

# An Overview of Meteorological Observations in BUFR Format (including “PrepBUFR” files)

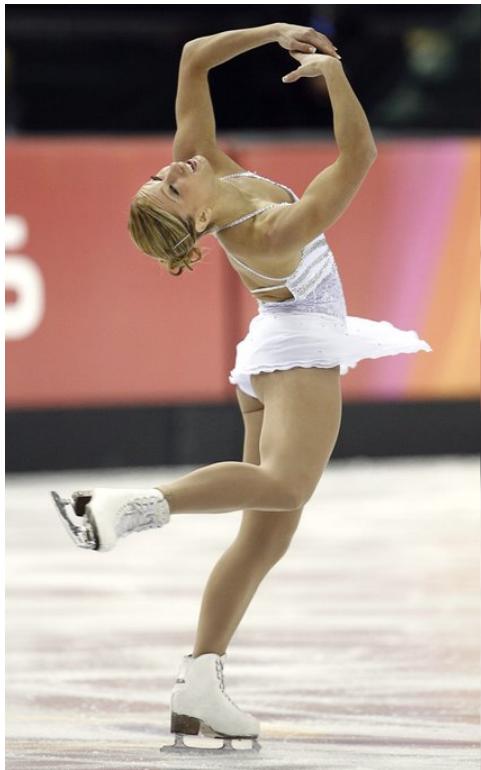
Stacie Bender  
NWS/NCEP/EMC

GSI Tutorial  
June 29, 2010



# Why use BUFR?

Flexibility



Compact Data Storage



WMO Standard



**World  
Meteorological  
Organization**

Weather • Climate • Water



# Common Feelings about BUFR

“I don’t have time to make an Olympic effort to learn how to deal with BUFR.”



“Using BUFR is like herding cats. There’s too much to learn at once.”



“BUFR drives me crazy!!”



**However, there is hope if you learn the basics and have some guidance!**

# NCEP BUFR Files: The Basics

## TOPICS COVERED:

- Obs processing/dataflow at NCEP: how BUFR fits into the “big picture”
- Interacting with BUFR files via the NCEP BUFRLIB software
  - BUFR Tables
  - Reading
  - Writing
  - Appending observations
- Where to go for help

## WHAT'S NOT COVERED:

- Exactly how to read and write all types of data available in BUFR format. (We'd be here until the end of the year!)



# Overview of observations processing and dataflow at NCEP

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

Tank files

(large BUFR files holding 24h of data)

Dump files

(duplicate-checked data from tanks, contain 1, 3, 6h blocks of data)

PrepBUFR files

(QC'd obs from dump files)



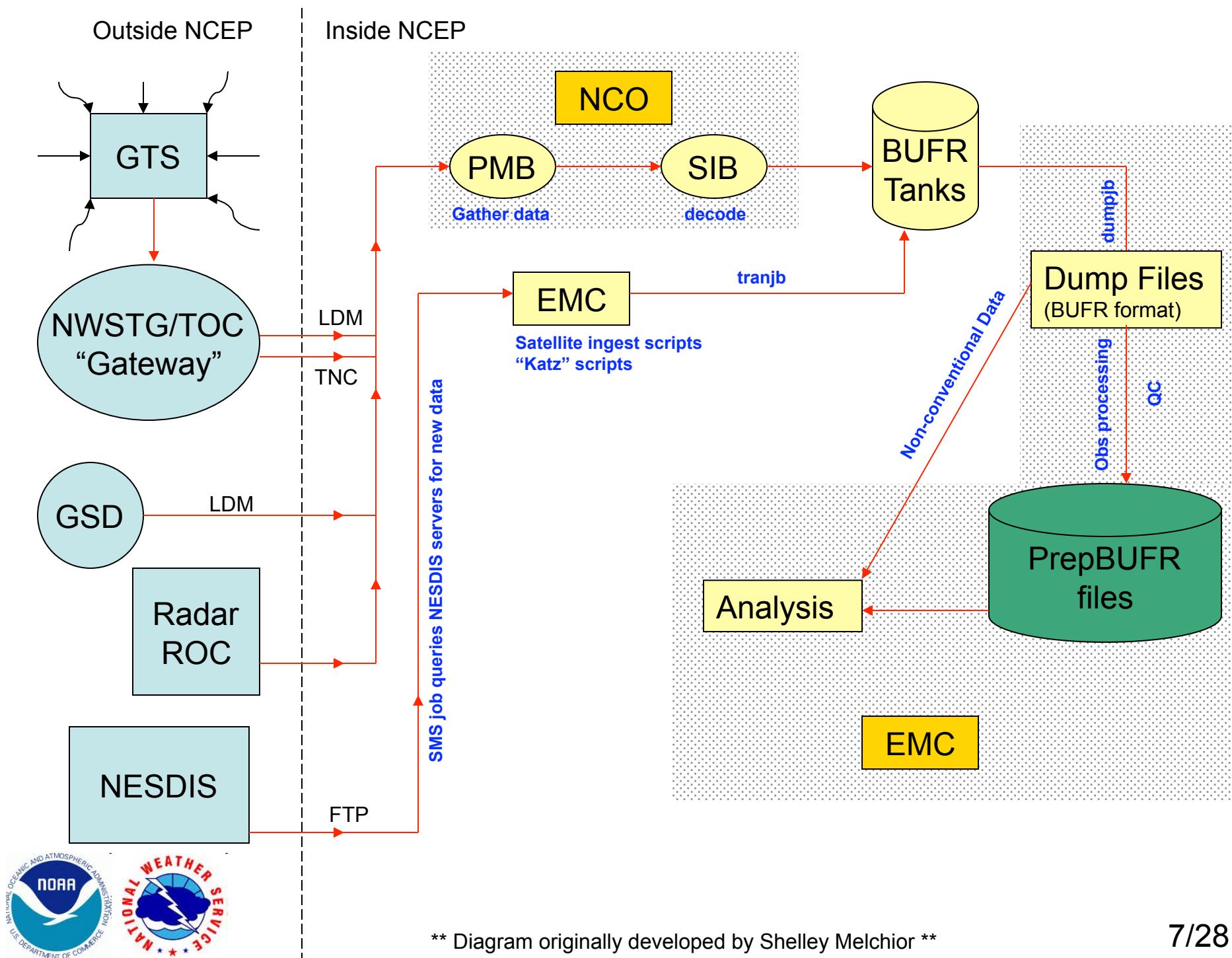
# BUFR vs. PrepBUFR

- BUFR = Binary Universal Form for the Representation of meteorological data (a WMO Standard)  
*\*\* BUFR is a data format. \*\**
- PrepBUFR = “prepared” or QC’d data in BUFR format (NCEP Convention/Standard)

*\*\* PrepBUFR is the NCEP term for QC’d data in BUFR format. \*\**

*\*\* PrepBUFR files are still BUFR files. \*\**





**GTS** = Global Telecommunications System

- World wide data gathers in GTS
- Sends data to NWSTG/TOC or the “Gateway”

**NWSTG/TOC** = NWS Telecommunication Gateway/Telecommunication Operations Center

- Intercepts GTS messages
- Sends data to **NCO** via TNC (TOC to NCEP Communications) line and via LDM (Local Data Manager)

**GSD** = NOAA/ESRL/GSD

- Provide Mesonet data to **NCO** via LDM several times hourly
- SIB converts data from netCDF to WMO BUFR

**Radar/ROC** = NOAA Radar Operations Center

- For more information on how Radar data is processed visit:
- [http://www.emc.ncep.noaa.gov/mmb/data\\_processing/data\\_processing/](http://www.emc.ncep.noaa.gov/mmb/data_processing/data_processing/)

**NESDIS** = National Environmental Satellite, Data, and Information Service

- Servers:
  - (a) dds/ddstest – serves up POES data (operational/test)
  - (b) satepsdist1e/satepsanone – serves up GOES data, winds, radiances, SST, etc ... (operational/test)
- **EMC** runs SMS jobs to query the NESDIS servers for new data



**NCO** = NCEP Central Operations

**PMB** = Production Management Branch, 2 Groups with direct EMC interaction:

- Data Flow - pull data from the outside
  - Interacts w/ the “Gateway”, LDM, ROC, etc ...
  - Data retrieval occurs continuously
  - Gather all the data then pass it off to SIB for decoding
- SPA – make sure code is running 24/7, implement changes

**SIB** = Systems Integration Branch, 1 Group with direct EMC interaction:

- Decoders - decodes data from native format to NCEP BUFR
- Decoders include:
  - ACARS, Aircraft, Aviation Weather (METAR), Bathymetry, Drifting Buoy, Land Sfc, Marine Sfc, NeXRAD Wind, Profiler, Rawinsonde, Satellite Wind, Supplementary Climatology, Tide Gauge
- Decoding operates continuously on the CCS machines (currently stratus and cirrus)
- NCEP BUFR files stored in the tanks on CCS machines (/dcom(dev)/us007003)
- “tranjb” is the process that takes a single BUFR file and appends it to the appropriate tank



## **EMC** = Environmental Modeling Center

- Runs SMS jobs to periodically query the NESDIS servers for new data
- Compares latest available data against a local history file to determine if data is new.
- Retrieves new data via FTP
- Convert native data to NCEP BUFR and stores them in the tanks
- “tranjb” is the process that takes a single BUFR file and appends it to the appropriate tank
- Processing runs on both NCEP CCS machines (at discrete time steps defined for each data type; not continuous)
- Responsible for dump file processing



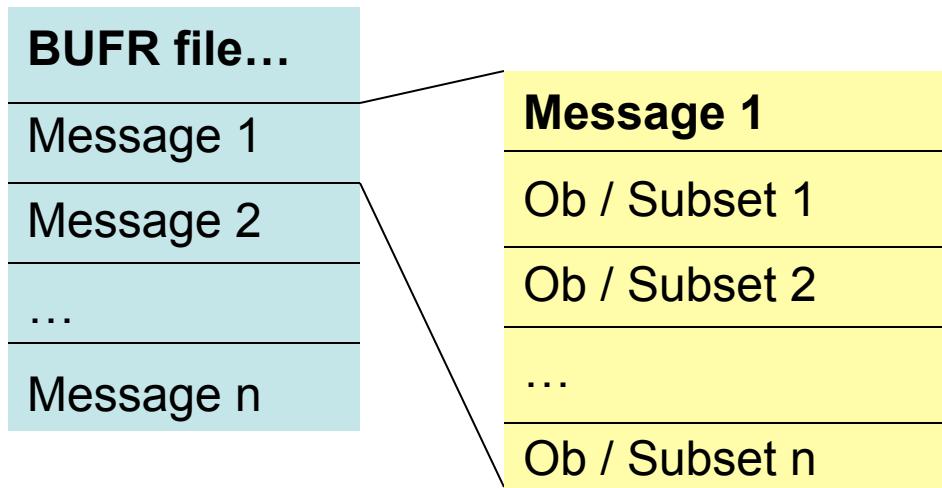
\*\* Information collected by Shelley Melchior and Dennis Keyser \*\*

# Focus: PrepBUFR files

Most GSI users are working with PrepBUFR files, so let's focus on those.

A bit of terminology: BUFR files (including “PrepBUFR” files) contain “messages”. Each message contains “subsets.” Each subset contains a meteorological observation.

In 1000 words:



# Tables: 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 PrepBUFR files, the tables defining the report structures are embedded at the top of the file.

An excerpt from the PrepBUFR table at  
[http://www.emc.ncep.noaa.gov/obs/examples/prepobs\\_prep.bufrtable](http://www.emc.ncep.noaa.gov/obs/examples/prepobs_prep.bufrtable) for the ADPUPA (upper-air) data type:

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

HEADR, PRSLEVEL, SST\_INFO, PREWXSEQ, CLOUDSEQ, CLOU3SEQ, SWINDSEQ, AFIC\_SEQ, TURB3SEQ expand further (see next slide).

Excellent reference for NCEP BUFR TABLES:

