GSI Setup, Run and Namelist

Hui Shao

GSI Community Tutorial, June 28-30, 2010, Boulder, CO

Outline

- Installing GSI
- Running GSI
- Basic runtime options

This talk is tailored based on GSI Community Release V2.0 and regional applications coupled with the WRF NMM/ARW.

Installing ...

Community GSI Release v2.0 Source Codes

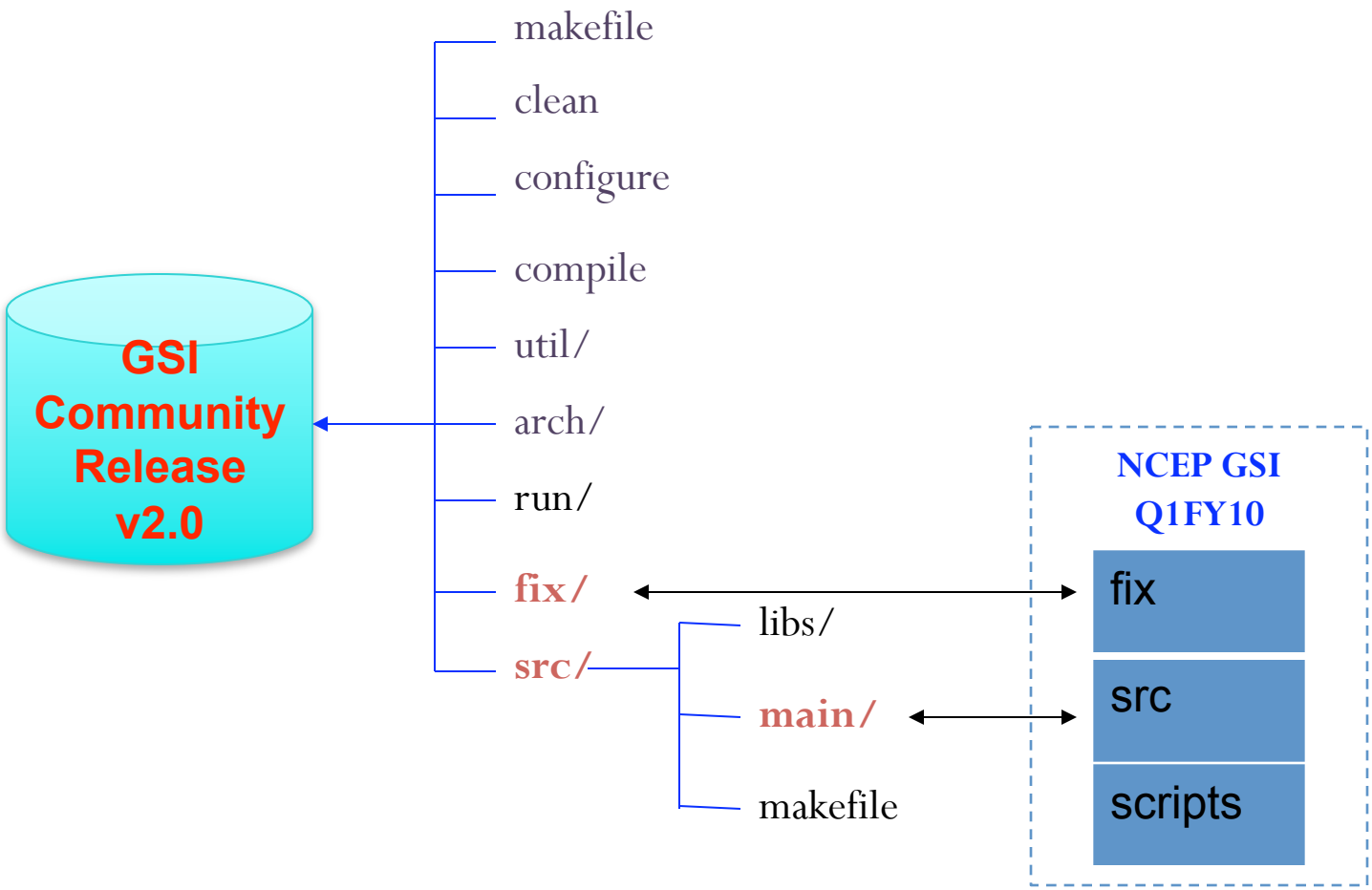
- Download GSI source code (comGSI_v2.tar.gz) from
  http://www.dtcenter.org/com-GSI/users/downloads/index.php

- Unpack the tarball, you may use the command
  "tar –zxvf comGSI_v2.tar.gz" or
  "gunzip comGSI_v2.tar.gz" and "tar –xvf comGSI_v2.tar"

The Community Release V2.0 corresponds to the GSI code in the NCEP Q1FY10 GFS implementation.

For people who are using the NCEP version of GSI, please be aware the installation procedure and portability of the GSI code might be different from the community release.

GSI Core and Extra Features



Fortran Compiler

- Supported and Tested Platforms
  - IBM: xlf
  - LINUX: pgi 7.0,8.0,9.0
  - LINUX: Intel 10.0 and 11.0
  - MAC OSX: PGI 8.0
- Supported and Tested by a 3rd party
  - SGI Altix: Intel 10.0

- Use "which" (e.g., which xlf) command to find the compiler path.
- To check version number, use "xlf –qversion", "pgf90 –V" and "ifort –v" for xlf, pgi and Intel Fortran compilers.

Please let us know of your experiences with various platforms and compilers.

External Libraries Required by GSI

- **netCDF (network Common Data Form) v3.6+ (not v4.0 yet):**
  - http://www.unidata.ucar.edu/software/netcdf/
- **WRF I/O API (WRF I/O Application Program Interface) v3.2:**

  WRF-ARW or NMM provided in the same tar file
  - http://www.mmm.ucar.edu/wrf/users/downloads.html or
  - http://www.dtcenter.org/wrf-nmm/users/downloads/index.php
- **ESSL (IBM) or LAPACK and BLAS (LINUX and MAC OSX)**
  - ESSL (Engineering and Scientific Subroutine Library):

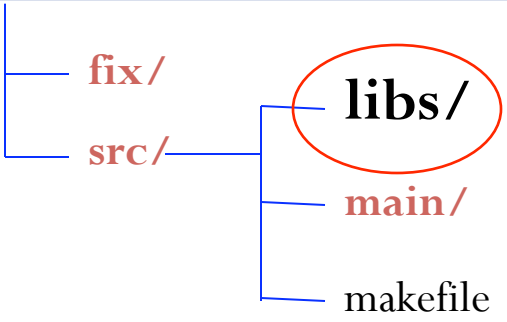
    http://www-03.ibm.com/systems/software/essl/index.html
    - ✓ Usually, the IBM compiler automatically sets up the ESSL. No manual set up required.
  - LAPACK (Linear Algebra PACKage): http://www.netlib.org/lapack/

    BLAS (Basic Linear Algebra Subprograms): http://www.netlib.org/blas/
    - ✓ LINUX: usually, the PGI compiler automatically sets up the libs.
    - ✓ LINUX: Intel compiler, please set up the libs manually.
    - ✓ MAC OSX: PGI compile, please set up the libs manually.

