

# Community Tools (1) - PrepBUFR/BUFR: Basic tools, NCEP data tank, and Obsproc

Ruifang Li\*, Ming Hu\*\*, and Yongrun Guo\*

\*NCAR/MMM

\*\*Developmental Testbed Center

# Topics covered

---

- NCEP observation data
  - Processing and dataflow
  - PrepBUFR/BUFR types and data servers
  - Basic concept of BUFR/PrepBUFR
- Community BUFR/PrepBUFR basic tools
  - 10 examples are released with community package
  - The detailed instructions for these examples are in the Chapter 8 of the GSI User's Guide for Release 3
- Research on WRFDA Obsproc to generate the PrepBUFR file

# NCEP observation data

---

Processing and dataflow

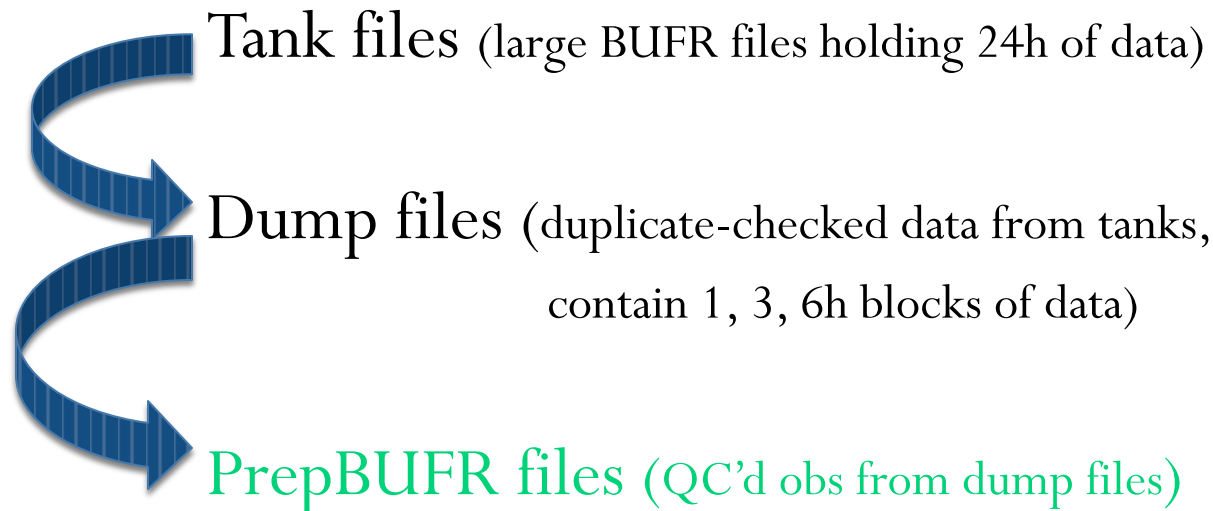
PrepBUFR/BUFR types and data servers

Basic concept of BUFR/PrepBUFR

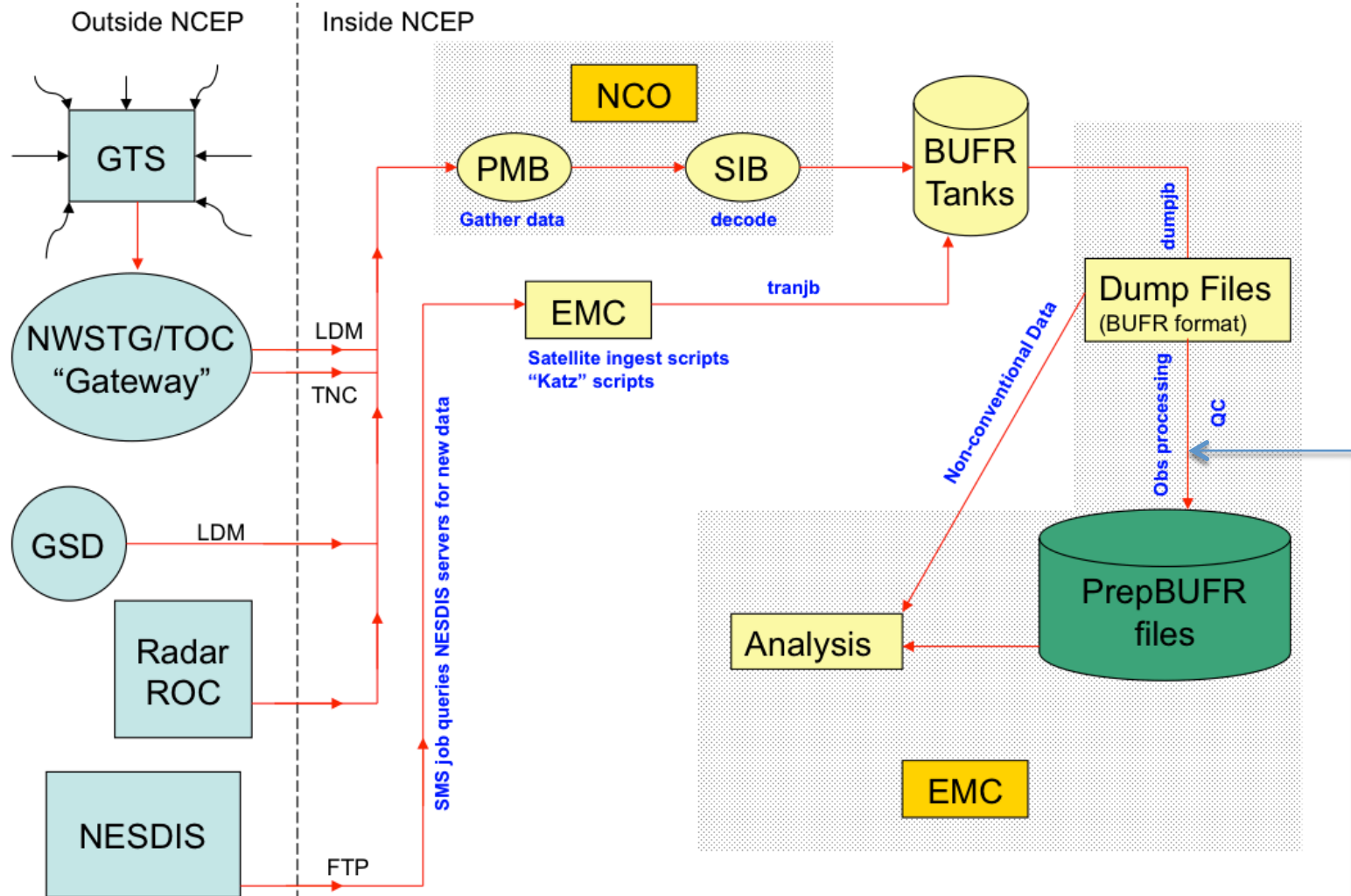
# Observation processing and dataflow at NCEP

