An Overview of Observational Data Processing at NCEP
(with information on BUFR Format including "PrepBUFR" files)

Dennis Keyser
NWS/NCEP/EMC

GSI Tutorial
August 6, 2013

TOPICS COVERED:

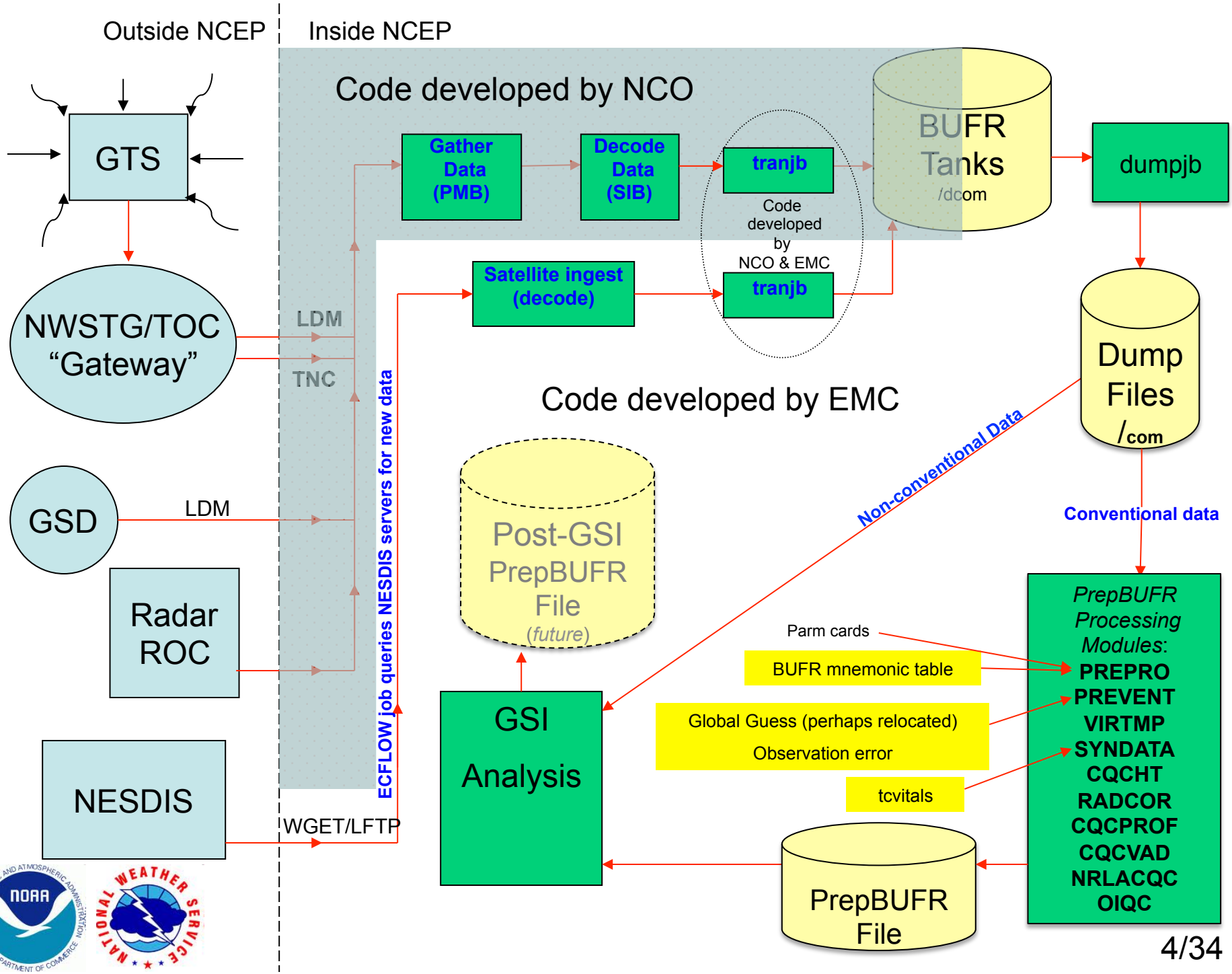
- Obs processing/dataflow at NCEP
- How BUFR fits into the “big picture”
- Interacting with BUFR files via NCEP BUFRLIB software
  - BUFR Tables
  - Reading
  - Writing
  - Appending observations
- Where to go for help

WHAT’S NOT COVERED:

- Details on how to read and write all types of BUFR data

Overview of observational processing and dataflow at NCEP

- Managed jointly by NCEP Central Operations (NCO) and NCEP/EMC
- Almost all observational data at NCEP eventually ends up in BUFR format (Binary Universal Form for the Representation of meteorological data)
  - Relies on NCEP BUFRLIB software (more about that later)
- Four stages:
  - Data flow into NCEP
  - Continuous decoding of data and accumulation into BUFR database or "tank" files (large files holding 24 hrs of data)
  - Network-specific generation of dump files (1 to 6 hr time-windowed, duplicate-checked BUFR data read from tanks)
  - Generation of PrepBUFR files (QC'd "conventional" obs from dump files, read by GSI)


Outside NCEP
Inside NCEP
Code developed by NCO
GTS
Gather Data (PMB)
Decode Data (SIB)
tranjb
Code developed by NCO & EMC
BUFR Tanks /dcom
dumpjb
NWSTG/TOC "Gateway"
LDM
TNC
Satellite ingest (decode)
tranjb
Dump Files /com
Code developed by EMC
GSD
LDM
Post-GSI PrepBUFR File (future)
Non-conventional Data
Conventional data
Radar ROC
ECFLOW job queries NESDIS servers for new data
Parm cards
BUFR mnemonic table
PrepBUFR Processing Modules:
PREPRO
PREVENT
VIRTMP
SYNDATA
CQCHT
RADCOR
CQCPROF
CQCVAD
NRLACQC
OIQC
GSI Analysis
Global Guess (perhaps relocated)
Observation error
tcvitals
NESDIS
WGET/LFTP
PrepBUFR File

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 see http://www.emc.ncep.noaa.gov/mmb/data_processing/data_processing/

NESDIS = National Environmental Satellite, Data, and Information Service

- Servers:
  - DDS – serves up POES data
  - SATEPSDIST1E – serves up GOES data
- **EMC** runs ECFLOW jobs to query the NESDIS servers for new data

NCO = NCEP Central Operations

PMB = NCO/Production Management Branch, two groups with 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 group – make sure production is running 24x7 & implement change

SIB = NCO/Systems Integration Branch, one group with EMC interaction:

- Decoders - translate data from their native format to NCEP BUFR, includes:
  ACARS (MCDRS), Aircraft (AIREP, PIREP, AMDAR, TAMDAR, RECCO), Land Sfc (SYNOP, METAR), RADAR (VAD winds, WSR-88D, Canadian, P3 TDR), Oceanographic (Bathy, TESAC, TRAKOB), Marine Sfc (Ship, Buoy, Tide Gauge, Coast Guard, CMAN), RARS, Profiler/RASS (NPN, MAP, Europe, Japan, Hong Kong), Upper Air (RAOB, PIBAL, DROP), GPS-IPW, GPS-RO, USGS River/Stream, SHEF (precip, land & marine sfc), Satellite Wind (JMA, EUMETSAT, India), CREX (marine sfc), Satellite Altimetry/Wave, Mesonet (incl. COOP, CRN, HYDRO, SNOW), SEVIRI, Lightning, LaRC Cloud, AIRNOW (ozone)
- Decoding operates continuously on both WCOSS machines (prod & dev)
- NCEP BUFR files stored in the database “tanks” on the WCOSS machines
- “tranjb” is the process that takes a single BUFR file and appends it to the appropriate database tank

EMC = Environmental Modeling Center

- Runs ECFLOW jobs periodically to query NESDIS servers for new data files
  - Compares list of files on server against local history file
- Retrieves new data via WGET or LFTP file transfer protocol
- Converts native data to NCEP BUFR and stores them in the database tanks
- "tranjb" is the process that takes a single BUFR file and appends it to the appropriate database tank
- Processing runs on both NCEP WCOSS machines (prod and dev) at discrete wall-clock times defined for each data type
- Satellite data types decoded here include:
  - 1B radiances from GOES (sounder and imager), SSM/IS, ATOVS (AMSU-A, AMSU-B, MHS, HIRS-3, HIRS-4), AQUA/TERRA (AIRS, AMSU-A, IASI), AVHRR/GAC, NPP (ATMS & CrIS)
  - Cloud data from GOES (via NESDIS & LaRC), global cloud analyses from AFWA & CLAVR
  - Temperature soundings from ATOVS
  - Sat-derived winds from GOES (IR, WV-img, WV-snd, VIZ), MODIS (IR, WV-img), AVHRR (IR)
  - Scatterometer winds from ASCAT, WindSat
  - Rainfall from TRMM/TMI
  - SST from POES (via NAVO & NESDIS), GOES, global SST analyses
  - Ozone from SBUV-2, GOME-2, OMI, MLS
  - Aerosol, Green Vegetation Fraction and smoke from NESDIS (Global)
  - Daily snow and ice analyses from NESDIS & USAF
  - Imager data (11 μ channel) from GOES

Database “Tanks”

- BUFR
- Arranged by UTC day and continuously grow throughout the day
- No QC (other than rudimentary checks inside decoders)
- No duplicate checking
- On WCOSS in /dcom/us007003/*yyyymmdd*/b*mmm*/xx*sss*

  (where *mmm* is WMO BUFR message type and *xxx* is local BUFR message subtype)

Dump Files

- BUFR
- Generated on production WCOSS machine from “tanks” at each network data cutoff time
- Time-windowed, geographically filtered (if RGL)
- Duplicate checked
- QC: NCEP/SDM purge (or keep) flags, reject list flags, and OPC marine flags on data
- Post-dump processing lists dump contents of files, sets applicable files to “restricted” and creates their non-restricted forms

Dump Files (cont.)

- On WCOSS in:

  /com/*NET*/prod/*RUN*.*yyyymmdd*/*MODEL*.t*cyc*z.*TYPE*.tm*MM*.bufr_d,
  where:

  – *NET*(*RUN*/*MODEL*) is either: **cdas**(**cdas**/**cdas**), **cfs**(**cdas**/**cdas1**), **dump**(**dump**/**dump**), **gfs**(**gdas**/**gdas1**), **gfs**(**gfs**/**gfs**), **nam**(**nam**/**nam**), **nam**(**ndas**/**ndas**), **rap**(**rap**/**rap**), **rap**(**rap_p**/**rap_p**), **rtma**(**rtma**/**rtma**)

  – *cyc* is cycle (hourly for *NET*=**dump**, **rap**, **rtma**; **00**, **06**, **12**, **18** all others)

  – *MM* is **00** for all types except *RUN*/*MODEL*=**ndas**/**ndas** where it can be **12**, **09**, **06** or **03**

  – *TYPE* is either: **1bamua**, **1bamub**, **1bhrs3**, **1bhrs4**, **1bmhs**, **adpsfc**, **adpupa**, **aircar**, **aircft**, **airsev**, **ascatt**, **ascatw**, **atms**, **atovs**, **avcsam**, **avcspm**, **bathy**, **cris**, **esamua**, **esamub**, **eshrs3**, **esmhs**, **geoimr**, **goesfv**, **goesnd**, **gome**, **gpsipw**, **gpsro**, **lghtng**, **lgycld**, **mls**, **msonet**, **mtiasi**, **nexrad**, **omi**, **osbuv8**, **proflr**, **radwnd**, **rassda**, **satwnd**, **sevcsr**, **sfcshp**, **sptrmm**, **ssmisu**, **tesac**, **trkob**, **trmm**, **vadwnd**, **wdsatr**, **wndsat** (types actually dumped depend upon the network)

PrepBUFR Processing/Files

- BUFR
- Generated on production WCOSS machine in each network from conventional data dump files
  - **adpsfc**, **adpupa**, **aircar**, **aircft**, **ascatw**, **atovs**, **goesnd**, **gpsipw**, **msonet**, **proflr**, **rassda**, **satwnd** (except for GFS/GDAS), **sfcshp**, **vadwnd**, **wndsat**
- Network-specific parm cards control processing
- Structure defined by input PrepBUFR mnemonic table
- BUFR message types can include:
  - **ADPUPA**, **AIRCAR**, **AIRCFT**, **SATWND**, **PROFLR**, **VADWND**, **SATEMP**, **ADPSFC**, **SFCSHP**, **SFCBOG**, **SPSSMI**, **SYNDAT**, **ERS1DA**, **GOESND**, **QKSWND**, **MSONET**, **GPSIPW**, **RASSDA**, **WDSATR**, **ASCATW** (what is present depends on the network)
  - Each message type has its own BUFR structure as defined in the PrepBUFR mnemonic table
- Post-processing sets files to restricted, creates non-restricted forms

PrepBUFR Processing/Files (cont.)

- On WCOSS in:
  /com/*NET*/prod/*RUN*.*yyyymmdd*/*MODEL*.t*cyc*z.prepbufr, where:
  - *NET*(*RUN*/*MODEL*) is either: **akrtma**(**akrtma**/**rtma**), (**cdas**(**cdas**/**cdas**), **cfs**(**cdas**/**cdas1**), **gfs**(**gdas**/**gdas1**), **gfs**(**gfs**/**gfs**), **gurtma**(**gurtma**/**gurtma**), **hirtma**(**hirtma**/**rtma**), **prrtma**(**prrtma**/**rtma**), **rtma2p5**(**rtma2p5**/**rtma2p5**)
  - *cyc* is cycle (hourly for *NET*=***rtma***; **00**, **06**, **12**, **18** all others)
- On WCOSS in: /com/*NET*/prod/*RUN*.*yyyymmdd*/*MODEL*.t*cyc*z.prepbufr.tm*MM*, where:
  - *NET*(*RUN*/*MODEL*) is either: **nam**(**nam**/**nam**), **nam**(**ndas**/**ndas**), **rap**(**rap**/**rap**), **rap**(**rap_p**/**rap_p**), **rtma**(**rtma**/**rtma**)
  - *cyc* is cycle (hourly for *NET*=**rap**, **rtma**; **00**, **06**, **12**, **18** all others)
  - *MM* is **00** for all types except *RUN*/*MODEL*=**ndas**/**ndas** where it can be **12**, **09**, **06** or **03**

Modules that run as part of the PrepBUFR processing

- PREPRO:
  - Reads in dumps, parm cards and PrepBUFR mnemonic table. Preforms rudimentary QC. Generates initial (pre-QC) PrepBUFR file.
- PREVENT:
  - Encodes global guess interpolated to obs locations into PrepBUFR file (used by subsequent QC modules).
    - Tropical cyclones may be relocated in guess in upstream processing.
  - Encodes observation error into PrepBUFR file (used by GSI).
  - Performs some rough quality control checks on surface pressure (vs. the background). Updates are encoded into PrepBUFR file.
- VIRTMP
  - Converts dry bulb temperature to virtual temperature and dewpoint temperature to specific humidity. Updates are encoded into PrepBUFR file.
- SYNDATA [runs in GFS/GDAS (for weak storms with no relocation upstream) and NAM/NDAS (for all storms)]
  - Reads in QC'd tropical cyclone records ("tcvitals"). Generates bogused wind profile reports in the vicinity of tropical storms. Generates bogused surface pressure and vertical profile moisture reports at storm center (NAM/NDAS only). Updates are encoded into PrepBUFR file.
  - Flags (for non-assimilation) all mass observations in the vicinity of each storm in the tcvitals file list. Updates are encoded into PrepBUFR file.
  - Flags (for non-assimilation) all dropwinsonde wind observations in the vicinity of each storm in the tcvitals file list. Updates are encoded into PrepBUFR file.

Modules that run as part of the PrepBUFR processing (cont.)

- CQCHT (does not run in RTMA network)
  - Performs complex quality control on rawinsonde heights and temperatures to identify and/or correct location, transcription and communications errors. Erroneous data that cannot be corrected are flagged for non-assimilation. Updates are encoded into PrepBUFR file.
  - Checks include hydrostatic, increment, horizontal statistical, vertical statistical, temporal, baseline and lapse rate.
- RADCOR (does not run in RTMA network)
  - Applies intersonde (radiation) corrections to CQCHT QC'd rawinsonde height and temperature data. The degree of correction is a function of the rawinsonde instrument type, sun angle and pressure level. Updates are encoded into PrepBUFR file.
- CQCPROF (does not run in RTMA network)
  - Performs complex quality control on wind profiler and SODAR data to identify erroneous data and remove it from consideration by the analyses. Updates are encoded into PrepBUFR file.
  - Checks include increment, vertical statistical, temporal statistical, and combined vertical-temporal.
- CQCVAD (does not run in RTMA network)
  - Performs complex quality control on Velocity Azimuth Display (VAD) winds from WSR-88D radars to identify erroneous data and remove it from consideration by the analyses. Updates are encoded into PrepBUFR file.
  - Checks include increment, vertical statistical, temporal statistical, and combined vertical-temporal. In addition, there is an algorithm to account for contamination due to the seasonal migration of birds.

Modules that run as part of the PrepBUFR processing (cont.)

- NRLACQC (does not run in RTMA network)
  - Performs quality control on all types of aircraft wind and temperature data. Reports failing QC are flagged if they cannot be rehabilitated. Duplicate reports are removed. Updates are encoded into PrepBUFR file.
  - Checks on tracks include duplicate report, spike, invalid data, stuck value, gross, inconsistent altitude or position, flight order, suspect data, reject list. The QC algorithm was developed by the NRL.

- OIQC (runs only in GFS & GDAS networks; output is not used by GSI because it performs its own internal QC)
  - Performs an OI-based QC on the full set of obs in the PrepBUFR file. A final quality decision is made based on the results from all prior platform-specific quality checks (see above) and from any manual quality marks attached to the data. . Updates are encoded into PrepBUFR file.
  - Checks include horizontal, vertical, geostrophic.

- The updated observation and quality mark information from the modules listed above is stored in replicated “event stacks”. They are arranged such that the first replication (i.e., the top of the stack) represents the final module’s observation/quality mark update. This is what is read by the analysis.
  - If information about previous module updates is needed, the full event stack can be unpacked. This allows one to view the record of every change to an observation throughout the course of the PrepBUFR processing.

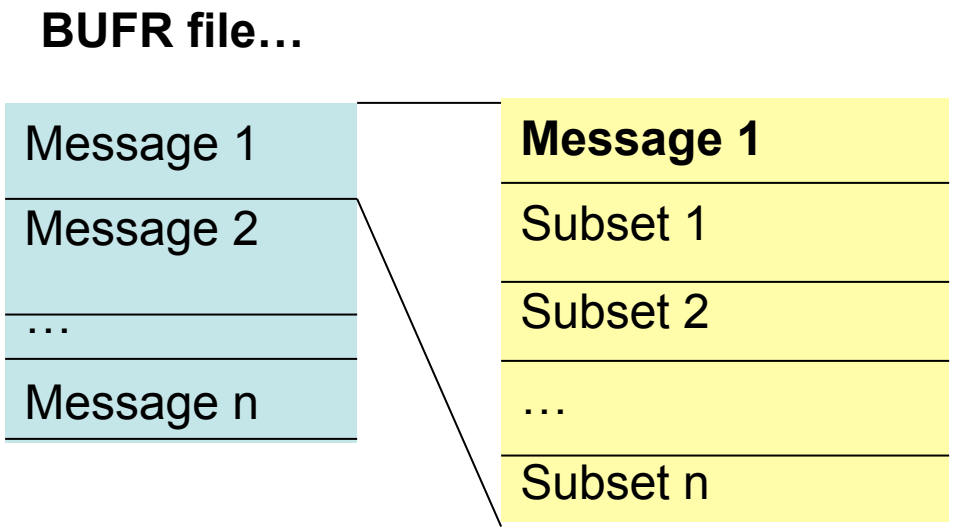
NCEP puts almost all obs into BUFR

- Advantages of BUFR:
  - Flexibility
  - Compact Data Storage
  - WMO Standard
- Features of NCEP BUFR files:
  - Use BUFRLIB routines to interface with files
  - BUFR table encoded into “dictionary” messages at top of file
    - Files are self defined, no need for external BUFR table to decode
  - Uses a single sequence descriptor to define complete subset structure in descriptor section (3) of BUFR messages (more on this later)
  - NCEP BUFR adheres to WMO standard

BUFR File Structure

All BUFR files (including tank, dump and PrepBUFR files) consist of a sequence of “messages”. Each message contains multiple “subsets” (or reports), each with the same structure (usually the same type of data). This structure is actually defined prior to the subsets within each message and consists of a series of “descriptors” whose attributes (e.g., scale, reference value, bit width, units) are defined in a BUFR table. Each subset contains all of the observations plus metadata making up a meteorological report.

(Note: Here, Message 1 represents the first “data” message. Recall that NCEP BUFR files normally have the BUFR table encoded into “dictionary” messages at their top.)



BUFR Tables: Define report structure in an NCEP BUFR file

The structure for the various types of reports/subsets are defined by "NCEP BUFR Mnemonic Table" text files when using the NCEP BUFRLIB software. Again, the tables defining the report structures are encoded in special "dictionary" messages at the top of the file.

An excerpt from PrepBUFR mnemonic table for ADPUPA (upper-air) message type (see http://www.emc.ncep.noaa.gov/mmb/data_processing/Obs_group_roundtable.doc/prepbufr_table.htm)

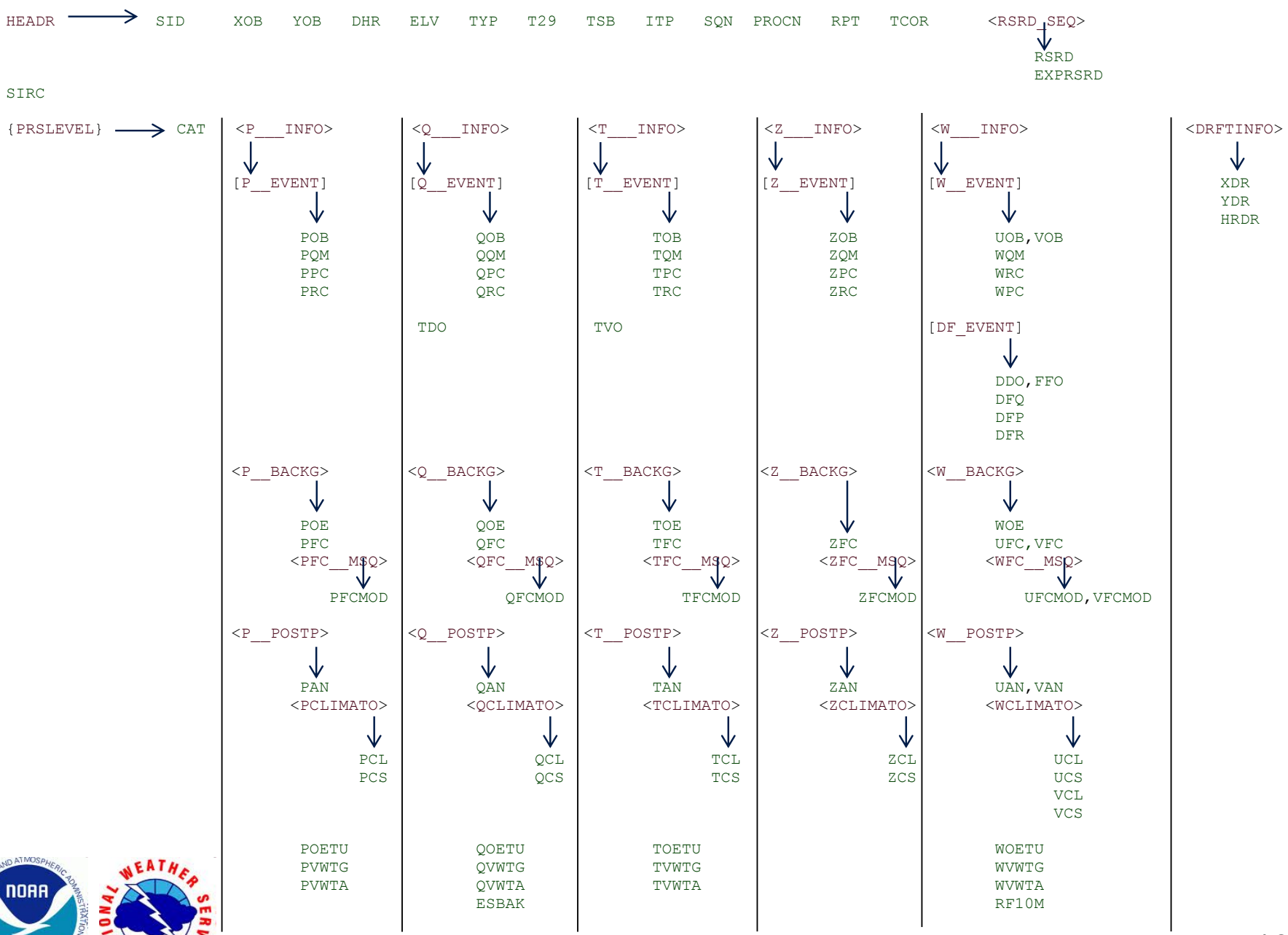
```
| ADPUPA   | HEADR   SIRC        {PRSLEVEL}  <SST_INFO>  <PREWXSEQ>  {CLOUDSEQ} |
| ADPUPA   |        <CLOU2SEQ>  <SWINDSEQ>  <AFIC_SEQ>  <TURB3SEQ>             |
```

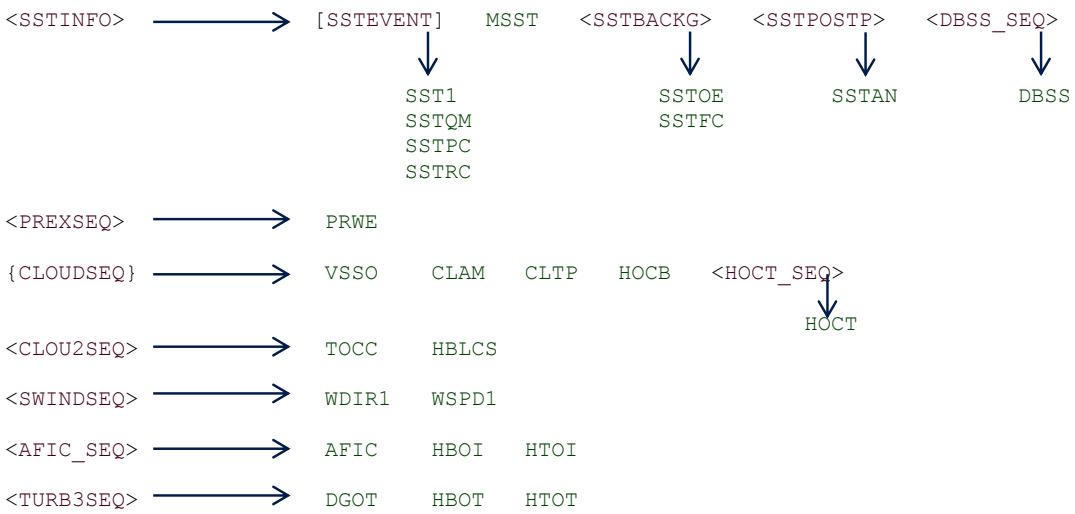
SIRC is a mnemonic defining the data (BUFR Table B) descriptor defined as "Solar and infrared radiation correction" (a code table value defined externally).