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



HEADR SIRC {PRSLEVEL}	-> SID	XOB	YOB	DHR	ELV	TYP	T29	TSB	ITP	SQN	PROCN	RPT	TCOR	<RSRD_SEQ> RSRD EXPRSRD
	-> CAT	<P__INFO>	<Q__INFO>	<T__INFO>	<Z__INFO>	<W__INFO>	<DRFTINFO> XDR YDR HRDR							
		[P__EVENT] POB	[Q__EVENT] QOB	[T__EVENT] TOB	[Z__EVENT] ZOB	[W__EVENT] UOB VOB								
		PQM	QQM	TQM	ZQM	WQM								
		PPC	QPC	TPC	ZPC	WPC								
		PRC	QRC	TRC	ZRC	WRC	[DF_EVENT] DDO FFO DFQ DFP DFR							
				TDO	TVO									
		<P__BACKG> POE	<Q__BACKG> QOE	<T__BACKG> TOE	<Z__BACKG> ZOE	<W__BACKG> WOE								
		PFC	QFC	TFC	ZFC	VFC								
		<PFC_MSQ> PFC_MOD	<QFC_MSQ> QFC_MOD	<TFC_MSQ> TFC_MOD	<ZFC_MSQ> ZFC_MOD	<WFC_MSQ> UFCMOD VFCMOD								
		<P__POSTP> PAN	<Q__POSTP> QAN	<T__POSTP> TAN	<Z__POSTP> ZAN	<W__POSTP> UAN VAN								
		<PCLIMATO> PCL	<QCLIMATO> QCL	<TCLIMATO> TCL	<ZCLIMATO> ZCL	<WCLIMATO> UCL UCS VCL VCS								
		PCS	QCS	TCS	ZCS									
<SST_INFO>	-> [SSTEVENT]	MSST	<SSTBACKG>	<SSTPOSTP>	<DBSS_SEQ>									
	SST1	SSTOE		SSTAN	DBSS									
	SSTQM	SSTFC												
	SSTPC													
	SSTRC													
<PREWXSEQ>	-> PRWE													
{CLOUDSEQ}	-> VSSO	CLAM	CLTP	HOCH		<HOCT_SEQ>								
						HOCT								
<CLOUD2SEQ>	-> TOCC	HBLCS												
<SWINDSEQ>	-> WDIR1	WSPD1												
<AFIC_SEQ>	-> AFIC	HBOI	HTOI											
<TURB3SEQ>	-> DGOT	HBOT	HTOT											

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.



# Replication: a way to efficiently store data in BUFR format

Two main types of replication:

- Fixed or standard: the number of repetitions is always constant
  - Example: satellite data from a fixed number of channels
  - Example: a set number of quality marks
- Delayed: the number of repetitions is not always constant
  - Example: levels in a radiosonde report



# Fixed or Standard Replication

- Values that are repeated a fixed number of times appear in the BUFR (or “[PrepBUFR](#)”) tables surrounded by quotation marks

Example: GOES IR winds

NC005015   BID RCPTIM	RPID	CORN									
NC005015   SAID OGCE	SCLF	SAZA	SSNX	SSNY							
NC005015   YEAR MNTH	DAYS	HOUR	MINU	SECO	TPHR	CLAT	CLON				
NC005015   SIDP SWCM	SCCF	SCBW	CCST	TCMD	LSQL	OFGI	SWQM				
NC005015   HAMD PRLC											
NC005015   WDIR "GQCPRMS"3											
NC005015   WSPD "GQCPRMS"3											
NC005015   "TWIND"4											
NC005015   "MDPT"10											

For example, GQCPRMS = OGCE GNAP PCCF, and these values are repeated 3 times.

where OGCE = Originating Center  
GNAP = Generating Application  
PCCF = Percent Confidence

- Fixed replication appears rarely, if ever, in [PrepBUFR](#) tables. However, it does occur in many of the dump files, many of which are read directly by the analysis step.



# Types of Delayed Replication

< MNEMONIC > = “one-bit replication”:

- values between angle brackets are either present or they are not.
- The values are repeated “zero or one” time(s).
- represents a way to “turn on or off” a set of values
- If values are not present, only one bit is used to represent the fact that the data are missing (efficient way to indicate missing data)

{ MNEMONIC } OR [ MNEMONIC ] = “8-bit delayed replication”:

- repeat the values within the brackets up to 255 times
- { } appear in both dump files and in [PrepBUFR](#) files.
- [ ] appear in only [PrepBUFR](#) files and are used specifically to represent quality decisions made on observations by different QC modules (hypothetically, you could have up to 255 quality control “events” on one observation – more on this in the next slide).

( MNEMONIC ) = “16-bit delayed replication”:

- repeat these values up to 65,535 times
- Use this if you might need to repeat values more than 255 times.



# QC Information in PrepBUFR files

- QC marks can be placed on observations by multiple QC modules.
- The QC information is stored in “event stacks”, which are basically a number of repetitions/replications of QC information.
- The first replication of the QC information represents the final QC decision regarding the observation. Most people use the final QC decision.
- If information about QC marks placed by specific QC modules is needed, the event stack can be unpacked in full.



# PrepBUFR event stacks: Previous example of ADPUPA data

(Slides 12 and 13)

HEADR SIRC {PRSLEVEL}	-> SID	XOB	YOB	DHR	ELV	TYP	T29	TSB	ITP	SQN	PROCN	RPT	TCOR	<RSRD_SEQ> RSRD EXPRS RD
	-> CAT	<P__INFO>	<Q__INFO>	<T__INFO>	<Z__INFO>	<W__INFO>	<DRFTINFO> XDR YDR HRDR							
		[P__EVENT] POB	[Q__EVENT] QOB	[T__EVENT] TOB	[Z__EVENT] ZOB	[W__EVENT] UOB VOB WQM WPC WRC								
		PQM PPC PRC	QQM QPC QRC	TQM TPC TRC	ZQM ZPC ZRC	[DF_EVENT] DDO FFO DFQ DFP DFR								
				TDO	TVO									
		<P__BACKG> POE PFC	<Q__BACKG> QOE QFC	<T__BACKG> TOE TFC	<Z__BACKG> ZOE ZFC	<W__BACKG> WOE UFC VFC								
		<PFC_MSQ> PFC_MOD	<QFC_MSQ> QFC_MOD	<TFC_MSQ> TFC_MOD	<ZFC_MSQ> ZFC_MOD	<WFC_MSQ> UFCMOD VFCMOD								
		<P__POSTP> PAN	<Q__POSTP> QAN	<T__POSTP> TAN	<Z__POSTP> ZAN	<W__POSTP> UAN VAN								
		<PCLIMATO> PCL PCS	<QCCLIMATO> QCL QCS	<TCCLIMATO> TCL TCS	<ZCLIMATO> ZCL ZCS	<WCCLIMATO> UCL UCS VCL VCS								
<SSST_INFO>	-> [SSSTEVENT] MSST SST1 SSTQM SSTPC SSTRC	<SSSTBACKG> SSTOE SSTFC	<SSSTPOSTP> SSTAN	<DBSS_SEQ> DBSS										
<PREWXSEQ> {CLOUDSEQ}	-> PRWE -> VSSO	CLAM	CLTP	HOCB	<HOCT_SEQ> HOCT									
<CLOUD2SEQ>	-> TOCC	HBLCS												
<SWINDSEQ>	-> WDIR1	WSPD1												
<AFIC_SEQ> <TURB3SEQ>	-> AFIC -> DGOT	HBOI HBOT	HTOI HTOT											

The report structure [T\_\_EVENT] represents a temperature “event stack” where:

TOB = temperature ob

TQM = temp. quality mark

TPC = temp. program code

TRC = temp. reason code

These four values can be repeated/replicated up to 255 times (i.e., there can be up to 255 QC “events”).



# NCEP BUFR LIB Software

- An overview is presented in this presentation, not every nitty gritty detail of how to read and write PrepBUFR files (unless you'd like to stay until the next millennium).
- Some preliminary information:
  - The BUFRLIB requires BUFR (or PrepBUFR) files to be FORTRAN-blocked before the BUFRLIB is used on them.
    - Almost always, any PrepBUFR file is already blocked for you and you can use the BUFRLIB routines directly.
    - If your BUFR (or PrepBUFR) file is not blocked, you can use the NCEP cwordsh utility to block it (<http://www.nco.ncep.noaa.gov/sib/decoders/BUFR LIB/toc/cwordsh/> ).
- There is detailed, clearly-explained, very useful BUFRLIB documentation written by NCO's Jeff Ator: <http://www.nco.ncep.noaa.gov/sib/decoders/BUFR LIB/>
  - Quality reference for:
    - Purposes/specific uses of major BUFRLIB routines
    - BUFRLIB routine arguments



# Commonly Used BUFRLIB Routines

- openbf
- readmg/ireadmg
- readsb/ireadsb
- openmb
- ufbint
- ufbrep
- ufbseq
- writsb
- closbf

Detailed documentation of the use of these routines is available at:  
<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/>



# Pseudo-code for READING

```
call openbf ! Opens a BUFR file for reading
do while(ireadmg.eq.0) ! Starts the read of BUFR messages
    do while(ireadsb.eq.0) ! Starts the read of BUFR subsets within the current message
        depending on data types and their report structures,
        call:
            - ufbint
            - ufbrep
            - ufbseq
        ^ These routines pull the data values from the BUFR
        subsets in the BUFR (or PrepBUFR) file.
    enddo ! ireadsb
enddo ! ireadmg
call closbf ! Close the BUFR file
```



# Pseudo-code for WRITING

**call openbf** ! Opens a BUFR file for writing

**call openmb** ! Opens a BUFR message for writing (it will also close any previously open BUFR messages if they are full.)

