the general GEOMS guidelines. The following tables show the file content:

- Table 2 shows the global attributes of LAS file: non-standard GEOMS attribute are highlighted with a different colour; if the attribute value does not depend on the measurement it is reported in value column;
- Table 3 displays the dimensions defined in the file;
- Table 4 displays the LAS file variables with a short description;

- Table 5 describes the Variable Attribute, according to GEOMS guidelines.

Table 2: LAS file global attributes. Non-standard GEOMS attributes are highlighted with a different colour

Attribute	Description
PI_NAME	The name of the instrument PI
PI_AFFILIATION	PI affiliation
PI_EMAIL	Pi email
PI_ADDRESS	Pi address
DO_NAME	The DO (Data Originator) is the person or the team that generated and quality controlled the data.
DO_AFFILIATION	DO affiliation
DO_EMAIL	DO email
DO_ADDRESS	DO address
DS_NAME	The DS (Data Submitter) is the person that submitted the data
DS_AFFILIATION	DS affiliation
DA_EMAIL	DS email

Issue 1.0

DS_ADDRESS	DS address
DATA_DESCRIPTION	DATA_DESCRIPTION contains a brief sentence summarizing the file's data content
DATA_DISCIPLINE	The discipline of which measurement belong
DATA_GROUP	The global attribute DATA_GROUP has a two-field entry, specifying the origin of the data (experimental, model, or a combination of both) and the spatial characteristics of the data.
DATA_LOCATION	A string that identifies the measurement site in the EVDC database
DATA_SOURCE	A string that identifies the BAQUNIN LIDAR channels in the EVDC database. The source name depends on the analysed wavelength
DATA_VARIABLES	The variables contained in the file
DATA_START_DATE	Start date and time of the measurement
DATA_STOP_DATE	Stop date and time of the measurement
DATA_FILE_LEVEL	Non-standard GEOMS attribute. It indicates the level of the analysis in the work frame of the BAQUNIN project. Level 1.0 indicates a standard analysis, level 1.5 indicates that the Lidar Ratio has been estimated using the aerosol optical depth from sun-photometer measurements
DATA_FILE_VERSION	DATA_FILE_VERSION specifies the version of the data. It is not associated with a scientific algorithm or a processing algorithm, the attribute entry specifies an arbitrary version of the file, beginning with 001(with leading zeroes). With each update the data file version shall be incremented by 1
DATA_MODIFICATIONS	The optional global attribute DATA_MODIFICATIONS is intended to describe the data modification history associated with DATA_FILE_VERSION found in the data file
DATA_CAVEATS	The optional global attribute DATA_CAVEATS refers to potential issues with the data in the current data file and shall inform the user to use this data with caution
DATA_RULE_OF_USE	DATA_RULES_OF_USE entry is the PI's guidelines for the data usage.
DATA_AKNOWLEDGEMENT	DATA_ACKNOWLEDGEMENT specifies the PI's "desired" acknowledgment.
DATA_QUALITY	The global attribute DATA_QUALITY specifies information on quality of the data.
DATA_TEMPLATE	DATA_TEMPLATE specifies information on applicable GEOMS templates for reported data.
DATA_PROCESSOR	GEOMS optional attribute. Actually not used
FILE_NAME	Name of the file
FILE_GENERATION_DATE	Date and time of generation of the file
FILE_ACCESS	FILE_ACCESS has a multi-field character string entry referring to the file project association in the data archive. FILE_ACCESS is used to define the file's access rights through data centre interfaces.
FILE_PROJECT_ID	Optional GEOMS attribute. Actually not used
FILE_DOI	Optional GEOMS attribute. Actually not used
FILE_ASSOCIATION	Optional GEOMS attribute. Actually not used

