ecCodes advanced features

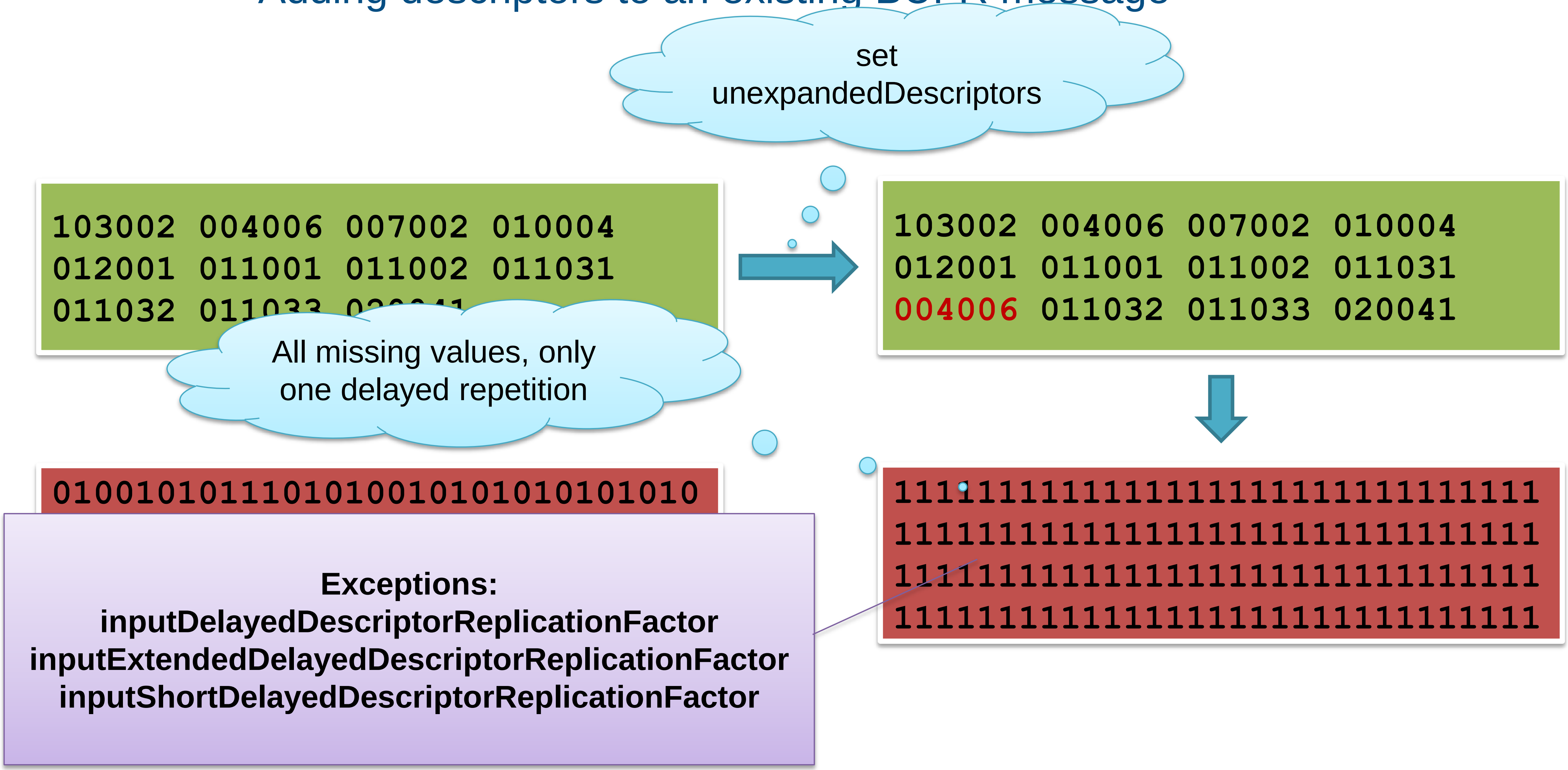
Computer User Training Course 2018

Shahram Najm

Development Section
Forecast Department

ECMWF

Adding descriptors to an existing BUFR message



set unexpandedDescriptors

```
103002 004006 007002 010004
012001 011001 011002 011031
011032 011033 020041
```

All missing values, only one delayed repetition

```
0100101011101010010101010101010
```

```
103002 004006 007002 010004
012001 011001 011002 011031
004006 011032 011033 020041
```

```
1111111111111111111111111111111
1111111111111111111111111111111
1111111111111111111111111111111
1111111111111111111111111111111
```

Exceptions:
inputDelayedDescriptorReplicationFactor
inputExtendedDelayedDescriptorReplicationFactor
inputShortDelayedDescriptorReplicationFactor

BUFR copy data

- Every time we set a list of unexpandedDescriptors the data section is re-built with missing values.

- How do I add some descriptors to an existing message?

  1. Read the BUFR message to be modified
  2. Clone to leave the original content untouched
  3. Get replication factors from original message
  4. Set replication factors in the clone
  5. Set new unexpandedDescriptors in the clone
  6. Use codes_bufr_copy_data to copy data from the original to the clone. This will copy only the keys from the original that are present in the clone taking into account the rank.
  7. Write the clone to a file

BUFR copy data practical

```
cd $SCRATCH
cp -r ~trx/ecCodes/2018/bufr_copy_data ./
cd bufr_copy_data
```

Add sequence 301128 (additional information on radiosonde ascent) to a TEMP BUFR message (unexpandedDescriptors=309052). Keep the same data as the original message.

1. Start from bufr_copy_data_skeleton.f90 or bufr_copy_data_skeleton.py
2. Use the input BUFR file temp_309052.bufr
3. Compare the output and original BUFR messages with bufr_compare

BUFR keys iterator

- Keys in a BUFR message are ordered and characterised by a position that we call “rank” (e.g. #4#pressure)
- Keys can have attributes (airTemperature->percentConfidence)
- ecCodes provides a keys iterator to go through all the keys including the rank and the attributes.
- See an example here: https://software.ecmwf.int/wiki/display/ECC/bufr_keys_iterator

High-level Pythonic interface

```
# See:
# https://software.ecmwf.int/wiki/display/ECC/High-level+Pythonic+Interface+in+ecCodes

import sys

from eccodes import BufrFile, BufrMessage

Filename = sys.argv[1]

with BufrFile(filename) as bufrs:

    for bufr in bufrs:

        bufr.unpack()

        for key in bufr.keys():

            print key, "=", bufr[key]
```

Making sample programs with bufr_dump

1. Get a BUFR file: `file.bufr`
2. Decide which language you prefer: **Fortran/Python/C** or **ecCodes filter** rules.
3. Use bufr_dump to generate code for encoding or decoding the message.
   - To have an example of encoding:

     ```
     bufr_dump –E”language” file.bufr
     ```

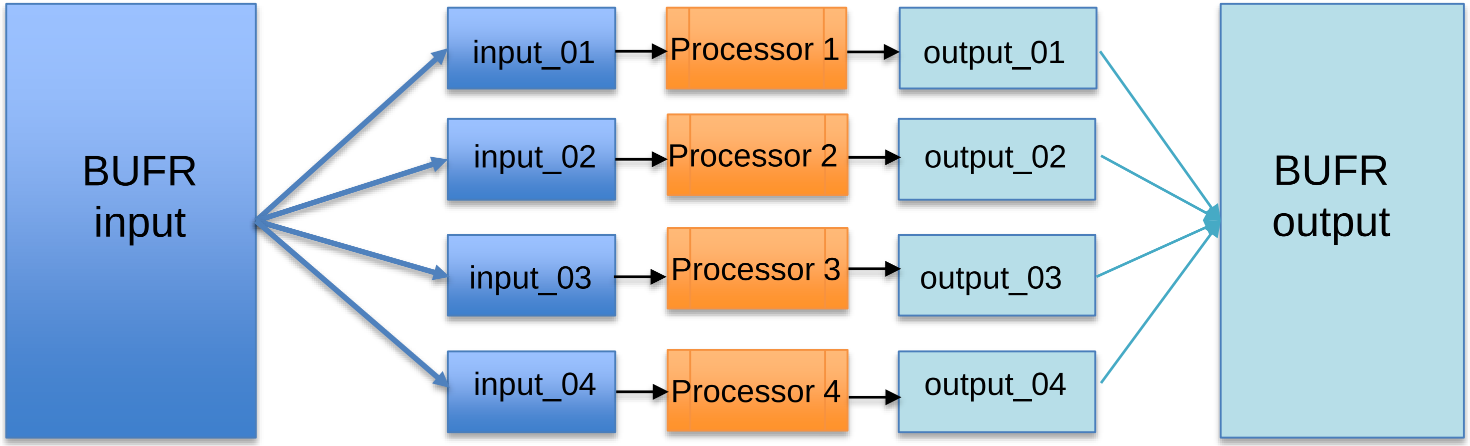
   - To have an example of decoding:

     ```
     bufr_dump –D”language” file.bufr
     ```

language = fortran
python
C
filter

Splitting files and processing in parallel

- There is another useful tool called codes_split_file which splits an input BUFR file into chunks of roughly the same size
- The output files are called *input*_01, *input*_02 etc
- Much faster than bufr_copy as no decoding of header is done
- Syntax: `codes_split_file [-v] nchunks input`
- Useful for parallelising operations where a large task is split into smaller ones which can be run on different processes


BUFR input
input_01
input_02
input_03
input_04
Processor 1
Processor 2
Processor 3
Processor 4
output_01
output_02
output_03
output_04
BUFR output