ecCodes BUFR encoding

Fortran 90 and Python API - part 1

Marijana Crepulja

Marijana.Crepulja@ecmwf.int

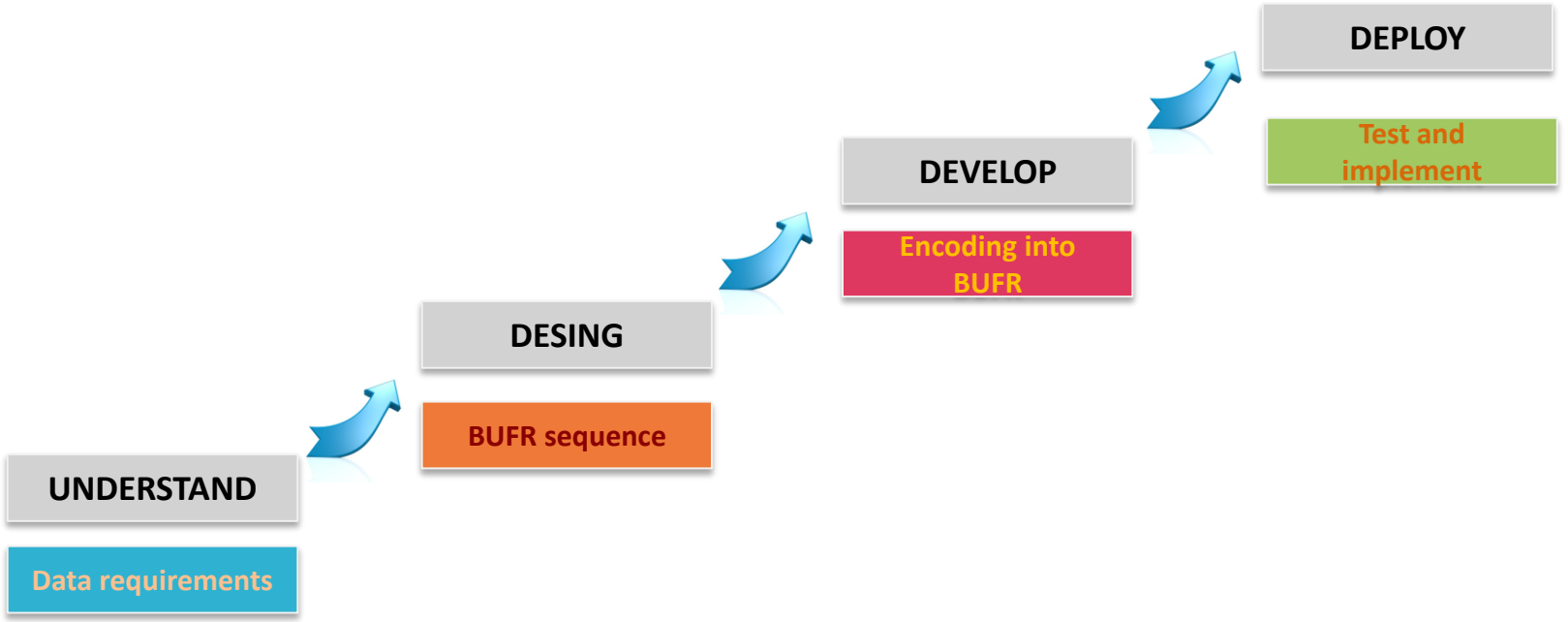
ECMWF

Introduction:

- Fortran 90 subroutines to encode BUFR data
- Python subroutines to encode BUFR data
- Practical examples

Encoding data into BUFR file

WMO Code-Tables combine authoritative definitions with encoding information unit of measure and precision, derived from ‘scale’, ‘reference value’ and ‘data width (bits)’



ecCodes BUFR F90 API

ecCodes Fortran BUFR encoding

- Getting a BUFR file structure from sample

Input arguments
Output arguments

```
call codes_new_from_sample (ibufr, 'sampleFile' , status)
```

ibufr: id of the message loaded in memory
sampleFile: name of the sample to get the BUFR message structure
status: CODES_SUCCESS if OK, integer value on error

sampleFile can be: 'BUFR4', 'BUFR3'

- Check status

```
if (status /= 0) then
    call codes_get_error_string (status, err_msg)
    print*,'ecCodesError: ', trim (err_msg), ' status=', status
    stop
 end if
```

ecCodes Fortran BUFR encoding scalars

- Set scalar value

call codes_set (ibufr, ‘key’, value, status)

Input arguments
Output arguments

ibufr: id of the message loaded in memory
key : variable name to be encoded
value: values of a variable to be encoded
status: CODES_SUCCESS if OK, integer value on error

value can be a integer(4), real(4), real(8) or a character.