---

- Managed jointly by NCEP Central Operations (NCO) and EMC
- Relies on NCEP BUFRLIB software
- Three stages:



# Data processing system at NCEP



Dennis Keyser's website on PREPBUFR PROCESSING AT NCEP:  
[http://www.emc.ncep.noaa.gov/mmb/data\\_processing/prepbufr.doc/document.htm](http://www.emc.ncep.noaa.gov/mmb/data_processing/prepbufr.doc/document.htm)

\*\* Diagram originally developed by Shelley Melchior \*\*



Developmental Testbed Center

# Operation BUFR/PrepBUFR files

- File name convention
  - gdas1.t00z.prepbufr.nr
  - gfs.t00z.gpsro.tm00.bufr\_d
  - ndas.t18z.lbamub.tm03.bufr\_d
  - nam.t00z.aircar.tm00.bufr\_d.nr
- Data coverage and cut off time
  - GDAS: covers global, latest 6 hours data
  - GFS: covers global, 2:45 hours data
  - NDAS: covers North America, longer cut off time than NAM
  - NAM: covers North America, shorter cut off time comparing to others

1<sup>st</sup> section: operation system

2<sup>st</sup> section: analysis time

3<sup>st</sup> section: data type

4<sup>st</sup>, 5<sup>st</sup> sections:

nr: non-restricted data

bufr\_d: bufr format

tm\*: indicate the catch up  
cycle analysis time,

=0 analysis time = 2<sup>st</sup> section

>0 analysis time = 2<sup>st</sup> section - 4<sup>st</sup> section

For example:

ndas.t18z.lbamub.tm03.bufr\_d

has analysis time=18z-03z=15z

See GSI User's Guide Section 8.2

# Operation BUFR/PrepBUFR files

---

- NCEP BUFR/PrepBUFR files are available for community users
- Resources listed in User's Guide Section 8.3
  - NCEP NOMADS Site:  
BUFR/PrepBufr for GDAS (Global) - 1 month buffer:  
<http://nomads.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/>  
BUFR/PrepBufr for NDAS (North America) - 1 month buffer:  
<http://nomads.ncep.noaa.gov/pub/data/nccf/com/nam/prod/>
  - NCDC NOMADS Site:  
BUFR/PrepBufr for GDAS (Global) - archive starting May 2007:  
<http://nomads.ncdc.noaa.gov/data/gdas/>
  - NCAR/CISL Research Data Archive (RDA) Site:  
**DS337.0**: NCEP ADP Global Upper Air and Surface Observations (PrepBUFR and NetCDF PB2NC Output) - archive starting May 1997:  
<http://dss.ucar.edu/datasets/ds337.0/>



# What is BUFR/PrepBUFR

---

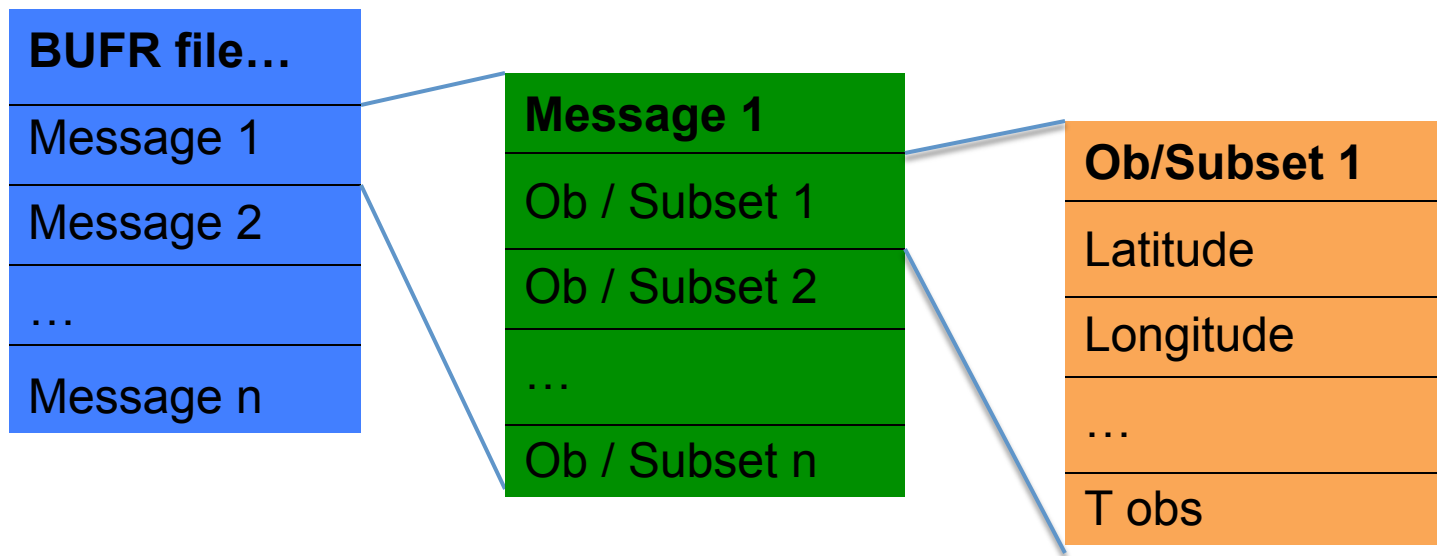
- **B**inary **U**niversal **F**orm for the **R**epresentation of meteorological data (BUFR)
  - BUFR is a “self-descriptive” table driven code form
    - The form and content of the data contained within a BUFR message are described within the BUFR message itself
  - BUFR is one of the code forms WMO recommends for the representation and exchange of observational data
- 
- **PrepBUFR is the NCEP term for “prepared” or QC’d data in BUFR format (NCEP convention/standard)**
  - **PrepBUFR file is still a BUFR file**



# BUFR/PrepBUFR file structure

A bit of terminology:

BUFR files (including “PrepBUFR” files) contain “messages”. Each message contains “subsets.” Each subset contains a meteorological observation.