HEADR, **PRSLEVEL**, **SST_INFO**, **PREWXSEQ**, **CLOUDSEQ**, **CLOU3SEQ**, **SWINDSEQ**, **AFIC_SEQ** & **TURB3SEQ** are mnemonics defining sequence (BUFR Table D) descriptors which expand further into other sequence and data mnemonics (see next two slides).

Excellent references for NCEP BUFR TABLES:
http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/toc/dfbftab/

http://www.emc.ncep.noaa.gov/mmb/data_processing/NCEP_BUFR_File_Structure.htm


HEADR
SID
XOB
YOB
DHR
ELV
TYP
T29
TSB
ITP
SQN
PROCN
RPT
TCOR
<RSRD_SEQ>
RSRD
EXPRSRD
SIRC
{PRSLEVEL}
CAT
<P___INFO>
[P__EVENT]
POB
PQM
PPC
PRC
<P__BACKG>
POE
PFC
<PFC__MSQ>
PFCMOD
<P__POSTP>
PAN
<PCLIMATO>
PCL
PCS
POETU
PVWTG
PVWTA
<Q___INFO>
[Q__EVENT]
QOB
QQM
QPC
QRC
TDO
<Q__BACKG>
QOE
QFC
<QFC__MSQ>
QFCMOD
<Q__POSTP>
QAN
<QCLIMATO>
QCL
QCS
QOETU
QVWTG
QVWTA
ESBAK
<T___INFO>
[T__EVENT]
TOB
TQM
TPC
TRC
TVO
<T__BACKG>
TOE
TFC
<TFC__MSQ>
TFCMOD
<T__POSTP>
TAN
<TCLIMATO>
TCL
TCS
TOETU
TVWTG
TVWTA
<Z___INFO>
[Z__EVENT]
ZOB
ZQM
ZPC
ZRC
<Z__BACKG>
ZFC
<ZFC__MSQ>
ZFCMOD
<Z__POSTP>
ZAN
<ZCLIMATO>
ZCL
ZCS
<W___INFO>
[W__EVENT]
UOB,VOB
WQM
WRC
WPC
[DF_EVENT]
DDO,FFO
DFQ
DFP
DFR
<W__BACKG>
WOE
UFC,VFC
<WFC__MSQ>
UFCMOD,VFCMOD
<W__POSTP>
UAN,VAN
<WCLIMATO>
UCL
UCS
VCL
VCS
WOETU
WVWTG
WVWTA
RF10M
<DRFTINFO>
XDR
YDR
HRDR



Here, **RED** mnemonics represent sequences (BUFR Table D) & **GREEN** mnemonics represent data values (BUFR Table B).

Here is the same ADPUPA message type as on slide 18, now expanded out to its full set of component data mnemonics:

```
| ADPUPA   | SID XOB YOB DHR ELV TYP T29 TSB ITP SQN PROCN RPT TCOR <RSRD      |
| ADPUPA   | EXPRSRD> SIRC {CAT <[POB PQM PPC PRC] <POE PFC <PFCMOD>> <PAN     |
| ADPUPA   | <PCL PCS> POETU PVWTG PVWTA>> <[QOB QQM QPC QRC] TDO <QOE QFC     |
| ADPUPA   | <QFCMOD>> <QAN <QCL QCS> QOETU QVWTG QVWTA ESBAK>> <[TOB TQM TPC  |
| ADPUPA   | TRC] TVO <TOE TFC <TFCMOD>> <TAN <TCL TCS> TOETU TVWTG TVWTA>>    |
| ADPUPA   | <[ZOB ZQM ZPC ZRC] <ZFC <ZFCMOD>> <ZAN <ZCL ZCS>>> <[UOB VOB WQM  |
| ADPUPA   | WPC WRC] [DDO FFO DFQ DFP DFR] <WOE UFC VFC <UFCMOD VFCMOD>> <UAN |
| ADPUPA   | VAN <UCL UCS VCL VCS> WOETU WVWTG WVWTA RF10M >> <XDR YDR HRDR>}  |
| ADPUPA   | <[SST1 SSTQM SSTPC SSTRC] MSST <SSTOE SSTFC> <SSTAN> <DBSS>>      |
| ADPUPA   | <PRWE> {VSSO CLAM CLTP HOCB <HOCT>} <TOCC HBLCS> <WDIR1 WSPD1>    |
| ADPUPA   | <AFIC HBOI HTOI> <DGOT HBOT HTOT>                                 |
```

It is long and detailed! The NCEP BUFRLIB compacts this into just the single descriptor defining the sequence mnemonic "ADPUPA". Thus, only one value is encoded in the descriptor section (3) of each ADPUPA BUFR message!

