

**BUFR, NETCDF & HDF5
DATA PREPROCESSING
FOR ARPEGE/ALADIN/AROME.**

PARAM.CFG FILE (FOR OBS CONVERT)

VERSION ANGLAISE / ENGLISH VERSION

v. 1.0.0

CONTENTS

1	Fist of all.....	4
1.1	Used writing rules.....	4
2	Introduction.....	4
3	History.....	4
4	Aim of the param.cfg file.....	4
5	Common writing rules used in param.cfg.....	4
6	For BUFR files.....	5
6.1	Description block syntax.....	5
6.2	BUFR label – /BUFR label (or BEGIN label – END label).....	5
6.3	nb1 nb2 nb3 nb4 nb5.....	5
6.4	encoding (key “model”).....	6
6.5	encoding (key “codage”).....	6
6.6	control.....	6
6.7	offset.....	6
6.8	Values (no longer used).....	6
6.9	Information field.....	6
7	For NETCDF(4) Files.....	7
7.1	Description block syntax.....	7
7.2	NETCDF label – /NETCDF label.....	7
7.3	nb1 nb2 nb3 nb4.....	7
7.4	dimension.....	7
7.5	genattrib.....	7
7.6	variable.....	8
7.7	channel.....	8
8	For HDF5 files.....	9
8.1	Description block syntax.....	9
8.2	HDF5 label – /HDF5 label.....	9
8.3	nb1 nb2 nb3.....	9
8.4	groupe.....	9
8.5	Genattrib.....	9
8.6	dataset.....	10

Annexe – 1 : param.cfg files

1	For some BUFR files.....	13
2	For SEVIRI NETCDF file.....	16
3	For HDF5 files.....	18

1 Fist of all

1.1 Used writing rules

- File names, jobs and programs are written in **bold** font.
- Block and key labels used in **param.cfg** file are written in *italic*.
- Examples are written using the `Courier New` font.
- In examples, optional entries are inserted within square brackets.
- Likewise, in examples, items written using the *Courier New italic* font must be replaced by their values.

2 Introduction

The purpose of this guide is to explain how to write a **param.cfg** file, used as an input file for **ObsConvert** (still under development) from CY48.

Knowledge of BUFR, NETCDF and HDF5 files formats is required.

3 History

Version 1.0.0 (16/03/2022) :

– first English release.

4 Aim of the param.cfg file

This file consists of blocks containing the required records allowing **ObsConvert** to recognize and process (or reject !) an input data file (BUFR, NETCDF or HDF5 format).

This implies that a particular block and its records are related to the format of a particular file and the data-type it contains.

Furthermore, some records may be added to be used in decoding subroutines of **ObsConvert** (following the programmer's own choices !).

5 Common writing rules used in param.cfg

- No blank is allowed before a record or a block label.
- Length of records can be up to 255 characters.
- Items *indice*, *posbufr*, *valeur*, *number* and *FXY* are integers.
- Items *chaîne* is a string and must be surrounded with “ ”.
- A comment line must begin with #.
- A comment cannot follow a record on the same line.
- Items of a record are separated by one or more blank character.
- The file is case sensitive.

6 For BUFR files

6.1 Description block syntax

```

BUFR label
nb1 nb2 nb3 nb4 0
codage      1      +/-FXY1 [information]
...
[codage  indicen +/-FXYn [information]]
[control  1      valeur [information]]
...
[control  indicen valeur [information]]
[offset   1      valeur [information]]
...
[offset  indicen valeur [information]]
[values  posbufr FXY [information]]
...
[values  posbufr FXY [information]]
/BUFR label

```

or

```


BUFR label
0 nb2 nb3 nb4 1
model      0      +/-FXY
[control  1      valeur [information]]
...
[control  indicen valeur [information]]
[offset   1      valeur [information]]
...
[offset  indicen valeur [information]]
[values  posbufr FXY [information]]
...
[values  posbufr FXY [information]]
/BUFR label

```

6.2 BUFR label – /BUFR label (or BEGIN label – END label)

Respectively open and close a description block for BUFR file which data-type is given by *label*. *label* is a string up to 16 characters and must match one of the listed sensors in the ExpandBufrFile() subroutine (see *bator_decodbufr_mod.F90*).

For backward compatibility with CY42, previous writings, *BEGIN label* and *END label* are also allowed.

 It is allowed to create several description blocks for a particular data-type. In this case, the list of « *codage* » keys must be different from one block to another one.