# BUFR/PrepBUFR tools

---

In this section, we will learn how to:

- encode and decode a simple BUFR file
- append a data to an existing BUFR file
- understand BUFR table
- encode, decode, and append GSI PrepBUFR files

# All tools based on NCEP BUFRLIB

---

- There is detailed, clearly-explained, very useful BUFRLIB documentation written by NCO's Jeff Ator:

<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/>

- Quality reference for:
  - Purposes/specific uses of major BUFRLIB routines
  - BUFRLIB routine arguments
- The BUFRLIB requires BUFR (or **PrepBUFR**) files to be FORTRAN-blocked before they are used by BUFRLIB
  - Almost always, any **BUFR** file is already blocked for you and you can use the BUFRLIB routines directly.
  - If your **BUFR** file is not blocked, you can use the NCEP cwordsh utility to block it (<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/toc/cwordsh/> ).

# Encode

## (write observations to a BUFR file)

---

A simplest case:

One temperature observation

See GSI User's Guide Section 8.1.2



```

program bufr_encode_sample  (34 Lines)
!
! example of writing one value into a bufr file
!
implicit none

```

```

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

```

```

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

```

```

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

```

```

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out,msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call clobf(unit_out)

```

```

end program

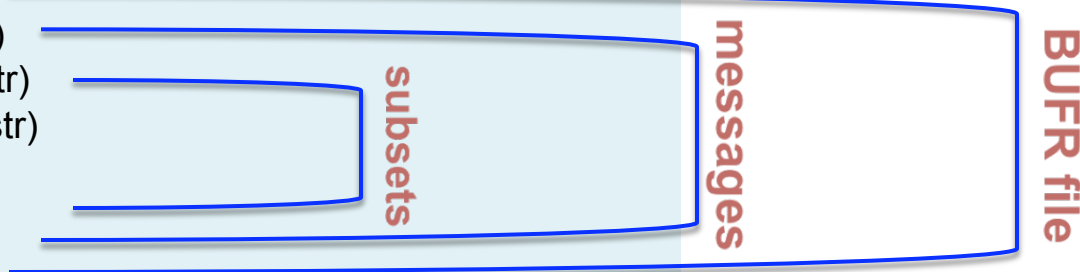
```

"A **BUFR file** contains one or more **BUFR messages**, each containing one or more **BUFR data subsets**, each containing one or more BUFR data values

Define mnemonics,  
data arrays, and  
assign obs value

Open BUFR tables file

Open BUFR file



```

program bufr_encode_sample  (34 Lines)
!
! example of writing one value into a bufr file
!
implicit none

```

```

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

```

```

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

```

```

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

```

```

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out, msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

```

```

end program

```

## Define mnemonics

String of blank-separated **mnemonics** associated with data array

Mnemonics:

XOB: Longitude  
YOB: Latitude  
DHR: obs time – cycle time  
TOB: temperature

## Setup data

Data written to subset

## Set date

```
program bufr_encode_sample (34 Lines)
!  
! example of writing one value into a bufr file  
!  
implicit none
```

```
character(80):: hdst='XOB YOB DHR'  
character(80):: obstr='TOB'  
real(8) :: hdr(3),obs(1,1)
```

```
character(8) msgtype  
integer :: unit_out=10,unit_table=20  
integer :: idate,iret
```

```
! set data values
```

```
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
```

```
obs(1,1)=287.15
```

```
idate=2008120100 ! YYYYMMDDHH
```

```
msgtype='ADPUPA' ! upper-air reports
```

```
! encode
```

```
open(unit_table,file='prepobs_prep.bufrtable')
```

```
open(unit_out,file='sample.bufr',action='write',form='unformatted')
```

```
call datelen(10)
```

```
call openbf(unit_out,'OUT',unit_table)
```

```
call openmb(unit_out, msgtype,idate)
```

```
call ufbint(unit_out,hdr,3,1,iret,hdst)
```

```
call ufbint(unit_out,obs,1,1,iret,obstr)
```

```
call writsb(unit_out)
```

```
call closmg(unit_out)
```

```
call closbf(unit_out)
```

```
end program
```

YYYYMMDDHH instead of  
YYMMDDHH

# BUFR file

```
program bufr_encode_sample (34 Lines)
!  
! example of writing one value into a bufr file  
!
```

```
implicit none
```

```
character(80):: hdst='XOB YOB DHR'  
character(80):: obstr='TOB'  
real(8) :: hdr(3),obs(1,1)
```

```
character(8) msgtype  
integer :: unit_out=10,unit_table=20  
integer :: idate,iret
```

```
! set data values  
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1  
obs(1,1)=287.15  
idate=2008120100 ! YYYYMMDDHH  
msgtype='ADPUPA' ! upper-air reports
```

```
! encode  
open(unit_table,file='prepobs_prep.bufrtable')  
open(unit_out,file='sample.bufr',action='write',form='unformatted')  
call datelen(10)  
call openbf(unit_out,'OUT',unit_table)  
  call openmb(unit_out, msgtype,idate)  
  call ufbint(unit_out,hdr,3,1,iret,hdst)  
  call ufbint(unit_out,obs,1,1,iret,obstr)  
  call writsb(unit_out)  
  call closmg(unit_out)  
call closbf(unit_out)
```

```
end program
```

Open BUFR tables file  
(can find it under directory  
./fix)

Open BUFR file  
(unformatted binary file for  
write)

**OPENBF ( LUBFR, CIO, LUNDX )**  
Input arguments:  
**LUBFR**=INTEGER: Logical unit for  
BUFR file  
**CIO** = 'IN' or 'OUT' or 'APN'  
**LUNDX**=INTEGER: Logical unit for  
BUFR tables

**CLOSBF ( LUBFR )**  
LUBFR=INTEGER: Logical unit for  
BUFR file



```

program bufr_encode_sample  (34 Lines)
!
! example of writing one value into a bufr file
!
implicit none

```

```

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

```

```

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

```

```

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

```

```

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write' ,form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out, msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

```

```

end program

```

# Message

**ADPUPA:** UPPER-AIR (RAOB, PIBAL, RECCO, DROPS) REPORTS

**ADPSFC:** SURFACE LAND (SYNOPTIC, METAR) REPORTS

**OPENMB ( LUBFR, CSUBSET, IDATE )**

Input arguments:

**LUBFR=INTEGER:** Logical unit for BUFR file

**CSUBSET=CHAR\*(\*):** Table A mnemonic for type of BUFR message to be opened

**IDATE=INTEGER:** Date-time to be stored within Section 1 of BUFR message

**CLOSMG ( LUBFR )**

**LUBFR=INTEGER:** Logical unit for BUFR file

```

program bufr_encode_sample (34 Lines)
!
! example of writing one value into a bufr file
!
implicit none