Type of the values depend on the variable declaration.

```
integer   (kind=4)             ::  integer_values
real      (kind=8)             ::  real_values
character (len=string_size)    ::  string_values
```

ecCodes Fortran BUFR encoding arrays

- Set array values

call codes_set (ibufr, ‘key’, value, status)

Input arguments
Output arguments

ibufr: id of the message loaded in memory
key : variable name to be encoded
value: values of an array to be encoded
status: CODES_SUCCESS if OK, integer value on error

value can be a array of integer(4), real(4), real(8)

Type of the values depends on array declaration.

```
integer    (kind=4),        dimension(:), allocatable ::  integer_array
real       (kind=8),        dimension(:), allocatable ::  real_array
```

if(allocated(array)) deallocate(array)

ecCodes Fortran BUFR encoding string arrays

Input arguments
Output arguments

- Set string array values

call codes_set_string_array (ibufr, ‘key’, value, status)

ibufr: id of the message loaded in memory
key : variable name to be encoded
value: values of string array to be encoded
status: CODES_SUCCESS if OK, integer value on error

Type of the value depends on the variable declaration.

```
character (len=string_size), dimension(:), allocatable :: string_array
allocate   (string_array)

…

if(allocated(string_array)) deallocate(string_array)
```

ecCodes Fortran BUFR encoding header

Input arguments
Output arguments

- Setting BUFR tables

  call codes_set (ibufr, 'masterTablesVersionNumber', value)

  call codes_set (ibufr, 'localTablesVersionNumber', value)

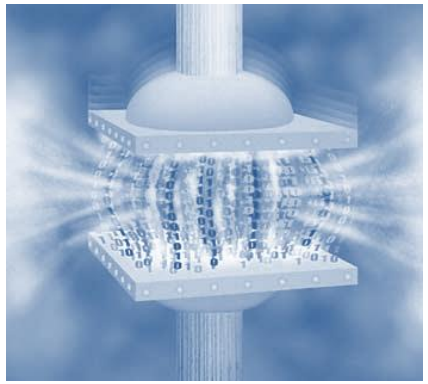
- Setting data compression

  call codes_set (ibufr, 'compressedData', value)

  value: 1 - compressed
  0 - uncompressed

- Setting number of subsets

  call codes_set (ibufr, 'numberOfSubsets', value)



ecCodes Fortran BUFR encoding data structure

Input arguments
Output arguments

- Setting unexpanded descriptors with known sequence

```
unexpandedDescriptors = 309052

call codes_set (ibufr, 'unexpandedDescriptors', unexpandedDescriptors)
```

- Setting array of the unexpanded descriptors

```
unexpandedDescriptors =
(/301051,4006,7002,10004,12001,11001,11002,11031,11032,11033,20041/)

call codes_set (ibufr, 'unexpandedDescriptors', unexpandedDescriptors)
```

ecCodes Fortran BUFR encoding

- Open file in write mode
  call codes_open_file (ifile, filename, mode, status)

Input arguments
Output arguments

ifile: object of the opened file to be used in all the file functions.
filename: name of the file to be open
mode: open mode can be 'r' (read) or 'w' (write)
status: CODES_SUCCESS if OK, integer value on error

- Set pack data section
  call codes_set(ibufr, ‘pack’, 1)
- Write encoded data into a file and close
  call codes_write(ibufr, filename)
- Close the file
  call codes_close(filename)
- Release the message
  call codes_release(ibufr)

ecCodes Fortran BUFR encoding

```
call codes_open_file(bufrout, outfile_name, 'w')

  call codes_bufr_new_from_samples(ibufr,'BUFR4',status)

  if (status/=CODES_SUCCESS) then
      print *,' ERROR creating BUFR from sample `
      stop 1
  endif

  call codes_set(ibufr,'masterTablesVersionNumber',27)

  call codes_set(ibufr,'unexpandedDescriptors',309052)

  call codes_set(ibufr,'shipOrMobileLandStationIdentifier','Reading')

  call codes_set(ibufr,'pack',1)

  call codes_write(ibufr,bufrout)

  call codes_release(ibufr)

  call codes_close_file(bufrout)
```

ecCodes BUFR Python API

Python BUFR encoding

- Getting a BUFR file structure from sample

  ibufr = codes_new_from_sample (‘sampleFile’)

  Input arguments
  Output arguments

  ibufr: id of the message loaded in memory
  sampleFile: name of the sample to get the BUFR message structure

  sampleFile can be: BUFR4, BUFR3

- Check status

```
if ibufr is None:
    print 'Not able to codes_bufr_new_from_sample'
    sys.exit()
```