6.3 *nb*₁ *nb*₂ *nb*₃ *nb*₄ *nb*₅

These values give number of records *codage*, *control*, *offset*, *values* and *model* contained in the block description :

- *nb1* for number of records for *codage*. Must be > 0 or, if there is one *model* must be $= 0$.
- *nb2* for number of records for *control*. Must be ≥ 0 .
- *nb3* for number of records for *offset*. Must be ≥ 0 .
- *nb4* for number of records for *values*. Must be ≥ 0 . (no longer used)
- *nb5* can only takes 0 (no *model*) or 1 (*model* exists).

6.4 encoding (key “model”)

It is the first way for Bator to recognize a BUFR file. For a particular file (in section 3) it may exist a characteristic unexpanded descriptor (or template) listed in WMO BUFR_table D. If it is the case, you can use it as *model*, avoiding you to use *codage* keys. See example page 13.

6.5 encoding (key “codage”)

Bator uses this key to recognize a BUFR file. For a particular file, the list of *codage* keys must strictly match the list (in number and order) of unexpanded descriptors inside the file (section 3 of a BUFR file). So, there is at least one *codage* key.

In some cases, BUFR files which contain the same data-type may have one or several different complementary unexpanded descriptors before and/or after the main sequence. To minimize number of blocks contained in **param.cfg** file, you can list these unexpanded descriptors preceding them with a minus character. If applied, these unexpanded descriptors will be ignored. See example page 13.

6.6 control

Optional key used to give an additional information which is missing in the BUFR file (number of channels, delayed replication factor position when more than one exists...). This facility may be used to enable dedicated decoding software to be more flexible.

6.7 offset

The value defined by this optional key gives the number of B descriptors to jump when an iteration is done in the decoding subroutine.

6.8 Values (useless in ObsConvert)

The values associated with this key give the position and the FXY of a descriptor (section 4 of a BUFR file) which value must be retrieved in the decoding subroutine.

Although its use is optional, it is strongly recommended.

By convention, if *posbuf* value is set to -1 for a given descriptor, its position must be calculated in the decoding subroutine,



A descriptor (FXY) must be unique. Nevertheless, if it is listed several time, only the last one will be considered.

6.9 Information field

This optional field, unused by **bator**, can be useful for the programmer. It gives information concerning the current record (WMO label...).

7 For NETCDF(4) Files

This file format is currently used for Lannion SEVIRI data preprocessing. The syntax of this description block can evolve when other data-type preprocessing will be planned. See example page [16](#).

7.1 Description block syntax

```
NETCDF label
nb1 nb2 nb3 nb4
dimension  'chaîne1'
...
[dimension 'chaînen']
{genattrib 'chaîne1'}
...
[genattrib 'chaînen']
variable   'chaîne1'
...
[variable  'chaînen']
[channel   'chaîne1']
...
[channel   'chaînen']
/NETCDF label
```

7.2 NETCDF label – /NETCDF label

Respectively open and close a description block for NETCDF4 file which data-type is given by *label*. *label* is a string up to 16 characters and must match one of the listed sensors in PrefetchNetcdf() subroutine (see bator_decodnetcdf_mod.F90).



a NETCDF description block must list all the first level items contained in the data file.

7.3 *nb*₁ *nb*₂ *nb*₃ *nb*₄

These values give the number of records of *dimension*, *genattrib*, *variable*, *channel* expected in the description block :

- *nb*₁ for number of records for *dimension*. Must be > 0.
- *nb*₂ for number of records for *genattrib*. Must be ≥ 0.
- *nb*₃ for number of records for *variable*. Must be > 0.
- *nb*₄ for number of records for *channel*. Must be ≥ 0.

7.4 *dimension*

The value of this key, *chaîne*, matches a dimension label used in the NETCDF data file. There must be as many records using this key as there are dimension labels in the NETCDF data file.

7.5 *genattrib*

The value of this key, *chaîne*, matches a global attribute label used in the NETCDF data file.

7.6 variable

The value of this key, *chaîne*, matches a variable label used in the NETCDF file. It is used to get meta-data like latitude, longitude and so on.

7.7 channel

Like *variable* key, the value of this key, *chaîne*, matches a variable label used in the NETCDF file. Its particularity is to concern data for a particular channel, instead of metadata.

8 For HDF5 files

8.1 Description block syntax

Syntaxe :

```
HDF5 label
nb1 nb2 nb3
[groupe      number  'chaîne1']
...
[groupe      number  'chaînen']
{genattrib   0      'chaîne1']
...
[genattrib   0      'chaînen']
[dataset     number  'chaîne1']
...
[dataset     number  'chaînen']
/HDF5 label
```

8.2 HDF5 label – /HDF5 label

Respectively open and close a description block for HDF5 file which data-type is given by *label*. *label* is a string up to 16 characters and must match one of the listed sensors in PrefetchHdf5() subroutine (see bator_decodhdf5_mod.F90).



a HDF5 description block must list all the first level items contained in the data file.

8.3 nb₁ nb₂ nb₃

These values give number of records *groupe*, *genattrib*, *dataset* contained in the block description :

- *nb1* for number of records for *groupe*. Must be ≥ 0 .
- *nb2* for number of records for *genattrib*. Must be ≥ 0 .
- *nb3* for number of records for *dataset*. Must be ≥ 0 .

8.4 groupe

This key needs two parameters :

- *number* is a unique identifier (integer) affected to the group and chosen by the programmer.
- *chaîne* is the label (or the constant part of the label) of the group in the HDF5 file.



If « chaîne » is the constant part of a group label, which must be at the beginning of this label, « number » must be negative. See example page 18.


8.5 Genattrib

The value of this key, *chaîne*, matches a top level attribute label used in the HDF5 file. Constant 0 is mandatory.

8.6 dataset

This key needs two parameters :

- *number* is a unique identifier (integer) affected to the dataset and chosen by the programmer.
- *chaîne* is the label (or the constant part of the label) of the group in the HDF5 file.

 *If « chaîne » is the constant part of the dataset label, which must be at the beginning of this label, « number » must be negative. See example page [18](#).*

ANNEXES

ANNEXE – 1 : PARAM.CFG FILES

1 For some BUFR files

```

BUFR synop1
0 1 0 0 1
model      0  307080
control    1  2 iteration's number for RR
/BUFR synop1

BUFR radome
0 1 0 0 1
model      0  307096
control    1  0
/BUFR radome

BUFR tesac
33 0 0 0 0
codage     1  001011
codage     2  301011
codage     3  301012
codage     4  301021
codage     5  306194
codage     6  002032
codage     7  002033
codage     8  022067
codage     9  022068
codage    10  103000
codage    11  031002
codage    12  007062
codage    13  022043
codage    14  022062
codage    15  002040
codage    16  002031
codage    17  002031
codage    18 -103000
codage    19 -031002
codage    20 -007062
codage    21 -022004
codage    22 -022031
codage    23 -022063
codage    24 -022228
codage    25 -222000
codage    26 -101000
codage    27 -031002
codage    28 -031031
codage    29 -001031
codage    30 -001032
codage    31 -101000
codage    32 -031002
codage    33 -033192
/BUFR tesac

BUFR hirs
0 0 0 0 1
model      1  310008

BUFR amdaromm
0 1 0 0 1
model      1  311001
control    1  0
/BUFR amdaromm
    
```

BUFR, NETCDF & HDF5 data preprocessing – param.cfg (ObsConvert)

```
BUFR amdaromm
0 1 0 0 1
model      1 311005
control    1 0
/BUFR amdaromm

BUFR amdaromm
0 1 0 0 1
model      1 311008
control    1 0
/BUFR amdaromm

BUFR amdaromm
0 1 0 0 1
model      1 311009
control    1 0
/BUFR amdaromm

BUFR amdaromm
0 1 0 0 1
model      1 311010
control    1 0
/BUFR amdaromm

# amdaromm from VHHHH (3)
BUFR amdaromm
15 1 0 0 0
codage     1 001008
codage     2 001023
codage     3 301021
codage     4 301011
codage     5 301013
codage     6 311006
codage     7 033025
codage     8 008004
codage     9 013002
codage    10 104000
codage    11 031001
codage    12 011075
codage    13 011076
codage    14 011037
codage    15 011077
control    1 0
/BUFR amdaromm

# amdaromm from VHHHH (1)
BUFR amdaromm
10 1 0 0 0
codage     1 001008
codage     2 001023
codage     3 301011
codage     4 301013
codage     5 301021
codage     6 008004
codage     7 102000
codage     8 031001
codage     9 311006
codage    10 013002
control    1 311008
/BUFR amdaromm

# amdaromm from VHHHH (2)
BUFR amdaromm
10 1 0 0 0
codage     1 001008
codage     2 001023
codage     3 301011
codage     4 301013
```

BUFR, NETCDF & HDF5 data preprocessing – param.cfg (ObsConvert)

```
codage      5  301021
codage      6  008004
codage      7  102000
codage      8  031001
codage      9  311007
codage     10  013002
control     1  311009
/BUFR amdaromm
```

2 For SEVIRI NETCDF file

```

NETCDF seviri
2 36 17 8
dimension  'nx'
dimension  'ny'
genattrib  'satid'
genattrib  'ncml_version'
genattrib  'nwc_saf_algorith_version'
genattrib  'title'
genattrib  'comment'
genattrib  'references'
genattrib  'created_by'
genattrib  'institution'
genattrib  'source'
genattrib  'Metadata_Conventions'
genattrib  'summary'
genattrib  'hrit_converted_by'
genattrib  'keywords'
genattrib  'keywords_vocabulary'
genattrib  'naming_authority'
genattrib  'cdm_data_type'
genattrib  'creator_name'
genattrib  'creator_url'
genattrib  'creator_email'
genattrib  'geospatial_vertical_min'
genattrib  'geospatial_vertical_max'
genattrib  'time_coverage_start'
genattrib  'time_coverage_end'
genattrib  'time_coverage_duration'
genattrib  'time_coverage_resolution'
genattrib  'licence'
genattrib  'Conventions'
genattrib  'Area_of_acquisition'
genattrib  'Scanning_direction'
genattrib  'history'
genattrib  'date_created'
genattrib  'id'
genattrib  'geospatial_lat_min'
genattrib  'geospatial_lat_max'
genattrib  'geospatial_lon_min'
genattrib  'geospatial_lon_max'
variable   'lat'
variable   'lon'
variable   'sat_azi_ang'
variable   'sat_zen_ang'
variable   'CTP'
variable   'CT'
variable   'CTP_QUALITY'
variable   'CT_QUALITY'
variable   'time'
variable   'dtime'
variable   'commentaires'
variable   'satellite'
variable   'geos'
variable   'ImageNavigation'
variable   'GeosCoordinateSystem'
variable   'y'
variable   'x'
channel    'IR_039'
channel    'WV_062'
channel    'WV_073'
channel    'IR_087'
channel    'IR_097'
channel    'IR_108'
channel    'IR_120'
channel    'IR_134'

```

```
/NETCDF seviri
```

3 For HDF5 files

```

HDF5 mtvza
0 1 31
groupe 1 'HDF4_DIMGROUP'
dataset -1 'm_m2_01_10.6V'
dataset -2 'm_m2_02_10.6H'
dataset -3 'm_m2_03_18.7V'
dataset -4 'm_m2_04_18.7H'
dataset -5 'm_m2_05_23.8V'
dataset -6 'm_m2_06_23.8H'
dataset -7 'm_m2_07_36.7V'
dataset -8 'm_m2_08_36.7H'
dataset -9 'm_m2_09_91.65V'
dataset -10 'm_m2_11_52_80V'
dataset -11 'm_m2_12_53_30V'
dataset -12 'm_m2_13_53_80V'
dataset -13 'm_m2_14_54_64V'
dataset -14 'm_m2_15_55_63V'
dataset -15 'm_m2_16_57_0.32_0.1H'
dataset -16 'm_m2_17_57_0.32_0.05H'
dataset -17 'm_m2_18_57_0.32_0.025H'
dataset -18 'm_m2_19_57_0.32_0.01H'
dataset -19 'm_m2_20_57_0.32_0.005H'
dataset -20 'm_m2_21_183_7.0V'
dataset -21 'm_m2_22_183_1.4V'
dataset -22 'm_m2_23_183_3.0V'
dataset -23 'm_m2_26_31.5V'
dataset -24 'm_m2_27_31.5H'
dataset -25 'm_m2_Julian Day'
dataset -26 'm_m2_Lattitude'
dataset -27 'm_m2_Longitude'
dataset -28 'm_m2_SunAzimuth'
dataset -29 'm_m2_SunZenith'
dataset -30 'm_m2_Time of day'
dataset -31 'm_m2_Surface'
/HDF5 mtvza

HDF5 odim
1 4 0
genattrib 0 'Conventions'
groupe 1 'what'
groupe 2 'where'
groupe 3 'how'
groupe -4 'dataset'
/HDF5 odim
    
```