```

```

character(80):: hdrstr='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

```

```

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

```

```

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

```

```

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out, msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdrstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

```

```

end program

```

## Data subsets

**UFBINT ( LUBFR, R8ARR, MXMN, MXLV, iret, CMNSTR )**

Input arguments:  
**CMNSTR**=CHAR\*(\*): String of blank-separated mnemonics associated with **R8ARR**

**R8ARR(MXMN, MXLV)=REAL\*8:**  
 Data values written to data subset

Output argument:  
**iret** = INTEGER Number of levels of data values written to data subset

**WRITSB ( LUBFR )**

Input argument:  
**LUBFR**=INTEGER; Logical unit for BUFR file

Write **data subset** from **memory** to BUFR file

# Decode

## (read observation from bufr file)

---

Read the data we just encoded

The code can read multi-message and multi-data subsets

See GSI User's Guide Section 8.1.1

## Encode

```

program bufr_encode_sample
!
! example of writing one value into a bufr file
!
implicit none

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write' &
,form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out,msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

end program

```

## Decode

```

program bufr_decode_sample
!
! example of reading observations from bufr
!
implicit none

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,10)

integer :: ireadmg,ireadsb
character(8) msgtype
integer :: unit_in=10
integer :: idate,iret,num_message,num_subset

! decode
open(unit_in,file='sample.bufr',action='read',form='unformatted')
call openbf(unit_in,'IN',unit_in)
call datelen(10)
num_message=0
msg_report: do while (ireadmg(unit_in,msgtype,idate) == 0)
num_message=num_message+1
num_subset = 0
write(*,'(I10,I4,a10)') idate,num_message,msgtype
sb_report: do while (ireadsb(unit_in) == 0)
num_subset = num_subset+1
call ufbint(unit_in,hdr,3,1,iret,hdst)
call ufbint(unit_in,obs,1,10,iret,obstr)
write(*,'(2I5,4f8.1)') num_subset,iret,hdr,obs(1,1)
enddo sb_report
enddo msg_report
call closbf(unit_in)

end program

```

## Decode

```
program bufr_decode_sample
!
! example of reading observations from bufr
!
implicit none

character(80):: hdst='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,10)

integer :: ireadmg,ireadsb
character(8) msgtype
integer :: unit_in=10
integer :: idate,iret,num_message,num_subset

! decode
open(unit_in,file='sample.bufr',action='read',form='unformatted')
call openbf(unit_in,'IN',unit_in)
call datelen(10)
num_message=0
msg_report: do while (ireadmg(unit_in,msgtype,idate) == 0)
  num_message=num_message+1
  num_subset = 0
  write*,'(I10,I4,a10)' idate,num_message,msgtype
  sb_report: do while (ireadsb(unit_in) == 0)
    num_subset = num_subset+1
    call ufbint(unit_in,hdr,3,1 ,iret,hdst)
    call ufbint(unit_in,obs,1,10,iret,obstr)
    write*,'(2I5,4f8.1)' num_subset,iret,hdr,obs(1,1)
  enddo sb_report
enddo msg_report
call closbf(unit_in)

end program
```

Read **message** and **subsets** from BUFR file into **memory**:

**IRET = IREADMG ( LUBFR, CSUBSET, IDATE )**

Input argument:

**LUBFR=INTEGER**: Logical unit for BUFR file

Output arguments:

**CSUBSET=CHAR\*(\*)**: Table A mnemonic for BUFR message

**IDATE=INTEGER**: Section 1 date-time for BUFR message

**IRET=INTEGER**: Return code:

0 = normal return

-1 = **no more** BUFR **messages** in BUFR file

**IRET = IREADSB ( LUBFR )**

Input argument:

**LUBFR=INTEGER**: Logical unit for BUFR file

Output arguments:

**IRET INTEGER** Return code:

0 = normal return

-1 = **no more** BUFR data **subsets** in current BUFR **message**

```

program bufr_decode_sample
!
! example of reading observations from bufr
!
implicit none

```

```

character(80):: hdsr='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,10)

integer :: ireadmg,ireadsb
character(8) msgtype
integer :: unit_in=10
integer :: idate,iret,num_message,num_subset

```

```

! decode
open(unit_in,file='sample.bufr',action='read',form='unformatted')
call openbf(unit_in,'IN',unit_in)
call datelen(10)
num_message=0
msg_report: do while (ireadmg(unit_in,msgtype,idate) == 0)
num_message=num_message+1
num_subset = 0
write(*,'(I10,I4,a10)') idate,num_message,msgtype
sb_report: do while (ireadsb(unit_in) == 0)
num_subset = num_subset+1
call ufbint(unit_in,hdr,3,1 ,iret,hdsr)
call ufbint(unit_in,obs,1,10,iret,obstr)
write(*,'(2I5,4f8.1)') num_subset,iret,hdr,obs(1,1)
enddo sb_report
enddo msg_report
call closbf(unit_in)

```

```

end program

```

"A **BUFR file** contains one or more **BUFR messages**, each containing one or more **BUFR data subsets**, each containing one or more BUFR data values

Define mnemonics,  
data arrays, and  
assign obs value

Open BUFR file

subsets

messages

BUFR file

# Append obs to existing prepbufr

---

Append one more temperature observation to the bufr file we just created

See GSI User's Guide Section 8.1.3

## Encode

```

program bufr_encode_sample
!
! example of writing one value into a bufr file
!
implicit none

character(80):: hdrstr='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

! set data values
hdr(1)=75.;hdr(2)=30.;hdr(3)=-0.1
obs(1,1)=287.15
idate=2008120100 ! YYYYMMDDHH
msgtype='ADPUPA' ! upper-air reports

! encode
open(unit_table,file='prepobs_prep.bufrtable')
open(unit_out,file='sample.bufr',action='write',
      ,form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',unit_table)
call openmb(unit_out, msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdrstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

end program

