

Observation Pre-processor for WRF-Var

Yong-Run Guo

National Center for Atmospheric Research
P.O.Box 3000, Boulder, CO 80307

Presented in WRF-Var Tutorial, 1 February 2010

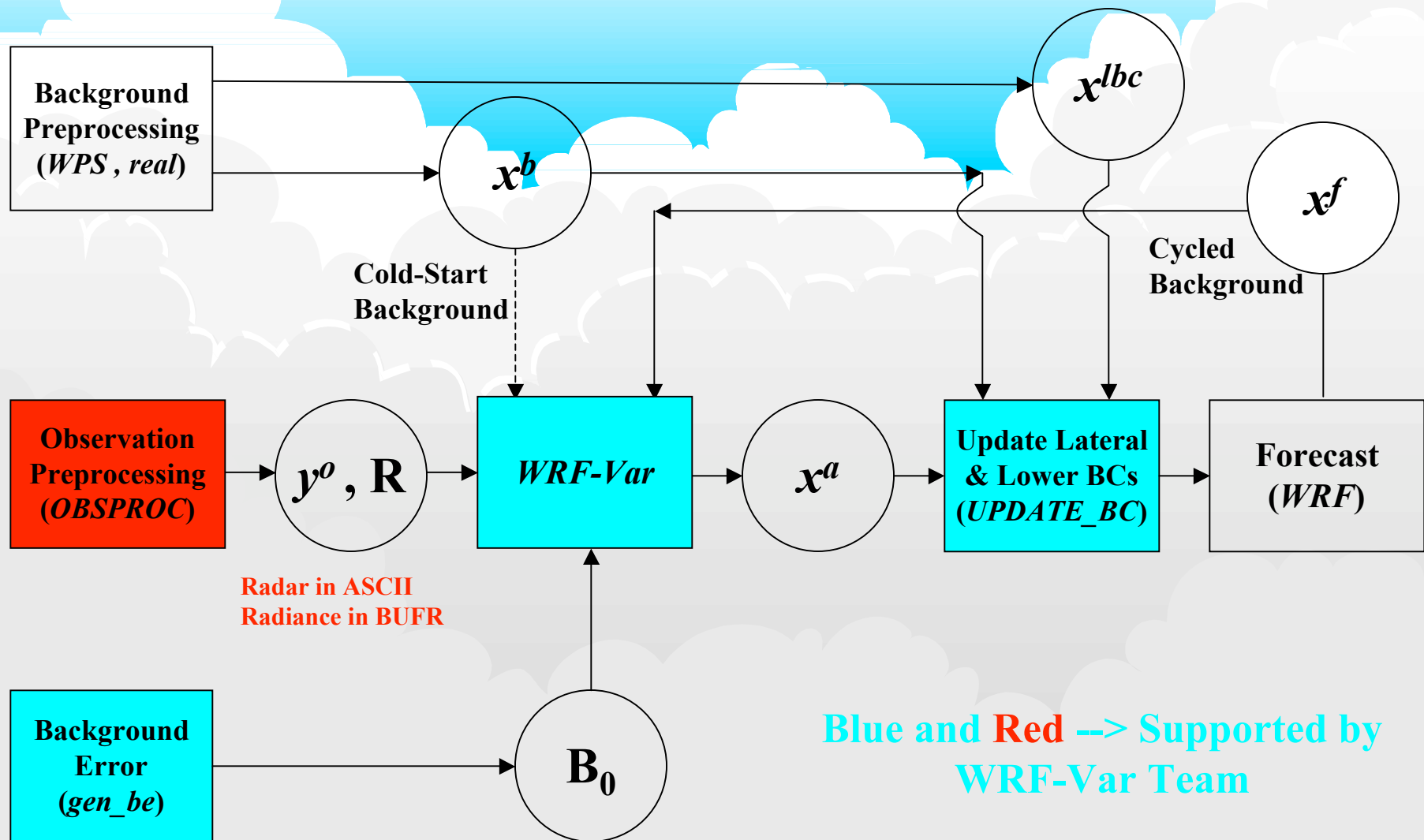
**Thanks to everyone in NCAR/MMM/WRFDA group, especially to
Hui-Chuan Lin, Ruifang Li, Meral Demirtas, Zhiquan Liu,
and Jim Bresch**

Outline

- *obsproc* and WRFVar
- Sources of the Observation data
- the LITTLE_R observation file to WRFDA/*obsproc*
- Output from WRFDA/var/*obsproc*
- Flow chart of OBS preprocessor (*obsproc*)
- How to plot the OBS distribution?
- Introduction to run *obsproc*

obsproc and WRFVar

WRF-Var in the WRF Modeling System



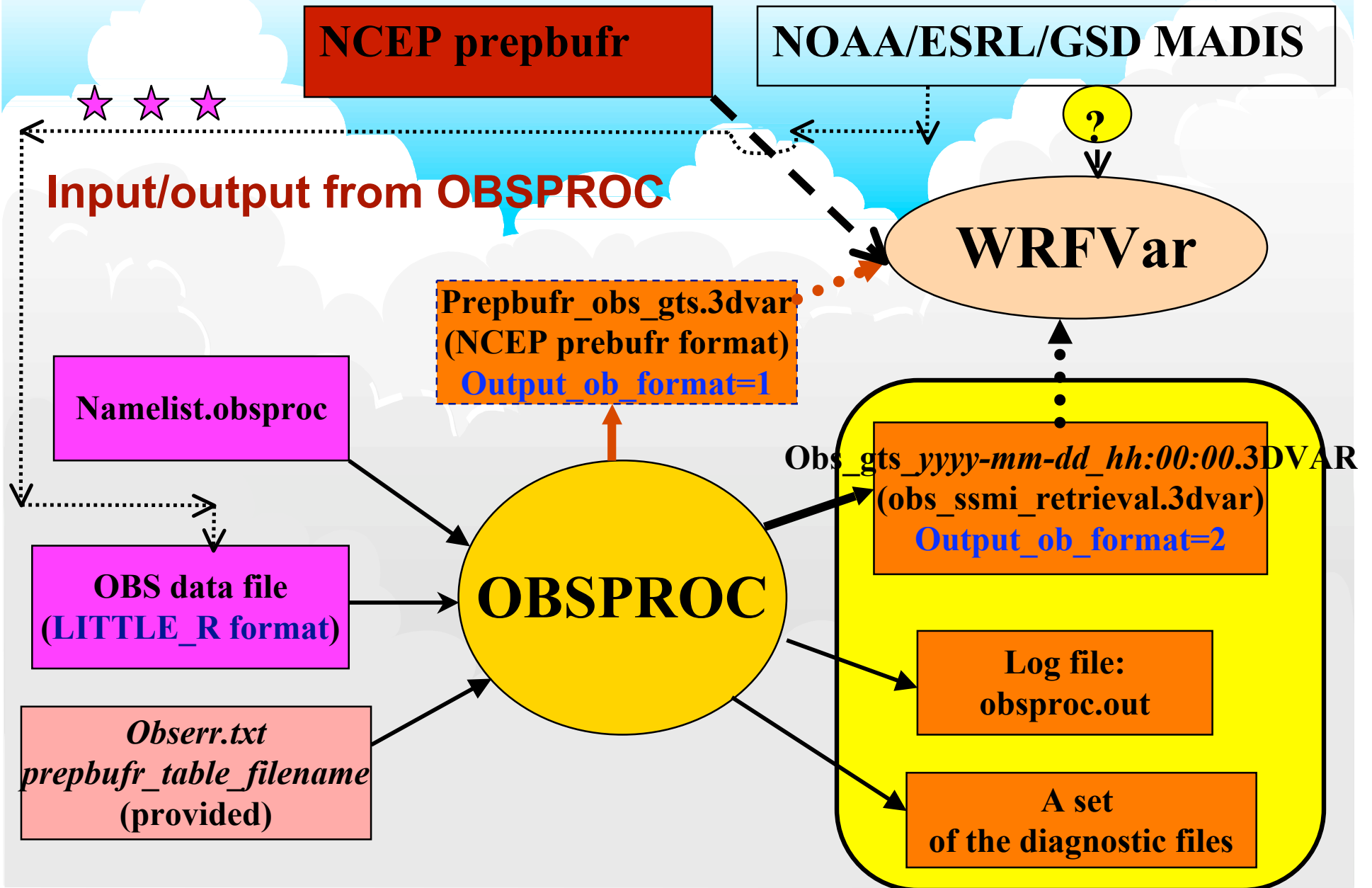
The purpose of obsproc:

to ingest the *intermediate format* (**LITTLE_R**) OBS data file and prepare the *OBS data file suitable for WRF-Var* needs (3DVAR, FGAT, 4DVAR, etc.)