To set up environment variables,
- csh/tcsh users: use "setenv" (e.g., setenv NETCDF $netcdfpath)
- bash/ksh users: use "export" (e.g., export NETCDF $netcdfpath)

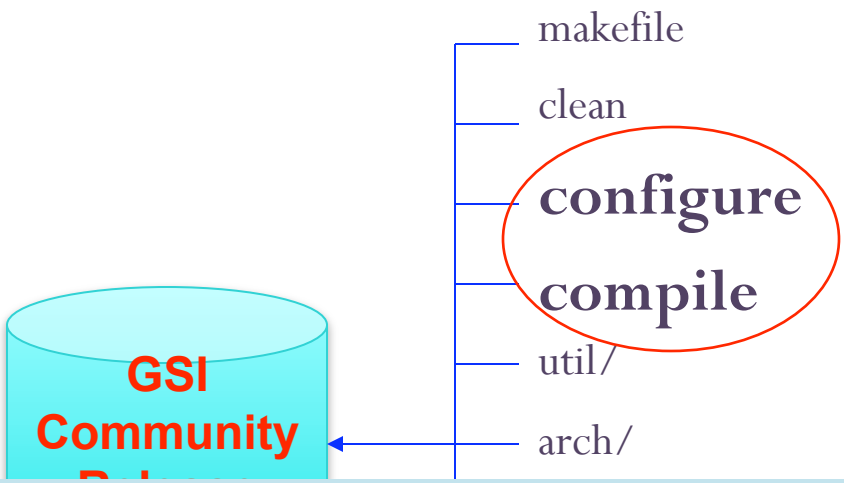
Internal Libraries Required by GSI

Name (~/src/libs)	Description
bufr	NCEP BUFR Library
crtm_gfsgsi	JCSDA Community Radiative Transfer Model (CRTM)
gfsio	Unformatted Fortran record for GFS I/O
sfcio	NCEP GFS surface file I/O module
sigio	NCEP GFS atmospheric file I/O module
sp	NCEP spectral-grid transforms (global application only)
w3	NCEP W3 library (date/time manipulation, GRIB)



No manual setup/compilation required if you use configure/compile utility provided in the GSI tarball.

Configure and Compile GSI



Functions:

- Checks the system hardware and software (*netCDF, WRF, and math libraries).*
- Offers the user choices for configuring GSI.
- Creates **configure.gsi**, which contains compilation options, paths, rules, etc. specific to your computer and compiler choice, and can be edited to change compilation options, if desired.
- Compiles both internal libraries and GSI source code and creates gsi.exe.

Configure GSI

> cd $your_sourcecode_dir/comGSI_v2

> ./configure

./configure.gsi will be created if successful.

- Example 1: NETCDF path was set up but not WRF directory

```
Will use NETCDF in dir: /usr/local/netcdf
** WARNING: No path to WRF_DIR and environment variable WRF_DIR not set.
** would you like to try to fix? [y]
```

- Example 2: Compilation options on IBM

```
Will use NETCDF in dir: /usr/local/netcdf
Will use WRF in dir: /ptmp/huishao/tutorial/comGSI_v2/../WRFV3
------------------------------------------------------------------------
Please select from among the following supported platforms.
   1. AIX 64-bit   (dmpar)
   2. AIX 32-bit  (dmpar)
Enter selection [1-2] :
```

dmpar: distributed-memory parallel

For unsupported platforms, refer to the following talk "GSI Software Design" by Don Stark.

Compile GSI

> cd $your_sourcecode_dir/comGSI_v2

> ./compile (or ./compile >& $your_log_filename)

~/comGSI_v2/run/gsi.exe will be created if successful.

- Get help with the compilation:

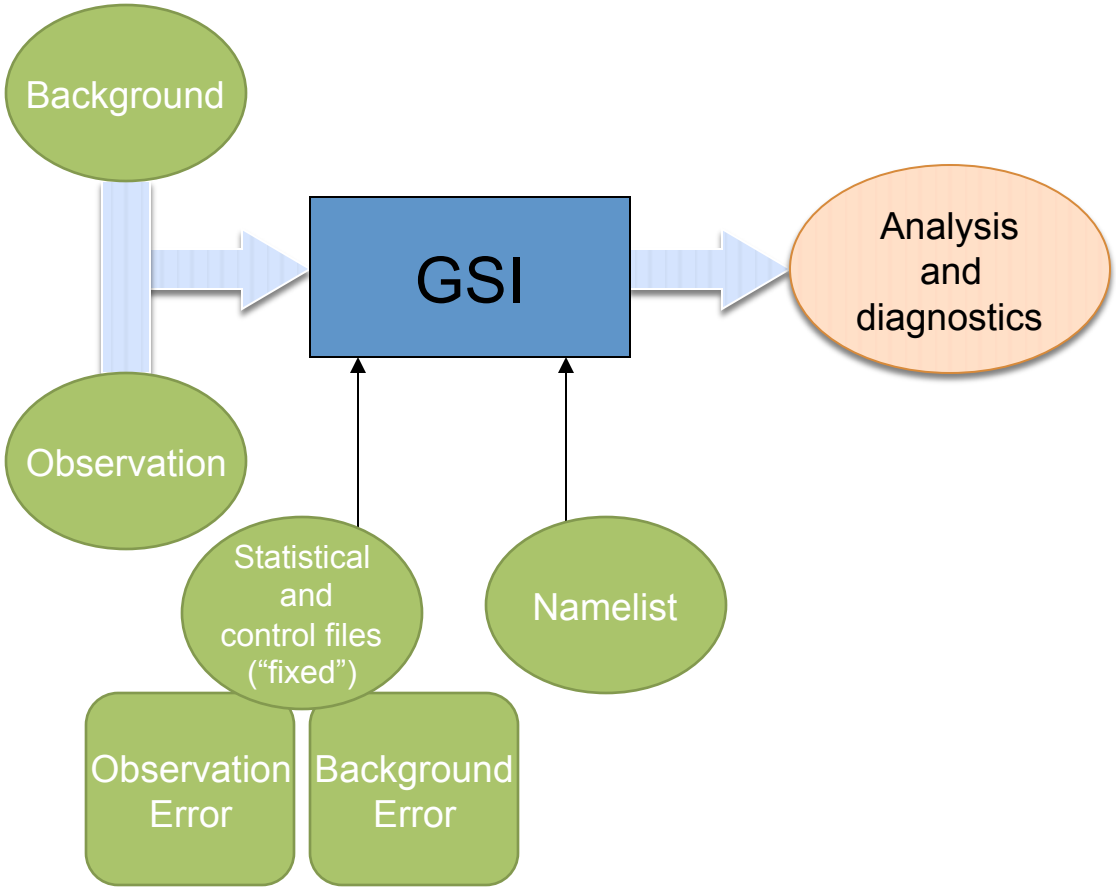
  >./compile –h

- Clean the GSI directory before making your next compilation:

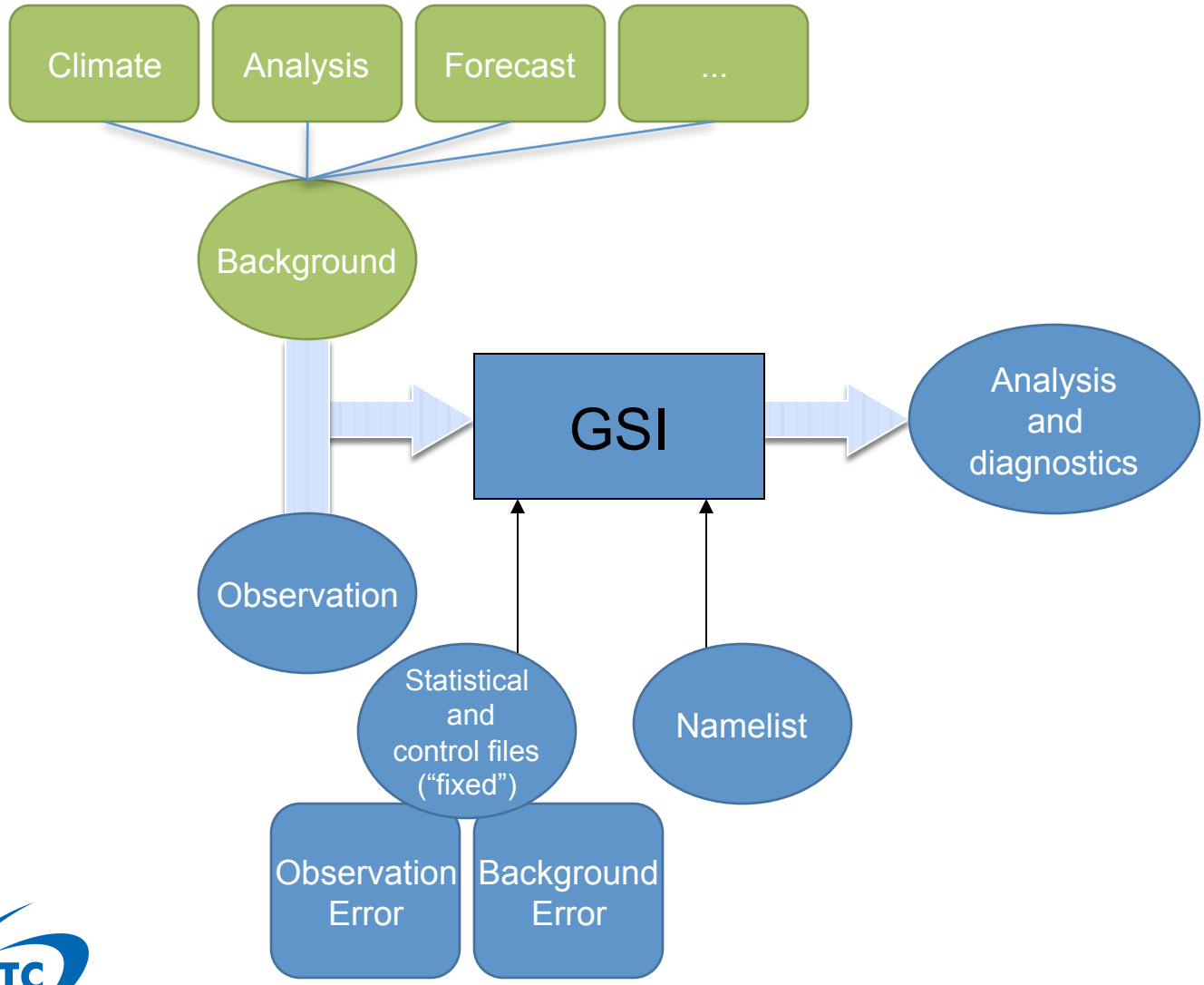
  > ./clean -a

Before You Run...