Replication: efficient data storage in BUFR

Two main types of replication:

- Fixed (or standard): the number of repetitions for a given sequence is always known ahead of time and is the same for every subset in a message
  - Example: satellite data from a fixed number of channels; a set number of quality marks
- Delayed: the number of repetitions for a given sequence is not known ahead of time and may vary for subsets in a message
  - Example: pressure levels in a radiosonde report

Only sequence mnemonics can be replicated in NCEP BUFRLIB!

Standard Replication/NCEP BUFR

- A sequence that is repeated a fixed number of times appears in the BUFR tables surrounded by quotation marks followed by the number of replications

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                                                         |

| GQCPRMS  | OGCE  GNAP   PCCF                                                 |
```

Here, sequence GQCPRMS (OGCE GNAP PCCF) is repeated 3 times.
   OGCE = Originating Center
   GNAP = Generating Application
   PCCF = Percent Confidence

- Fixed replication appears rarely, if ever, in the PrepBUFR table. However, it does occur in many of the tank and dump tables.

Delayed Replication/NCEP BUFR

< SEQ_MNEMONIC > ➔ one-bit delayed replication:

- The sequence is repeated either zero or one time(s)
- Represents a way to turn “on” or “off” a set of data values in the sequence
- Only 1 bit is needed to represent all data values in the sequence being missing (0 replications) rather than filling each value in the sequence with all bits “on” (which represents missing in BUFR)
  - This is an efficient way to store data sequences that are always missing (e.g., PIBAL mass data in ADPUPA message type)

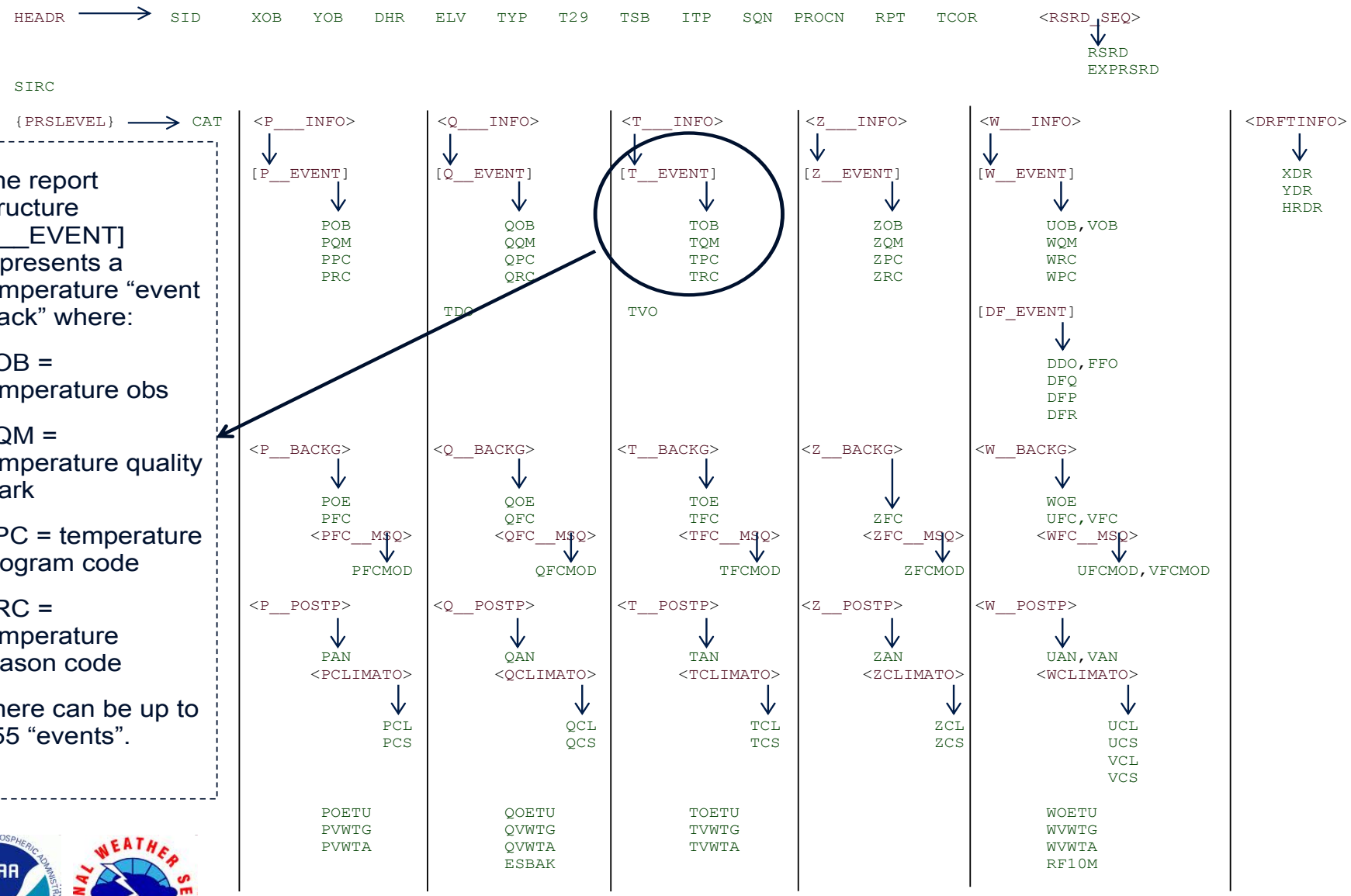
{ SEQ_MNEMONIC } or [SEQ_MNEMONIC] ➔ 8-bit delayed replication:

- The sequence can be repeated up to 255 times
- { } represent regular data replications (e.g., profile levels, radiance channels)
- [ ] represent replicated (stacked) “events” (only in PrepBUFR files)
  - Each event normally consists of a sequence representing the obs value & its quality mark, along with a program code & a reason code (providing metadata on what module updated the obs/quality mark and why it did what it did)
- { } and [ ] replications are treated differently by the BUFRLIB software

(SEQ_MNEMONIC) ➔ 16-bit delayed replication:

- Like { } except the sequence can be repeated up to 65,535 times

PrepBUFR event stacks: Previous example of ADPUPA message type



The report structure [T__EVENT] represents a temperature "event stack" where:

TOB = temperature obs

TQM = temperature quality mark

TPC = temperature program code

TRC = temperature reason code

There can be up to 255 "events".

Commonly Used BUFRLIB Routines

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

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

NCEP BUFRLIB Software

- A simple overview is presented in these slides. For more details, see http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/
  - Quality reference for:
    - Purposes/specific uses of major BUFRLIB routines
    - BUFRLIB routine arguments

- Some preliminary information:
  - Previous versions of BUFRLIB required BUFR files to be FORTRAN-blocked
  - The newest version (updated for the NCEP supercomputer move from CCS to WCOSS) will work with EITHER blocked or unblocked BUFR files
    - On WCOSS, all BUFR files are now unblocked (the default)

Pseudo-code for READING

call openbf ! Open BUFR file for reading