Basic function of the obsproc:

- Screening of the conventional observations (time window, domain, duplication, etc.), and keep the necessary information for WRF-Var assimilation
- Assign the observation errors to each of the observations
- Do the basic quality control (gross check and consistent check)
- Save *a processed OBS file*, which can be repeatedly used for multiple times of WRF-Var experiments.

**** to avoid to use any of the model meteorological fields in obsproc.**



Sources of the Observation data for WRF-Var

There are 3 observation data sources available:

- NCEP prepbufr files: real-time and archived;

Tested and used for ASR project (`ob_format=1`)

- NOAA/ESRL/GSD MADIS files: real-time and archived;

Not fully tested (`ob_format=3 ?`)

- NCAR archived observation data files (LITTLE_R format)
via obsproc (`ob_format=2`)

The impact of the different observation data sources on the final analysis should be investigated.

prepbufr observation data from NOAA ftp site:

You can download the NCEP real-time *prepbufr* observation data from

<ftp://ftp.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/>

There are data within the most recent 24 hours. The *prepbufr* observation files are *gdas1.thhz.prepbufr.unblok.nr* and *gdas1.thhz.prepbufr.nr*

More complete archived data can be found from NCAR,

<http://dss.ucar.edu/datasets/ds337.0>

ASR project people ([Hui-Chuan Lin](#), [Zhiquan Liu](#), et al.) has more experiences in using this data sources.

Dr. Zhiquan Liu also have the *prepbufr* data archived in NCAR MSS from 2006071800Z under his own directory. If users have the account in NCAR machines, they can get the data from NCAR MSS.

Or you can download the archived prepbuf observation data from NCAR MSS:

- ❖ `mss:/LIUZ/GDAS/yyyymm/yyyymddhh/gdas1.thhz.prepbuf.unblok.nr`
(using a script: *cwordsh* to add the blocking information to the BUFR file in Linux/PGI)

The `cwordsh.tar` file can be downloaded from

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

(See README_cwordsh after un-tarred the `cwordsh.tar` file).

To add the blocking information, you can run in Bourne shell

```
cwordsh "block" das1.thhz.prepbuf.unblok.nr das1.thhz.prepbuf.block.nr
```

Or for c-shell, it is

```
sh ./cwordsh "block" das1.thhz.prepbuf.unblok.nr \ das1.thhz.prepbuf.block.nr
```

The file: `das1.thhz.prepbuf.block.nr` can be used as the input file as *ob.buf*.

- ❖ `mss:/LIUZ/GDAS/yyyymm/yyyymddhh/gdas1.thhz.prepbuf.nr`
(can be directly used by machine IBM)

How to use the prebufr observations in WRF-Var?

Make sure WRFDA code is compiled with *setenv BUFR 1*

```
ln -sf gdas1.t${hh}z.prebufr.nr      ob.bufr
```

```
ln -sf gdas1.t${hh}z.gpsro.tm00.bufr_d  gpsro.bufr ← BUFR file for GPSRO
```

namelist.input

```
&wrfvar3
```

```
    ob_format = 1      ← prebufr observation file
```

```
&wrfvar4
```

```
    thin_mesh_conv = 20.0, 20.0, 20.0, ... ← thinning  
    with 20km distance
```

```
&wrfvar21
```

```
    time_window_min = 2002-08-02_22:30:00
```

```
&wrfvar22
```

```
    time_window_max = 2002-08-03_01:30:00
```

For more details, to see

<https://wiki.ucar.edu/display/~hclin/prebufr2wrfva>
r

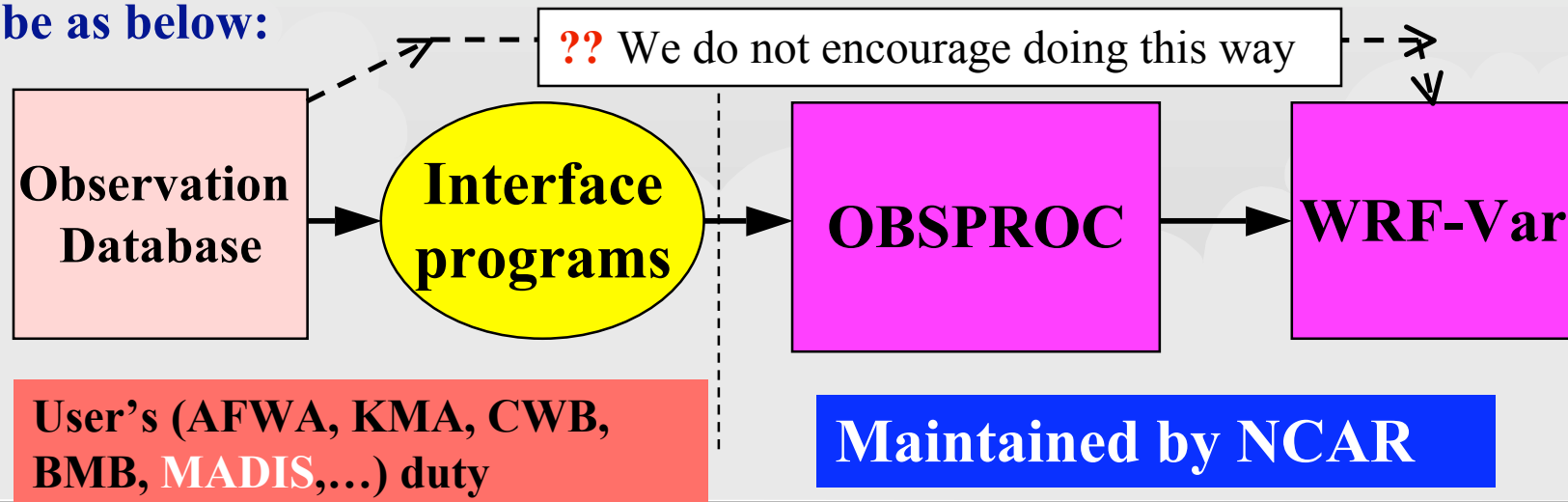
NOAA/ESRL/GSD MADIS (files available back to **July 1, 2001**)

The interface program to ingest the MADIS data to WRF-Var was developed by Michael F. Barth, NOAA/GSD. The information for MADIS data can be found from

<http://madis.noaa.gov/>

NCAR is merging the code to WRFDA released code as `ob_format=3`, but it is still in testing. We cannot guarantee it is working properly in this version of WRFDA.

The better way to ingest OBS data from a database to WRF-Var should be as below:



MADIS stands for *Meteorological Assimilation Data Ingest System*.

The source code can be obtained at

http://madis.noaa.gov/madis_api.html

From the “*Software Download*” section on that page, click “*Source Code-Only Package*”, the `madis-3.7.tar.gz` file will be downloaded. After un-tarred this file, you can find the files: *doc/README* and *doc/INSTALL.unix*, which contain all the information about how to have access to observation data from MADIS, and how to install the MADIS software.

To request real-time or archive access to any of MADIS datasets, please fill out the application form at

http://madis.noaa.gov/data_application.html

Recently, NCAR/MMM has developed the converter program to transfer the MADIS data to LITTLE_R format observation file.

As requested by AFWA project, the converted and tested observation types are

METAR,
Canadian SAO's,
ACARS,
SATWND,
Marine,
radiosonde.

It is possible to convert more MADIS observations to LITTLE_R file. If people is interested in doing it, please contact lir@ucar.edu (Ruifang Li), or look at the web pages:

https://wiki.ucar.edu/display/~lir/MADIS+to+little_r+converter

or

https://wiki.ucar.edu/display/mmm/MADIS_2_little_r_2_OBSPROC

NCAR archived observation files

- The NCEP ADP observation data (upper air from **20 December 1972** to 28 February 2007 and surface data from **10 February 1975** to 28 February 2007) may also be downloaded.

http://www.mmm.ucar.edu/mm5/mm5v3/data/free_data.html

Then use a MM5 utility FETCH to convert these ADP data to **LITTLE_R** format data.