FILE_META_VERSION	FILE_META_VERSION indicates the version of the metadata definitions used in the data file and the tool name used to generate the current HDF or netcdf data file.
LASER_WAVELENGHT	Non-standard GEOMS attribute. Indicates the laser wavelength observed
SIGNAL_FILE	Non-standard GEOMS attribute. Indicates the input averaged file path
VERTICAL_RESOLUTION_M	Non-standard GEOMS attribute. The averaged file vertical resolution [m]
REPETITION_RATE	Non-standard GEOMS attribute. The laser repetition rate [Hz]
LAS_VERSION	Non-standard GEOMS attribute. The version of LAS used for the analysis

Table 3: LAS file variables dimensions

Dimension name	Dimension size
nbin	The number of altitude grid points on which is calculated the signal, usually 3000
nprof	The number of profiles saved in the file.
maxnlayers	Maximum number of aerosol and cloud layers handled by LAS, equal to 5.
bottom_top	A dimension equal to 2. It is used to save the couple of bottom and top altitude of some special layers (see LAS documentation).

Table 4: LAS file variables

Variable Name	Type	Dimensions	Description
DATETIME	double	nprof	Date and time of the profile in MJD2K
DATETIME.START	double	nprof	Date and time of the start of the acquisition of the averaged profile in MJD2K
DATETIME.STOP	double	nprof	Date and time of the stop of the acquisition of the averaged profile in MJD2K
INTEGRATION.TIME	float	nprof	total integration time of the averaged profile
ALTITUDE	float	nbin	The altitude grid of the profiles above sea level [m]
LATITUDE	float	NA	Instrument latitude
LONGITUDE	float	NA	Instrument longitude
ALTITUDE.INSTRUMENT	float	NA	Instrument altitude above sea level [m]
WAVELENGTH_EMISSION	int	NA	Wavelength of the emitted beam
WAVELENGTH_DETECTION	int	NA	Wavelength detected
VOLUME.BACKSCATTER.RATIO	double	nprof, nbin	ratio between total backscatter coefficient and molecular backscatter coefficient
VOLUME.BACKSCATTER.RATIO_ UNCERTAINTY.COMBINED.STANDARD	double	nprof, nbin	volume backscatter ratio total uncertainty
AEROSOL.EXTINCTION.COEFFICIENT	float	nprof, nbin	extinction coefficient profiles

AEROSOL.EXTINCTION.COEFFICIENT_UNCERTAINTY.COMBINED.STANDARD	float	nprof, nbin	extinction coefficient profiles total uncertainty
AEROSOL.EXTINCTION.COEFFICIENT_UNCERTAINTY.RANDOM.STANDARD	float	nprof, nbin	extinction coefficient profiles random uncertainty
AEROSOL.EXTINCTION.COEFFICIENT_UNCERTAINTY.SYSTEMATIC.STANDARD	float	nprof, nbin	extinction coefficient profiles systematic uncertainty
AEROSOL.BACKSCATTER.COEFFICIENT	float	nprof, nbin	backscatter coefficient profiles
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.COMBINED.STANDARD	float	nprof, nbin	backscatter coefficient profiles total uncertainty
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.RANDOM.STANDARD	float	nprof, nbin	backscatter coefficient profiles random uncertainty
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.SYSTEMATIC.STANDARD	float	nprof, nbin	backscatter coefficient profiles systematic uncertainty
SIGNAL	double	nprof, nbin	signal profiles
SIGNAL_UNCERTAINTY.COMBINED.STANDARD	double	nprof, nbin	signal profiles total uncertainty
SIGNAL.BACKGROUND	double	nprof	The background of the signal measured for each profile
SIGNAL.BACKGROUND_UNCERTAINTY.RANDOM.STANDARD	double	nprof	The background uncertainty for each profile
BACKGROUNDLAYER.HEIGHT	double	nprof, bottom_top	Respectively the bottom and top altitudes of the vertical layer used to calculate the background
BACKGROUND.MULTIPLIER	float	nprof	A multiplicative constant used to eventually modify the original background computed by "background fit" or "last 100 points" methods
CLOUD.TOP.HEIGHT	double	nprof, maxnlayers	Top altitude of the cloud layers if presents
CLOUD.BOTTOM.HEIGHT	double	nprof, maxnlayers	bottom altitude of the cloud layers if presents
ALTITUDE.OVERLAP	double	nprof	LIDAR overlap altitude
AEROSOL.SCALE	double	nprof	altitudes scale used for extrapolation to ground of the volume backscattering ratio
AEROSOL.LAYER.BOTTOM.HEIGHT	double	nprof, maxnlayers	altitudes used as bottom of the aerosol layers in the calculation
AEROSOL.LAYER.TOP.HEIGHT	double	nprof, maxnlayers	altitudes used as top of the aerosol layers in the calculation
AEROSOL.LIDAR.FACTOR.HEIGHT	double	nprof, bottom_top	Respectively the bottom and top altitudes of the vertical layer used to compute the LIDAR factor in the calculation

