WRF-Var System

WRF Tutorial

July 25th 2006

Syed RH Rizvi

MMM Division, NCAR,

PO Box 3000, Boulder, Colorado,

80307-3000, USA

rizvi@ucar.edu

Web Site: http://www.mmm.ucar.edu/wrf/WG4

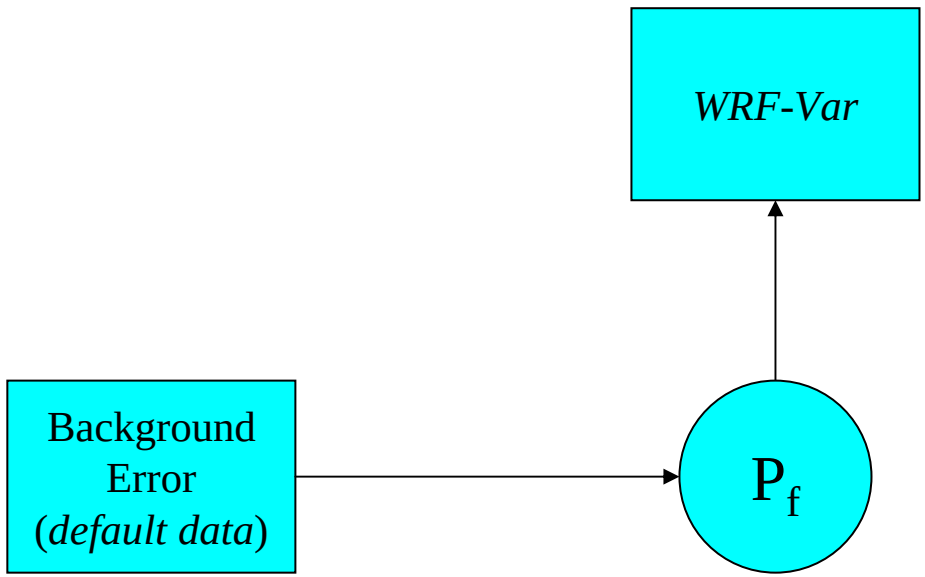
WRF-Var in the WRF Modeling System

WRF-Var in the WRF Modeling System

Blue = Supported by WRF-Var Team

WRF-Var in the WRF Modeling System

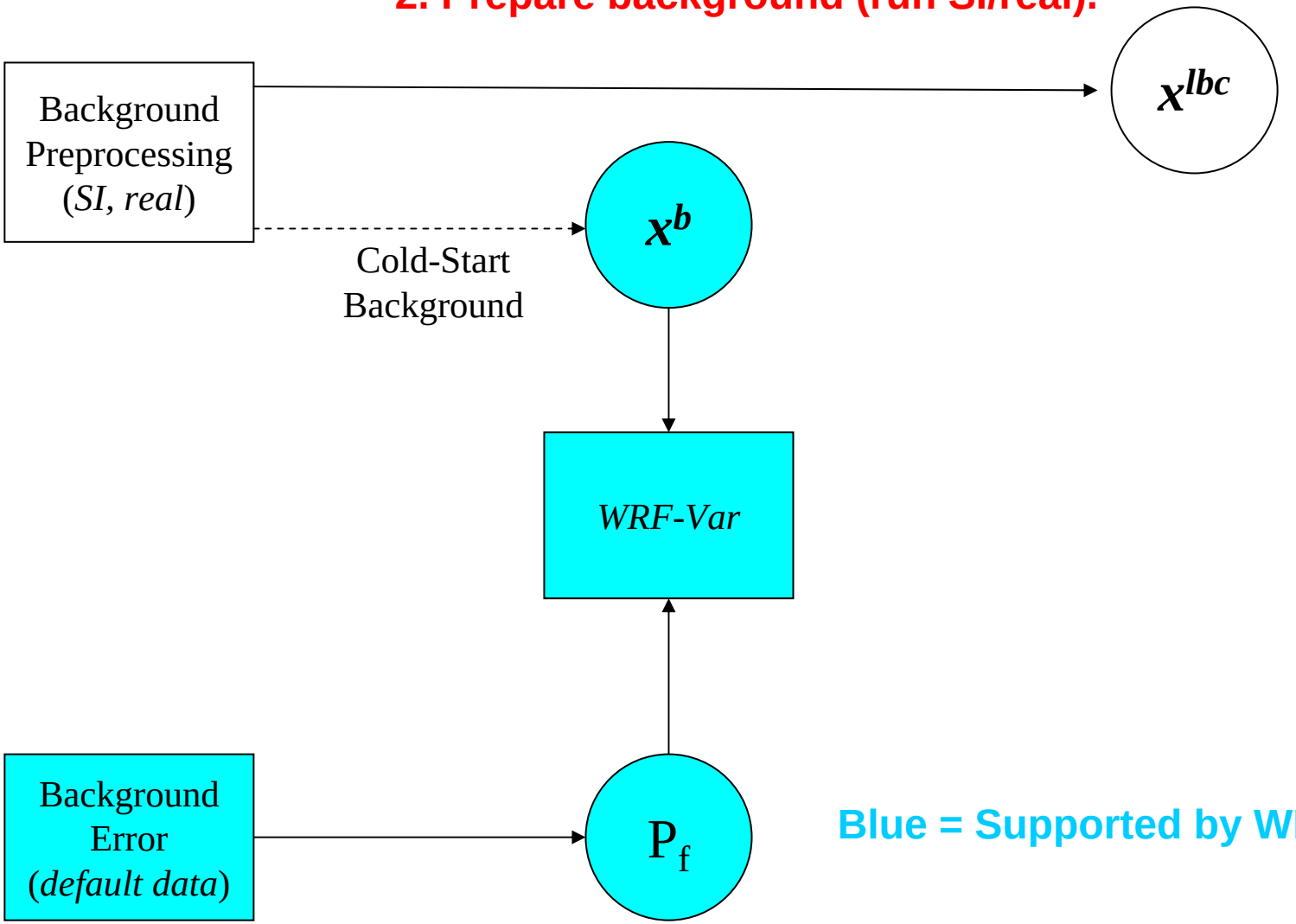
1. Prepare BE data (initially use default statistics)