do while(ireadmg.eq.0) ! Sequentially read in each BUFR message in the file

  do while(ireadsb.eq.0) ! Sequentially read in each subset in the current message

    depending on data types and their report structures, call either:

    - ufbint
    - ufbevn
    - ufbrep
    - ufbseq

    these routines pull data values from subsets in the BUFR file

  enddo ! ireadsb

enddo ! ireadmg

call closbf ! All messages and subsets read from BUFR file, close it

(for more information, see
http://www.emc.ncep.noaa.gov/mmb/data_processing/decode_only_BUFR_example.txt)

Pseudo-code for WRITING

call openbf ! Open BUFR file for writing

do while … ! Loop through each native-format "report" that is to be encoded into the BUFR file

put report into arrays that allow it to interface with BUFRLIB software

call openmb ! Opens a new BUFR message in output file for writing (if necessary)

depending on data types and their report structures, call either:

- ufbint
- ufbrep
- ufbseq

these routines store data values for report into BUFRLIB memory

call writsb ! Encode report into BUFR message in output file

enddo ! while …

call closbf ! All subsets written into BUFR file, close it

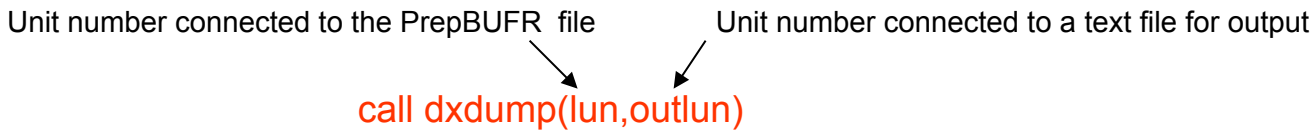
(for more information, see
http://www.emc.ncep.noaa.gov/mmb/data_processing/encode_only_BUFR_example.txt)

Appending data to an existing PrepBUFR file

Before appending any new data to a PrepBUFR file, examine its BUFR table.

Make sure the report structure in the existing file fits the report structure of the data you want to append. The report structure in the table can change from time to time 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 later when using BUFRLIB to decode the data in the file.

If you're not sure of the report structures within the PrepBUFR file, you can use the BUFRLIB routine *dxdump* to extract the table from the dictionary messages at the top of the file.



See http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/toc/other/#dxdump for more details.

Appending data to an existing PrepBUFR file (cont.)

The main coding difference between creating a new PrepBUFR file and appending data to an existing file lies in the second argument to the BUFRLIB subroutine *openbf* (*open BUFR file*):

! Create a new BUFR file for output

call openbf(<unit number connected to PrepBUFR file>,'OUT',<unit number connected to PrepBUFR table>)

vs.

! Append data to an existing BUFR file

call openbf(<unit number connected to PrepBUFR file>,'APX',<unit number connected to PrepBUFR table>)

See http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/toc/intro/#openbf for more details.

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
- Obs/QC Processing Forum (including BUFR):
  http://emc-ls-sand04.ncep.noaa.gov/forum/viewforum.php?f=25&sid
  - 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/

  "Table B" (defines mnemonics that represent various data values):
  http://www.emc.ncep.noaa.gov/mmb/data_processing/bufrtab_tableb.htm
- DTC Docs (BUFR code examples):
  http://www.dtcenter.org/com-GSI/BUFR/examples/index.php

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:
  - BUFRLIB support via online web form:

    http://www.nco.ncep.noaa.gov/sib/decoders/mail_bufrlib/

    Comments and questions sent via this form reach key contacts in EMC and NCO. Exchanges are then made via email to solve problems.
  - DTC GSI helpdesk

    Please send email to gsi_help@ucar.edu

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, BUFR 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. Please ask for help if you need it!

Questions?