ecCodes

Shahram Najm
Development Section
Forecast Department

What is ecCodes?

ecCodes is a package developed by ECMWF which provides an application programming interface and a set of tools for decoding and encoding messages in the following formats:

- **WMO FM-92 GRIB edition 1 and edition 2**
- **WMO FM-94 BUFR edition 3 and edition 4**
- **WMO GTS abbreviated header**

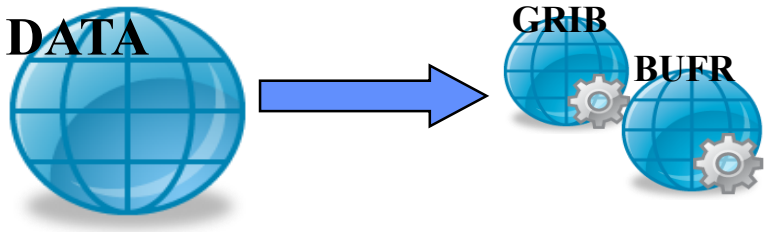
We will cover only GRIB and BUFR in this training course.

ecCodes: Formerly known as GRIB-API

- ecCodes is an evolution of GRIB-API (with additional support for BUFR)
- For GRIB encoding and decoding, ecCodes provides the same functionality as GRIB-API
- The current version of ecCodes is still beta but will shortly be production ready
- ecCodes will eventually replace GRIB-API and BUFRDC

GRIB and BUFR

GRIB and BUFR are both binary formats maintained by the World Meteorological Organization (WMO)



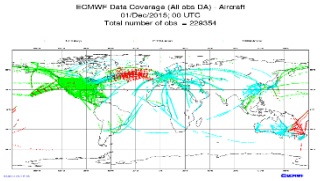
Acronyms:

GRIB => Initially "GRIdded Binary" but later expanded to "General Regularly-distributed Information in Binary form"

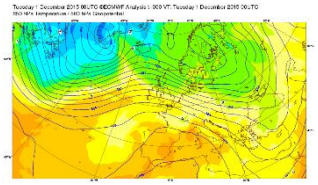
BUFR => "Binary Universal Form for the Representation of meteorological data"

GRIB and BUFR

- **BUFR (Binary Universal Form for the Representation of meteorological data) is**
  - **a flexible binary format**
  - **mainly used to encode in situ and satellite observations**
  - **can also represent forecast data**



- **GRIB (General Regularly-distributed Information in Binary form) is**
  - **designed to encode data produced by numerical weather prediction models**
  - **can also represent observations, but on a regularly distributed coverage**



GRIB and BUFR

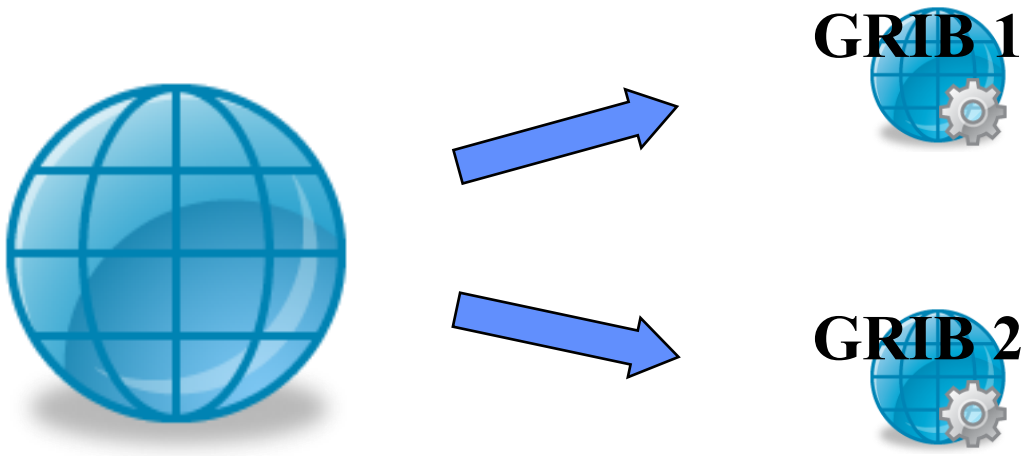
- In the first part of this course we are going to concentrate on GRIB and on Thursday we cover BUFR in detail
- But many of the features will apply equally to BUFR. This is because we aimed to provide a single programming interface to access both data formats in a consistent manner