Python BUFR encoding

Input arguments
Output arguments

- Set scalar value of integer, float or string

  codes_set (ibufr, ‘key’, value)

  ibufr: id of the message loaded in memory
  key : name of the variable key or shortName from Table B
  value: value to be encoded

- Set array values of integer, float or string

  codes_set_array (ibufr, ‘key’, values)

- Set by rank

  e.g. codes_set (ibufr, ‘#3#key’, values)

Python BUFR encoding header

Input arguments
Output arguments

- Setting BUFR tables

  codes_set(ibufr, 'masterTablesVersionNumber', value)

  codes_set(ibufr, 'localTablesVersionNumber', value)

- Setting data compression

  codes_set(ibufr, 'compressedData', value)

  value: 1 - compressed
  0 - uncompressed

- Setting number of subsets

  codes_set(ibufr, 'numberOfSubsets', value)

Python BUFR encoding

Input arguments
Output arguments

- Setting known BUFR sequence

  unexpandedDescriptors =309052

  codes_set(ibufr, 'unexpandedDescriptors', unexpandedDescriptors)

- Setting unexpanded descriptors

  unexpandedDescriptors = [301051,4006,7002,10004,12001,11001,11002,11031,11032,11033,20041]

  codes_set_array(ibufr, 'unexpandedDescriptors', unexpandedDescriptors)

Python BUFR encoding

- Open a file in write mode

  file = open(filename, mode)

  file: the file to be used in all file functions
  filename: name of the file to be open
  mode: open mode can be 'r' (read) or 'w' (write)

- Set pack for data section

  codes_set(ibufr, ‘pack’, 1)

- Write encoded data into a file

  codes_write(ibufr, file)