Blue = Supported by WRF-Var Team

WRF-Var in the WRF Modeling System

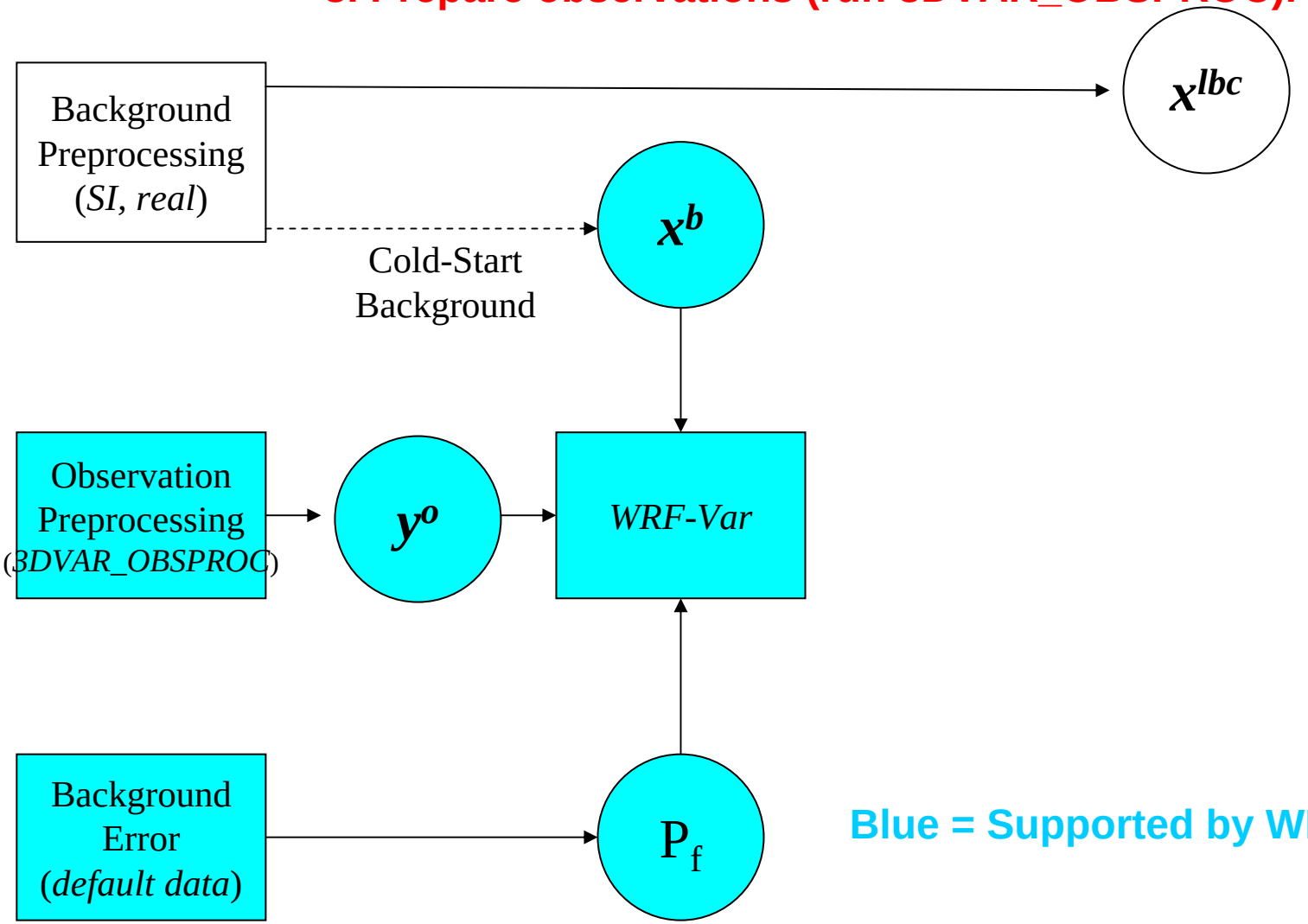
2. Prepare background (run SI/real).



Blue = Supported by WRF-Var Team

WRF-Var in the WRF Modeling System

3. Prepare observations (run 3DVAR_OBSPROC).

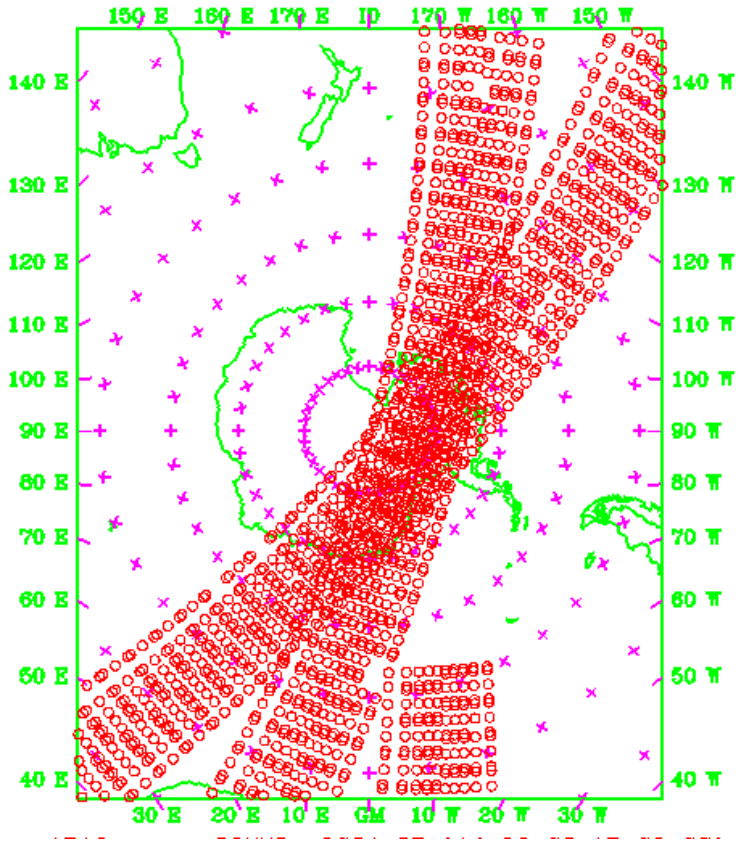


Blue = Supported by WRF-Var Team

Observation Preprocessing (*3DVAR_OBSPROC*)

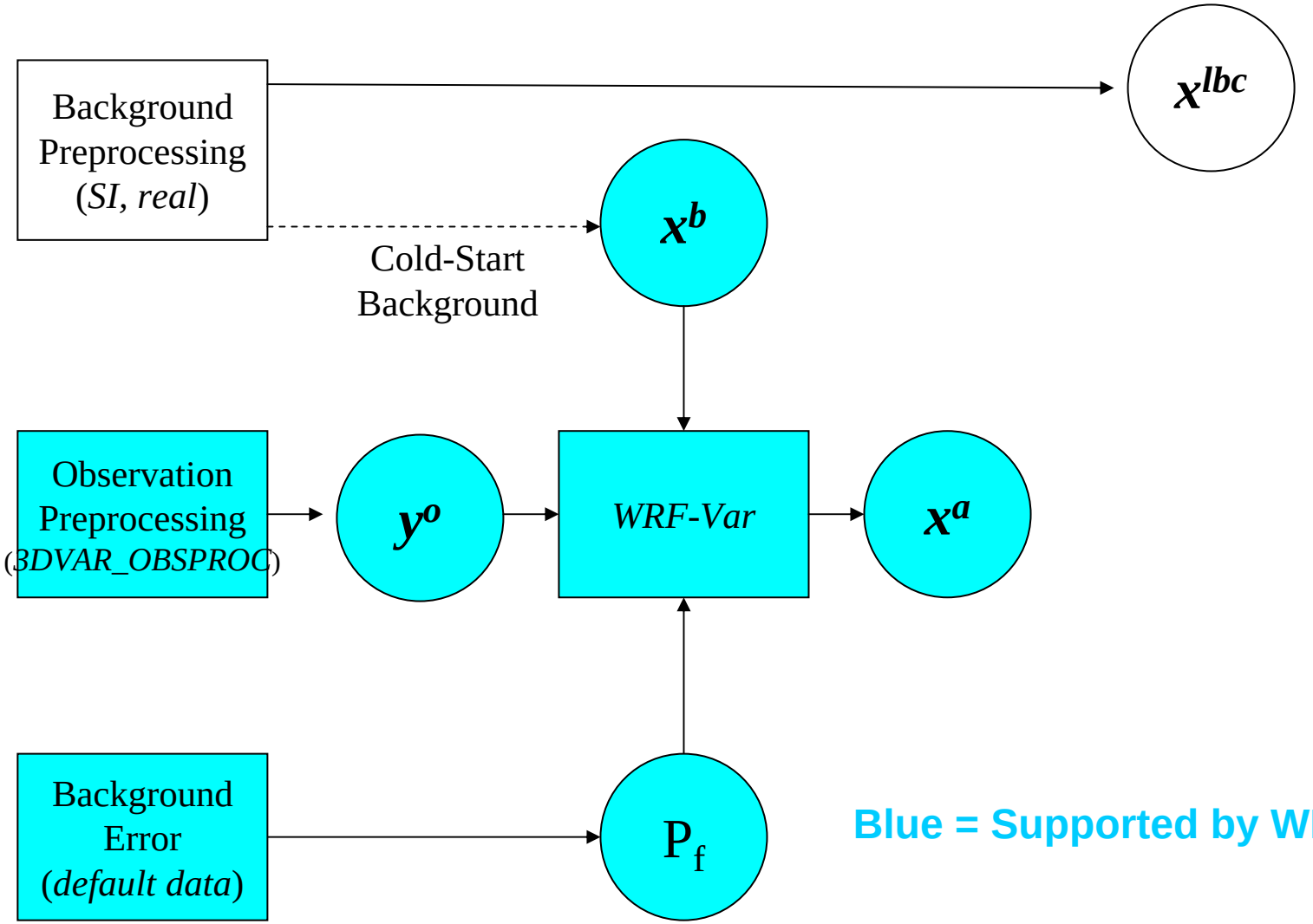
- Reads in observation files from decoders/GTS.
- Performs gross QC, e.g. domain/time, consistency, duplicate, merging.
- Simple thinning option.
- Assign observation errors.
- Outputs in text “3D-Var format” for further QC and assimilation in WRF-Var.
- Plots observation distributions.
- Note: Work under way to convert to BUFR, rather than text files.

Example thinned AIRS distribution
00 UTC 15th May 2004 (+/-2hrs):



WRF-Var in the WRF Modeling System

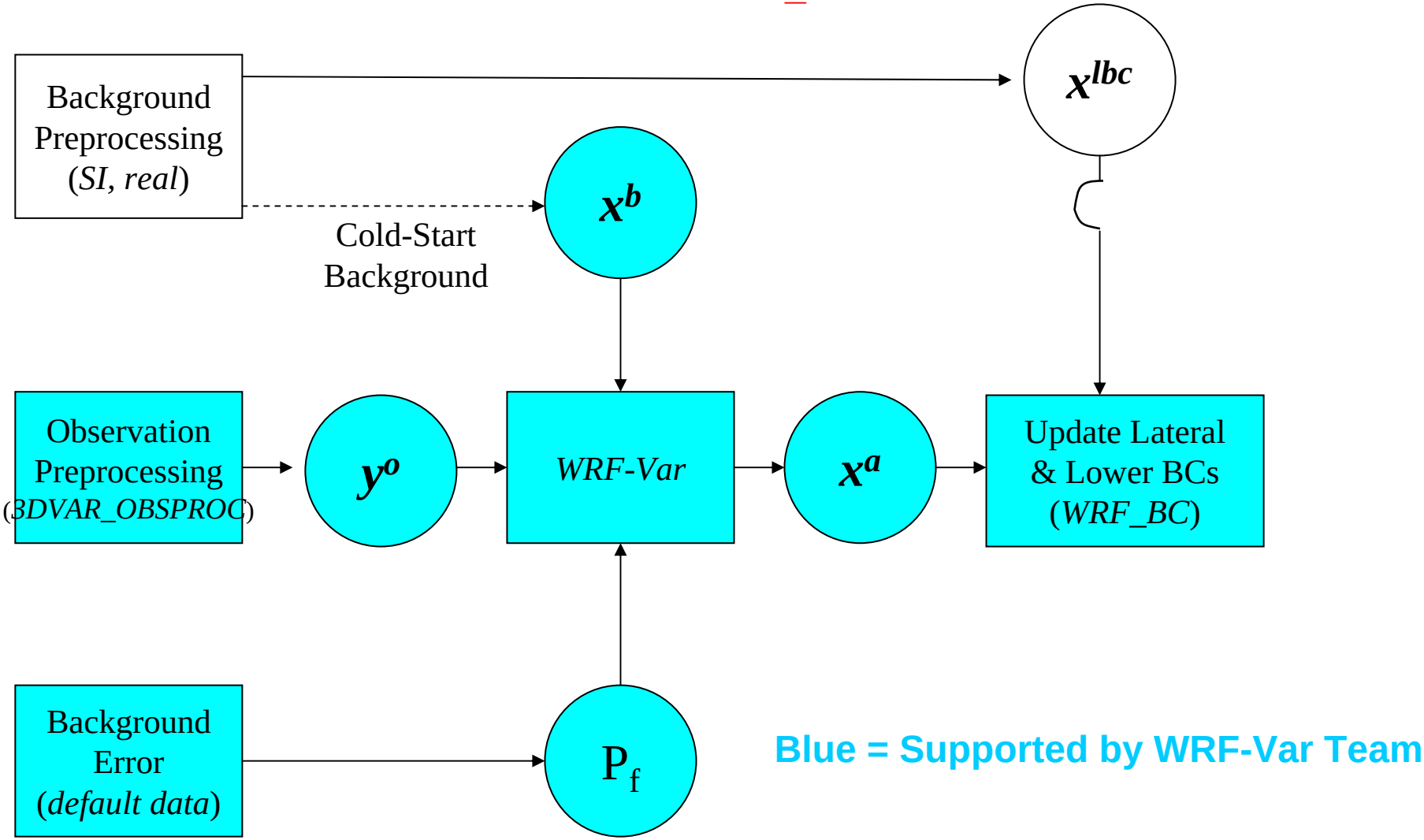
4. Run WRF-Var.



Blue = Supported by WRF-Var Team

WRF-Var in the WRF Modeling System

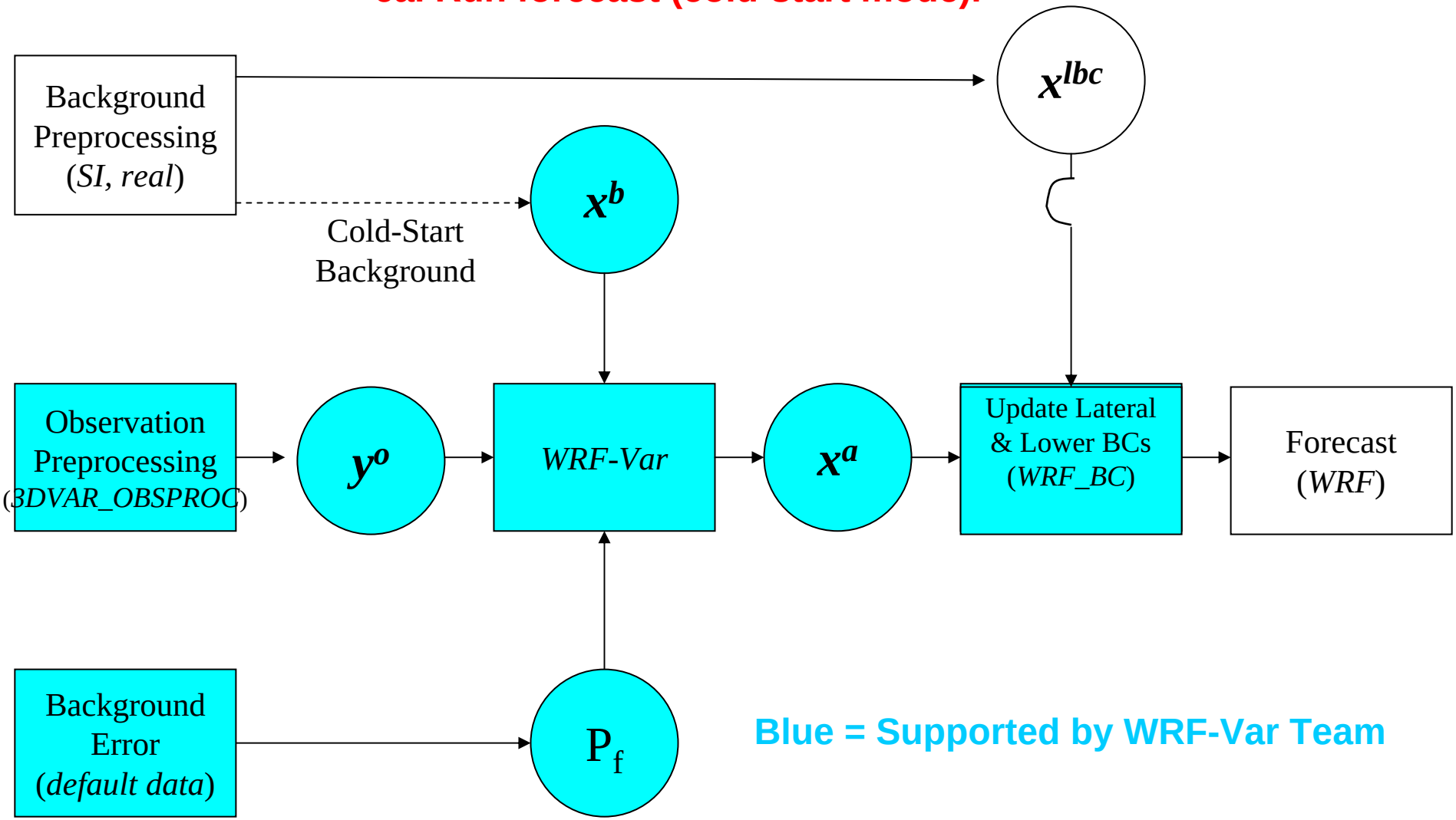
5. Run WRF_BC.



Blue = Supported by WRF-Var Team

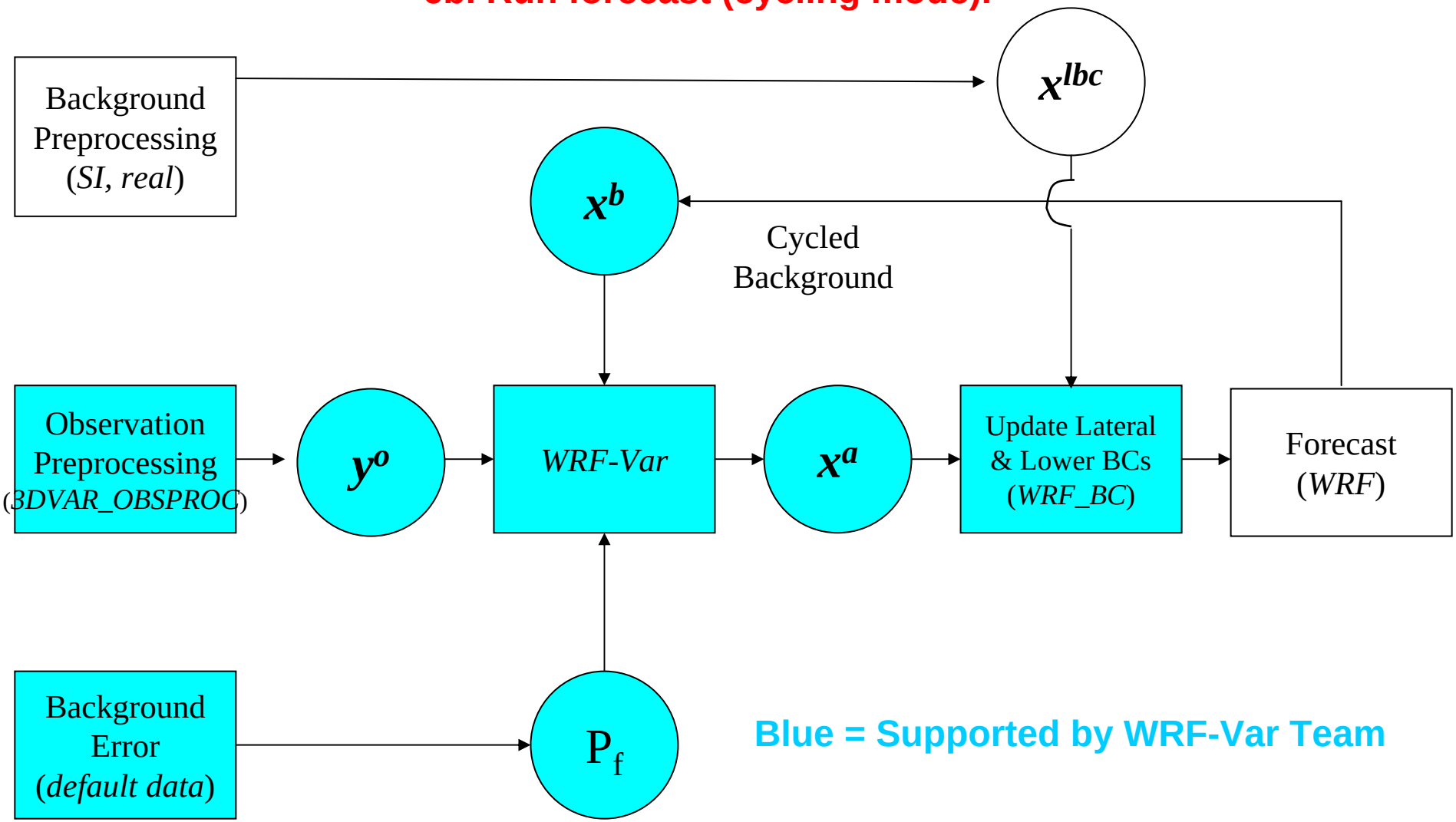
WRF-Var in the WRF Modeling System

6a. Run forecast (cold-start mode).



WRF-Var in the WRF Modeling System

6b. Run forecast (cycling mode).



Cold-Start/Cycling Use of WRF-Var

- Initial/“one-off”” tests of WRF-Var and WRF will adopt the “cold-start” method.

- However, there are a number of advantages in using the “cycling” approach:

1. The analysis is better balanced (background/subsequent model are the same).
2. Typically, the cycled background will contain higher resolution information.
3. Fewer spin-up problems (e.g. hydrometeors are initialized to zero in SI/real).
4. Improvements (and degradations!) will accumulate through subsequent cycles.
5. The assimilation is cleaner (no danger of assimilating observations twice).

Therefore, cycling WRF-Var/WRF is the ultimate goal!

- **Warning:** If you are only assimilating a sub-set of the observations available to the cold-start background (e.g. global analysis), you may find you cannot beat “noobs” (I.e. running WRF from real output). In this case, use “cold-start” mode.

Background Error (BE) Estimation in WRF-Var

The number 1 question from WRF-Var users is

“What background error covariances are best for my applicatio

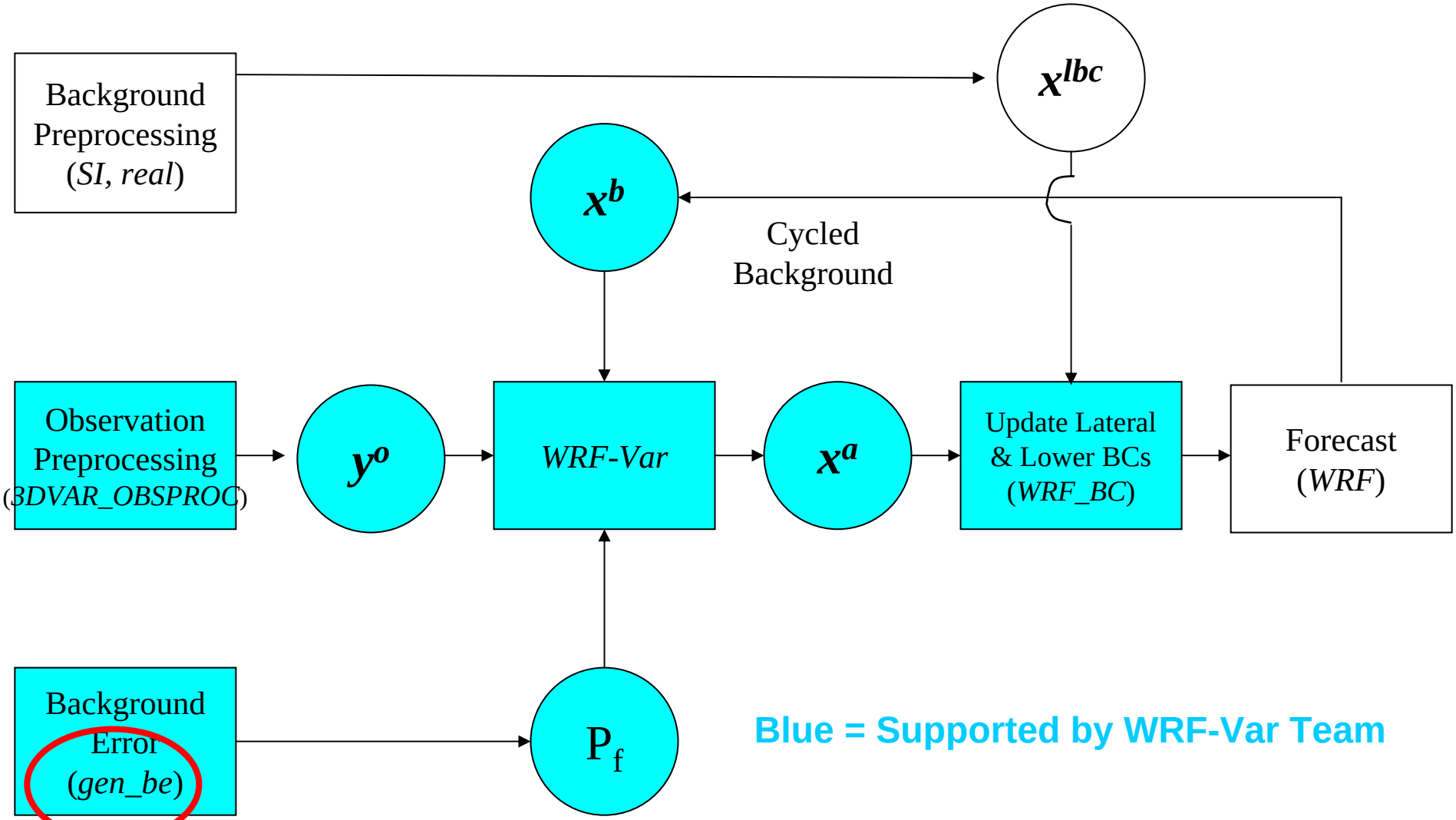
Procedure:

1. Use default statistics files supplied with code (MM5, GFS-based
2. Create your own, once you have run your system for ~a few we
3. Implement, tune, and iterate.

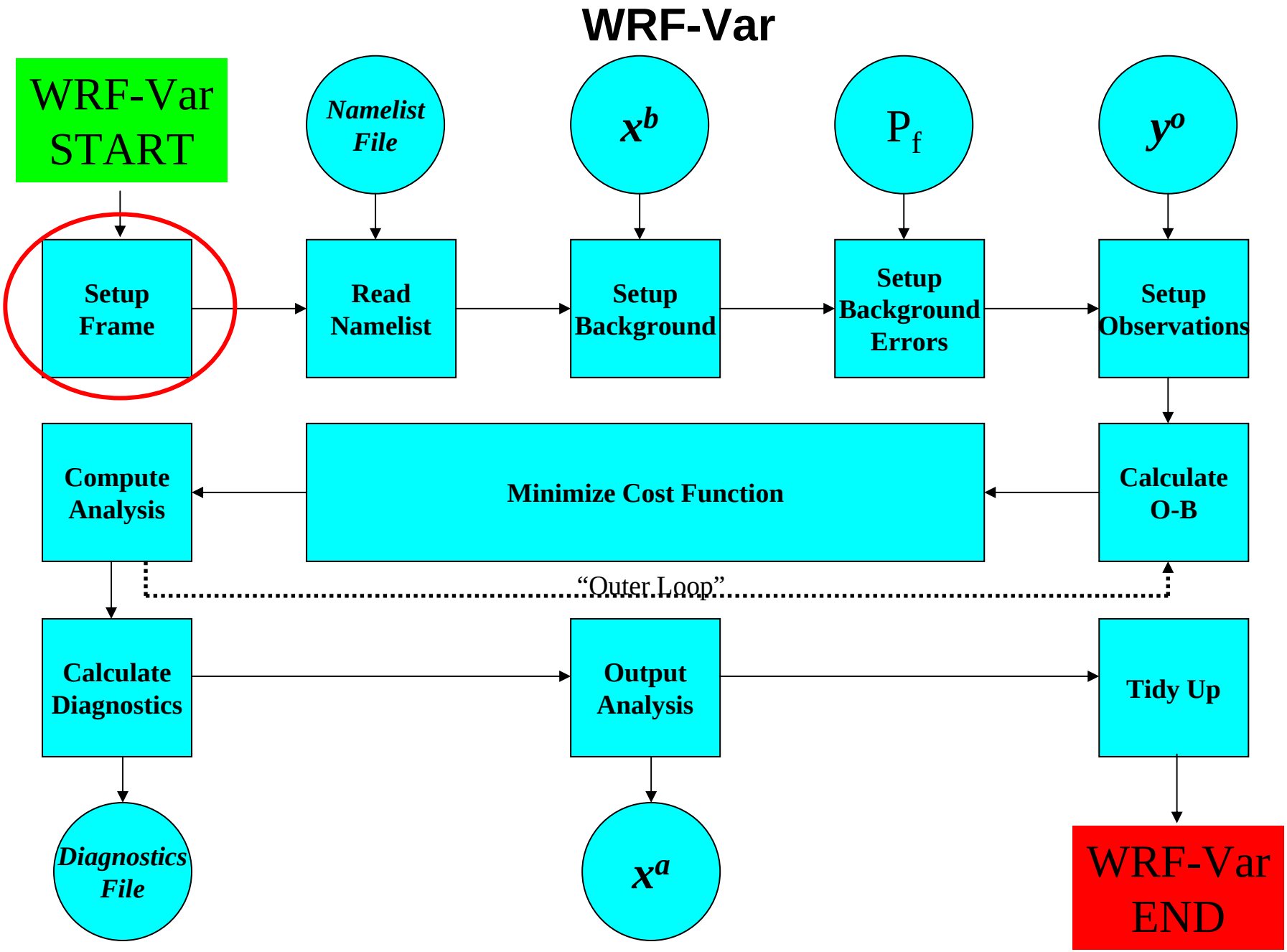
A new utility *gen_be* has been developed at NCAR to calculate BEs