! To store data in BUFRLIB memory, call:

- ufbint
- ufbrep
- ufbseq

**call wrtsb** ! Send data stored via “ufb” routines to output file

**call closbf** ! Close the BUFR file

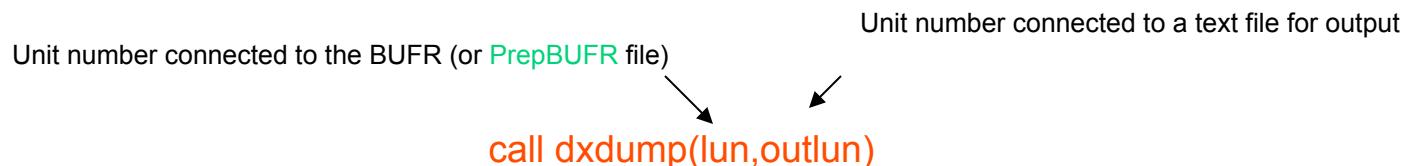


# Appending data to an existing file

*Before appending any new data, look at the BUFR table associated with the PrepBUFR file to which you're appending data.*

Make sure the report structure in the existing file fits the report structure of the data you want to append. Sometimes the report structure changes as new variables get added. *The report structures in the table used to build the original BUFR file should match the report structures of the data you want to add.* If not, you may get errors when using the BUFRLIB.

If you're not sure of the report structures within the PrepBUFR file, you can use the BUFRLIB routine *dxdump* to extract the table embedded within the PrepBUFR file before appending data.



# Appending data to an existing file (cont.)

The main difference between writing out a brand new BUFR file (or PrepBUFR file) lies in the second argument to the BUFRLIB subroutine *openbf* (*open BUFR file*):

! Open brand new BUFR file for output

```
call openbf(unit number for BUFR/PrepBUFR file, 'OUT', unit number for BUFR/PrepBUFR table)
```

! Append data to existing BUFR file

```
call openbf(unit number for BUFR/PrepBUFR file, 'APN', unit number for BUFR/PrepBUFR table)
```

VS



# Online sources of help and information

- NCO BUFRLIB Documentation:  
<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/>
- WMO General BUFR Docs:  
[http://www.wmo.int/pages/prog/www/WMOCodes/Guides/BUFRCREXPreface\\_en.html](http://www.wmo.int/pages/prog/www/WMOCodes/Guides/BUFRCREXPreface_en.html)
- Obs/QC Processing Forum (including BUFR):  
<http://optimus.ncep.noaa.gov/forum/viewforum.php?f=29>
  - may need to be inside NCEP firewall for access
- EMC Docs:  
Overall description of how data is processed at NCEP:  
[http://www.emc.ncep.noaa.gov/mmb/data\\_processing/data\\_processing/](http://www.emc.ncep.noaa.gov/mmb/data_processing/data_processing/)  
and  
“Table B” (defines mnemonics that represent various data values):  
[http://www.emc.ncep.noaa.gov/mmb/data\\_processing/bufrtab\\_tableb.htm](http://www.emc.ncep.noaa.gov/mmb/data_processing/bufrtab_tableb.htm)



# What if the online docs don't answer your question?

Some problems and challenges faced by those working with files in BUFR format are best handled on a case-by-case basis.

How to get help and information from real, live humans (yes, there are humans at NCEP!):

BUFRLIB support via online web form:

[http://www.nco.ncep.noaa.gov/sib/decoders/mail\\_bufrlib/](http://www.nco.ncep.noaa.gov/sib/decoders/mail_bufrlib/) - comments

and questions sent via this form reach several of us in EMC and NCO. Exchanges are then made via email to solve problems.

Send email to the GSI Helpdesk: [gsi\\_help@ucar.edu](mailto:gsi_help@ucar.edu), where your request will be routed appropriately.





# In a nutshell...



- The BUFR format for meteorological data is flexible and powerful but sometimes confusing for new (and even experienced!) users.
- NCEP uses the BUFRLIB software and BUFR tables to interact with BUFR files (including [PrepBUFR](#) files).
- Working with BUFR files requires some patience, especially at first.
- Often, the BUFR format and the BUFRLIB software are easiest to learn by:
  - working through examples
  - experimentation
  - not being afraid to dig in and try out your own code
- Help is available! Most experienced users still remember the growing pains they felt while learning to use BUFR files. Please ask for help if you need it!



# Questions?

