bufr_filter practicals I

Roberto Ribas

Forecast department
Copernicus Production section

ECMWF

Preparations:

Copy the directory bufr_tools_filter_adv in your local directory

cd $SCRATCH

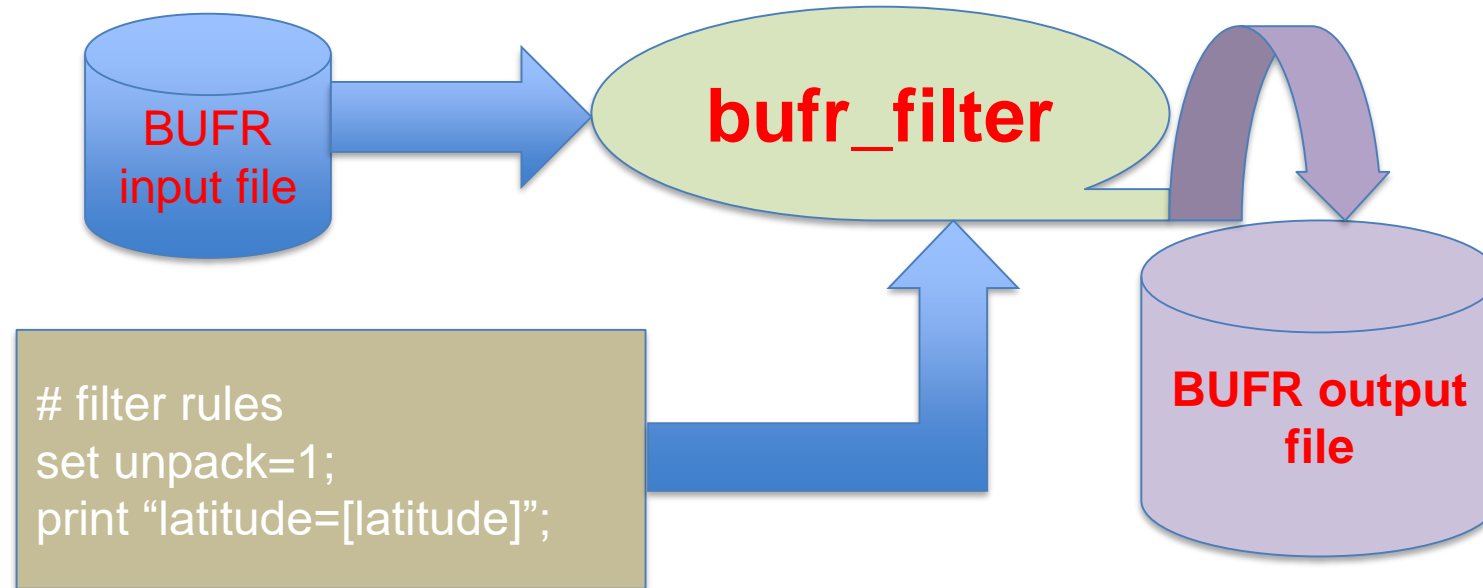**cp –r   ~trx/ecCodes/2019/bufr_tools_filter_adv   ./**

cd bufr_tools_filter_adv

bufr_filter reminder

To run bufr_filter we need a BUFR file and a filter (text file with bufr_filter rules) and we may produce an output BUFR file, or just print some data.

bufr_filter –o output.bufr filter_text_file input.bufr

BUFR input file → **bufr_filter** → BUFR output file

```
# filter rules
set unpack=1;
print "latitude=[latitude]";
```

ECMWF

bufr_filter to inspect files(exercise 1)

1. Inspect the contents of a BUFR file. Create a filter file and use it to print the following keys for the **synop1.bufr** file:

   unexpandedDescriptors

   expandedDescriptors

   expandedAbbreviations

   expandedNames

2. Print the **latitude,longitude,airTemperature** data (what happens if you don't set unpack=1 in your filter?).

3. For **airTemperature**, print the **units,code,width,reference** and **scale** (which are the attributes of the key airTemperature) you can access them via the operator arrow "→".

bufr_filter to inspect files (exercise 2)

Use **bufr_filter** to print the attribute **percentConfidence** of the key **pressureReducedToMeanSeaLevel** from the file:

synop_with_confidence.bufr

Print all the attributes of **pressureReducedToMeanSeaLevel**.