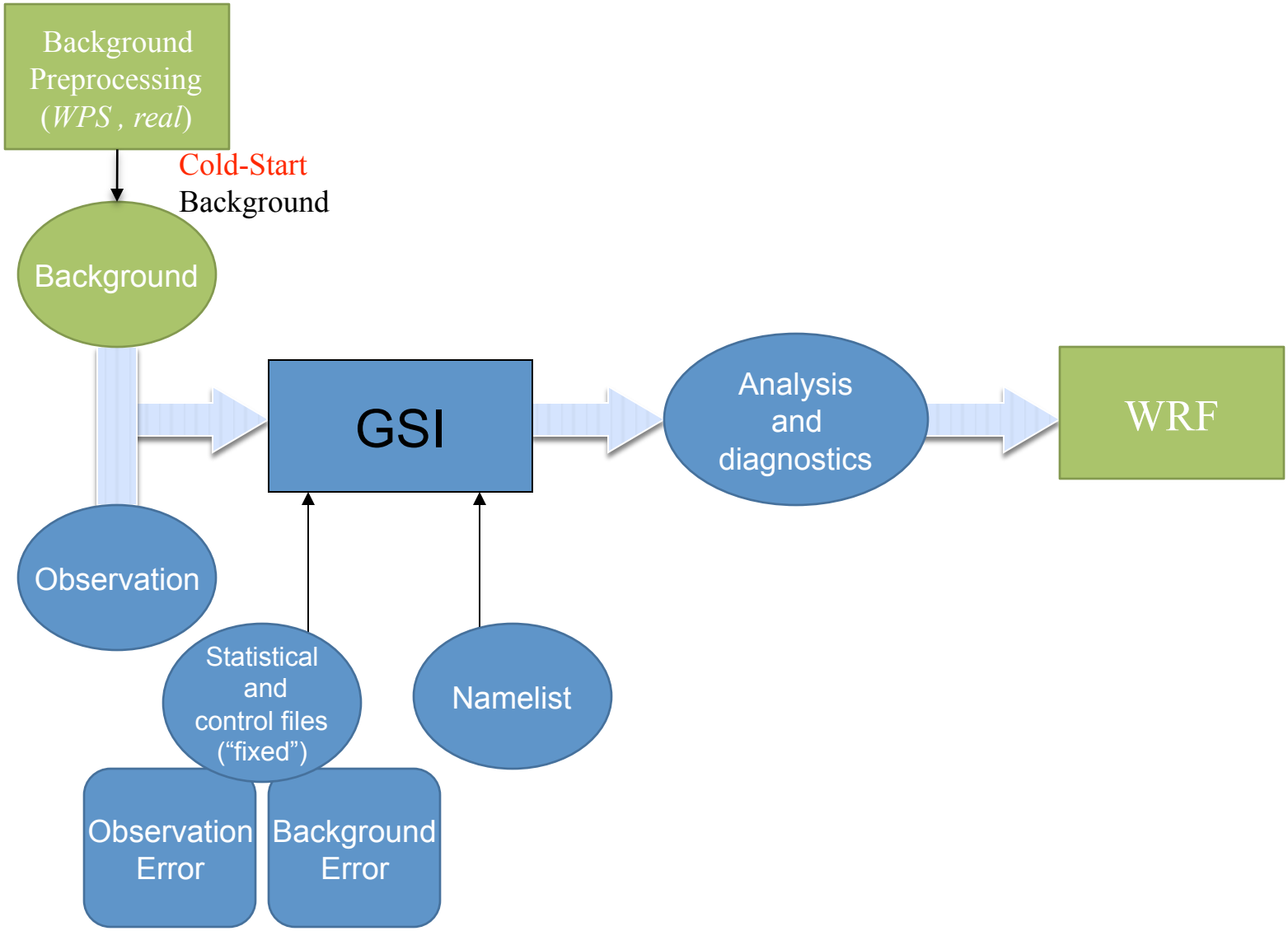
GSI Input and Output



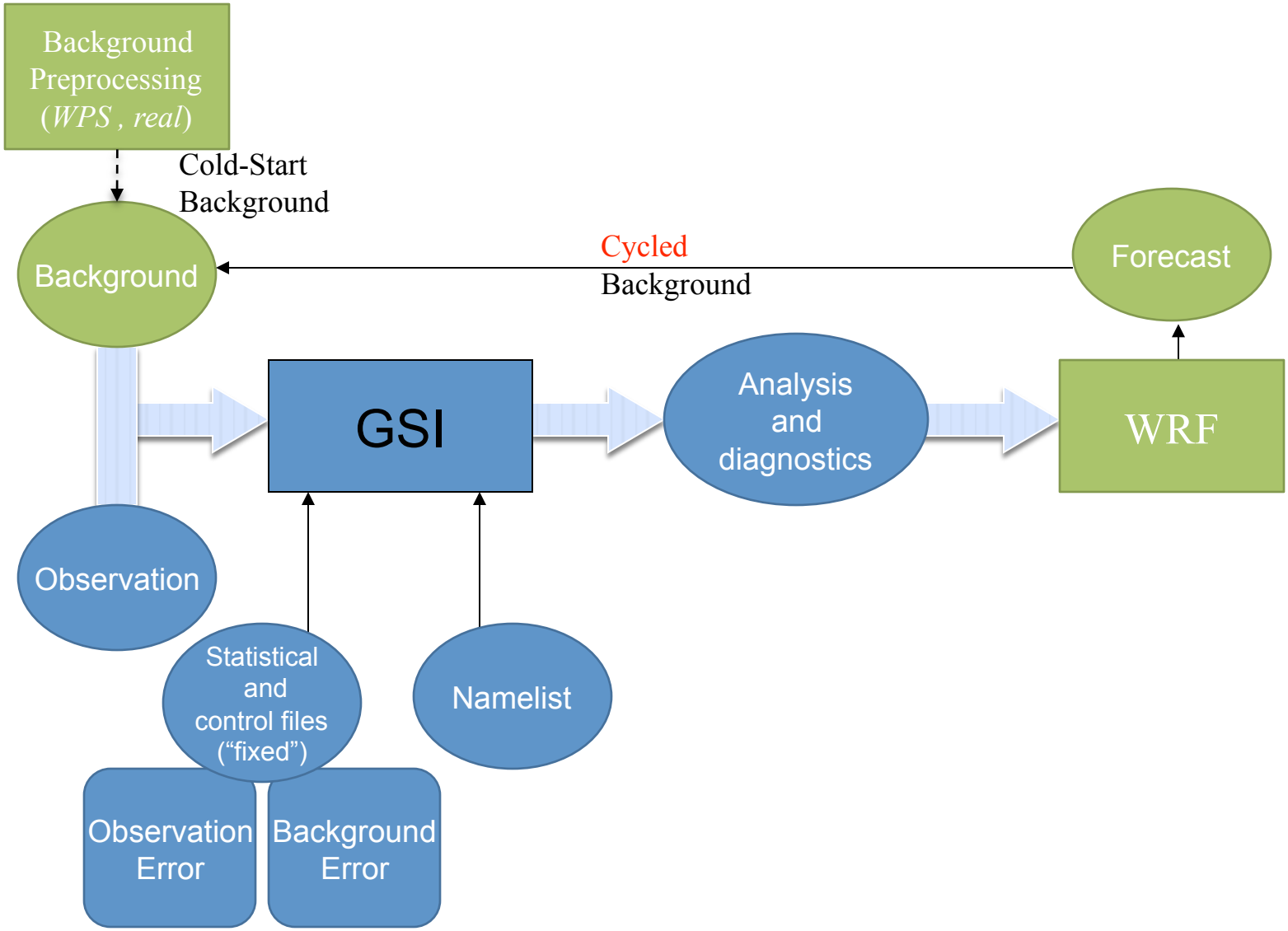
GSI Background



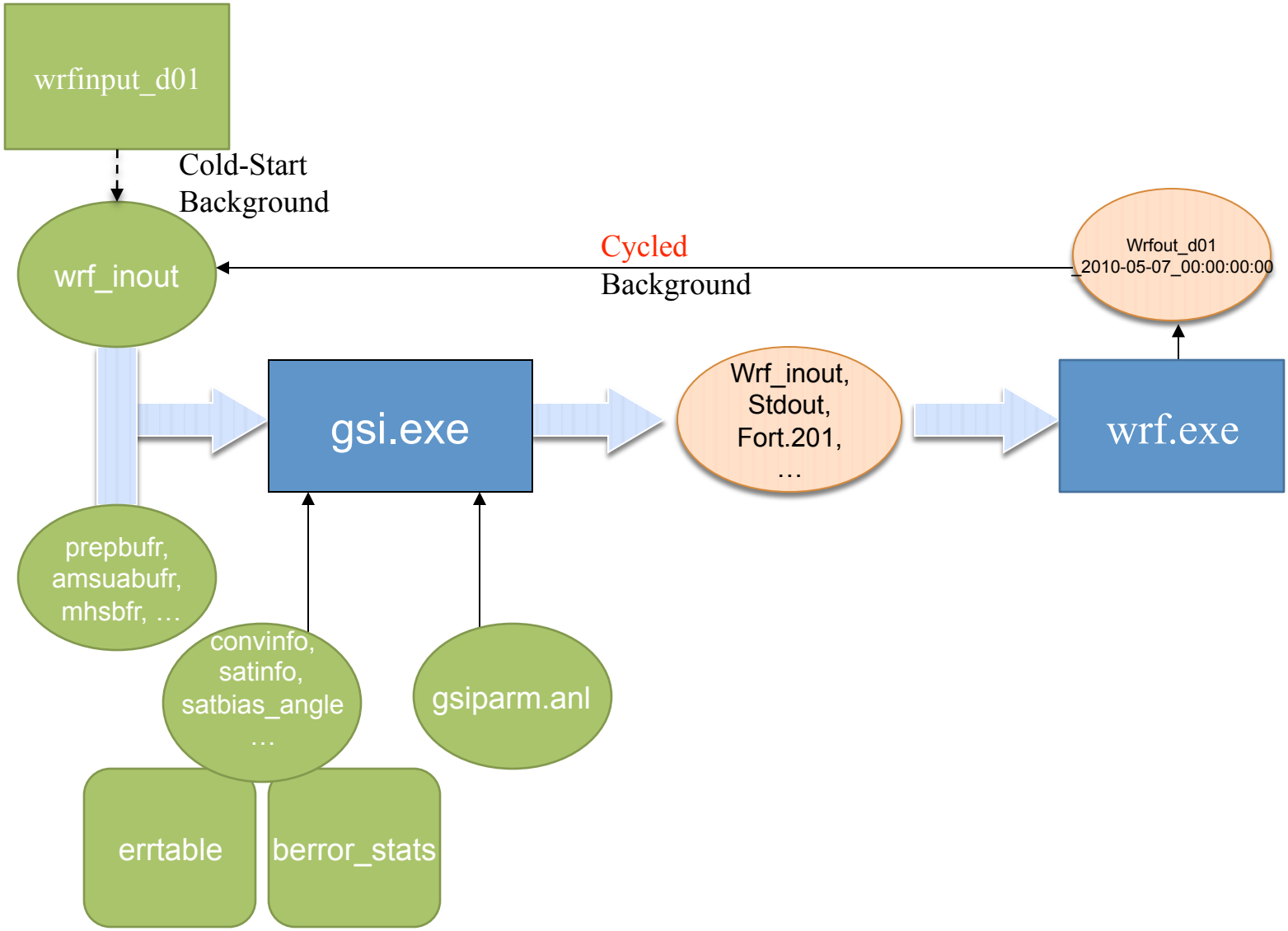
Example: "Cold-start" vs cycling in a WRF system



Example: "Cold-start" vs cycling in a WRF system



Example: GSI Input and Output in a WRF System



Observation File Names Used in GSI

(GSI Users' Manual-Chapter 3)