```

## Append

```

program bufr_append_sample

!sample of appending one observation into bufr file
implicit none
character(80):: hdrstr='XOB YOB DHR'
character(80):: obstr='TOB'
real(8) :: hdr(3),obs(1,1)

character(8) msgtype
integer :: unit_out=10,unit_table=20
integer :: idate,iret

! set data values
hdr(1)=85.0;hdr(2)=50.0;hdr(3)=0.2
obs(1,1)=300.0
idate=2008120101 ! YYYYMMDDHH
msgtype='ADPSFC' ! surface land reports

! get bufr table from existing bufr file
open(unit_table,file='prepobs_prep_app.bufrtable')
open(unit_out,file='sample.bufr',status='old',form='unformatted')
call openbf(unit_out,'IN',unit_out)
call dxdump(unit_out,unit_table)
call closbf(unit_out)

! append
open(unit_out,file='sample.bufr',status='old',form='unformatted')
call datelen(10)
call openbf(unit_out,'APN',unit_table)
call openmb(unit_out, msgtype,idate)
call ufbint(unit_out,hdr,3,1,iret,hdrstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call writsb(unit_out)
call closmg(unit_out)
call closbf(unit_out)

end program

```



## Append

```
program bufr_append_sample
!
```

```
! sample of appending one observation into bufr file
```

```
implicit none
```

```
character(80):: hdrstr='XOB YOB DHR'
```

```
character(80):: obstr='TOB'
```

```
real(8) :: hdr(3),obs(1,1)
```

```
character(8) msgtype
```

```
integer :: unit_out=10,unit_table=20
```

```
integer :: idate,iret
```

```
! set data values
```

```
hdr(1)=85.0;hdr(2)=50.0;hdr(3)=0.2
```

```
obs(1,1)=300.0
```

```
idate=2008120101 ! YYYYMMDDHH
```

```
msgtype='ADPSFC' ! surface land reports
```

```
! get bufr table from existing bufr file
```

```
open(unit_table,file='prepobs_prep_app.bufrtable')
```

```
open(unit_out,file='sample.bufr',status='old',form='unformatted')
```

```
call openbf(unit_out,'IN',unit_out)
```

```
call dxdump(unit_out,unit_table)
```

```
call closbf(unit_out)
```

```
! append
```

```
open(unit_out,file='sample.bufr',status='old',form='unformatted')
```

```
call datelen(10)
```

```
call openbf(unit_out,'APN',unit_table)
```

```
call openmb(unit_out, msgtype,idate)
```

```
call ufbint(unit_out,hdr,3,1,iret,hdrstr)
```

```
call ufbint(unit_out,obs,1,1,iret,obstr)
```

```
call writsb(unit_out)
```

```
call closmg(unit_out)
```

```
call closbf(unit_out)
```

```
end program
```

Appending requires the report structure (BUFR table) of the data you want to append fits the report structure in the existing file. So we use the following subroutine to retrieve BUFR table from the existing BUFR file:

***DXDUMP ( LUBFR, LDXOT )***

Input arguments:

**LUBFR:** INTEGER Logical unit for BUFR file

**LDXOT:** INTEGER Logical unit for output

BUFR tables file

# Test results (Basic Practice case 0):

---

**./bufr\_encode\_sample.exe**

This should generate a bufr file *sample.bufr*

**./bufr\_decode\_sample.exe**

This reads one observation from *sample.bufr*, and write out observation:

```
2008120100    1  ADPUPA
      1      1    75.0    30.0    -0.1    287.1
```

**./bufr\_append\_sample.exe**

Now, append a new observation to *sample.bufr*.

**./decode\_sample.exe**

Read *sample.bufr* and show two observations in it:

```
2008120100    1  ADPUPA
      1      1    75.0    30.0    -0.1    287.1
2008120101    2  ADPSFC
      1      1    85.0    50.0     0.2    300.0
```

# Examples for GSI BUFR/PrepBUFR files

Code name	Illustrated process function
<i>prepbufr_decode_all.f90</i>	<ul style="list-style-type: none"><li>• read BUFR table from an existing prepbufr file</li><li>• read all observation information used by GSI analysis from an existing prepbufr file.</li></ul>
<i>prepbufr_encode_surface.f90</i>	<ul style="list-style-type: none"><li>• write a surface observation into a new prepbufr file</li></ul>
<i>prepbufr_encode_upperair.f90</i>	<ul style="list-style-type: none"><li>• write a upper air observation into a new prepbufr file</li></ul>
<i>prepbufr_append_upperair.f90</i>	<ul style="list-style-type: none"><li>• read BUFR table from an existing prepbufr file</li><li>• append a upper air observation into an existing prepbufr file</li></ul>
<i>prepbufr_append_surface.f90</i>	<ul style="list-style-type: none"><li>• read BUFR table from an existing prepbufr file</li><li>• append a surface observation into an existing prepbufr file.</li></ul>
<i>prepbufr_append_retrieve.f90</i>	<ul style="list-style-type: none"><li>• read BUFR table from an existing prepbufr file</li><li>• append retrieved data into an existing prepbufr file.</li></ul>
<i>bufr_decode_radiance.f90</i>	<ul style="list-style-type: none"><li>• read BUFR table from an existing radiance bufr file</li><li>• read radiance data from an existing radiance bufr file.</li></ul>

- These files have **the same structure** and call the same BUFRLIB **subroutines/functions** as three simple examples
- The only difference is the **mnemonic lists** used in these files are much longer

# MNEMONIC in prepbufr\_decode\_all.f90

In GSI, *read\_prepbufr.f90* reads prepbufr file. The following mnemonic lists came from *read\_prepbufr.f90* and are used in PrepBUFR sample code like: prepbufr\_decode\_all.f90

```
integer, parameter :: mxmn=35, mxlv=250
character(80):: hdst='SID XOB YOB DHR TYP ELV SAID T29'
character(80):: obstr='POB QOB TOB ZOB UOB VOB PWO CAT PRSS'
character(80):: qcstr='PQM QQM TQM ZQM WQM NUL PWQ      '
character(80):: oestr='POE QOE TOE NUL WOE NUL PWE      '
real(8) :: hdr(mxmn), obs(mxmn,mxlv), qcf(mxmn,mxlv), oer(mxmn,mxlv)
```

```
call ufbint(unit_in,hdr,mxmn,1,iret,hdst)
call ufbint(unit_in,obs,mxmn,mxlv,iret,obstr)
call ufbint(unit_in,oer,mxmn,mxlv,iret,oestr)
call ufbint(unit_in,qcf,mxmn,mxlv,iret,qcstr)
```

