BUFR with ecCodes extra features

GTS and METAR decoding

Enrico Fucile

enrico.fucile@ecmwf.int

What is ecCodes?

E CODES = GRIB-API (GRIB decoder) + BUFR decoder + **GTS headers decoder**

+ METAR decoder + ???? decoder

GTS header

http://www.wmo.int/pages/prog/www/ois/Operational_Information/Publications/WMO_386/AHLsymbols/AHLsymbols_en.html

T1 T2 A1 A2 ii CCCC YYGGgg (BBB)

T1 **alpha** character that designates the general code form of the contents of the bulletin (Coded or plain text).
T2 **alpha** characher that designates the data type.

A1 **alpha** character that designates the geographical area the content of the bulletin covers.
A2 **alpha** character that designates the geographical area, or may define the forecast period.

ii **numeric** set of two characters.

CCCC identification of the processing center that generated the bulletin.

YYGGgg day, hour, and minute the bulletin was prepared or the reporting time of the contained reports in it.

(BBB) optional group of **alpha** characters

GTS header decoding and encoding tools

gts_<tool> [options] gts_file [gts_file] … [output_gts]

gts_<tool> -h (help on how to use the command)

- to inspect the GTS headers: **gts_dump, gts_ls, gts_get**
- to copy selected GTS messages: **gts_copy**
- To dump the GTS headers: **gts_dump**
- To compare two GTS files: **gts_compare**
- A filter providing a basic language to inspect, change and select GTS messages: **gts_filter**

GTS Python interface

```
fh_in = open(fpath, 'r')
gts = codes_gts_new_from_file(fh_in)
gtsKeys = ['TT', 'AA', 'II', 'CCCC', 'YY', 'GG', 'gg', 'BBB']
for key in gtsKeys:
  value = codes_get(gts, key)
  print value,’=’,key
```

What is ecCodes?

E CODES = GRIB-API (GRIB decoder) + BUFR decoder + GTS headers decoder

+ **METAR decoder** + ???? decoder

METAR

METAR LBBG 041600Z 12012MPS 1400 BKN022 OVC050 M04/M07 Q1020=

- **LBBG** – ICAO airport code for Burgas Airport
- **041600Z** - time of the observation. Day 04 of the month, 1600 UTC
- **12012MPS** – wind direction is from 120°, wind speed of 12 m/s
- **1400** - prevailing visibility is 1,400 m
- **BKN022** - broken (over half the sky) cloud layer with its base at 2,200 ft (670 m) above ground level.
- **OVC050** - unbroken cloud layer (overcast) with its base at 5,000 ft (1,500 m) above ground level.
- **M04/M07** indicates the temperature is −4 °C and the dew point is −7 °C.
- **Q1020** indicates the current altimeter setting (QNH) is 1,020 hPa.

METAR decoding only tools

metar_<tool> [options] metar_file [metar_file] … [output_metar]

metar_<tool> -h (help on how to use the command)

- to inspect a METAR: **metar_dump, metar_ls, metar_get**
- to copy selected METAR messages: **metar_copy**
- To dump a METAR: **metar_dump**
- To compare two METAR files: **metar_compare**
- A filter providing a basic language to inspect, select METAR messages: **metar_filter**
- **All the keys are "read only" and therefore cannot be set. ecCodes provides only decoding for METAR**

METAR Python interface

```
fh_in = open(fpath, 'r')
metar = codes_metar_new_from_file(fh_in)
metarKeys = ['CCCC','day', 'hour', 'minute', 'windSpeed',
'windDirection','temperature','dewPointTemperature']
for key in metarKeys:
    value = codes_get(metar, key)
    print value,’=’,key
```