WMO Binary Codes

- Fully describes GRIB and BUFR coding standards
- It is the only authoritative source for the WMO binary codes GRIB and BUFR
- It is publicly accessible on the WMO web site: http://www.wmo.int/pages/prog/www/WMOCodes.html
- A revision of the full manual is published every three years
- A new version of the tables which are part of the manual is released externally twice a year
- Latest version of the tables is accessible in several formats from the WMO web site: http://www.wmo.int/pages/prog/www/WMOCodes/WMO306_vI2/LatestVERSION/LatestVERSION.html

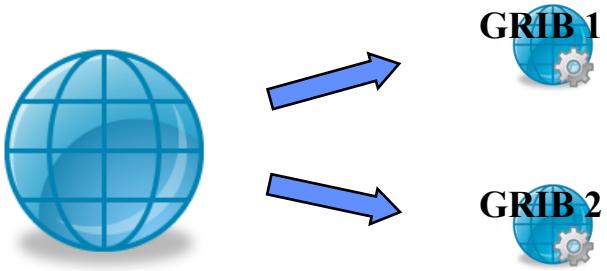
GRIB edition 1 vs. 2

- Two different versions of the GRIB coding standard are available at the moment (edition 1 and 2)
- The coding principles in both editions are similar, but their implementation is very different