# MNEMONIC used in prepbuf

String name	MNEMONIC	NUMBER	DESCRIPTION
hdstr	SID	001192	STATION IDENTIFICATION
	XOB	006002	LONGITUDE
	YOB	005002	LATITUDE
	DHR	004192	OBSERVATION TIME MINUS CYCLE TIME
	TYP	001193	PREPBUFR REPORT TYPE
	ELV	010194	STATION ELEVATION
	SAID	001007	SATELLITE IDENTIFIER (SATELLITE REPORTS ONLY)
obstr	POB	007192	PRESSURE OBSERVATION
	QOB	013192	SPECIFIC HUMIDITY OBSERVATION AFTER "VIRTMP" STEP - ALWAYS RECALCULATED FROM QUALITY CONTROLLED VIRTUAL TEMPERATURE DATA)
	TOB	012192	TEMPERATURE OBSERVATION(AFTER "PREPRO" STEP - REPORTED TEMP, EITHER SENSIBLE OR VIRTUAL DEPENDING UPON DATA TYPE AFTER "VIRTMP" STEP - VIRTUAL TEMPERATURE IF MOISTURE AVAILABLE, OTHERWISE SENSIBLE)
	ZOB	010196	HEIGHT OBSERVATION
	UOB	011003	U-COMPONENT WIND OBSERVATION
	VOB	011004	V-COMPONENT WIND OBSERVATION
	PWO	013213	TOTAL PRECIPITABLE WATER OBSERVATION
	CAT	001194	PREPBUFR DATA CATEGORY
	PRSS	010195	SURFACE PRESSURE OBSERVATION

# MNEMONIC used in prepbuf (continue)

qcstr	PQM	007193	PRESSURE (QUALITY) MARKER
	QQM	013193	SPECIFIC HUMIDITY (QUALITY) MARKER
	TQM	012195	TEMPERATURE (QUALITY) MARKER
	ZQM	010197	HEIGHT (QUALITY) MARKER
	WQM	011192	WIND (QUALITY) MARKER
	NUL		
	PWQ	013214	TOTAL PRECIPITABLE WATER (QUALITY)MARKER
oestr	POE	007197	PRESSURE OBSERVATION ERROR
	QOE	013198	RELATIVE HUMIDITY OBSERVATION ERROR
	TOE	012200	TEMPERATURE OBSERVATION ERROR
	NUL		
	WOE	011199	WIND OBSERVATION ERROR
	NUL		
	PWE	013219	TOTAL PRECIPITABLE WATER OBSERVATION ERROR

more information can be found in:

`./fix/prepobs_prep.bufrrtable`

Including:

Scale, Reference, Bite Number, and Units

# Understand BUFR table

---

The difference between simple sample cases and real cases is the data structure, which defined by BUFR table

# BUFR table

---

- Defining report structures in any kind of BUFR file (“PrepBUFR” files too!)
  - Report structures for various types of observations / subsets are defined by “NCEP BUFR Tables” when using the NCEP BUFRLIB software.
- In NCEP BUFR files, the tables defining the report structures are embedded at the top of the file
- Excellent reference for NCEP BUFR TABLES:  
<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/toc/dfbftab/>



# BUFR Table Example

## Message type

MNEMONIC	NUMBER	DESCRIPTION
ADPUPA	A48102	UPPER-AIR (RAOB, PIBAL, RECCO, DROPS) REPORTS
AIRCAR	A48103	MDCRS ACARS AIRCRAFT REPORTS

## Sequence (structure)

MNEMONIC	SEQUENCE
ADPUPA	HEADR SIRC {PRSLEVEL} <SST_INFO> <PREWXSEQ> {CLOUDSEQ}
ADPUPA	<CLOU2SEQ> <SWINDSEQ> <AFIC_SEQ> <TURB3SEQ>
HEADR	SID XOB YOB DHR ELV TYP T29 TSB ITP SQN PROCN RPT
HEADR	TCOR <RSRD_SEQ>

# BUFR Table Example

## MNEMONIC Definition

SID	001194	STATION IDENTIFICATI	
XOB	006240	LONGITUDE	
YOB	005002	LATITUDE	
DHR	004215	OBSERVATION TIME MINUS CYCLE TIME	
ELV	010199	STATION ELEVATION	
TYP	055007	PREPBUFR REPORT TYP	

## MNEMONIC Unit

MNEMONIC	SCAL	REFERENCE	BIT	UNITS
SID	0	0	64	CCITT IA5
XOB	2	-18000	16	DEG E
YOB	2	-9000	15	DEG N
DHR	3	-24000	16	HOURS
ELV	0	-1000	17	METER
TYP	0	0	9	CODE TABL

# BUFR Table Example

MNEMONIC	SEQUENCE
ADPUPA	HEADR SIRC {PRSLEVEL} <SST_INFO> <PREWXSEQ> {CLOUDSEQ}
ADPUPA	<CLOU2SEQ> <SWINDSEQ> <AFIC_SEQ> <TURB3SEQ>

HEADR	SID	XOB	YOB	DHR	ELV	TYP	T29	TSB	ITP	SQN
HEADR	PROCN	RPT	TCOR	<RSRD_SEQ>						

PRSLEVEL	CAT	<P__INFO>	<Q__INFO>	<T__INFO>	<Z__INFO>	<W__INFO>
PRSLEVEL		<DRFTINFO>				

P__INFO	[P__EVENT]		<P__BACKG>	<P__POSTP>
Q__INFO	[Q__EVENT]	TDO	<Q__BACKG>	<Q__POSTP>
T__INFO	[T__EVENT]	TVO	<T__BACKG>	<T__POSTP>
Z__INFO	[Z__EVENT]		<Z__BACKG>	<Z__POSTP>

P__EVENT		POB		PQM	PPC	PRC
Q__EVENT		QOB		QQM	QPC	QRC
T__EVENT		TOB		TQM	TPC	TRC
Z__EVENT		ZOB		ZQM	ZPC	ZRC

P__BACKG	POE	PFC	<PFC__MSQ>
Q__BACKG	QOE	QFC	<QFC__MSQ>
T__BACKG	TOE	TFC	<TFC__MSQ>
Z__BACKG		ZFC	<ZFC__MSQ>

P__POSTP	PAN	<PCLIMATO>	POETU
Q__POSTP	QAN	<QCLIMATO>	QOETU
T__POSTP	TAN	<TCLIMATO>	TOETU
Z__POSTP	ZAN	<ZCLIMATO>	

**No need to be intimidated!**

The fully expanded report structure may look confusing, but it all boils down to individual data values.

Expanding all the sequences, even if you need to do it by hand, will help you easily understand exactly what is contained within the report.



# WRFDA Obsproc

---

Research on Obsproc generating prepbufr file

# WRFDA Obsproc

---

- Obsproc is a community tool to ingest the *intermediate format (LITTLE\_R) OBS data file* and *prepare the OBS data file suitable for WRFDA needs (3DVAR, FGAT, 4DVAR, etc.)*
- Obsproc writes *LITTLE\_R* observation to ASCII data for WRFDA use. It also provides functionality to generate Prepbufr file. We run experiments in GSI to test this functionality.

# Experiment Setup

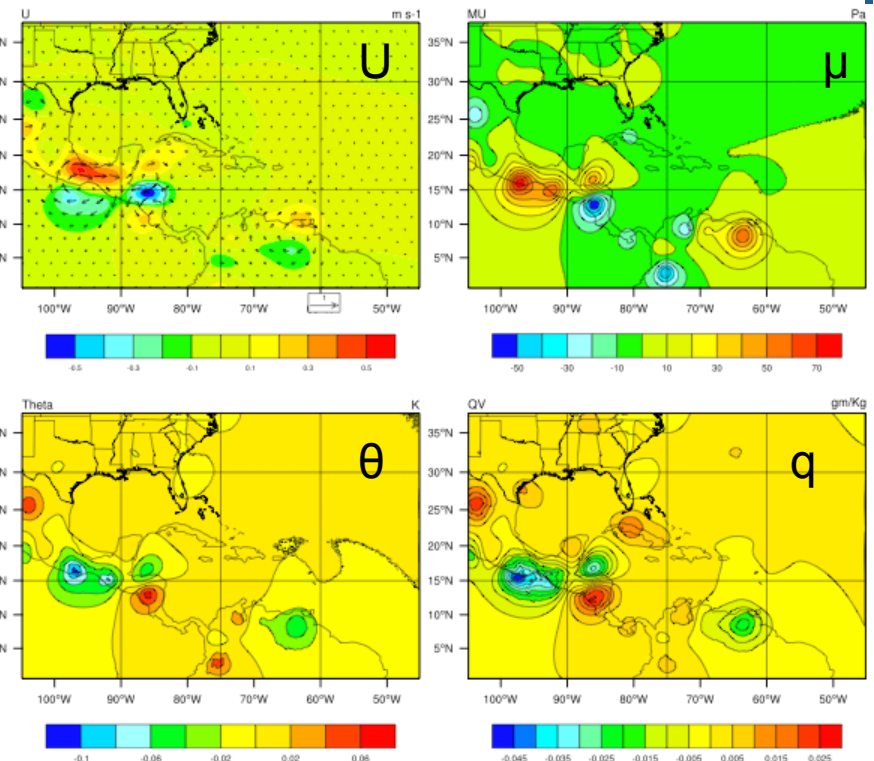
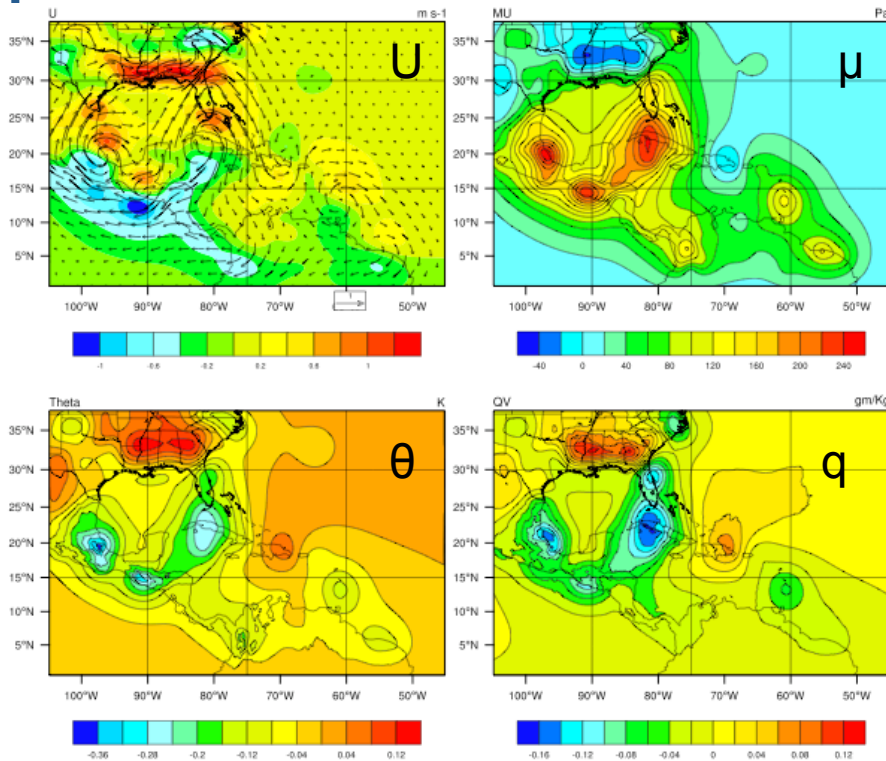
---

- Time: 12 UTC 15 August 2007
- Domain: T8-- 418x280x56,  $ds=15$  km,  
 $p_{top} = 1000$  Pa
- Observations:
  1. NCEP GDAS with modified *convinfo* file for only ADPSFC assimilated
  2. Obsproc ADPSFC
  3. NCEP GDAS with modified *convinfo* file for only ADPUPA assimilated
  4. Obsproc ADPUPA