AEROSOL.LIDAR.RATIO	float	nprof, maxnlayers	The LIDAR Ratio used in the computation for each aerosol layer (sorted according to increasing altitude).
AEROSOL.LIDAR.RATIO_ UNCERTAINTY.COMBINED.STANDARD			The LIDAR Ratio uncertainty calculated for each aerosol layer (sorted according to increasing altitude).
AEROSOL.OPTICAL.DEPTH	double	nprof, maxnlayers	The optical depth retrieved in the computation for each aerosol layer
AEROSOL.OPTICAL.DEPTH_ UNCERTAINTY.COMBINED.STANDARD	double	nprof, maxnlayers	The optical depth total uncertainty retrieved in the computation for each aerosol layer
CLOUD.LIDAR.RATIO	double	nprof, maxnlayers	The LIDAR Ratio retrieved in the computation for each cloud layer (sorted according to increasing altitude).
CLOUD.LIDAR.RATIO_ UNCERTAINTY.COMBINED.STANDARD	double	nprof, maxnlayers	The uncertainty of the LIDAR Ratio retrieved in the computation for each cloud layer
CLOUD.FLAG	int	nprof, maxnlayers	Flag indicating the computation mode used to retrieve the cloud. Flag values are: 0 = "single cloud"; 2 = "topcloud"
CLOUD.OPTICAL.DEPTH	double	nprof, maxnlayers	The optical depth retrieved in the computation for each cloud layer (sorted according to increasing altitude)
CLOUD.OPTICAL.DEPTH_ UNCERTAINTY.COMBINED.STANDARD	double	nprof, maxnlayers	The uncertainty of the optical depth calculated for each cloud layer (sorted according to increasing altitude)
LIDAR_CALIBRATION.FACTOR	double	nprof	The LIDAR calibration constant of each profile
LIDAR_CALIBRATION.FACTOR_ UNCERTAINTY	double	nprof	The LIDAR calibration constant uncertainty of each profile
AEROSOL.LIDAR.FACTOR	double	nprof	The LIDAR factor of each profile
AEROSOL.LIDAR.FACTOR_ UNCERTAINTY	double	nprof	LIDAR factor uncertainty of each profile
TOTAL.OPTICAL.DEPTH	float	nprof	total optical depth of the profile
TOTAL.OPTICAL.DEPTH_ UNCERTAINTY.COMBINED.STANDARD	float	nprof	total optical depth uncertainty
FLAG.ANALYZED	int	nprof	flag that indicates if a certain profile has been analysed. Flag values are: 0 = not yet analysed; 1 = analysed

Issue 1.0

FLAG.COMPUTATION	int	nprof	flag that indicates the retrieval algorithm employed. Flag values are: 0 = factor retrieval; 1 = C retrieval
FLAG.MERGEDSIGNAL	int	nprof	flag that indicates if the signal is the result of merging two channels measurements. Flag values are: 0 = no merging operation; 1= signal merged S1S2 mode; 2 = signal merged S2S1 mode
MERGINGLAYER.HEIGHT	double	nprof, bottom_top	Respectively the bottom and top altitudes of the layer where the profiles have been merged
FLAG.SYNCHSIGNAL	int	nprof	Flag that indicates that the signal has been synchronized with another channel. Flag values are: 0 = no synchronization; 1 = signal S1 synchronized on S2; 2 = signal S2 synchronized on S1
SYNCHLAYER.HEIGHT	int	nprof, bottom_top	Respectively the bottom and top altitudes of the layer where the profiles have been synchronized
SYNCH.BINSHIFT	int	nprof	The shift of the signal in number of bins
FLAG.CONVERGENCE	int	nprof	A flag to recognize the convergence mode used for the profile's retrieval: 0 = integral convergence; 1 = punctual convergence
FLAG.LR	int	nprof	A flag to recognize the LR mode used for the aerosol calculation: 0 = LR as user input; 1 = LR obtained from CIMEL tau, 2 = LR obtained from POM tau
INDEPENDENT.OPTICAL.DEPTH	double	nprof	Optical depth measured by sun-photometers and used to calculate aerosol LR; it is equal to 0 if no independent measurements were available
INDEPENDENT.OPTICAL.DEPTH_ UNCERTAINTY	double	nprof	Uncertainty of the optical depth measured by sun-photometers
SMOOTH.HEIGHT	float	nprof	The altitude above which the results are eventually vertically smoothed
SMOOTH.BINS	int	nprof	The number of points averaged in the vertical smoothing process; if this value is 0 no smoothing process occurred

SHOTS.NUMBER	int	nprof	The number of laser shots accumulated to composing the signal
RETRIEVAL.MODE	int	nprof	Flag that indicates the retrieval mode used for obtaining the data. The flag values are: -1 = not analysed, 0 = elastic-LR retrieval, 1 = RAMAN-LR retrieval
ANGSTROM.COEFFICIENT	float	nprof	angstrom coefficient used in the RAMAN retrieval if RETRIEVAL.MODE flag is 1
ANGSTROM.COEFFICIENT_ UNCERTAINTY	float	nprof	angstrom coefficient uncertainty
OVERLAP.OPTICAL.DEPTH	double	nprof	Optical depth measured from the ground to the overlap altitude. The extinction coefficient below the overlap altitude is not a direct measurement, but an extrapolation

Table 5: Variable attributes

Variable Attribute	Description
VAR_NAME	The name of the variable
VAR_SIZE	Size of the variable
VAR_DESCRIPTION	A short description of the variable
VAR_NOTES	Notes about variable usage or calculation
VAR_DEPEND	The dependency of the variable with respect of other variables in the file
VAR_DATA_TYPE	The variable numeric representation
VAR_UNITS	Variable units
VAR_SI_CONVERSION	String that describes how convert the variable units in SI units. The string is composed by 3 camps separated by ";". A conversion has the general form $y = a + bx$, where x are the actual units and y are the desired units. The string first camp contains the a parameter, the string second camp the b parameter and the third camp the SI units
VAR_VALID_MIN	The minimum acceptable value of the variable
VAR_VALID_MAX	The maximum acceptable value of the variable
VAR_FILL_VALUE	This value is assigned to variable elements affected by some problem or not calculated

LAS output file name is in the format *[location]_[average time in minutes]min_[signal type]_back_[start date in format yyyymmddHHMMSS]_lev[level]_[version number].nc.* The signal type strings are described in Table 1 with the addition of some *special strings* described in

Table 6 used to mark the 532 nm merged signal and the 355 nm signal analysed using the N2d RAMAN; the level camp of the file name refers to the analysis level (see above), while the version number is a three digit integer referring to the global attribute *DATA_FILE_VERSION*.

Issue 1.0

Table 6: Special signal strings

Special string	Description
532MTAN	532 nm signal obtained merging the *532Hitan* and *532tan* measurements
532MPAN	532 nm signal obtained merging the *532Hipan* and *532pan* measurements
355phRAMANN2d	355 nm signal analysed together with the 386 nm N2 RAMAN signal

Issue 1.0

6. BAQUNIN DB FILE

The *BAQUNIN DB file* refers to the file published on the BAQUNIN project website archive at https://www.baqunin.eu/products/aerosol-back-lidar/. The file format resembles the GEOMS aerosol LIDAR measurement template, named *GEOMS-TE-LIDAR-AEROSOL-004;* detailed information about GEOMS LIDAR template, attributes and variables is available at https://avdc.gsfc.nasa.gov/index.php?site=701636862#GEOMS-EX-LIDAR-GA.

BAQUNIN DB file name is assigned automatically using underscore separated global attribute entries (lowercase) and the file extension *".nc*": *DATA_DISCIPLINE* last camp ("groundbased"), *DATA_SOURCE*, *DATA_LOCATION* ("rome.sapienza"), *DATA_START_DATE* (format yyymmddtHHMMSS), *DATA_STOP_DATE* (format yyymmddtHHMMSS), *DATA_FILE_LEVEL*, *DATA_FILE_VERSION*.

The BAQUNIN DB file differs from a standard GEOMS LIDAR file in the following points:

- the GEOMS template is intended to record the result of the retrieval of a **single** profile, while the BAQUNIN DB file contains the results of all the profiles analysed and recorded in the LAS file
- the BAQUNIN DB file contains LAS_VERSION and DATA_FILE_LEVEL non-standard global attributes, described in the paragraph 5
- the BAQUNIN DB file name contains the reference to the DATA_FILE_LEVEL attribute
- the BAQUNIN DB file contains the information about Lidar Ratio uncertainty in the non-standard variable *AEROSOL.LIDAR.RATIO_UNCERTAINTY.COMBINED*.*STANDARD.*

See Appendix A for the list of LAS variables transferred to the DAQUNIN DB file.

The software used to convert the LAS file into the BAQUNIN DB file is the Python3 script *LAS2GEOMS.py*.

Issue 1.0

7. EVDC FILES

The *EVDC files* refers to the files published in the EVDC archive. Following all the GEOMS guidelines, these files record the result of the retrieval of a single signal profile, a measurement session usually can produce several EVDC files. The EVDC files present the non-standard GEOMS attributes *LAS_VERSION* and *DATA_FILE_LEVEL*. The presence of these two non-standard attributes is accepted by the EVDC archive Quality Checker Tool.

The EVDC file is produced extracting a single profile results from the BAQUNIN DB file or the average of more profiles of the BAQUNIN DB file. In the last case, the EVDC file is referred to a measurement time longer than the BAQUNIN DB file profiles integration time (see the variable INTEGRATION.TIME).

EVDC file name follows strictly the GEOMS guideline is constructed as the BAQUNIN DB file name, without the *DATA_FILE_LEVEL* reference.

The software used to extract or average the information of the BAQUNIN DB file to produce the EVDC file is the Python3 script *LAS2GEOMS_singleprofile.py.*

See Appendix A for the list of LAS variables transferred to the EVDC file.

8. APPENDIX

8.1 Appendix A: LAS files variables transferred into the BAQUNIN DB and EVDC files

The following table shows the list of LAS file variables that are transferred to the BAQUNIN DB and EVDC files. As described in the paragraph 7, the BAQUNIN DB files contains the result of all profiles of the measurement session, instead in each EVDC files only a single profile is present.

Table 7 LAS file variables transferred to the BAQUNIN DB and EVDC files

LAS Variable Name	BAQUNIN DB	EVDC
DATETIME	☑	☑
DATETIME.START	☑	☑
DATETIME.STOP	☑	☑
INTEGRATION.TIME	☑	☑
ALTITUDE	☑	☑
LATITUDE	☑	☑
LONGITUDE	☑	☑
ALTITUDE.INSTRUMENT	☑	☑
WAVELENGTH_EMISSION	☑	☑
WAVELENGTH_DETECTION	☑	☑
VOLUME.BACKSCATTER.RATIO		
VOLUME.BACKSCATTER.RATIO_UNCERTAINTY.COMBINED.STANDARD		
AEROSOL.EXTINCTION.COEFFICIENT	☑	☑
AEROSOL.EXTINCTION.COEFFICIENT_ UNCERTAINTY.COMBINED.STANDARD	☑	☑
AEROSOL.EXTINCTION.COEFFICIENT_ UNCERTAINTY.RANDOM.STANDARD	☑	☑
AEROSOL.EXTINCTION.COEFFICIENT_UNCERTAINTY.SYSTEMATIC.STANDARD	☑	☑
AEROSOL.BACKSCATTER.COEFFICIENT	☑	☑
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.COMBINED.STANDARD	☑	☑
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.RANDOM.STANDARD	☑	☑
AEROSOL.BACKSCATTER.COEFFICIENT_UNCERTAINTY.SYSTEMATIC.STANDARD	☑	☑
SIGNAL		
SIGNAL_UNCERTAINTY.COMBINED.STANDARD		
SIGNAL.BACKGROUND		
SIGNAL.BACKGROUND_UNCERTAINTY.RANDOM.STANDARD		
BACKGROUNDLAYER.HEIGHT		
BACKGROUND.MULTIPLIER		
CLOUD.TOP.HEIGHT		
CLOUD.BOTTOM.HEIGHT		
ALTITUDE.OVERLAP		
AEROSOL.SCALE		
AEROSOL.LAYER.BOTTOM.HEIGHT		
AEROSOL.LAYER.TOP.HEIGHT		
AEROSOL.LIDAR.FACTOR.HEIGHT		

Issue 1.0

AEROSOL.LIDAR.RATIO	☑	☑
AEROSOL.LIDAR.RATIO_UNCERTAINTY.COMBINED.STANDARD	☑	
AEROSOL.OPTICAL.DEPTH		
AEROSOL.OPTICAL.DEPTH_UNCERTAINTY.COMBINED.STANDARD		
CLOUD.LIDAR.RATIO		
CLOUD.LIDAR.RATIO_UNCERTAINTY.COMBINED.STANDARD		
CLOUD.FLAG		
CLOUD.OPTICAL.DEPTH		
CLOUD.OPTICAL.DEPTH_UNCERTAINTY.COMBINED.STANDARD		
LIDAR_CALIBRATION.FACTOR		
LIDAR_CALIBRATION.FACTOR_UNCERTAINTY		
AEROSOL.LIDAR.FACTOR		
AEROSOL.LIDAR.FACTOR_UNCERTAINTY		
TOTAL.OPTICAL.DEPTH		
TOTAL.OPTICAL.DEPTH_UNCERTAINTY.COMBINED.STANDARD		
FLAG.ANALYZED		
FLAG.COMPUTATION		
FLAG.MERGEDSIGNAL		
MERGINGLAYER.HEIGHT		
FLAG.SYNCHSIGNAL		
SYNCHLAYER.HEIGHT		
SYNCH.BINSHIFT		
FLAG.CONVERGENCE		
FLAG.LR		
INDEPENDENT.OPTICAL.DEPTH		
INDEPENDENT.OPTICAL.DEPTH_UNCERTAINTY		
SMOOTH.HEIGHT		
SMOOTH.BINS		
SHOTS.NUMBER		
RETRIEVAL.MODE		
ANGSTROM.COEFFICIENT		
ANGSTROM.COEFFICIENT_UNCERTAINTY		
OVERLAP.OPTICAL.DEPTH		

Issue 1.0

9. REFERENCES

R1. The Python Language reference, available at: https://docs.python.org/3/reference

R2. LabView documentation, available at website: https://zone.ni.com/reference/en-XX/help/371361R-01/lvconcepts/labview_documentation_resources/

R3. NETCDF4 library documentation, available at https://unidata.github.io/netcdf4-python/netCDF4

R4. GEOMS reference: https://evdc.esa.int/documentation/geoms/

R5. GEOMS templates: https://avdc.gsfc.nasa.gov/index.php?site=701636862#GEOMS-EX-LIDAR-GA

R6. LICEL Transient Recorder system specifics: www.licel.com.

R7. BAQUNIN website: https://www.baqunin.eu

R8. EVDC website: https://evdc.esa.int/

Issue 1.0

End of Document