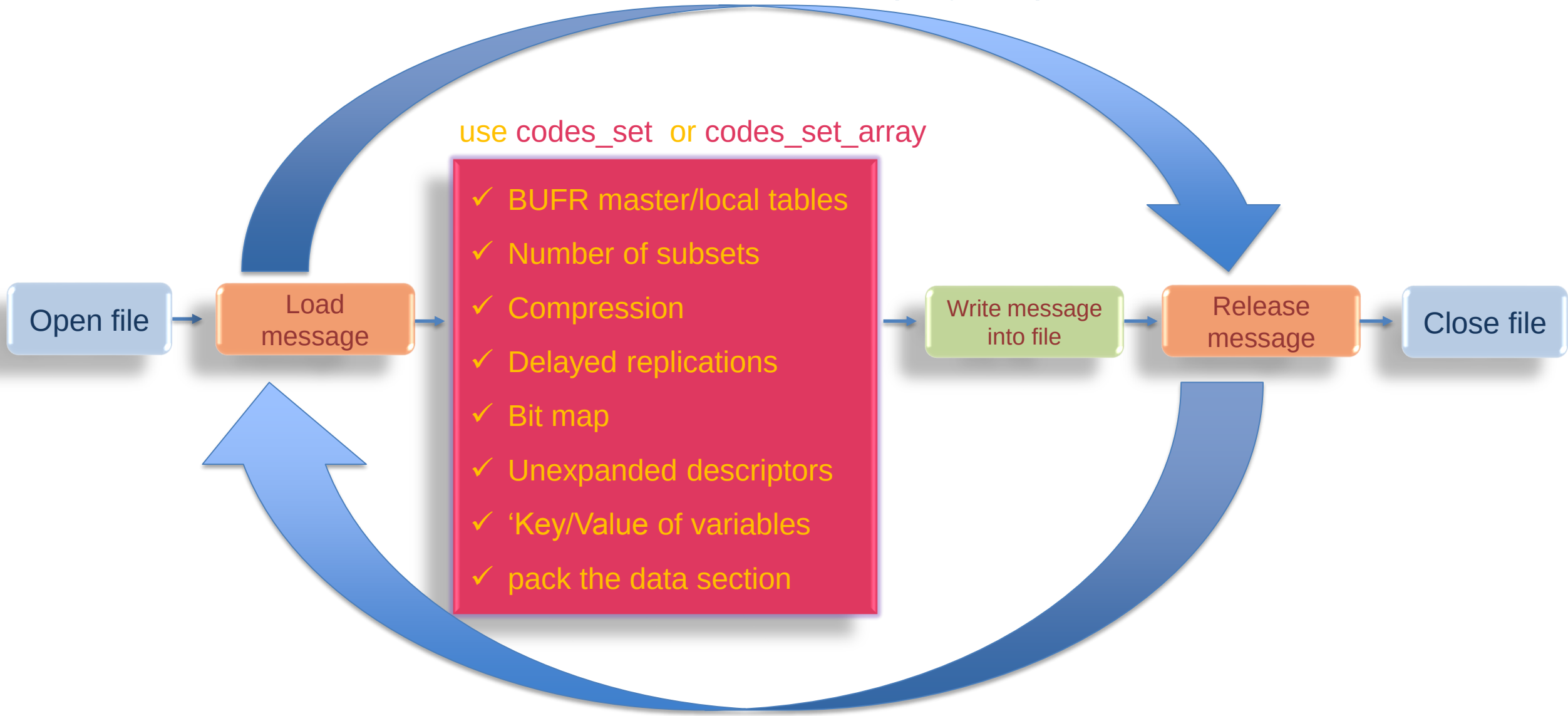
- Release the BUFR message

  codes_release(ibufr)

Input arguments
Output arguments

On error throw exception
CodesInternalError

Encode into BUFR: step by step



Practical

- Navigate to your $SCRATCH

  cd $SCRATCH

- Copy the material for the practical

  cp -r ~trx/ecCodes/2018/bufr_api_encode .

- There are subdirectories for F90 and python
  cd F90
  cd python

- The directories are named by practical number

- Have a look at the README

- Have fun

Practical 1: Encoding TEMP into a BUFR file

1. Open file in write mode
2. Create new BUFR message from sample
3. Set ‘masterTablesVersionNumber’ to 27
4. Set unexpandedDescriptors to known sequence 309052
5. Set values for:

   ‘shipOrMobileLandStationIdentifier‘
   ‘latitude‘,
   ‘longitude‘
   ‘height‘
6. ‘**pack**’ the data section
7. Write the created message into the file
8. Release the message
9. Close the output file
10. Check the new file with bufr_dump

codes_bufr_new_from_samples

codes_set

codes_set (ibufr,'pack',1)

codes_open_file

codes_write

codes_release

Practical 2: Encode multisubset BUFR message

- Create BUFR file with known sequence 309052 for TEMP
  - set ‘numberOfSubsets‘ to 2
  - set in the second subset values of

    ‘shipOrMobileLandStationIdentifier‘
    ‘latitude‘
    ‘longitude‘
    ‘height‘

- Explore
  - set the key and its value for variable of your choise
  - set the value of variable that is out of range
  - use bufr_dump to have a look at the message

Practical 3: Encoding compressed BUFR message

1. Open file in write mode
2. Create new BUFR message from sample
3. Set ‘numberOfSubsets’ to 2
4. Set BUFR message as compressed
5. Set unexpandedDescriptors array 301011, 001015, 301021
6. Set values for ‘year’, ‘month’, ‘day’
7. Set values for ‘latitude’, ‘longitude’, ‘stationOrSiteName’
8. ‘pack’ the data section
9. Write the created message into the file
10. Release the message
11. Close the output file
12. Check the new file with bufr_dump

codes_open_file

codes_bufr_new_from_samples

codes_set

codes_set (ibufr,'pack',1)

codes_write

codes_release

Practical 4: Encode BUFR message with replication

- Using array of unexpanded descriptors
  106002 008002 104003 005002 006002 010002 012001
- Set some values for the second and the last instance of
  - latitude (005002)
  - longitude (006002)
  - nonCoordinateHeight (010002)
  - airTemperature (012001)

codes_open_file

codes_bufr_new_from_samples

codes_set

codes_set (ibufr,'pack',1)

codes_write

codes_release

References

- ecCodes

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

- BUFR tables

  https://software.ecmwf.int/wiki/display/ECC/BUFR+tables


рахмат
danke
謝謝
ngiyabonga
Баярлалаа
spas
teşekkür ederim
спасибо
mersi
kia ora
barka
welalin
tack
faafetai lava
vinaka
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
kiitos
dankie
спасибі
blagodaram
thank you
gracias
nanni
nandri
dhanyavad
хвала
asante
manana
mauruuru
bedankt
hvala
köszönöm
dankon
aciu
obrigada
tenki
murakoze
bayarlalaa
gràcie
enkosi
akun
djiere dieuf
tau
mochchakkeram
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
chokrane
sobodi
děkuji
sukriya
kop khun krap
taiku
obrigado
sagolun
arigatô
takk
dakujem
trugarez
mèsi
ありがとう
tanemirt
grazie
didi madloba
najis tuke
terima kasih
rahmet
shukriya
mercé
мерси
kam sah hamnida
rahmat
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci