

# Prebufnr converter for GSI

Ming Hu

Developmental Testbed Center

# What is this talk for?

---

- GSI is using BUFR/PrepBUFR
  - Conventional data : PrepBUFR
  - Satellite: BUFR
  - Radar: BUFR
- If you have data that are not in BUFR format, you have to convert them to PrepBUFR/BUFR. In this talk, we will learn how to:
  - encode and decode a simple BUFR file
  - append a data to an existing BUFR file
  - encode, decode, and append GSI prepbuf

# Encode

(write observations to a BUFR file)

A simplest case:  
One temperature observation

```

program encode_sample
! write one observation into prepbuf file
implicit none

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,irate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
irate=2008120100
subset='ADPUPA'

```

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

```

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buf',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
  call openmb(unit_out,subset,irate)
    call ufbint(unit_out,hdr,4,1,irate,hdstr)
    call ufbint(unit_out,obs,1,1,irate,obstr)
    call ufbint(unit_out,oer,1,1,irate,oestr)
    call ufbint(unit_out,qcf,1,1,irate,qcstr)
    call writsb(unit_out,ibfmsg,irate)
  call closmg(unit_out)
call closbf(unit_out)

```

BUFR tables file

BUFR file

data subsets

messages

BUFR file

end program

```
program encode_sample
! write one observation into prepbufr file
implicit none
```

```
character(80):: hstr='XOB YOB ELV DHR'
character(80):: ostr='TOB', qcstr='TQM', oestr='TOE'
```

```
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)
```

```
INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4
```

```
character(8) subset
integer :: unit_out=10,irate,iret
```

```
hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
irate=2008120100
subset='ADPUPA'
```

```
open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
call openmb(unit_out,subset,irate)
call ufbint(unit_out,hdr,4,1,irate,hstr)
call ufbint(unit_out,obs,1,1,irate,ostr)
call ufbint(unit_out,oer,1,1,irate,oestr)
call ufbint(unit_out,qcf,1,1,irate,qcstr)
call writsb(unit_out,ibfmsg,irate)
call closmg(unit_out)
call closbf(unit_out)
```

```
end program
```

## Define mnemonics

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

Data array

BUFR message size limit of 10000 bytes

### Mnemonics:

XOB: X

YOB: Y

ELV: Z

DHR: t

TOB: temperature

TQM: obs error

TOE: obs quality

## Set data

```
program encode_sample
! write one observation into prepbufr file
implicit none

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB', qcstr='TQM', oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret
```

```
hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
```

Data values written  
to data subset

```
ideate=2008120100
subset='ADPUPA'
```

```
open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
```

```
call openmb(unit_out,subset,ideate)
call ufbint(unit_out,hdr,4,1,iret,hdstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call ufbint(unit_out,oer,1,1,iret,oestr)
call ufbint(unit_out,qcf,1,1,iret,qcstr)
call writsb(unit_out,ibfmsg,iret)
call closmg(unit_out)
call closbf(unit_out)
```

```
end program
```

YYYYMMDDHH instead of  
YYMMDDHH

# BUFR file

```
program encode_sample
! write one observation into prepbufr file
implicit none

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,irate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
irate=2008120100
subset='ADPUPA'

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
  call openmb(unit_out,subset,irate)
  call ufbint(unit_out,hdr,4,1,irate,hdstr)
  call ufbint(unit_out,obs,1,1,irate,obstr)
  call ufbint(unit_out,oer,1,1,irate,oestr)
  call ufbint(unit_out,qcf,1,1,irate,qcstr)
  call writsb(unit_out,ibfmsg,irate)
  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

# Message

```
program encode_sample
! write one observation into prepbufr file
implicit none

character(80):: hdst='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
ideate=2008120100
subset='ADPUPA'

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
call openmb(unit_out,subset,ideate)
  call ufbint(unit_out,hdr,4,1,iret,hdst)
  call ufbint(unit_out,obs,1,1,iret,obstr)
  call ufbint(unit_out,oer,1,1,iret,oestr)
  call ufbint(unit_out,qcf,1,1,iret,qcstr)
  call writsb(unit_out,ibfmsg,iret)
call closmg(unit_out)
call closbf(unit_out)

end program
```

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



# Data subsets

```
program encode_sample
! write one observation into prepbufr file
implicit none

character(80):: hdstr='XOB YOY ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
ideate=2008120100
subset='ADPUPA'

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buf',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
call openmb(unit_out,subset,ideate)
call ufbint(unit_out,hdr,4,1,iret,hdstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call ufbint(unit_out,oer,1,1,iret,oestr)
call ufbint(unit_out,qcf,1,1,iret,qcstr)
call writsb(unit_out,ibfmsg,iret)
call closmg(unit_out)
call closbf(unit_out)

end program
```

## **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

```

program encode_sample
! write one observation into prepbuf file
implicit none

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
ideate=2008120100
subset='ADPUPA'

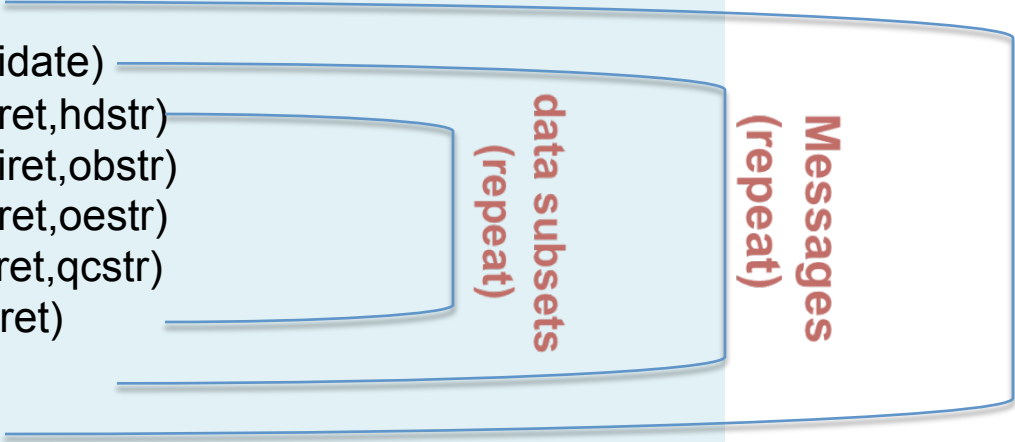
open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
  call openmb(unit_out,subset,ideate)
    call ufbint(unit_out,hdr,4,1,iret,hdstr)
    call ufbint(unit_out,obs,1,1,iret,obstr)
    call ufbint(unit_out,oer,1,1,iret,oestr)
    call ufbint(unit_out,qcf,1,1,iret,qcstr)
    call writsb(unit_out,ibfmsg,iret)
  call closmg(unit_out)
call closbf(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**

- Repeat:
- Message
  - Data subset
  - Levels



end program

# Decode

(read observation from bufr file)

Read the data we just encoded  
The code can read multi-message and multi-data  
subsets

```

program encode_sample
! write one observation into prepbufr file
implicit none

```

## Encode

```

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

```

```

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

```

```

character(8) subset
integer :: unit_out=10,ideate,iret

```

```

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
ideate=2008120100
subset='ADPUPA'

```

```

open(22,file='prepobs_prep.bufrrtable',action='read')
open(unit_out,file='singleT.bufrr',action='write',form='unformatted')
call datelen(10)

```

```

call openbf(unit_out,'OUT',22)

```

```

call openmb(unit_out,subset,ideate)

```

```

call ufbint(unit_out,hdr,4,1,iret,hdstr)
call ufbint(unit_out,obs,1,1,iret,obstr)
call ufbint(unit_out,oer,1,1,iret,oestr)
call ufbint(unit_out,qcf,1,1,iret,qcstr)

```

```

call writsb(unit_out,ibfmsg,iret)

```

```

call closmg(unit_out)
call closbf(unit_out)

```

```

end program

```

```

program decode_sample
! read observation from prepbufr
implicit none

```

## Decode

```

character(80):: hdstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

```

```

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

```

```

INTEGER :: ireadmg,ireadsb

```

```

character(8) subset
integer :: unit_in=10,ideate,iret,nmsg,ntb

```

```

open(unit_in,file='singleT.bufrr',form='unformatted')

```

```

call openbf(unit_in,'IN',unit_in)

```

```

call datelen(10)

```

```

nmsg=0

```

```

msg_report: do while (ireadmg(unit_in,subset,ideate) == 0)

```

```

nmsg=nmsg+1

```

```

ntb = 0

```

```

sb_report: do while (ireadsb(unit_in) == 0)

```

```

ntb = ntb+1

```

```

call ufbint(unit_in,hdr,4,1,iret,hdstr)

```

```

call ufbint(unit_in,obs,1,1,iret,obstr)

```

```

call ufbint(unit_in,oer,1,1,iret,oestr)

```

```

call ufbint(unit_in,qcf,1,1,iret,qcstr)

```

```

write(*,'(I10,8f8.1)') ideate,hdr,obs(1,1),oer(1,1),qcf(1,1)

```

```

enddo sb_report

```

```

write(*,*) 'subset num=',ntb

```

```

enddo msg_report

```

```

write(*,*) 'message num=',nmsg

```

```

call closbf(unit_in)

```

```

end program

```



## Decode

```
program decode_sample
! read observation from prepbuf
implicit none

character(80):: hstr='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4
INTEGER :: ireadmg,ireadsb

character(8) subset
integer :: unit_in=10,odate,iret,nmsg,ntb

open(unit_in,file='singleT.bufr',form='unformatted')
call openbf(unit_in,'IN',unit_in)
call datelen(10)
nmsg=0
msg_report: do while (ireadmg(unit_in,subset,odate) == 0)
nmsg=nmsg+1
ntb = 0
sb_report: do while (ireadsb(unit_in) == 0)
ntb = ntb+1
call ufbint(unit_in,hdr,4,1,iret,hstr)
call ufbint(unit_in,obs,1,1,iret,obstr)
call ufbint(unit_in,oer,1,1,iret,oestr)
call ufbint(unit_in,qcf,1,1,iret,qcstr)
write*,'(I10,8f8.1)' odate,hdr,obs(1,1),oer(1,1),qcf(1,1)
enddo sb_report
write(*,*) 'subset num=',ntb
enddo msg_report
write(*,*) 'message num=',nmsg
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**

# Append obs to existing prepbufr

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

## Encode

```
program encode_sample
! write one observation into prepbuf file
implicit none

character(80):: hdst='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret

hdr(1)=175.0;hdr(2)=30.0;hdr(3)=500.0;hdr(4)=10.0
obs(1,1)=287.0;oer(1,1)=1.0;qcf(1,1)=1.0
ideate=2008120100
subset='ADPUPA'

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
call openmb(unit_out,subset,ideate)
call ufbint(unit_out,hdr,4,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call ufbint(unit_out,oer,1,1,iret,oestr)
call ufbint(unit_out,qcf,1,1,iret,qcstr)
call writsb(unit_out,ibfmsg,iret)
call closmg(unit_out)
call closbf(unit_out)

end program
```

## Append

```
program append_sample
! write one observation into prepbuf file
implicit none

character(80):: hdst='XOB YOB ELV DHR'
character(80):: obstr='TOB',qcstr='TQM',oestr='TOE'
real(8) :: hdr(4),obs(1,1),oer(1,1),qcf(1,1)

INTEGER, PARAMETER :: MXBF = 160000,ibfmsg = MXBF/4

character(8) subset
integer :: unit_out=10,ideate,iret

hdr(1)=275.0;hdr(2)=50.0;hdr(3)=700.0;hdr(4)=1.0
obs(1,1)=297.0;oer(1,1)=2.0;qcf(1,1)=2.0
ideate=2008120100
subset='ADPUPA'

open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='singleT.buftr',status='old',form='unformatted')
call datelen(10)
call openbf(unit_out,'APN',22)
call openmb(unit_out,subset,ideate)
call ufbint(unit_out,hdr,4,1,iret,hdst)
call ufbint(unit_out,obs,1,1,iret,obstr)
call ufbint(unit_out,oer,1,1,iret,oestr)
call ufbint(unit_out,qcf,1,1,iret,qcstr)
call writsb(unit_out,ibfmsg,iret)
call closmg(unit_out)
call closbf(unit_out)

end program
```

# Test results (practice section):

---

**./encode\_sample.exe**

This should generate a bufr file *singleT.bufr*

**./decode\_sample.exe**

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

```
2008120100 175.0 30.0 500.0 10.0 287.0 1.0 1.0
subset num= 1
message num= 1
```

**./append\_sample.exe**

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

**./decode\_sample.exe**

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

```
2008120100 175.0 30.0 500.0 10.0 287.0 1.0 1.0
subset num= 1
2008120100 275.0 50.0 700.0 1.0 297.0 2.0 2.0
subset num= 1
message num= 2
```



# Prepbufr file for GSI

I wrote these samples based on:

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

In GSI, `read_prepbufr.f90` reads prepbufr file

# MNEMONIC in read\_prepbufr.f90

---

```
data hdstr    /'SID XOB YOB DHR TYP ELV SAID  '/
data obstr    /'POB QOB TOB ZOB UOB VOB PWO CAT PRSS' /
data qcstr    /'PQM QQM TQM ZQM WQM NUL PWQ      '/
data oestr    /'POE QOE TOE NUL WOE NUL PWE      '/

data drift    /'XDR YDR HRDR                      '/
data sststr   /'MSST DBSS SST1 SSTQM SSTOE        '/
data satqcstr /'RFFL QIFY QIFN EEQF' /
data prvstr   /'PRVSTG' /
data sprvstr  /'SPRVSTG' /
data levstr   /'POB' /
```

# MNEMONIC list

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 list, 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

```

program encode_prepbuf
! write one observation into prepbuf file
implicit none

character(80):: hdstr='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(8),obs(35,1),qcf(35,1),oer(35,1)

REAL,PARAMETER :: MXBF    = 160000
INTEGER          :: ibfmsg = MXBF/4

character(8)  :: subset,c_sid
integer       :: unit_out=10,odate,iret

real(8)       :: rstation_id
equivalence(rstation_id,c_sid)
!
READ IN DATA and QUALITY into hdr, obs, oer, qcf
! write observation into prepbuf file
open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='prepbuf.test',action='write',form='unformatted')
call datelen(10)
call openbf(unit_out,'OUT',22)
  call openmb(unit_out,subset,odate)
  call ufbint(unit_out,hdr,8,1,iret,hdstr)
  call ufbint(unit_out,obs,9,1,iret,obstr)
  call ufbint(unit_out,oer,7,1,iret,oestr)
  call ufbint(unit_out,qcf,7,1,iret,qcstr)
  call writsb(unit_out,ibfmsg,iret)
  call closmg(unit_out)
call closbf(unit_out)

end program

```

```

program decode_prepbuf
! read observation from prepbuf
implicit none
character(80):: hdstr='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):: ostr='POE QOE TOE NUL WOE NUL PWE      '
real(8)  :: hdr(8),obs(35,1),qcf(35,1),oer(35,1)

REAL,PARAMETER :: MXBF = 160000
INTEGER          :: ibfmsg = MXBF/4, ireadm,ireadsb
character(8)     :: subset, c_sid
integer          :: unit_in=10, idate, nmsg, ntb, i, iret
real(8)          :: rstation_id
equivalence(rstation_id,c_sid)
!
open(unit_in,file='prepbuf.test',form='unformatted')
call openbf(unit_in,'IN',unit_in)
call datelen(10)
nmsg=0
msg_report: do while (ireadm(unit_in,subset,idate) == 0)
nmsg=nmsg+1
ntb = 0
sb_report: do while (ireadsb(unit_in) == 0)
ntb = ntb+1
call ufbint(unit_in,hdr,8,1,iret,hdstr)
call ufbint(unit_in,obs,9,1,iret,obstr)
call ufbint(unit_in,oer,7,1,iret,oestr)
call ufbint(unit_in,qcf,7,1,iret,qcstr)
rstation_id=hdr(1)
write idate,c_sid, hdr(i), obs,oer,qcf out
enddo sb_report
write(*,*) 'subset num=',ntb
enddo msg_report
write(*,*) 'message num=',nmsg
call closbf(unit_in)
end program

```

```

program append_prepbufr
! Append observations into prepbufr file
implicit none

character(80):: hdstr='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(8),obs(35,1),qcf(35,1),oer(35,1)

REAL,PARAMETER :: MXBF    = 160000
INTEGER          :: ibfmsg = MXBF/4

character(8)  :: subset,c_sid
integer       :: unit_out=10,idate,iret

real(8)       :: rstation_id
equivalence(rstation_id,c_sid)
!
READ IN DATA and QUALITY into hdr, obs, oer, qcf
! write observation into prepbufr file
open(22,file='prepobs_prep.bufhtable',action='read')
open(unit_out,file='prepbufr.test',status='old',form='unformatted')
call datelen(10)
call openbf(unit_out,'APN',22)
  call openmb(unit_out,subset,idate)
  call ufbint(unit_out,hdr,8,1,iret,hdstr)
  call ufbint(unit_out,obs,9,1,iret,obstr)
  call ufbint(unit_out,oer,7,1,iret,oestr)
  call ufbint(unit_out,qcf,7,1,iret,qcstr)
  call writsb(unit_out,ibfmsg,iret)
  call closmg(unit_out)
call closbf(unit_out)

end program

```

# Prepbufr convertor test cases

- ./encode\_prepbufr.exe  $\longrightarrow \Delta T = 2.26255$
- ./append\_prepbufr.exe  $\longrightarrow \Delta T = -2.28900$
- ./decode\_prepbufr.exe

## One or two temperature observations in the generated prepbufr file

```
2010050700    HU80503      285.0      37.0      0.0      120.0 99999997952.0 99999997952.0 99999997952.0
  obs=      500.0      100.0     -10.0      0.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
  oer=       1.0      0.1      1.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
  qcf=       1.0 99999997952.0 1.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
subset num=           1

2010050700    HU80505      285.0      37.0      0.0      120.0 99999997952.0 99999997952.0 99999997952.0
  obs=      900.0      100.0     16.0      0.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
  oer=       1.0      0.1      1.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
  qcf=       1.0 99999997952.0 1.0 99999997952.0 99999997952.0 99999997952.0 99999997952.0
subset num=           1

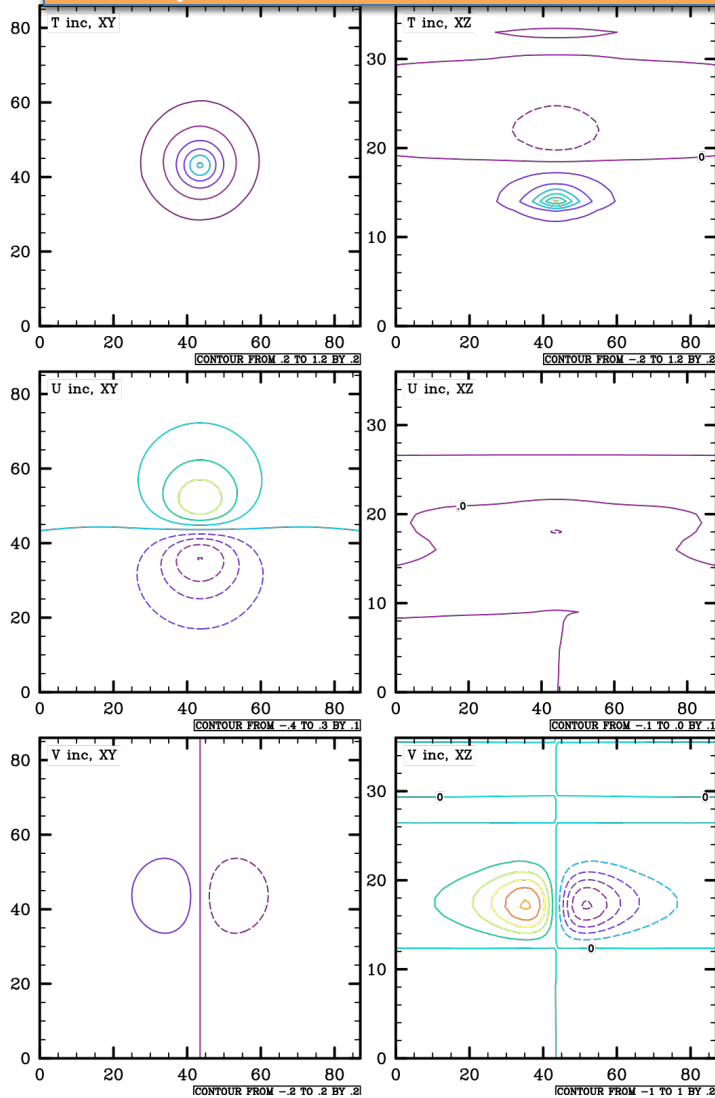
message num=           2
```

```
character(80):: hdstr='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      '
```

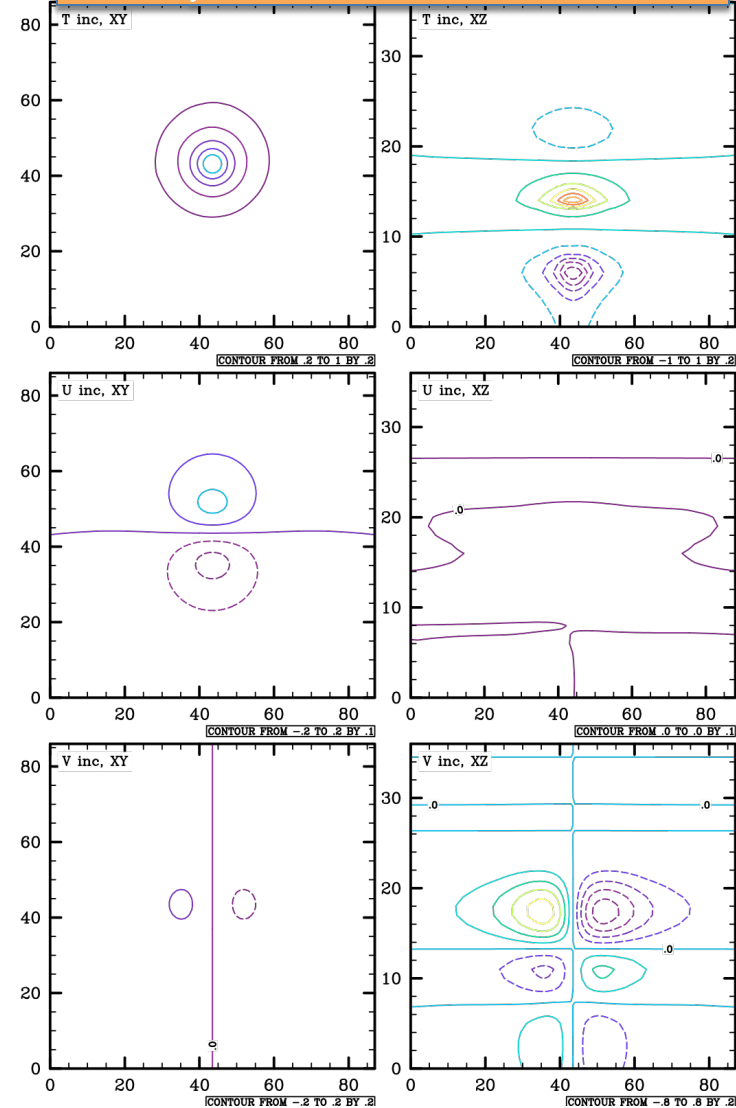


# GSI analysis with generated prepbufr

Analysis increment from 1 obs



Analysis increment from 2 obs



# Summary

# Summary

---

- Based on example, we can:
  - Write data into a new prepbufr file
  - Append data into a existing prepbufr file
  - Read data from prepbufr file
- But data are complex:
  - Each kind of data needs a code to write
  - Didn't touch quality control and quality mark issues
- Other prepbufr convertor available for community
  - OBSPROC

Questions?

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