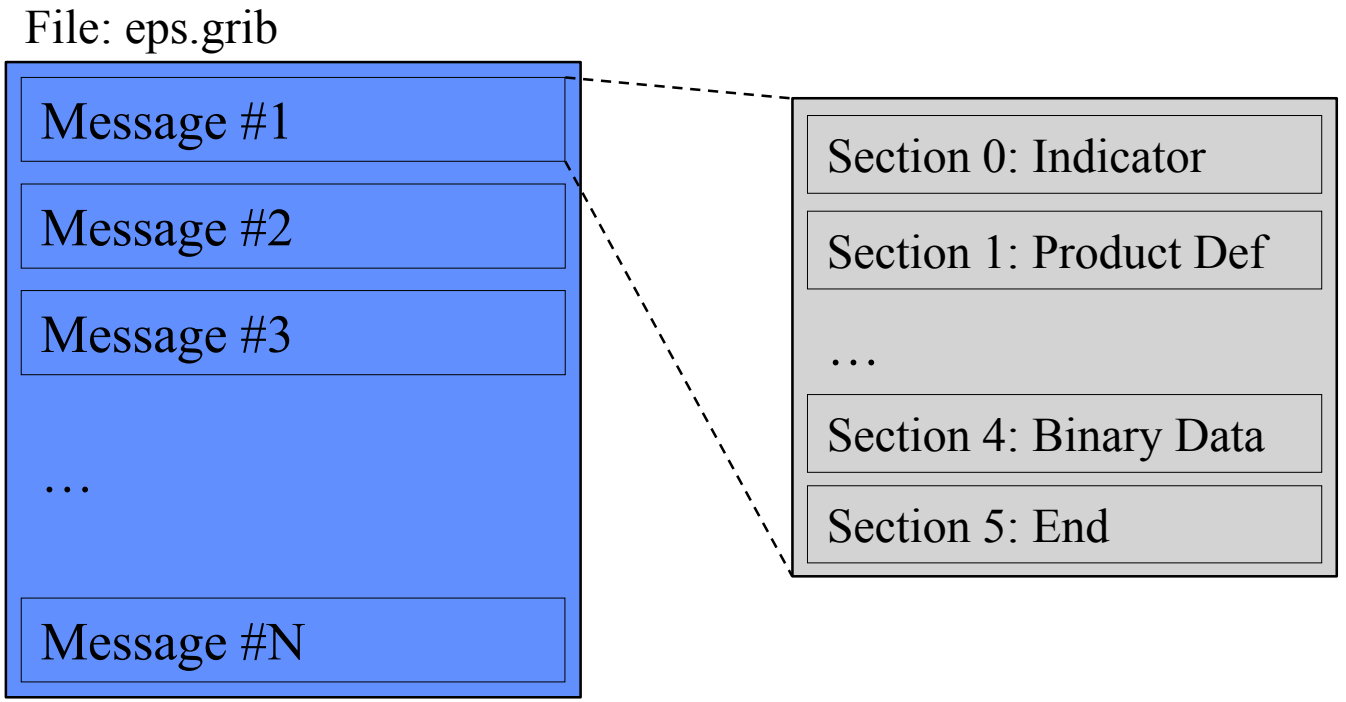
GRIB edition 1 vs. 2

- Different structure. They both have sections (with different meaning)
- GRIB 2 is more flexible because of its template/table structure
- In GRIB 2 several variables are defined with more precision (e.g. angles are in micro-degrees)
- In GRIB 2 the description of the data (parameter, time, statistics,…) is more complex and is template/table based (prone to become even more complex)



GRIB Structure

- A file may contain one or more GRIB messages
- Each message contains several sections
- Note: A file can contain a mix of editions 1 and 2



GRIB 1 vs. 2: Different Sections

GRIB 1

SECTION 0 Indicator

SECTION 1 Product definition

SECTION 2 Grid Description

SECTION 3 Bitmap

SECTION 4 Binary Data

SECTION 5 End (7777)

GRIB 2

SECTION 0 Indicator

SECTION 1 Identification

SECTION 2 Local Use

SECTION 3 Grid Definition

SECTION 4 Product Definition

SECTION 5 Data Representation

SECTION 6 Bitmap

SECTION 7 Binary Data

SECTION 8 End (7777)

GRIB 1 vs. 2: Legacy

- Please note that GRIB edition 1 is a legacy WMO Code which is not accepted for GTS exchange anymore
- New versions of the tables/templates are only issued for GRIB edition 2

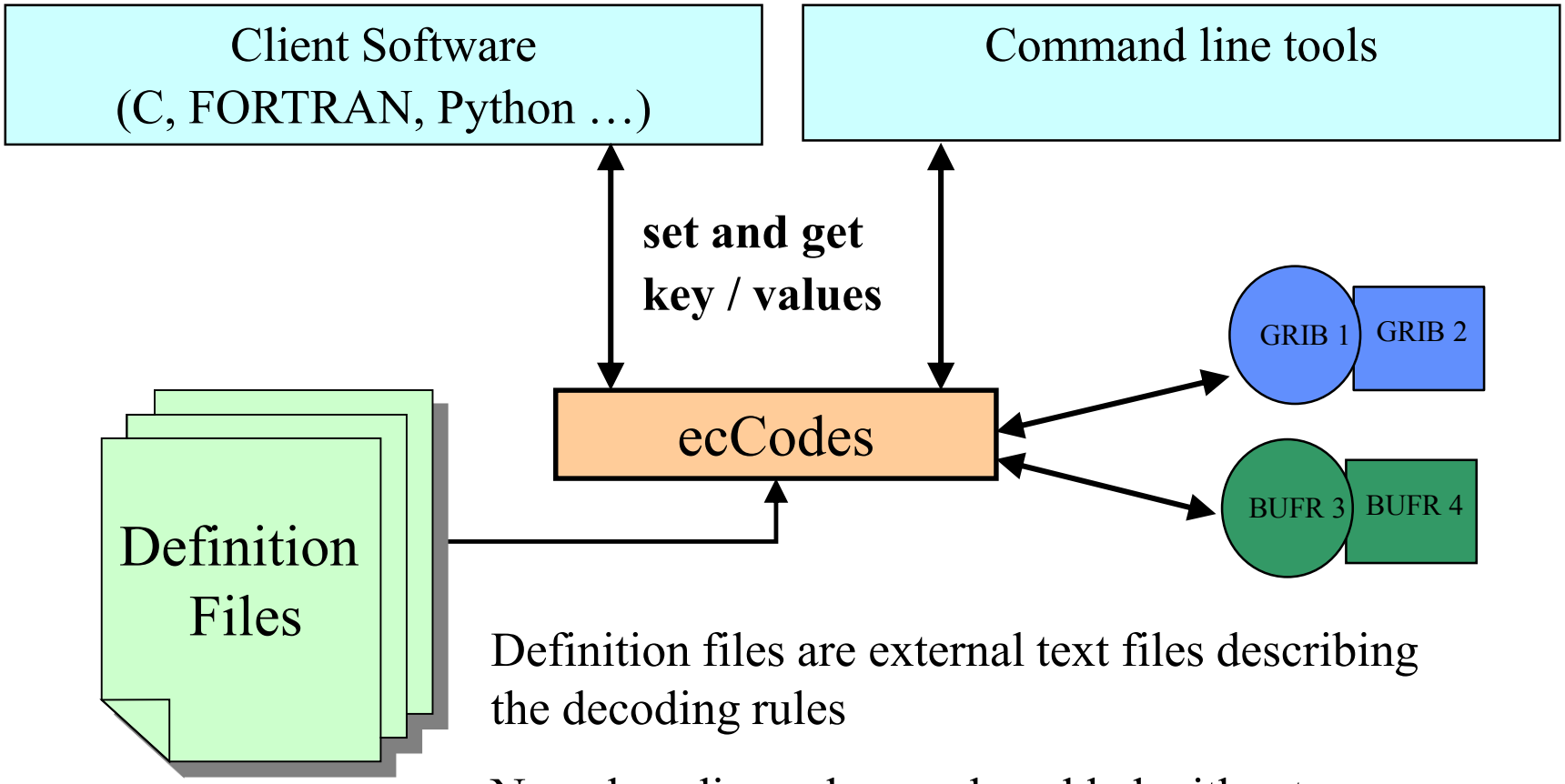
ecCodes: requirements

- ecCodes main requirement is to decode/encode both editions with the SAME function calls
- ecCodes has to be flexible enough to be easily updated with new template and tables
- ecCodes should hide the binary layer of the message, providing the user a higher level of access
- ecCodes must provide a way to convert data between the different editions
- ecCodes provides interfaces for Fortran, C and Python

ecCodes vs. Older software

- Previous Fortran-based packages provided array-based access to the message.
  - ksec2(2) => Number of points along a parallel
  - ksec2(3) => Number of points along a meridian
  - …
- ecCodes provides a key/value based approach
  - NumberOfPointsAlongAParallel => Number of points along a parallel
  - NumberOfPointsAlongAMeridan => Number of points along a meridian
  - …

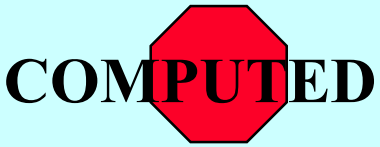
ecCodes: Design



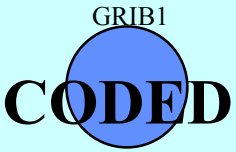
Definition files are external text files describing the decoding rules