GSI Name	Content	Example file from operation
prepbufr	Conventional observations, including ps, t, q, pw, uv, spd, dw, sst, from observation platforms such as METAR, sounding, et cl.	gdas1.t12z.prepbufr
amsuabufr	AMUS-A radiances (brightness temperatures) same for other radiance files from satellite NOAA-16, 17,18, and metop-a	gdas1.t12z.1bamua.tm00.bufr_d
amsubbufr	AMUS-B observation from Satellite NOAA-15, 16,17	gdas1.t12z.1bamub.tm00.bufr_d
radarbufr	Radar radial velocity Level 2.5	ndas1.t12z. radwnd. tm12.bufr_d
gpsrobufr	gps_ref'	gdas1.t12z.gpsro.tm00.bufr_d
ssmirrbufr	pcp_ssmi	gdas1.t12z.spssmi.tm00.bufr_d
tmirrbufr	pcp_tmi	gdas1.t12z.sptrmm.tm00.bufr_d
sbuvbufr	sbuv2 observation from satellite NOAA16, 17, 18	gdas1.t12z.osbuv8.tm00.bufr_d
hirs2bufr	hirs2 observation from satellite NOAA14	gdas1.t12z.1bhrs2.tm00.bufr_d
hirs3bufr	hirs3 observation from satellite NOAA16, 17	gdas1.t12z.1bhrs3.tm00.bufr_d
hirs4bufr	hirs4 observation from satellite NOAA 18 and metop-a	gdas1.t12z.1bhrs4.tm00.bufr_d
airsbufr	Airs observation from satellite AQUA	gdas1.t12z.airsev.tm00.bufr_d
msubufr	Msu observation from satgellite NOAA 14	gdas1.t12z.1bmsu.tm00.bufr_d
airsbufr	Amsua and AIRS radiances from satellite AQUA	gdas1.t12z.airsev.tm00.bufr_d
mhsbufr	Microwave Humidity Sounder observation from NOAA 18 and METOP-A	gdas1.t12z.1bmhs.tm00.bufr_d
ssmitbufr	Ssmi observation from satellite f13 f14 f15	gdas1.t12z.ssmit.tm00.bufr_d
amsrebufr	AMSR-E radiance from satellite AQUA	gdas1.t12z.amsre.tm00.bufr_d

DTC Develop

"Fixed" File Names Used in GSI
(GSI Users' Manual-Chapter 3)

File name in GSI	Example files used by NAM	Content
berror_stats	nam_nmmstat_na	background error covariance
errtable	nam_errtable.r3dv	Observation error table
Observation data control file (more detailed explanation in Section 3.5)		
convinfo	nam_regional_convinfo.txt	Conventional observation information file
satinfo	nam_regional_satinfo.txt	satellite channel info file
pcpinfo	nam_global_pcpinfo.txt	precipitation rate observation info file
ozinfo	nam_global_ozinfo.txt	ozone observation information file
mesonetuselist	nam_mesonet_uselist.txt	
Bias correction used by radiance analysis		
satbias_angle	nam_global_satangbias.txt	satellite scan angle dependent bias correction file
satbias_in	ndas.t06z.satbias.tm03	satellite variational bias correction coefficient file
Radiance coefficient used by CRTM		
EmisCoeff.bin	EmisCoeff.bin	IR surface emissivity coefficient file
AerosolCoeff.bin	AerosolCoeff.bin	Aerosol coefficients
CloudCoeff.bin	CloudCoeff.bin	Cloud scattering and emission coefficients
${satsen}.SpcCoeff.bin	${satsen}.SpcCoeff.bin	Sensor spectral response characteristics
${satsen}.TauCoeff.bin	${satsen}.TauCoeff.bin	Transmittance coefficients

DTC Developme

Get What You Need…

- Check GSI executable has been created appropriately
  - ~/comGSI_v2/run/gsi.exe
- Obtain background files
  - GSI has been used for different applications
    - WRF NMM input field in binary format (IBM: NCEP, DTC)
    - WRF NMM input field in netCDF format (IBM: NCEP, all the supported platforms: DTC)
    - WRF ARW input field in binary format (IBM: NCEP, DTC)
    - WRF ARW input field in netCDF format (IBM: NCEP, all the supported platforms:DTC)
    - GFS input field in binary format (IBM: NCEP, DTC)
    - GMAO global model input field in binary format (LINUX: GMAO)
    - …

- Obtain observation files.
  - The NCAR CISL Research Data Archive (RDA):
    http://dss.ucar.edu/
  - NCEP public server:
    http://www.emc.ncep.noaa.gov/data/
  - NCAR Mass Storage System (MSS): authorized access
  - GSD Mass Storage System: authorized access
  - Or any place you may obtain
    - PREPBUFR formatted data: conventional (including satellite derived data)
    - BUFR formatted data: satellite radiance, radar reflectivity, GPS radio occultation(RO) refractivity/bending angle

GSI can run pseudo single observation test (PSOT) without real observations.

- Check statistical and control files under ~/comGSI_v2/fix
  - Background errors (BEs)
    - nam_nmmstat_na (nam_glb_berror.f77_Little_Endian): computed from the NCEP's NAM statistics
    - nam_glb_berror.f77 (nam_glb_berror.f77_Little_Endian): computed from the NCEP's GFS statistics
  - Observation errors
    - nam_errtable.r3dv: used at NCEP for the NAM application.
    - prepobs_errtable.global: used at NCEP for the GFS (global) application.
  - Observation data control file (info files)
    - global_convinfo.txt: conventional obs. information file
    - global_pcpnfo.txt: satellite channel information file
    - global_satinfo.txt: precipitation rate observation information file
  - Bias correction used by radiance analysis
    - global_satangbias.txt: satellite scan angle dependent bias correction file
    - ndas.t06z.satbias.tm03: satellite variational bias correction coefficient file
  - Radiance coefficient used by CRTM (~/comGSI_v2/fix/crtm_gfsgsi)

If you are working on a non-NAM domain, the statistical and control files generated based on global applications might be good ones to start with.

- We do provide a sample run script for you to start with
  - ~/comGSI_v2/run/run_gsi.ksh

Run_gsi.ksh: Structure

- Ask for computer resources to run GSI — Experiment Setup
- Set machine dependent compilation option
- Set experimental variables (experiment name, analysis time, background and observation)
- Check the definition of required variables — Setup check
- Generate a run directory for GSI (working or temp directory)
- Copy GSI executable to run directory — Input files
- Copy background file to run directory
- Copy or link observations to run directory
- Copy fixed files to run directory
- Generate namelist for GSI
- Run the GSI executable — Run and Save
- Save the GSI analysis results

Run_gsi.ksh: Experiment Setup
(An Example from Practical Sessions)

```
#
 GSIPROC=1
 ARCH='LINUX_PGI'
#
 ANAL_TIME=2010050700
#
 WORK_ROOT=/data1/class05/GSI/gps_arw_${ANAL_TIME}
#
 BK_FILE=/wrfhelp/GSI/Datafiles/2010050700/arw_wrfinput_d01_2010-05-07_00:00:00
 OBS_ROOT=/wrfhelp/GSI/Datafiles/2010050700/obs
 PREPBUFR=/wrfhelp/GSI/Datafiles/2010050700/obs/le_nam.t00z.prepbufr.tm00.nr
 FIX_ROOT=/data1/class05/GSI/comGSI_v2/fix
 GSI_EXE=/data1/class05/GSI/comGSI_v2/run/gsi.exe
#
 bk_core=ARW
#
 bkcv_option=NAM
```

Number of processors/compiler

Analysis time (YYYYMMDDHH)

Set up your working directory

Set up GSI input directories

Which WRF dynamic core (ARW or NMM) was used to generate the background Or which format (ARW or NMM) was adopted for the background?