Note: the **ADP_observation** files stored in NCAR MSS are COS-Blocked file. Nowadays, NCAR no longer supports this cosblock stripping transfer to/from MSS. Before using FETCH, you must use two steps to get the cosblock files as below:

```
msrcp msws:/DSS/file1 file1  
cosconvert -b file1
```

The utility “cosconvert”, etc. can be downloaded from

http://dss.ucar.edu/libraries/io/cos_blocking/utls

- Conventional **LITTLE_R** observation data can also be downloaded from NCAR MSS:

mss:/BRESCH/RT/DATA/yyyymm/obs/yyyymmddhh.gz
(available starting from **2003040800Z**, every 6 hours)

mss:/RTFCST/ARCHIVE/RAIN/init/yyyymmddhh.tar (available
starting from **1999012400Z to 2004072612Z**, every 12
hours)

the LITTLE_R observation file to WRFDA/obsproc

Input obs file: LITTLE_R format

- The *input file* to OBSPROC is the LITTLE_R format (see below)
This is a **report-based** file, so all types of the observation data can be easily 'ca'ted together to form a monolithic file, which is an ASCII file and easily to read, edit,.....
- Users' duty is just to develop **an interface program** to convert their own observations in any format to the LITTLE_R(ASCII) file. The LITTLE_R obs file can be processed by OBSPROC, and an observation file suitable for WRFVar assimilation will be produced.
- The available NCEP ADP observation data in NCAR/MSS have the longest archive history: upper air since 1972, and surface since 1975.

Structure of the input OBS (LITTLE_R) file

- *OBS decoded file* in LITTLE_R format is composed of the *Reports*
- *Report* is composed of the *Records* (*header, data, ..., and ending*) and 3 *tail integers* (3I7): -888888-888888-888888
- *Record* is composed of the *fields*
 - The *fields* in the *header record* (Fortran format in parenthesis)
 - The *fields* in the *data record* (Fortran format in parenthesis)
 - The *fields* in the *ending record*

The details for each of *records* are described below:

The fields in the header record:

No	Field	No	Field	No	Field
1	Latitude (f20.5)	2	Longitude (f20.5)	3	ID (a40)
4	Name (a40)	5	Platform (a40)	6	Source (a40)
7	Elevation (f20.5)	8	Num_vld_fld (i10)	9	Num_error (i10)
10	Num_warning (i10)	11	Seq_num (i10)	12	Num_dupd (i10)
13	Is_sound (L10)	14	Bogus (L10)	15	Discard (L10)
16	Valid_time%sut (i10)	17	Valid_time%julian (i10)	18	Valid_time%date_char(a20)
19	Slp%data (f13.5)	20	Slp%qc (i7)	21	Ref_pres%data (f13.5)
22	Ref_pres%qc (i7)	23	Ground_t%data (f13.5)	24	Ground_t%qc (i7)
25	SST%data (f13.5)	26	SST%qc (i7)	27	Psfc%data (f13.5)
28	Psfc%qc (i7)	29	Precip%data (f13.5)	30	Precip%qc (i7)
31	T_max%data (f13.5)	32	T_max%qc (i7)	33	T_min%data (f13.5)
34	T_min%qc (i7)	35	T_min_night%data (f13.5)	36	T_min_night%qc (i7)
37	P_tend03%data (f13.5)	38	P_tend03%qc (i7)	39	P_tend24%data (f13.5)
40	P_tend24%qc (i7)	41	Cloud_cvr%data (f13.5)	42	Cloud_cvr%qc (i7)
43	Celling%data (f13.5)	44	Celling%qc (i7)	45	Pw%data (f13.5)
46	Pw%qc (i7)	47	Tb19v%data (f13.5)	48	Tb19v%qc (i7)
49	Tb19h%data (f13.5)	50	Tb19h%qc (i7)	51	Tb22v%data (f13.5)
52	Tb22v%qc (i7)	53	Tb37v%data (f13.5)	54	Tb37v%qc (i7)
55	Tb37h%data (f13.5)	56	Tb37h%qc (i7)	57	Tb85v%data (f13.5)
58	Tb85v%qc (i7)	59	Tb85h%data (f13.5)	60	Tb85h%qc

The fields in the data record (Fortran format in parenthesis)

No	Field	No	Field
1	Pressure%data (f13.5)	2	Pressure%qc (i7)
3	Height%data 9f13.5)	4	Height%qc (i7)
5	Temperature%data (f13.5)	6	Temperature%qc (i7)
7	Dew_point%data (f13.5)	8	Dew_point%qc (i7)
9	Speed%data (f13.5)	10	Speed%qc (i7)
11	Direction%data (f13.5)	12	Direction%qc (i7)
13	U%data (f13.5)	14	U%qc (i7)
15	V%data (f13.5)	16	V%qc (i7)
17	RH%data (f13.5)	18	RH%qc (i7)
19	Thickness%data (f13.5)	20	Thickness%qc (i7)

The fields in the ending record

No	field	No	field	No	field	No	field
1	-777777.00000	2	0	3	-777777.00000	4	0
5	-888888.00000	6	0	7	-888888.00000	8	0
9	-888888.00000	10	0	11	-888888.00000	12	0
13	-888888.00000	14	0	15	-888888.00000	16	0
17	-888888.00000	18	0	19	-888888.00000	20	0

Remarks

1. The tail fields in the header record are not need to be all filled in. For example, if no SSMI Tb (brightness temperature) available, the header record may only have 46 fields.
2. For certain type of observations, the some of the fields in data record are just used as the storage, not the actual data as the field's name. For example, for QuikScat SeaWind, the fields: U%data and V%data are used to store the observation errors of speed and direction, respectively. If there is no observation errors available, the missing value of -88.0 should be assigned to U%data and V%data.
3. For certain types of observations, such as GPSREF, etc., the observation data are not the wind, temperature, moisture, etc., so specific arrangements are made with the fields to hold the refractivity, perigee point location, etc.

GPS RO data format

Content of the level record in little_r file:

Press .	Geo height	Temp .	Dew-p	speed	Dir.	u	v	rh	thick
Miss.	height	miss	Refractivity	Impact parameter	Azimuth angle	latitud e	longitude	Bending angle	Opt. bending

The units of parameters for GPSRO data in little_r file:

press	latitude	longitud e	height	temp	Refractivity	Azimuth angle	Impact parameter *1.e-3	Bending angle*1.e7
Ref. Atmos	N	E	m	miss	N	Deg.		rad

Output from WRFDA/var/obsproc

1. NCEP *prepBUFR* format (partly tested yet!)

In obsproc namelist.obsproc

&record9

prepbufr_output_filename='prepbufr_obs_gts.3dvar'

prepbufr_table_filename='prepbufr_table_filename'

output_ob_format=1 (or 3)

In WRFVar namelist.input,

&wrfvar3

ob_format=1,

The prepbufr file is a binary file, and there are block and unblock files. See

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

2, ASCII format --- Easy to manipulate: read, edit, etc. and *endian* independent (fully supported)

In obsproc namelist.obsproc

&record9

output_ob_format=2 (or 3)

```
;      Select the obs_gts (ASCII) files used for 3DVAR, FGAT,  
      and 4DVAR:  
      use_for      = '3DVAR', ; '3DVAR' obs file, same as  
                           before, default  
                           ; 'FGAT ' obs files for FGAT  
                           ; '4DVAR' obs files for 4DVAR  
      ;      num_slots_past and num_slots_ahead are used ONLY for  
FGAT      and 4DVAR:  
      num_slots_past    = 3, ; the number of time slots before  
                           time_analysis  
      num_slots_ahead   = 3, ; the number of time slots after  
                           time_analysis
```

In WRFVar namelist.input,

&wrfvar3

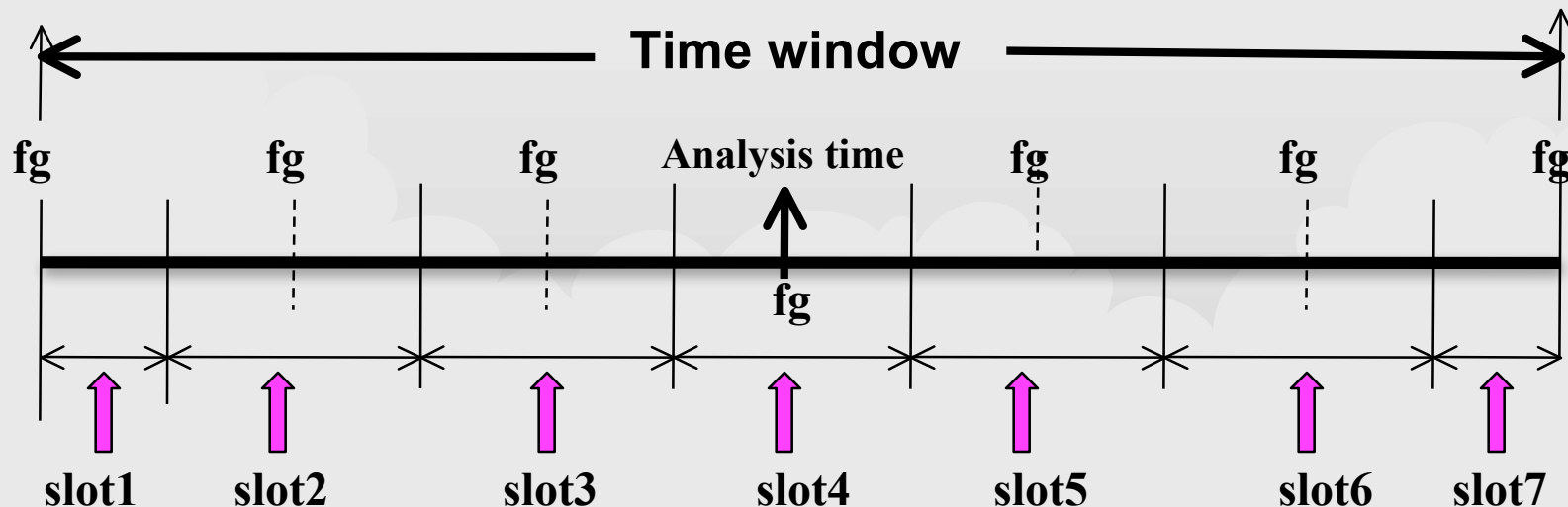
ob_format=2,

3DVAR, FGAT, and 4DVAR have the different requirements:

3DVAR ---- not allowed the time duplicate observation **within time window**

FGAT ---- multiple time slots within the time window, but not allowed the time duplicate observation **within time window**
(*First Guess at Appropriate Time*)

4DVAR ---- multiple time slots within the time window, but not allowed the time duplicate observation **within each of the time slots**.



Output filenames for WRFVar

For 3DVAR,

obs_gts_YYYY-mm-dd_hh:00:00.3DVAR

For FGAT,

obs_gts_YYYY-mm-dd_hh:00:00.FGAT

For 4DVAR,

obs_gts_YYYY-mm-dd_hh:00:00.4DVAR

1, *Obs_gts_yyyy-mm-dd_hh:00:00.3DVAR* and *obs_ssmi_retrieval.3dvar*

Header: the information for this OBS file and data format

Data : *header* record and *data* records for each of levels

- These are the OBS input file to WRF-Var program
- *obs_ssmi_retrieval.3dvar* needed only when SSMI retrieval data (Sea surface wind speed and PW) available
- These files can be used as input to MAP_plot to obtain the gmeta plot file with NCAR GRAPHICS

2, *obsproc.out* ---- a program execution log file

The printing out from the program execution. It can be used to monitor the program execution and to identify the troubles if any

3, Diagnostic files depended on the print switches in namelist

obsproc.exe >&! obsproc.out

File: *obs_gts_yyyy-mm-dd_hh:00:00.3DVAR*

```
TOTAL = 8169, MISS. = -888888.,
SYNOP = 1432, METAR = 164, SHIP = 86, BUOY = 0, TEMP = 179, AMDAR = 0,
AIREP = 265, PILOT = 0, SATEM = 0, SATOB = 6043, GPSPW = 0, SSMT1 = 0,
SSMT2 = 0, TOVS = 0, QSCAT = 0, PROFL = 0, OTHER = 0,
PHIC = 28.50, XLONC = 116.00, TRUE1 = 10.00, TRUE2 = 45.00, XIM11 = 1.00, XJM11 = 1.00,
TSO = 275.00, TLP = 50.00, PTOP = 7000., PS0 = 100000.,
IXC = 67, JXC = 81, IPROJ = 1, IDD = 1, MAXNES = 10,
NESTIX = 67, 67, 67, 67, 67, 67, 67, 67, 67, 67,
NESTJX = 81, 81, 81, 81, 81, 81, 81, 81, 81, 81,
NUMC = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
DIS = 135.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
NESTI = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
NESTJ = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
INFO = PLATFORM, DATE, NAME, LEVELS, LATITUDE, LONGITUDE, ELEVATION, ID.
SRFC = SLP, PW (DATA, QC, ERROR).
EACH = PRES, SPEED, DIR, HEIGHT, TEMP, DEW PT, HUMID (DATA, QC, ERROR)*LEVELS.
INFO_FMT = (A12, 1X, A19, 1X, A40, 1X, I6, 3(F12.3, 11X), 6X, A5)
SRFC_FMT = (F12.3, I4, F7.2, F12.3, I4, F7.2)
EACH_FMT = (3(F12.3, I4, F7.2), 11X, 3(F12.3, I4, F7.2), 11X, 1(F12.3, I4, F7.2)))
```

Number of observations

Model domain information

Data format

```
#-----#
FM-35 TEMP 1999-08-19 11:00:00 AHMADABAD / INDIA 17 23.070 72.630 55.000 42647
-888888.000 -88 200.00 -888888.000 -88 0.20
100000.000 0 100.00 3.000 0 1.10 225.000 0 5.00 53.000 0 7.00 305.550 0 1.00 298.550 0 1.00 65.984 0 15.00
92500.000 0 100.00 3.000 0 1.10 220.000 0 5.00 745.000 0 7.48 300.550 0 1.00 295.550 0 1.00 73.714 0 12.60
91600.000 0 100.00 -888888.000 -88 1.10 -888888.000 -88 5.00 830.000 0 7.54 300.150 0 1.00 295.150 0 1.00 73.653 0 12.30
87300.000 0 100.00 -888888.000 -88 1.10 -888888.000 -88 5.00 1251.000 0 7.84 297.350 0 1.00 291.350 0 1.00 68.712 0 10.82
85000.000 0 100.00 6.000 0 1.10 250.000 0 5.00 1498.000 0 8.00 295.350 0 1.00 290.750 0 1.00 74.809 0 10.00
79200.000 0 100.00 -888888.000 -88 1.13 -888888.000 -88 5.00 2104.000 0 8.22 290.750 0 1.00 288.550 0 1.00 86.755 0 10.00
78000.000 0 100.00 -888888.000 -88 1.18 -888888.000 -88 5.00 2238.000 0 8.27 292.950 0 1.00 281.950 0 1.00 48.599 0 10.00
70700.000 0 100.00 -888888.000 -88 1.39 -888888.000 -88 5.00 3076.000 0 8.57 289.950 0 1.00 268.950 0 1.00 23.245 0 10.00
70000.000 0 100.00 2.000 0 1.40 30.000 0 5.00 3168.000 0 8.60 -888888.000 -11 1.00 -888888.000 -11 1.00 -888888.000 -11 10.00
61100.000 0 100.00 -888888.000 -88 1.76 -888888.000 -88 5.00 4312.000 3 10.01 282.350 0 1.00 271.350 0 1.00 45.937 0 10.00
60000.000 0 100.00 -888888.000 -88 1.80 -888888.000 -88 5.00 4463.000 3 10.20 281.350 0 1.00 272.350 0 1.00 52.920 0 10.00
57400.000 0 100.00 -888888.000 -88 1.90 -888888.000 -88 5.00 4828.000 3 10.66 279.150 0 1.00 274.150 0 1.00 70.188 0 10.00
52700.000 0 100.00 -888888.000 -88 2.14 -888888.000 -88 5.00 5526.000 3 11.55 275.550 0 1.00 272.650 0 1.00 81.128 0 10.00
50300.000 0 100.00 -888888.000 -88 2.28 -888888.000 -88 5.00 5902.000 3 12.04 273.350 0 1.00 269.650 0 1.00 76.187 0 10.00
50200.000 0 100.00 -888888.000 -88 2.29 -888888.000 -88 5.00 5918.000 3 12.06 273.150 0 1.00 269.450 0 1.00 76.157 0 10.00
50000.000 0 100.00 3.000 0 2.30 95.000 0 5.00 5950.000 0 12.10 272.850 0 1.00 269.250 0 1.00 76.682 0 10.00
46900.000 0 100.00 2.000 0 2.42 100.000 0 5.00 6461.000 3 12.90 270.450 0 1.00 266.250 0 1.00 72.904 0 10.00
FM-35 TEMP 1999-08-19 11:00:00 BOMBAY / SANTACRUZ / INDIA 25 19.120 72.850 14.000 430
```

Types of observations to be processed

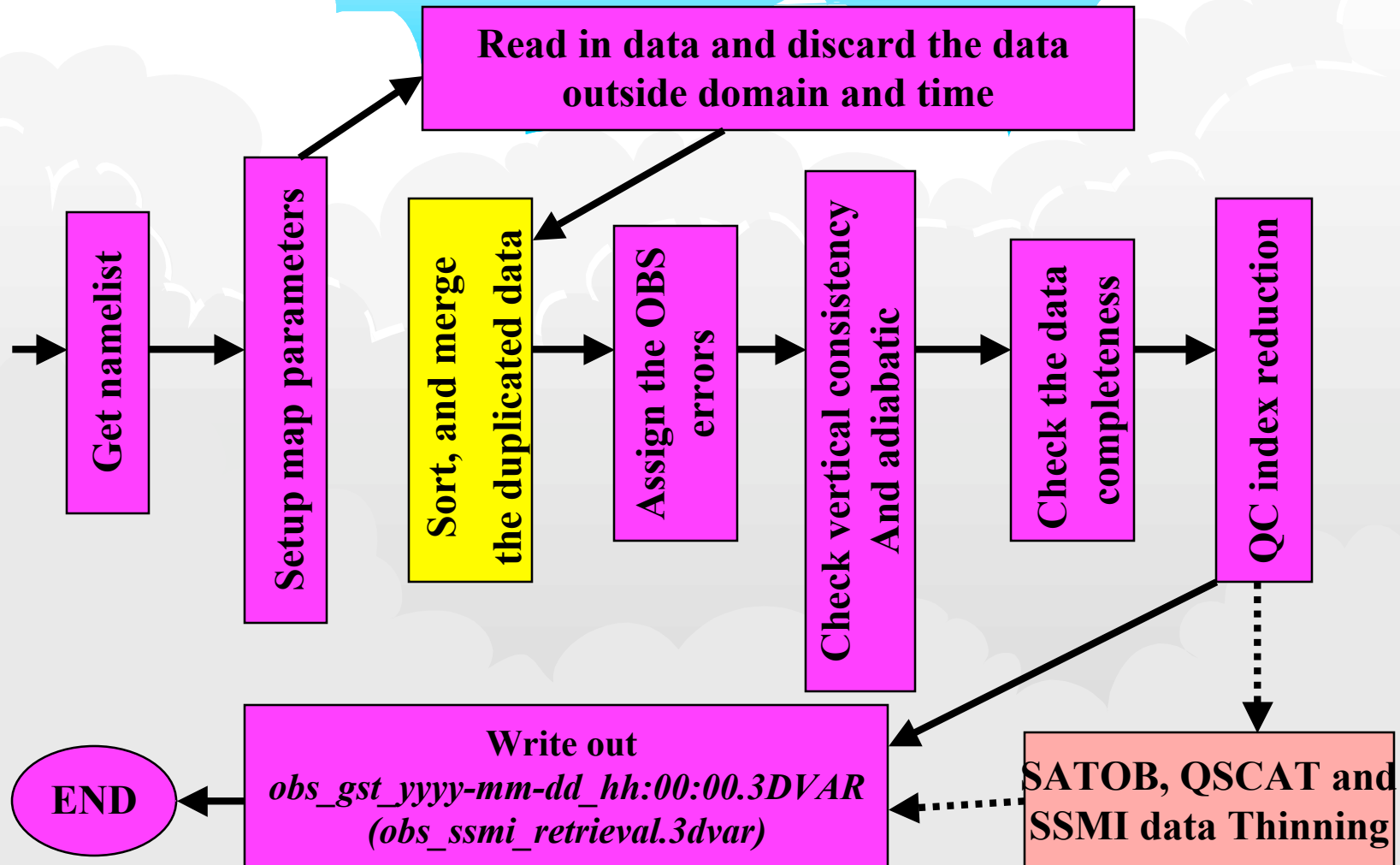
→ 18 types (SYNOP, SHIPS, METAR, TEMP, TAMDAR, AIREP, PILOT, AMDAR, PROFL, SATOB, SATEM, SSMT1, SSMT2, SSMI, GPSPW/GPSZD, GPSRF, QSCAT, BOGUS)

Each type of the observations is identified by its WMO code in WRF-Var. If the standard WMO code is not available to a **new** data type, user should assign a 3-digit code to that data type.

WMO code for each type of observations

No.	Name	WMO code	WMO code name
1	SYNOP	12, 14	SYNOP, SYNOP MOBIL
2	SHIP	13	SHIP
3	METAR	15, 16	METAR, SPECI
4	PILOT	32, 33, 34	PILOT, PILOT SHIP, PILOT MOBIL
5	SOUND	35, 36, 37, 38	TEMP, TEMP SHIP, TEMP DROP, TEMP MOBIL
6	AMDAR	42	AMDAR
7	SATEM	86	SATEM
8	SATOB	88	SATOB
9	AIREP	96, 97	AIREP
10	TAMDAR	101	TAMDAR
11	GPSPW	111	GPSPW (Ground-based GPS precipitable water)
12	GPSZD	114	GPSZD (Ground-based GPS Zenith Total Delay)
13	GPSRF	116	GPSRF (Space-based GPS Refractivity)
14	SSMT1	121	SSMT1
15	SSMT2	122	SSMT2
16	SSMI	125	SSMI
17	PROFL	132	WIND PROFILER
18	BOGUS	135	TCBOU (Typhoon bogus), BOGUS (other bogus)
19	QSCAT	281	Quik SCAT level-2B SeaWind
20	OTHER		UNKNOWN

Flow chart of OBS preprocessor for 3DVAR



Tasks of the OBS preprocessor: obsproc

- 1, To perform a time-windowed and, in case of regional application (`domain_check_h = .TRUE.`), geographically-filtered dump of the ingested observations

Currently, the *time-check for observation data was also in WRF-Var assimilation code*, so to select the observation data within a suitable time-window in OBSPROC is not so strict.

For the regional application with the `I PROJ = 1` (Lambert conformal), 2 (Polar Stereographic), or 3 (Mercator), there is a geographic-filtered performed based on the model domain settings. *For the global application of WRF-Var, it should set `I PROJ = 0` and no geographic-filtered is performed.*

Gross check during the data ingestion:

Any mistakes (unexpected, no logical, ...) could be happened in the raw observation data, the data screening through the gross check is a tedious work.

- Ignore the data with the invalid WMO code.
- Any data values in header record > 888887 or < -888887 or $\text{pressure\%data} \leq 0.0$, etc., will be regarded as missing.
- Elevations for SHIP and BUOY data outside the Great Lakes are always set to zero. If the pressure $< 85,000$ Pa for SHIP and BUOY, the data are tossed out.
- Gross pressure/height consistent check based on the reference atmosphere defined by namelist variables: `base_pres`, `base_temp`, and `base_lapse`
- If both pressure and height are missing, the whole data are discarded.
-

Tasks of the OBS preprocessor: OBSPROC (cont.)

2, Sort and merge the duplicated data

- To retrieve the pressure or height based on the observed information with the hydrostatic assumption
- To re-order (from bottom to top) and merge the data reports with the same ***platform, time, and location*** based on the pressure.
- To remove the duplicate reports of observations:
for 3DVAR and FGAT only observations ***nearest to the analysis time*** are kept,
for 4DVAR, the observations ***nearest to the central time of each of the time slots*** are kept.

Tasks of the OBS preprocessor: OBSPROC (cont.)

3, To assign the observation errors to the different types of observations

Sources of the observations errors:

- Directly from the observation reports (GPS PW/ZTD, QSCAT, etc.)
- US Air Force (AFWA) OBS error file: *obserr.txt*
- NCEP OBS error (Parrish and Derber 1992)

4, To perform the quality control (QC) for soundings

- Vertical consistency check: super adiabatic check and wind shear check (qc_test_vert_consistency=.true.)
- Dry convective adjustment (qc_test_convective_adj=.true.)
- To discard the data above the model top ($p < p_{top}$) in the upper-air observations (remove_above_lid = .true.)

Tasks of the OBS preprocessor: OBSPROC (cont.)

8, To complete thinning with the *SATOB*, *SSMI*, and *QSCAT* data

The data points nearest to the model grid-points will be picked up for assimilation for SATOB, SSMI, and QSCAT.

9, To write out the OBS files in ASCII format as the WRF-Var input

→ GTS data (*obs_gts_YYYY-mm-dd_hh:00:00.3DVAR*): pressure, Wind, height, temperature, dew-point, RH, thickness, etc.

→ SSMI data (*obs_ssmi_retrieval.3dvar*): PW and surface wind speed

10, To plot the distribution for each type of observations

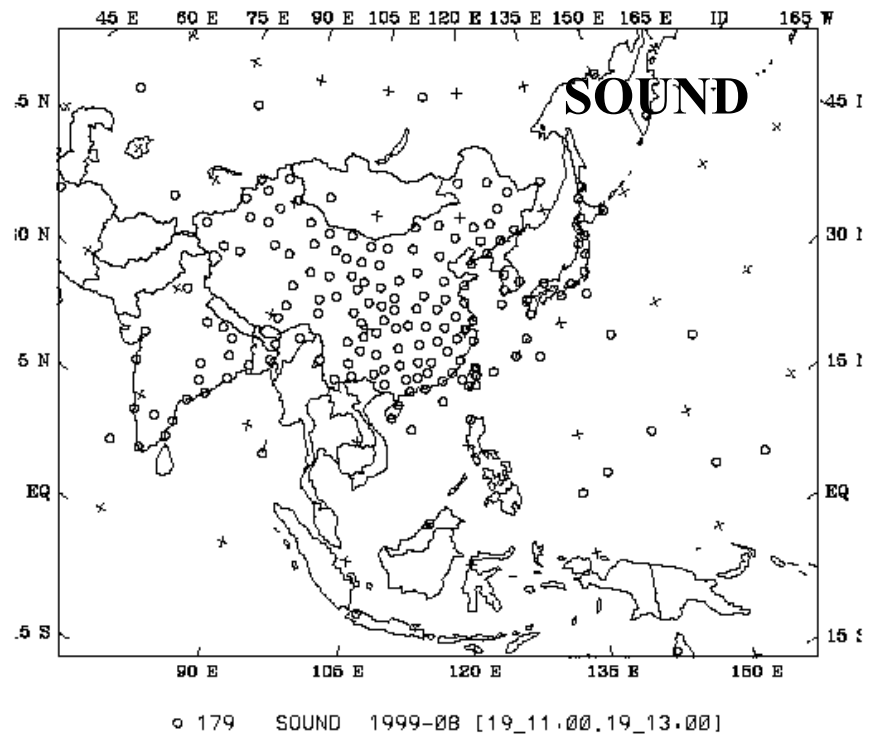
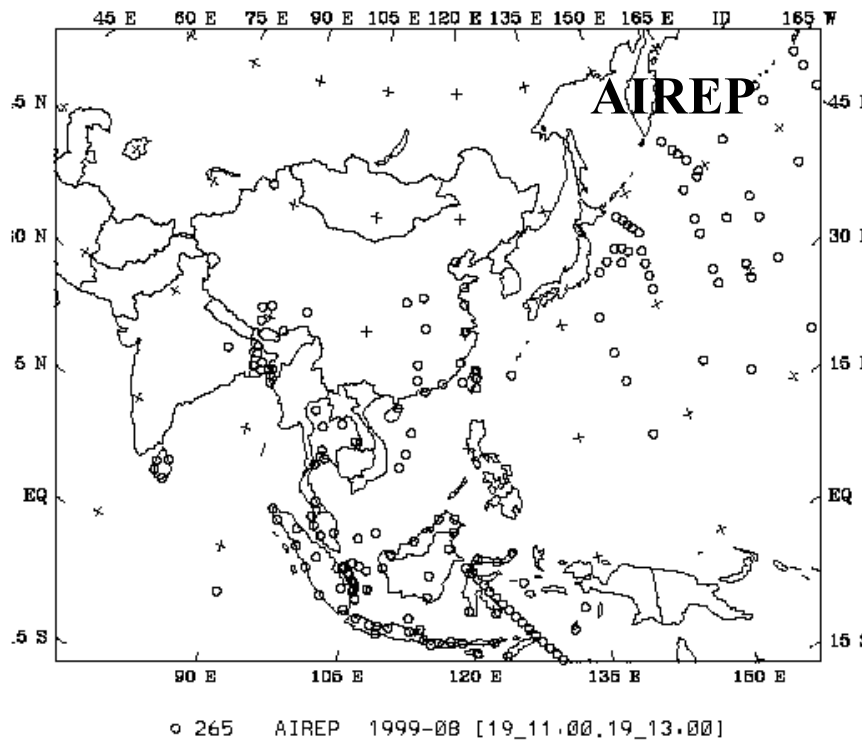
→MAP_plot

How to plot the OBS distribution?

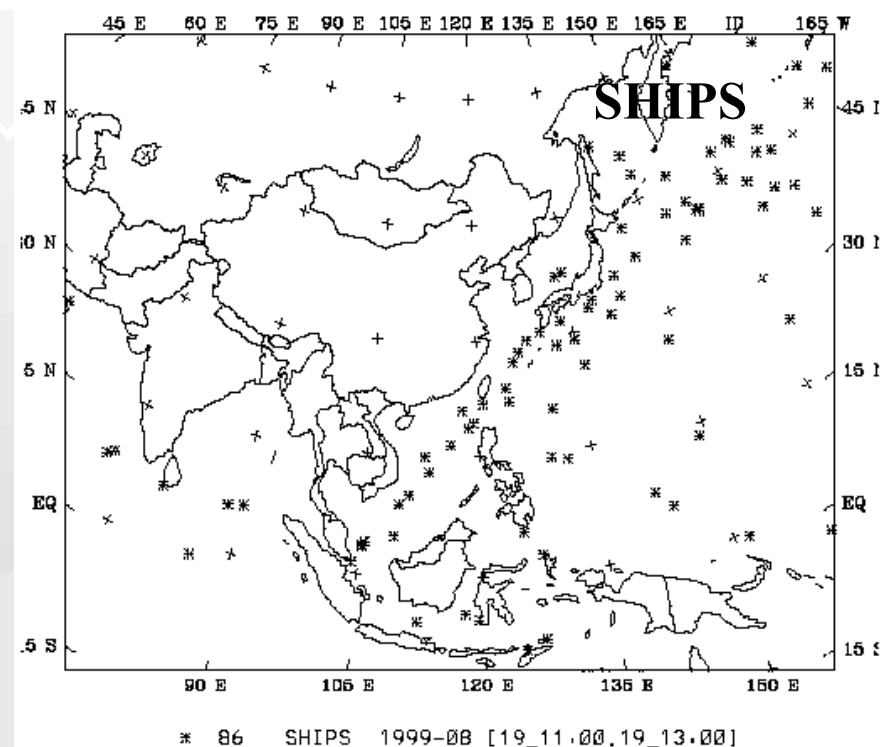
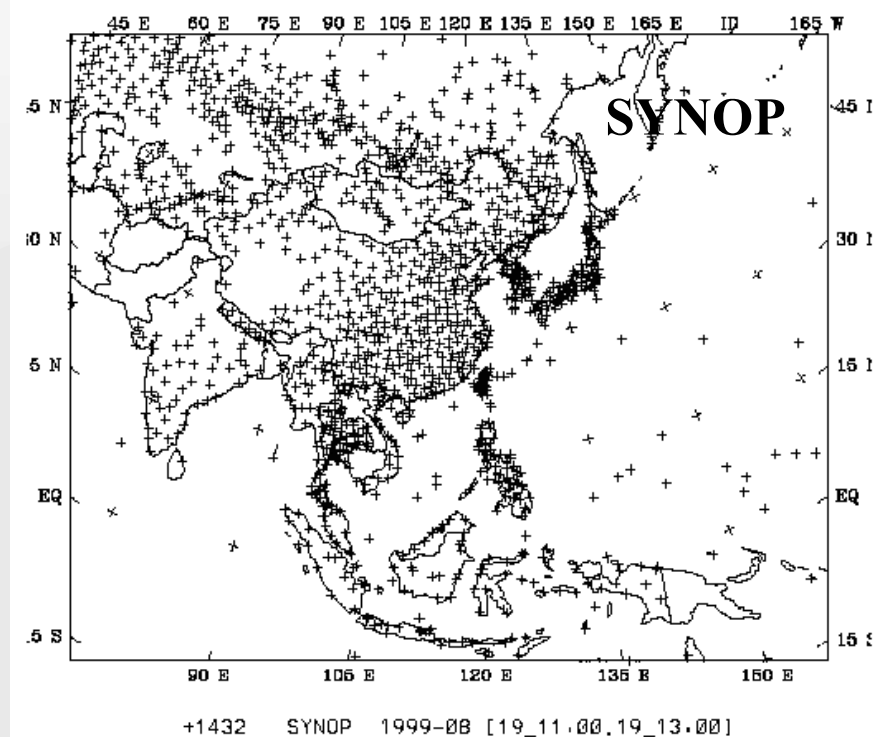
- Go to the directory **WRFDA/var/obsproc/MAP_plot**
- Modify the shell script **Map.csh**
 - » To fill in **TIME_ANALYSIS**, etc., and **OBSDATA** file name
- Run shell script **Map.csh**
 - » You will have a gmeta file: **gmeta.\${TIME_ANALYSIS}** to show the the distribution of observations contained in **OBSDATA** file.

Or copy the **Map.csh** from **WRFDA/var/obsproc/MAP_plot** to your working directory, edit it, and run it.

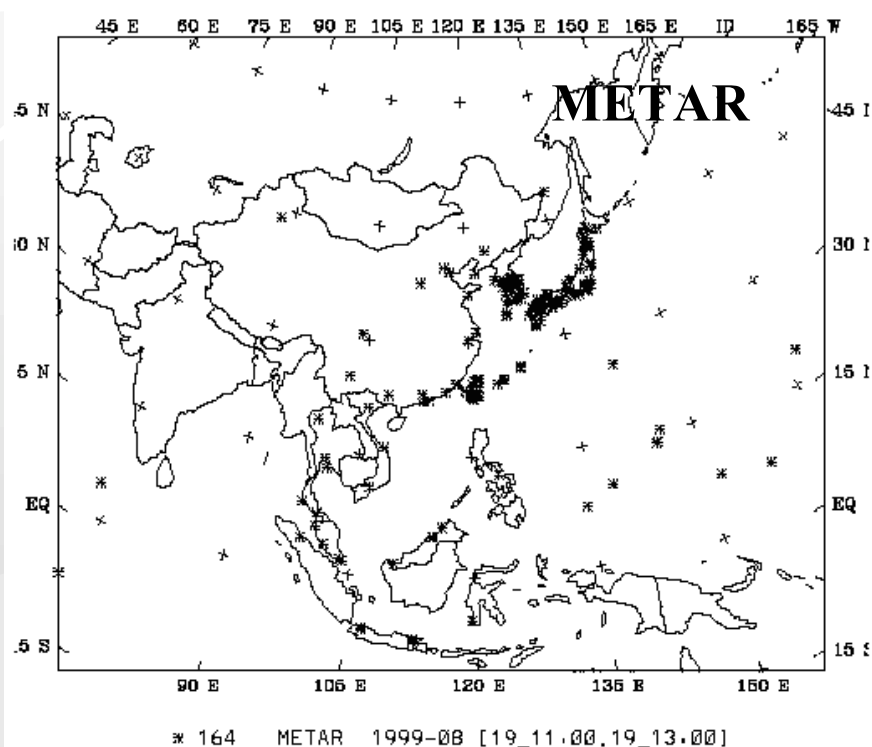
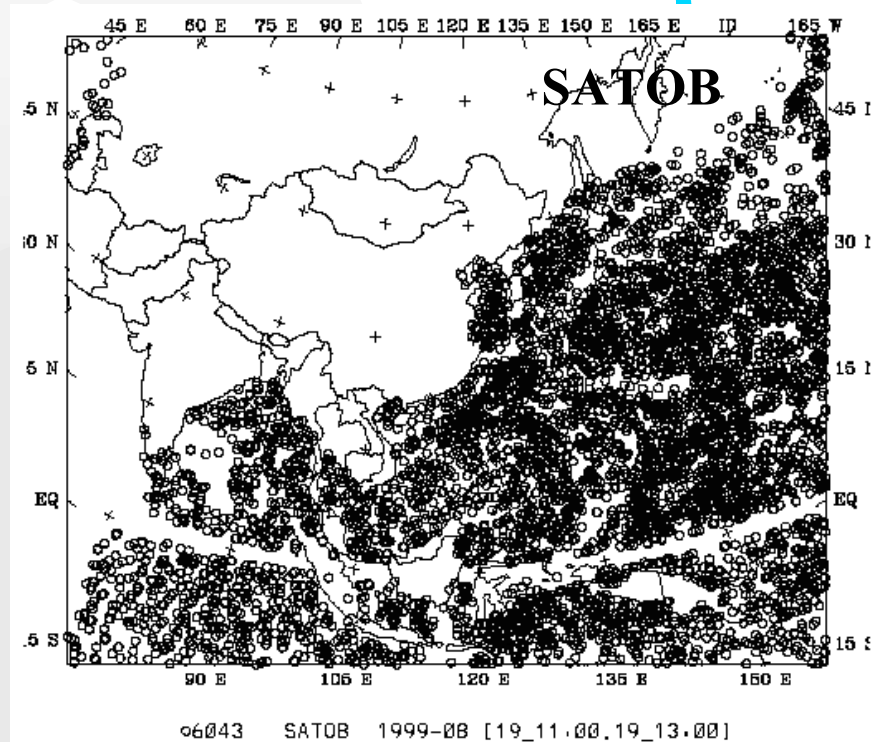
Distribution for each type of observations



Distribution for each type of observations



Distribution for each type of observations



Introduction to run *obsproc*

➤ Compiling the *obsproc*

```
cd /wrfhelp/SOURCE_CODE/WRFDA/var/obsproc  
make (or install WRFDA in your system, the obsproc.exe is created  
automatically)
```

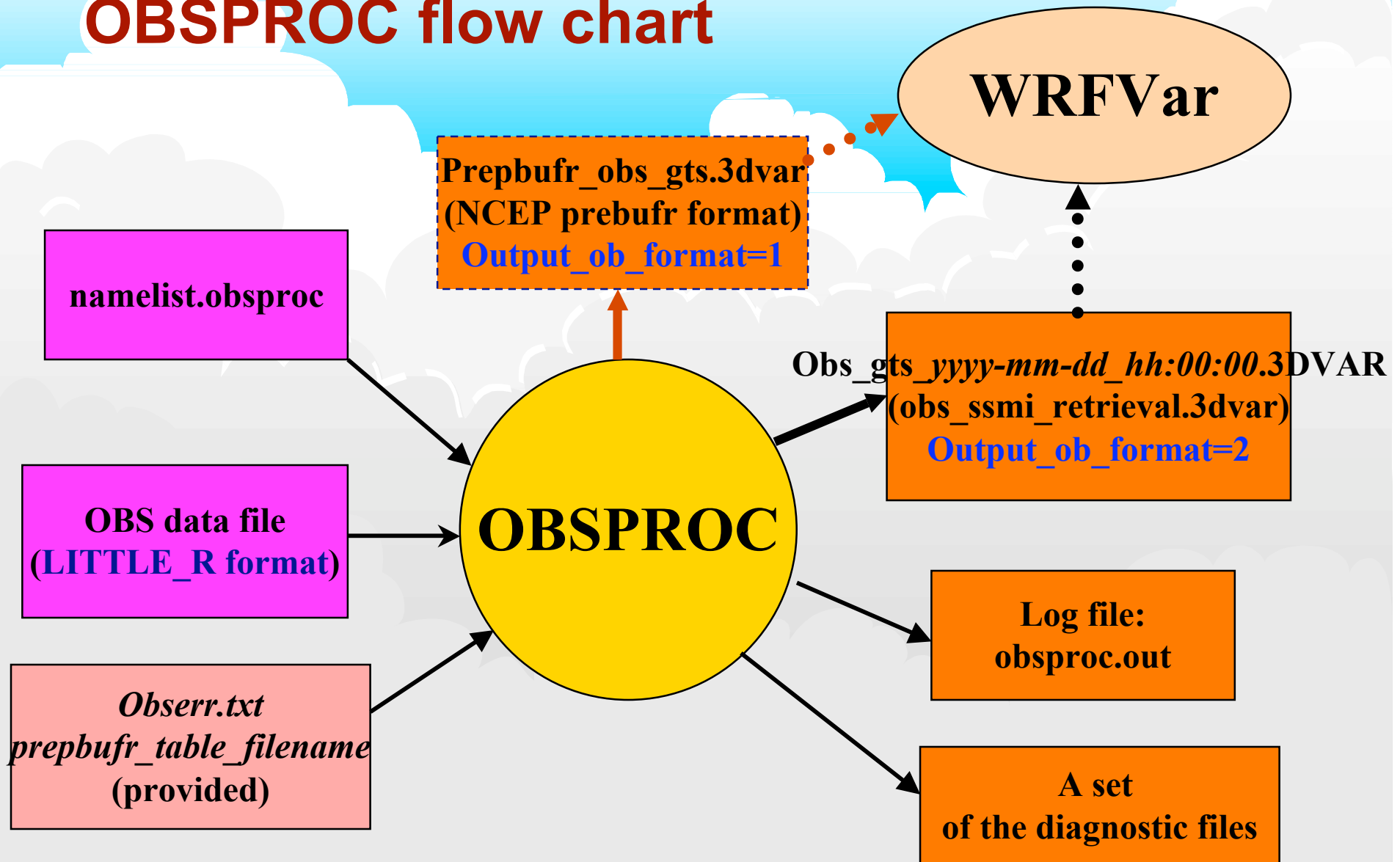
➤ Edit the *namelist.obsproc*

OBS data file: `/wrfhelp/DATA/VAR/Con200/ob/2007010200/ob.little_r`
`/wrfhelp/SOURCE_CODE/WRFDA/var/obsproc /obsproc.exe >&! obsproc.out`

➤ Plot the horizontal distribution of the observations

```
cp /wrfhelp/SOURCE_CODE/WRFDA/var/obsproc /MAP_plot/Map.csh  Map.csh  
edit MAP.csh  
Map.csh
```

OBSPROC flow chart



■ Input files for OBS preprocessor (obsproc)

3 Input files or 4 files if prebufr obs file need to be produced.

- OBS decoded file (Reports) in little_r format

A report (F90 pointer linking structure)

- ❖ header record (fields)
- ❖ Level1 data record (fields)
-
- ❖ Level*n* data record (fields)
- ❖ Ending record (fields)
- 3 Integers in format(3i7)

- Namelist file (*namelist.obsproc*) (See: README.namelist)

Record1: input file names

Record2: analysis times

Record3: Maximum number of observations allowed

Record4: quality control switches

Record5: print switches

Record6: define the reference state: ptop, etc.

Record7: Geographic parameters

Record8: Domain settings

Record9: Output format: prebufr, ascii, or both

- AFWA OBS errors file: *obserr.txt* (provided by 3DVAR system)
- Prebufr table file: *prebufr_table_filename* (used only for output_ob_format=1 or 3)

OBSPROC namelist variables.

&record1

obs_gts_filename	name and path of decoded observation file
fg_format	'MM5' for MM5 application, 'WRF' for WRF application
obserr.txt	name and path of observational error file
first_guess_file	name and path of the first guess file (Only for MM5 application)

&record2

time_window_min	Beginning of time window (included) as ccyy-mm-dd_hh:mn:ss
time_analysis	Ananalysis time as ccyy-mm-dd_hh:mn:ss
time_window_max	End of time window (included) as ccyy-mm-dd_hh:mn:ss

**** Note :** Only observations between [time_window_min, time_window_max] will kept.

&record3`max_number_of_obs`

Maximum number of observations to be loaded, i.e. in domain and time window, this is independent of the number of obs actually read.

`fatal_if_exceed_max_obs`

.TRUE.: will stop when more than `max_number_of_obs` are loaded
.FALSE.: will process the first `max_number_of_obs` loaded observations.

&record4`qc_test_vert_consistency`

.TRUE. will perform a vertical consistency quality control check on sounding

`qc_test_convective_adj`

.TRUE. will perform a convective adjustment quality control check on sounding

`qc_test_above_lid`

.TRUE. will flag the observation above model lid

`remove_above_lid`

.TRUE. will remove the observation above model lid

`domain_check_h`

.TRUE. will discard the observations outside the domain

`Thining_SATOB`

.FALSE.: no thinning for SATOB data.

.TRUE.: thinning procedure applied to SATOB data.

`Thining_SSMI`

.FALSE.: no thinning for SSMI data.

.TRUE.: thinning procedure applied to SSMI data.

`Thining_QSCAT`

.FALSE.: no thinning for SATOB data.

.TRUE.: thinning procedure applied to SSMI data.

&record6

x_left	West border of sub-domain, not used
x_right	East border of sub-domain, not used
y_bottom	South border of sub-domain, not used
y_top	North border of sub-domain, not used
ptop	Reference pressure at model top
ps0	Reference sea level pressure
base_pres	Same as ps0. User must set either ps0 or base_pres.
ts0	Mean sea level temperature
base_temp	Same as ts0. User must set either ts0 or base_temp.
tlp	Temperature lapse rate
base_lapse	Same as tlp. User must set either tlp or base_lapse.
pis0	Tropopause pressure, the default = 20000.0 Pa
base_tropo_pres	Same as pis0. User must set either pis0 or base_tropo_pres
tis0	Isothermal temperature above tropopause (K), the default = 215 K.
base_start_temp	Same as tis0. User must set either tis0 or base_start_temp.

&record7

IPROJ	Map projection (0 = Cylindrical Equidistance, 1 = Lambert Conformal, 2 = Polar stereographic, 3 = Mercator)
PHIC	Central latitude of the domain
XLONC	Central longitude of the domain
TRUELAT1	True latitude 1
TRUELAT2	True latitude 2
MOAD_CEN_LAT	The central latitude for the Mother Of All Domains
STANDARD_LON	The standard longitude (Y-direction) of the working domain.

&record8

IDD	Domain ID ($1 \leq ID \leq \text{MAXNES}$), Only the observations geographically located on that domain will be processed. For WRF application with $\text{XLONC} \neq \text{STANDARD_LON}$, set $\text{IDD}=2$, otherwise set 1.
MAXNES	Maximum number of domains as needed.
NESTIX	The I(y)-direction dimension for each of the domains
NESTJX	The J(x)-direction dimension for each of the domains
DIS	The grid size for each of the domains. For WRF application, always set $\text{NESTIX}(1)$, $\text{NESTJX}(1)$, and $\text{DIS}(1)$ based on the information in wrfinput.
NUMC	The mother domain ID number for each of the domains
NESTI	The I location in its mother domain of the nest domain's low left corner -- point (1,1)
NESTJ	The J location in its mother domain of the nest domain's low left corner -- point (1,1). For WRF application, $\text{NUMC}(1)$, $\text{NESTI}(1)$, and $\text{NESTJ}(1)$ are always set to be 1.

&record9

prebufr_output_filename	Name of the prebufr OBS file.
prebufr_table_filename	'prebufr_table_filename'; not change
output_ob_format	output 1, prebufr OBS file only; 2, ASCII OBS file only; 3, Both prebufr and ASCII OBS files.
use_for	'3DVAR' obs file, same as before, default 'FGAT ' obs files for FGAT '4DVAR' obs files for 4DVAR
num_slots_past	the number of time slots before time_analysis
num_slots_ahead	the number of time slots after time_analysis
write_synop	If keep synop obs in obs_gts (ASCII) files.writer
write_ship	If keep ship obs in obs_gts (ASCII) files.
write_metar	If keep metar obs in obs_gts (ASCII) files.
write_buoy	If keep buoy obs in obs_gts (ASCII) files.
write_pilot	If keep pilot obs in obs_gts (ASCII) files.
write_sound	If keep sound obs in obs_gts (ASCII) files.
write_amdar	If keep amdar obs in obs_gts (ASCII) files.
write_tamdar	If keep tamdar obs in obs_gts(ASCII) files.
write_satem	If keep satem obs in obs_gts (ASCII) files.
write_satob	If keep satob obs in obs_gts (ASCII) files.
write_airep	If keep airep obs in obs_gts (ASCII) files.
write_gpssp	If keep gpssp obs in obs_gts (ASCII) files.
write_gpsztd	If keep gpsztd obs in obs_gts (ASCII) files.
write_gpsref	If keep gpsref obs in obs_gts (ASCII) files.
write_gpseph	If keep gpseph obs in obs_gts (ASCII) files.

```
write_ssmt1  
write_ssmt2  
write_ssmi  
write_tovs  
write_qscat  
write_profl  
write_bogus  
write_airs
```

If keep ssmt1 obs in obs_gts (ASCII) files.

If keep ssmt2 obs in obs_gts (ASCII) files.

If keep ssmi obs in obs_gts (ASCII) files.

If keep tovs obs in obs_gts (ASCII) files.

If keep qscat obs in obs_gts (ASCII) files.

If keep profile obs in obs_gts (ASCII) files.

If keep bogus obs in obs_gts (ASCII) files.

If keep airs obs in obs_gts (ASCII) files.