New decoding rules can be added without rebuilding ecCodes

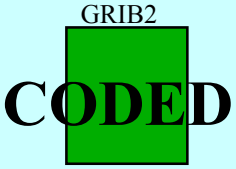
GRIB 1 vs. 2



latitudeOfFirstGridPointInDegrees=40

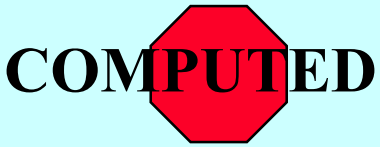


latitudeOfFirstGridPoint=40000



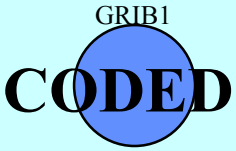
latitudeOfFirstGridPoint=40000000

GRIB 1 vs. 2

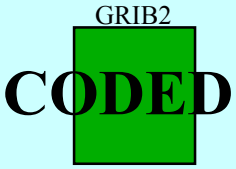


gridType=reduced_gg

gg = Gaussian Grid



dataRepresentationType=4
numberOfPointAlongAParallel=MISSING
ijDirectionIncrementGiven=0
pl ={…}



gridDefinitionTemplateNumber=40
numberOfPointsAlongAParallel=MISSING
iDirectionIncrementGiven=0
iDirectionIncrement=MISSING
pl={…}

ecCodes: available interfaces

- C native interface exposes all the functionalities (the engine itself is written in C)
- Fortran 90 provides an easy access to the main functionalities of the library from Fortran 90
- Python provides access to the C functions from Python
- Tools provide a command line interface to the API
- All the interfaces provide a way to set/get the same key/values pairs from the messages

ecCodes: available interfaces (C)

```
h = codes_handle_new_from_file(context,in,PRODUCT_GRIB, &err);
codes_get_double(h, "latitudeOfFirstGridPointInDegrees", &lat1);
codes_set_long(h, "centre", centre);
codes_set_string(h, "date", date, &len);
codes_handle_delete(h);
```

```
h = codes_handle_new_from_file(context,in,PRODUCT_BUFR, &err);
codes_get_long(h, "stationNumber",  &stationNum);
codes_get_double(h, "airTemperatureAt2M", &airTemp);
codes_set_long(h, "bufrHeaderCentre", 222);
codes_handle_delete(h);
```

Note: The functions with the “grib_” prefix are also supported

ecCodes: available interfaces (Fortran 90)

```
call codes_new_from_file(ifile, igrib, CODES_PRODUCT_GRIB, iret)
call codes_get(igrib, 'latitudeOfFirstGridPointInDegrees', lat1)
call codes_set(igrib, 'centre', centre)
call codes_set(igrib, ‘date', ’20070212’)
call codes_release(igrib)
```

```
call codes_new_from_file(ifile, ibufr, CODES_PRODUCT_BUFR, iret)
call codes_get(ibufr, 'stationNumber', stationNumber)
call codes_get(ibufr, 'airTemperatureAt2M', airTemp)
call codes_set(ibufr, 'bufrHeaderCentre', 222)
call codes_release(ibufr)
```

Note: The functions with the “grib_” prefix are also supported

ecCodes: available interfaces (Python)

```
gid = codes_new_from_file(f, CODES_PRODUCT_GRIB)
lat = codes_get(gid, 'latitudeOfFirstGridPointInDegrees')
codes_set(gid, 'centre', centre)
codes_set(gid, 'date', date)
codes_release(gid)
```

```
bid = codes_new_from_file(f, CODES_PRODUCT_BUFR)
stationNumber = codes_get(bid, 'stationNumber')
airTemp = codes_get(bid, 'airTemperatureAt2M')
codes_set(bid, 'bufrHeaderCentre', 222)
codes_release(bid)
```

Note: The functions with the “grib_” prefix are also supported

ecCodes: available interfaces (tools)

```
grib_get –p latitudeOfFirstGridPointInDegrees input.grib
grib_set –s centre=ecmf,date=20070212 input.grib out.grib

bufr_get -p bufrHeaderCentre,typicalMonth input.bufr
bufr_set -s typicalMonth=12 input.bufr out.bufr
```

Help and Support

- **For issues, bugs and requests:**
  Software.Support@ecmwf.int

- **Wiki:**
  https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home

- **Please use the Forums for general discussions:**
  https://software.ecmwf.int/wiki/display/ECC/Forums

☺

Questions ?