Which default background error file (GLOBAL or NAM) will be used? If you generate your own BE file, please adopt it accordingly instead of using this option.

Run_gsi.ksh: Setup Check

```
case $ARCH in
  'IBM_LSF')
     BYTE_ORDER=Big_Endian
     RUN_COMMAND="mpirun.lsf "
...

 if [ ! "${ANAL_TIME}" ]; then
  echo "ERROR: \$ANAL_TIME is not defined!"
  exit 1
 fi
...

  workdir=${WORK_ROOT}
  rm -rf ${workdir}
  mkdir -p ${workdir}
  cd ${workdir}
  echo ${workdir}
...
```

Check compiler setup

- Load Sharing Facility (LSF) is a third party job scheduling system used by DTC on IBM. If you are using LoadLeveler, please adopt it accordingly.

Check experimental setup

Create your own working directory. From this point on, the GSI run will be done in this working directory.

Run_gsi.ksh: Input Files

```
cp ${GSI_EXE} gsi.exe
cp ${BK_FILE} wrf_inout
ln -s ${PREPBUFR} prepbufr
# ln -s ${OBS_ROOT}/ndas.t12z.1bamua.tm12.bufr_d_le amsuabufr
if [ ${bkcv_option} = GLOBAL ] ; then
  if [ ${BYTE_ORDER} = Little_Endian ] ; then
    BERROR=${FIX_ROOT}/nam_glb_berror.f77_Little_Endian
…
  fi
  OBERROR=${FIX_ROOT}/prepobs_errtable.global
else
…
fi
…
cp $BERROR   berror_stats
cp $OBERROR  errtable
cp $CONVINFO convinfo
cp $SATANGL  satbias_angle
cp $SATINFO  satinfo
…
```

The background file will be replaced after running GSI, therefore please use "copy" instead of "link".

Run_gsi.ksh: Input Files (cont)

Create the GSI namelist

```
#
cat << EOF > gsiparm.anl
 &SETUP
  miter=2,niter(1)=10,niter(2)=10,
  write_diag(1)=.true.,write_diag(2)=.false.,write_diag(3)=.true.,
  gencode=78,qoption=2,
  factqmin=0.0,factqmax=0.0,deltim=$DELTIM,
  ndat=62,npred=5,iguess=-1,
  oneobtest=.false.,retrieval=.false.,
  nhr_assimilation=3,l_foto=.false.,
  use_pbl=.false.,
 /
 &GRIDOPTS
  JCAP=$JCAP,JCAP_B=$JCAP_B,NLAT=$NLAT,NLON=$LONA,nsig=$LEVS,hybrid=.true.,
```

Run_gsi.ksh: Run and …

```
case $ARCH in
  'IBM_LSF')
     ${RUN_COMMAND} ./gsi.exe < gsiparm.anl > stdout 2>&1  ;;
  * )
     ${RUN_COMMAND} ./gsi.exe > stdout 2>&1  ;;
Esac
…
#
cp stdout      stdout.anl.${ANAL_TIME}
cp wrf_inout   wrfanl.${ANAL_TIME}
ln fort.201    fit_p1.${ANAL_TIME}
ln fort.202    fit_w1.${ANAL_TIME}
ln fort.203    fit_t1.${ANAL_TIME}
ln fort.204    fit_q1.${ANAL_TIME}
ln fort.207    fit_rad1.${ANAL_TIME}
…
```

If running in distributed-memory mode, you need to set up the computer resources (e.g., processor numbers, memory, wallclock…) based on the platform you are using.

At the end of the standard output, stdout, you should find

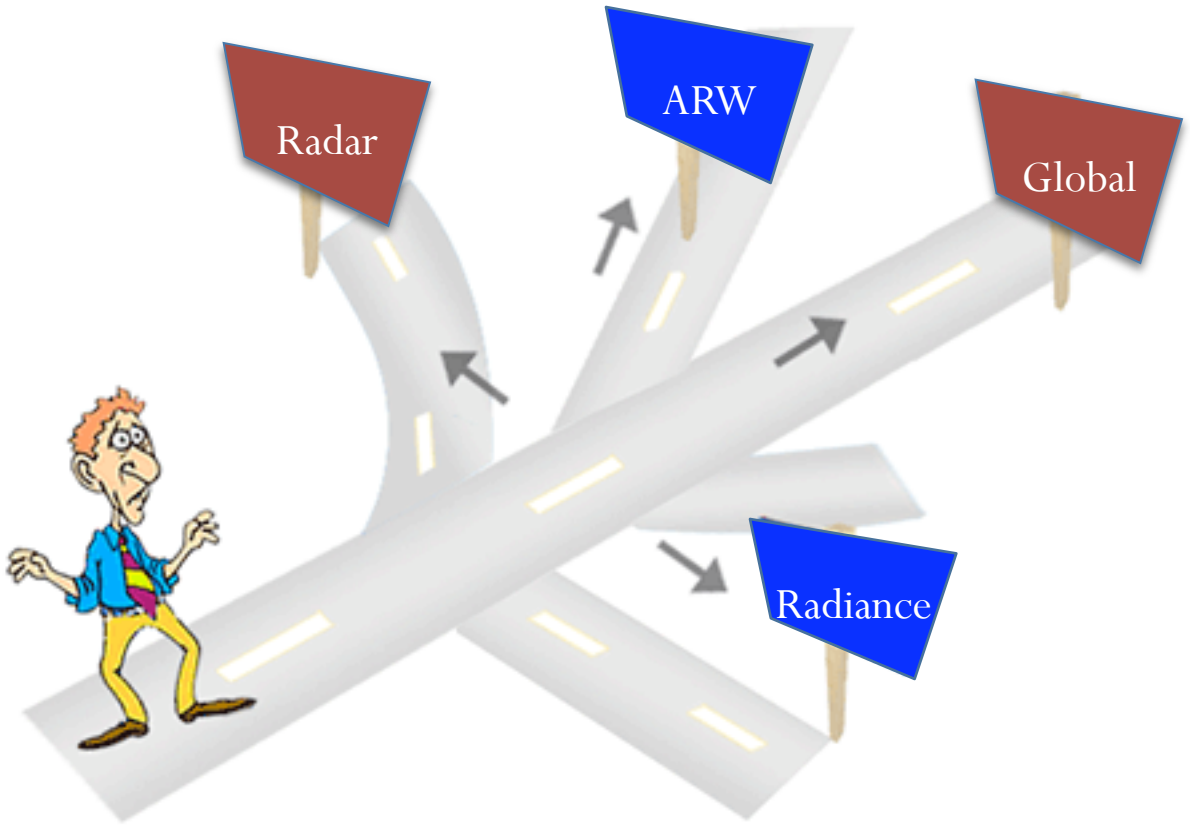
```
 Begin Jo table outer loop
    Observation Type           Nobs                    Jo          Jo/n
surface pressure               7001    2.7252434250153715E+03       0.389
temperature                    4243    2.4056108310665422E+03       0.567
wind                          16522    7.3449376699398827E+03       0.445
moisture                       1658    5.1611478560903402E+02       0.311
gps                            1012    4.8580642818843398E+03       4.800
                               Nobs                    Jo          Jo/n
           Jo Global          30436    1.7849970993515170E+04       0.586
 End Jo table outer loop


     ENDING DATE-TIME    MAY 27,2010  13:31:55.235  147  THU   2455344
     PROGRAM GSI_ANL HAS ENDED.  IBM RS/6000 SP
* . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * .
```