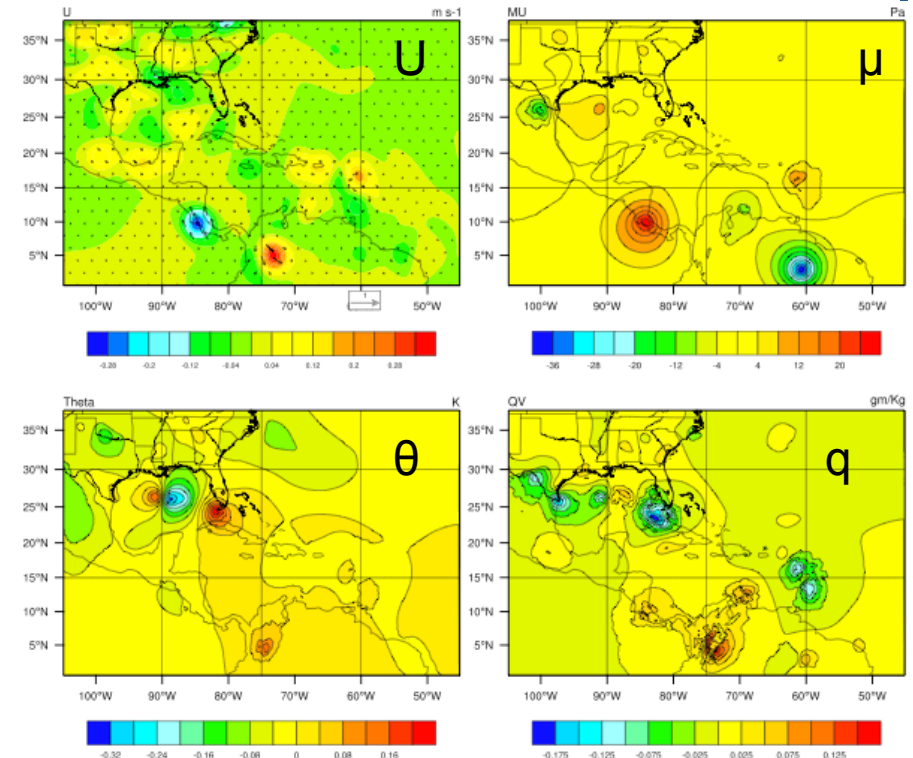
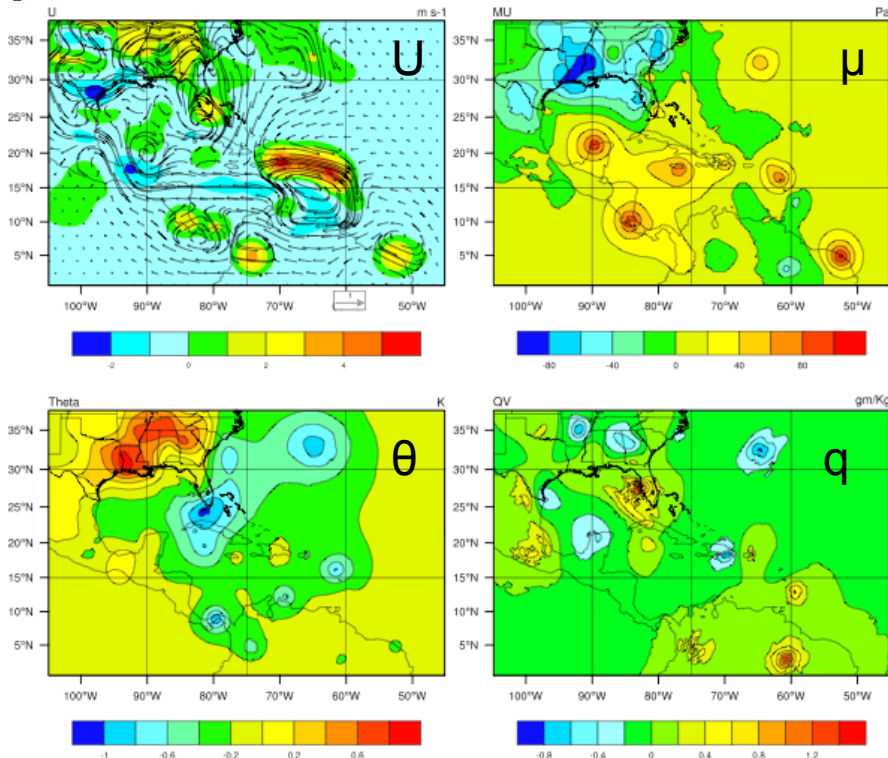
# Increments of $U$ , $\mu$ , $\theta$ , and $q$ at level=1 from assimilation of obsproc **ADPSFC** observations with GSI

Differences	U (m/s)		$\mu$ (Pa)		$\theta$ (K)		Q (g/kg)	
	min	max	min	max	min	max	min	max
Obsproc/ADPSFC - FG	-1.15	1.28	-41.6	254.8	-0.38	0.14	-0.17	0.13
Obsproc/ADPSFC – GDAS/ADPSFC	-0.57	0.52	-59.3	76.1	-0.11	0.07	-0.05	0.03



# Increments of $U$ , $\mu$ , $\theta$ , and $q$ at level=25 from assimilation of obsproc **ADPUPA** observations with GSI

Differences	U (m/s)		$\mu$ (Pa)		$\theta$ (K)		Q (g/kg)	
	min	max	min	max	min	max	min	max
Obsproc/ADPUPA - FG	-2.82	5.61	-97.55	102.43	-1.04	0.93	-0.82	1.44
Obsproc/ADPUPA – GDAS/ADPUPA	-0.29	0.35	-37.91	21.10	-0.33	0.22	-0.20	0.16





# Obsproc experiment discussion

---

- Based on the GSI (o-g) statistics, the increment of  $U$ ,  $\mu$ ,  $\theta$ , and  $q$  of obsproc ADPUPA(ADPSFC) – GDAS ADPUPA(ADPSFC), WRFDA obsproc produced the ADPSFC and ADPUPA prepBUFR files close to the NCEP GDAS data.
- NCEP prepBUFR TYP, CAT information, and QC flags are important and complicate. In the conversion from GDAS to LITTLE\_R, we keep those information as most as possible, otherwise, we cannot obtain the acceptable results.
- Here, as the first step, we only worked on the ADPSFC and ADPUPA data. If resources are available, the work can be done on other observations type by type.

# Summary

---

- Lots of operation observation data files available
- DTC has document and cases for users to learn
  - Basic structure and processing of BUFR files
  - Write data into a new BUFR/PrepBUFR file
  - Append data into an existing BUFR/PrepBUFR file
  - Read data from a BUFR/PrepBUFR file
- But data are complex:
  - Each type of data needs a code to write
  - Quality control and quality marker issues
- Research on Obsproc PrepBUFR data

# Reference and Thanks

---

- Some of slides are modified or come from Ming Hu, Stacie Bender's 2010 GSI tutorial presentation, Yong-Run Guo's 2010 WRFDA tutorial presentation.
- Ming Hu: **Community Tools (1): PrepBUFR Converter**
- Stacie Bender: **Observations: PrepBUFR and BUFR**
  - Download from [http://www.dtcenter.org/com-GSI/users/docs/presentations\\_2010.php](http://www.dtcenter.org/com-GSI/users/docs/presentations_2010.php)
- Yong-Run Guo: **Observation Pre-processing**
  - Download from [http://www.mmm.ucar.edu/wrf/users/wrfda/Tutorials/2010\\_Aug/tutorial\\_presentation\\_summer\\_2010.html](http://www.mmm.ucar.edu/wrf/users/wrfda/Tutorials/2010_Aug/tutorial_presentation_summer_2010.html)

# Questions?

[gsi\\_help@ucar.edu](mailto:gsi_help@ucar.edu)