Hint use **bufr_dump** with the option **–jf** to display all attributes and send it to a file.

bufr_dump –jf synop_with_confidence.bufr > out.json

Use **kate** to view the file **out.json**.

bufr_filter access by rank/condition (exercise 3)

Print **latitude,longitude, height** from the file **temp.bufr**.

Get a dump of the file in JSON and compare it with the result of the **bufr_filter**. Alternatively you can use option **–p** of **bufr_dump** to see the keys.

1.- By using **bufr_filter** and accessing keys by rank find the second instance of   **windSpeed** and **windDirection**.

2.- By using **access by condition**(Using /key=value/ syntax), print **windSpeed** and **windDirection** for specific pressure (for example 66850 Pa) and **airTemperature** and **pressure** for specific geopotential levels (for example **nonCoordinateGeopotentialHeight** value of 35 gpm or 4199 gpm).

bufr_filter to inspect uncompressed data (exercise 4)

bufr_filter can work with uncompressed data.

Use the file **synop_multi_subset.bufr** to print the following information

- **compressedData**.
- **stationNumber,stationOrSiteName,latitude,longitude,airTemperature,**

dewPointTemperature for subsets number 3 and 5.

Check your results with the JSON output of **bufr_dump** and **bufr_dump -p**.

bufr_filter to inspect compressed data (exercise 5)

BUFR files can contain compressed data, **bufr_filter** works also with compressed data.

Use the file **scatterometer.bufr** to retrieve:

- **compressedData** key.

- **latitude,longitude** keys.

- the **backscatter** for all subsets with **beamIdentifier 2**. You can use some formatting to improve the output.

Hint: Refer back to the **bufr_filter** introduction slides for formatting options.

Check your results with the output of **bufr_dump,** the option **–p** may be handy.

BUFR Subset Extraction

We start to use the –o option !!!

bufr_filter to extract subsets

Sometimes we may need to retrieve only a small number of subsets. To do so we have to set some keys.

To extract only the fourth subset.

```
set unpack=1;
set extractSubset=4;
set doExtractSubsets=1;
write;
```

To extract from subset 3 to subset 10.

```
set unpack=1;
set extractSubsetIntervalStart=3;
set extractSubsetIntervalEnd=10;
set doExtractSubsets=1;
write;
```

Remark: Remember to *set doExtractSubsets* to 1 before writing. This filtering works for **compressed** and **uncompressed** data.

Bufr_filter to extract subsets

• The subset extraction can be done for specific subsets for example the 1st, 4th, and 8th subsets.

```
set unpack=1;
set extractSubsetList={1,4,8};
set doExtractSubsets=1;
write;
```

Extracting the 4th subset

```
set unpack=1;
set extractSubset=4;
set doExtractSubsets=1;
write;
```

bufr_filter -o aa.b ex6.1.flt amv2_87.bufr

ECMWF
EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

bufr_filter to extract subsets in an area

bufr_filter can be used to extract subsets within an area defined by a bounding box.

This feature only works for **compressed data**.

This may be useful with satellite data, to retrieve all the subsets in a certain area defined by a bounding box. Bear in mind, that latitudes/longitudes in the output file are not necessarily ordered.

bufr_filter to extract subsets in an area

```
set unpack=1;
transient originalNumberOfSubsets=numberOfSubsets;
set extractAreaNorthLatitude=26.0;
set extractAreaSouthLatitude=23.0;
set extractAreaEastLongitude=-35.0;
set extractAreaWestLongitude=-55.0;
set doExtractArea=1;
if ( extractedAreaNumberOfSubsets != 0) {
    write;
}
print "extracted [extractedAreaNumberOfSubsets] of [originalNumberOfSubsets]
subsets";
```

bufr_filter to extract subsets in an area

Comments

The bounding box of the region must be set, through the keys

extractAreaNorthLatitude
extractAreaSouthLatitude
extractAreaEastLongitude
extractAreaWestLongitude

The if statement avoids writing if the extraction does not find any subsets in the given area.