And…

You may have the following files after running GSI successfully in your working directory :

```
be1105en:/ptmp/huishao/tutorial/comet_practice/gps_arw_2010050700/temp>ls
GSI_inc_xymap_arw_7.ncl*    fort.209                    obs_input.0009
GSI_radiance_arw_19.pdf     fort.210                    obs_input.0010
berror_stats*               fort.211                    ozinfo*
convinfo*                   fort.212                    pcpbias_out
diag_conv_anl.2010050700    fort.213                    pcpinfo*
diag_conv_ges.2010050700    fort.214                    pe0000.conv_01
errtable*                   fort.215                    pe0000.conv_03
fit_p1.2010050700           fort.220                    pe0000.obs_setup
fit_q1.2010050700           fsize_gpsrobufr0000         prepobs_prep.bufrtable*
fit_rad1.2010050700         fsize_prepbufr0000          satbias_angle*
fit_t1.2010050700           gps.ncl                     satbias_in
fit_w1.2010050700           gsi.exe*                    satbias_out
fort.201                    gsiparm.anl                 satinfo*
fort.202                    obs_input.0001              siganl
fort.203                    obs_input.0002              sigf03
fort.204                    obs_input.0003              stdout
fort.205                    obs_input.0004              stdout.anl.2010050700
fort.206                    obs_input.0005              wrf_inout*
fort.207                    obs_input.0007              wrfanl.2010050700*
fort.208                    obs_input.0008
```

Please refer to the talk "GSI diagnostics" for more information.



Options?

The GSI namelist (gsiparm.anl) and the "info" files (convinfo, satinfo, …) help the user to configure a GSI run without recompiling the code.

- GSI namelist:
  - Specific Fortran 90 namelist format:

```
&namelistname        - start
…
/                    - end
```

  - Description of the GSI namelist variables is given in GSI User's Guide (Chapter 3) and gsimod.F90.
- "Info" file:
  - Control the data usage, time window, gross quality control based on data types.
  - Description of the "info" files is given in GSI User's Guide (Chapter 4) .

GSI Namelist

1. SETUP: General control variables.
2. GRIDOPTS: Grid setup variable, including global regional specific namelist variables.
3. BKGERR:Background error related variables.
4. ANBKGERR: Anisotropic background error related variables.
5. JCOPTS: Constraint term in cost function (Jc)
6. STRONGOPTS: Strong dynamic constraint.
7. OBSQC: Observation quality control variables.
8. OBS_INPUT: Input data control variables.
9. SUPEROB_RADAR: Level 2 BUFR file of radar wind superobs.
10. LAG_DATA: Lagrangian data assimilation related variables.
11. SINGLEOB_TEST: Pseudo single observation test setup.

GSI Namelist - SETUP

General Control Variables

```
&SETUP
  miter=2,niter(1)=10,niter(2)=10,
  write_diag(1)=.true.,write_diag(2)=.false.,write_diag(3)=.true.,
  gencode=78,qoption=2,
  factqmin=0.0,factqmax=0.0,deltim=$DELTIM,
  ndat=62,npred=5,iguess=-1,
  oneobtest=.false.,retrieval=.false.,
  nhr_assimilation=3,l_foto=.false.,
  use_pbl=.false.,
/
```

GSI Namelist - SETUP(cont.)

- **miter**: Number of outer loops of analysis.
- **niter(1)** : Maximum iteration number of 1st inner loop iterations. Inner loop will stop either reaching this maximum number or reaching convergence condition or broken.
- **niter(2)** : Maximum iteration number of 2nd inner loop iterations. Inner loop will stop either reaching this maximum number or reaching convergence condition or broken.

GSI Namelist-SETUP(cont.)

- **write_diag**: Logical to write out diagnostic files.

  Example, for a two outer loop run (miter=2),

  - write_diag(1)=.true.: Produce diagnostics file containing the background information.
  - write_diag(2)=.true.: Produce diagnostic file containing the intermediate analysis information after the first outer loop.
  - write_diag(3)=.true. : Produce diagnostic file containing the final analysis information after the second outer loop.

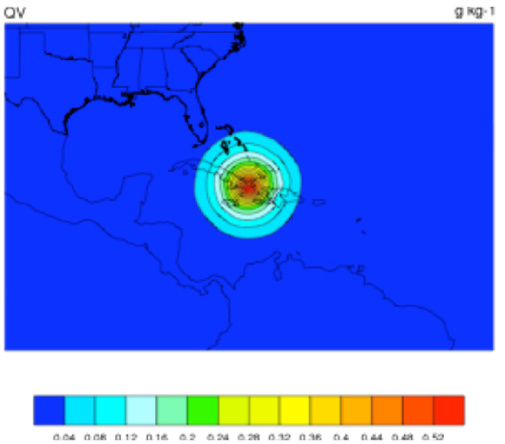
- **ndat**: number of observation variables (not observation types). This number should be consistent with the number of dimensions of **dfile** in the namelist OBS_INPUT. So if adding a new observation variable, **ndat** must be incremented by one and one new line must be added to OBS_INPUT. The maximum value of **ndat** is 100.

GSI Namelist-SETUP(cont.)

- **qoption**: Option for moisture analysis variable.
  - qoption=1: The moisture analysis variable is pseudo-relative humidity, q/q_{satg}. The saturation specific humidity, q_{satg}, is computed from the guess and held constant during the inner loop. Thus, the RH control variable can only change via changes to specific humidity, q.
  - qoption=2: The moisture analysis variable is normalized RH. In this formulation, GSI allows RH to change in the inner loop via changes to surface pressure (pressure), temperature, or specific humidity.

GSI Namelist
- SETUP(cont.) and SINGLEOB_TEST

- **oneobtest**: Logical to run pseudo single observation test (PSOT).
  - oneobtest=.true.: Run PSOT. You also need to modify the following namelist to set up the pseudo single observation.




```
&SINGLEOB_TEST
  maginnov=1.0,magoberr=0.8,oneob_type='t',
  oblat=38.,oblon=279.,obpres=500.,obdattim=${ANAL_TIME},
  obhourset=0.,
/
```

For PSOT, ~/comGSI_v2/fix/prepobs_prep.bufrtable is also required.

GSI Namelist-GRIDOPTS

Grid setup variables, including regional specific variables

```
&GRIDOPTS
 JCAP=$JCAP,JCAP_B=$JCAP_B,NLAT=$NLAT,NLON=$LONA,nsig=$LEVS,hybrid=.true.,
 wrf_nmm_regional=${bk_core_nmm},wrf_mass_regional=${bk_core_arw},
 diagnostic_reg=.false.,
 filled_grid=.false.,half_grid=.true.,netcdf=.true.,
/
```

- **wrf_nmm_regional**: If true, background comes from WRF NMM. When using WRF ARW or performing a global analysis, set it to false.
- **wrf_mass_regional**: If true, background comes from WRF ARW. When using WRF NMM or performing a global analysis, set it to false.
- **netcdf**: If true, wrf files are in NETCDF format, otherwise wrf files are in binary format. This option only works when performing a regional GSI analysis.

GSI Namelist-BKGERR

```
&BKGERR
  as=${as_op}
  vs=${vs_op}
  hzscl=${hzscl_op}
  bw=0.,fstat=.true.,
/
```

- Refer to the talk "Background Errors Estimation and Tuning"

GSI Namelist-OBSQC

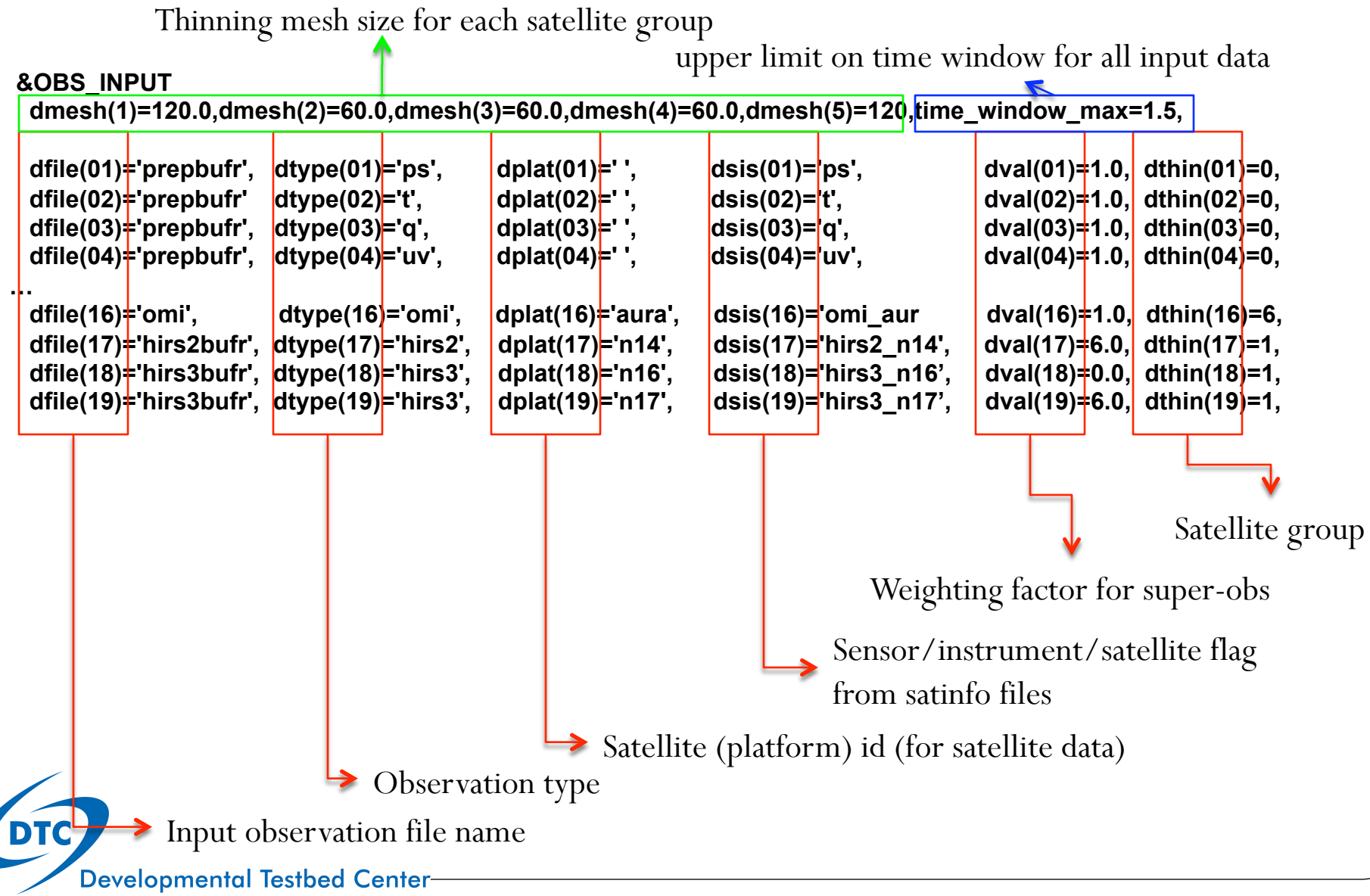
Observation Quality Control Variables

```
&OBSQC
 dfact=0.75,dfact1=3.0,noiqc=.false.,c_varqc=0.02,vadfile='prepbufr',
/
```

- Parameters used for gross error checks are defined in "info" files.
- Parameters above are used for nonlinear (variational) quality control (QC).
- In the namelist SETUP, **niter_no_qc** defines number of inner iterations without nonlinear QC for each outer iteration.

GSI Namelist-OBS_INPUT

Input Data Control



Info files-convinfo

- Control the usage of conventional data (t, q, ps, radial wind, …) and GPS RO refractivity and bending angle.
- Description can be find in GSI User's Guide (chapter 4) and ~/comGSI_v2/fix/global_convinfo.txt.

!otype	type	sub	iuse	twindow	numgrp	ngroup	nmiter	gross	ermax	ermin	var_b	var_pg	ithin	rmesh	pmesh	npred
tcp	112	0	1	3.0	0	0	0	75.0	5.0	1.0	75.0	0.000000	0	0.	0.	0
ps	120	0	1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
ps	132	0	-1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
ps	180	0	1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
ps	181	0	1	3.0	0	0	0	3.6	3.0	1.0	3.6	0.000300	0	0.	0.	0
ps	182	0	1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
ps	183	0	-1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
ps	187	0	1	3.0	0	0	0	4.0	3.0	1.0	4.0	0.000300	0	0.	0.	0
t	120	0	1	3.0	0	0	0	8.0	5.6	1.3	8.0	0.000001	0	0.	0.	0
t	126	0	-1	3.0	0	0	0	8.0	5.6	1.3	8.0	0.001000	0	0.	0.	0
t	130	0	1	3.0	0	0	0	7.0	5.6	1.3	7.0	0.001000	0	0.	0.	0

iuse=1: use data

iuse=0: do not use data

iuse=-1: monitor data

Gross Error Parameter

gross: gross error

ermax: max error

ermin: min error

If(abs(simulated)-observation)/obserrx > gross, the observation is rejected, where obserrx = max(ermin,min(ermax,obs error)

Info files-satvinfo

- Control the usage of satellite radiance data (AMSU-A, AMSU-B, HIRS3, …).
- You may find some description of the file in radinfo.f90.

!sensor/instr/sat	chan	iuse	error	ermax	var_b	var_pg
amsua_n15	1	1	3.000	4.500	10.00	0.000
amsua_n15	2	1	2.000	4.500	10.00	0.000
amsua_n15	3	1	2.000	4.500	10.00	0.000
amsua_n15	4	1	0.600	2.500	10.00	0.000
amsua_n15	5	1	0.300	2.000	10.00	0.000
amsua_n15	6	1	0.230	2.000	10.00	0.000
amsua_n15	7	1	0.250	2.000	10.00	0.000
amsua_n15	8	1	0.275	2.000	10.00	0.000
amsua_n15	9	1	0.340	2.000	10.00	0.000
amsua_n15	10	1	0.400	2.000	10.00	0.000
amsua_n15	11	-1	0.600	2.500	10.00	0.000

iuse:

=-2: do not use

=-1: monitor if diagnostics produced

=0: monitor and use in QC only

=1: use data with complete quality control

=2 use data with no airmass bias correction

=3 use data with no angle dependent bias correction

=4 use data with no bias correction

ermax: max error (for QC)

error: variance for each satellite channel

Questions? …