A *transient* variable is used to keep track of the number of subsets in the message and see how many have been selected (through the key extractedAreaNumberOfSubsets).

bufr_filter to extract subsets in an area (exercise 6)

Using the file **amv2_87.bufr** and **bufr_filter** do an area extraction for a relevant area.

Use **bufr_dump** to see the ranges of latitudes and longitudes.

How many subsets were selected?

bufr_filter to extract subsets in a time range

To select subsets in a given time range the following keys are provided:

extractDateTimeYearStart
extractDateTimeMonthStart
extractDateTimeDayStart
extractDateTimeHourStart
extractDateTimeMinuteStart
extractDateTimeSecondStart
extractDateTimeYearEnd
extractDateTimeMonthEnd
extractDateTimeDayEnd
extractDateTimeHourEnd
extractDateTimeMinuteEnd
extractDateTimeSecondEnd
extractedDateTimeNumberOfSubsets
doExtractDateTime   must be set to 1 to extract

bufr_filter to extract subsets in a time range

To do the actual selection don't forget to use

> **set doExtractDateTime=1**;

Remarks

The full start time and the full end time must be specified to actually do the extraction.

As with area selection, this feature works only with **compressed data**.

bufr_filter to extract subsets in a time range (exercise 7)

Using the file scatterometer.bufr and **bufr_filter**, get the number of subsets where seconds is in the interval [26,30]. How many subsets do you get? Write the result into a file, and using **bufr_dump** check that the subsets are within the time interval.

bufr_filter for simple thinning

With high resolution data, it may be needed to reduce the number of observations.

To allow thinning, the following keys are provided.

- **simpleThinningSkip**

- **doSimpleThinning**

```
# allows thinning of the BUFR file
set unpack=1;
set simpleThinningSkip=5; # take subsets 1,7,13
set doSimpleThinning=1;  # does the actual thinning
write;
```

It only works for compressed data.

bufr_filter for simple thinning (exercise 8)

Use the file scatterometer.bufr and **bufr_filter** to thin the observations taking observations 1,7,13 etc. Check your results with **bufr_dump**.

creating BUFR messages with bufr_filter

Using **bufr_filter** we can create new messages. To do so, we need:

- An input BUFR file.

- Set **compressedData** (1→ for compressed data, 0→ for uncompressed)

- Set the key **unexpandedDescriptors** to the list of descriptors of the new message.

creating BUFR messages with bufr_filter

Remarks

When we create a new message with **bufr_filter** by setting the **unexpandedDescriptors**, the library is using the input message only as a seed, to select the proper tables etc.

Once we set **unexpandedDescriptors** the section 3 is **fully set**. We can not change any key in section 3 after setting the **unexpandedDescriptors**. **The keys that affect the structure of the message (e.g. compressedData, delayedDescriptorReplicationFactor, bit maps etc.) should be set BEFORE setting unexpanded descriptors**.

If the input message has a very old table version (11 for example) the setting of **unexpandedDescriptors** doesn't work.

creating BUFR messages with bufr_filter (exercise 9a)

We can use **bufr_filter** to create a new BUFR message by setting the values of some keys. In particular, the **unexpandedDescriptors** keys allows us to create a new BUFR message.

1.-Set the **unexpandedDescriptors** key to the following list

{106002, 008002 ,104003 ,005002, 006002 ,010002, 012001}

As an input file you can use **synop.bufr**. This input file is used as a seed, to select the proper tables.

By setting the key **unexpandedDescriptors** we are actually creating a new message, with all the key values are set to MISSING (null).

2.-Check the original and the newly created messages with **bufr_dump**.

3.-Using a different rules file, print the following keys for the new message :

unexpandedDescriptors, expandedDescriptors, expandedAbbreviations

Creating BUFR messages with bufr_filter (exercise 9b)

4. Set some values to the message created in the previous practical for the second instance of **latitude, longitude, nonCoordinateHeight** and **airTemperature**.

5. How many subsets do you have in your file?

What if (exercise 10)

Sometimes we have to deal with old messages, or messages that use old tables. In this case, we may not get the right results.

Using the file **old_amdar.bufr** as an input and **bufr_filter**, try to create a new message with the following *unexpandedDescriptors*:

{106002, 008002, 104003, 005002, 006002, 010002,012001}

Does it work? What happened? Check with **bufr_dump** the **old_amdar.bufr** file.