WRF-Var in the WRF Modeling System

7. WRF-Var/WRF Ultimate Configuration!



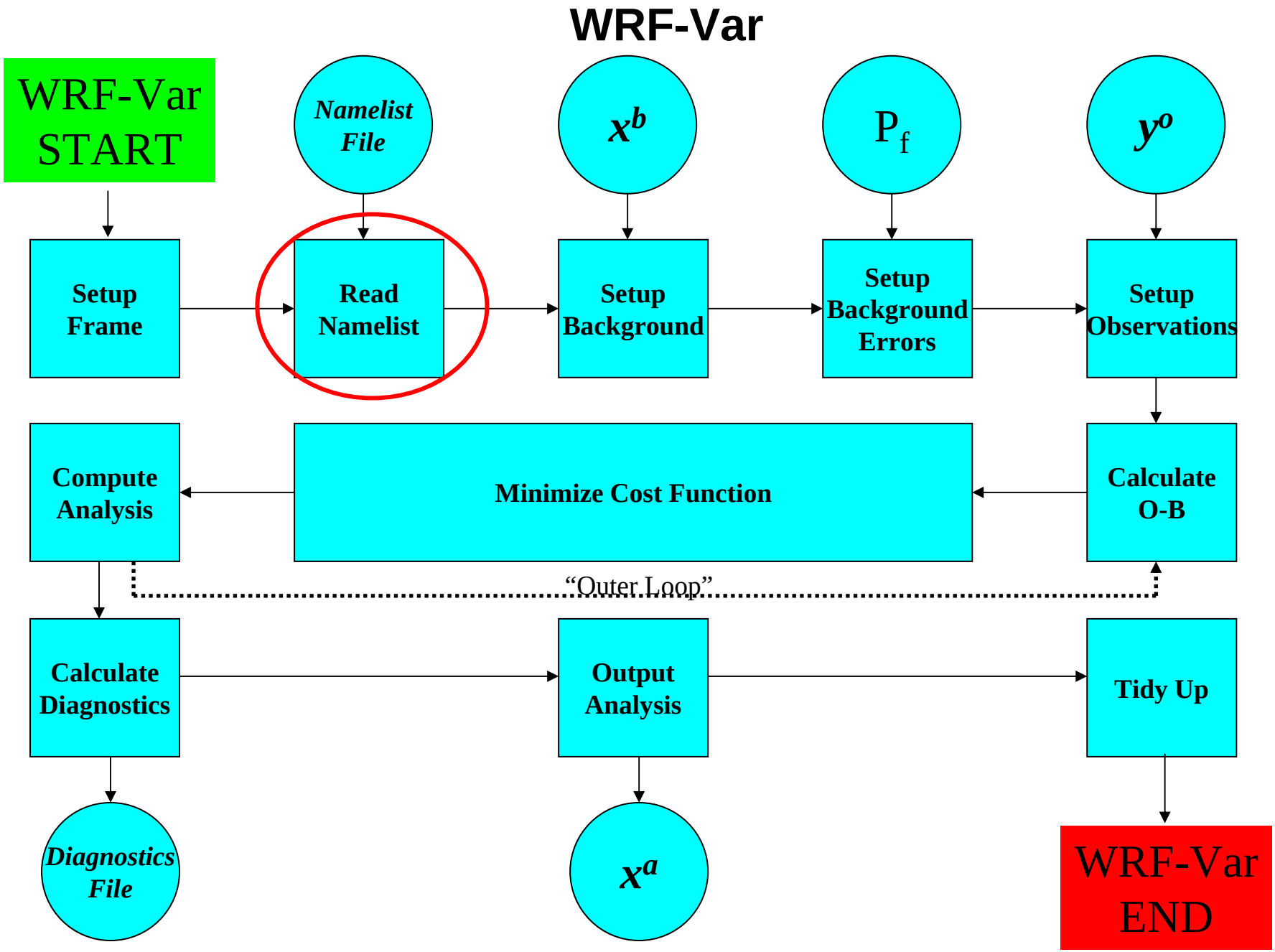
WRF-Var Code Overview


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Frame

- **Reads grid dimensions from “namelist.input” file.**

- **Use WRF framework’s distributed memory capability to initialize tile, memory, patch dimensions, etc.**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

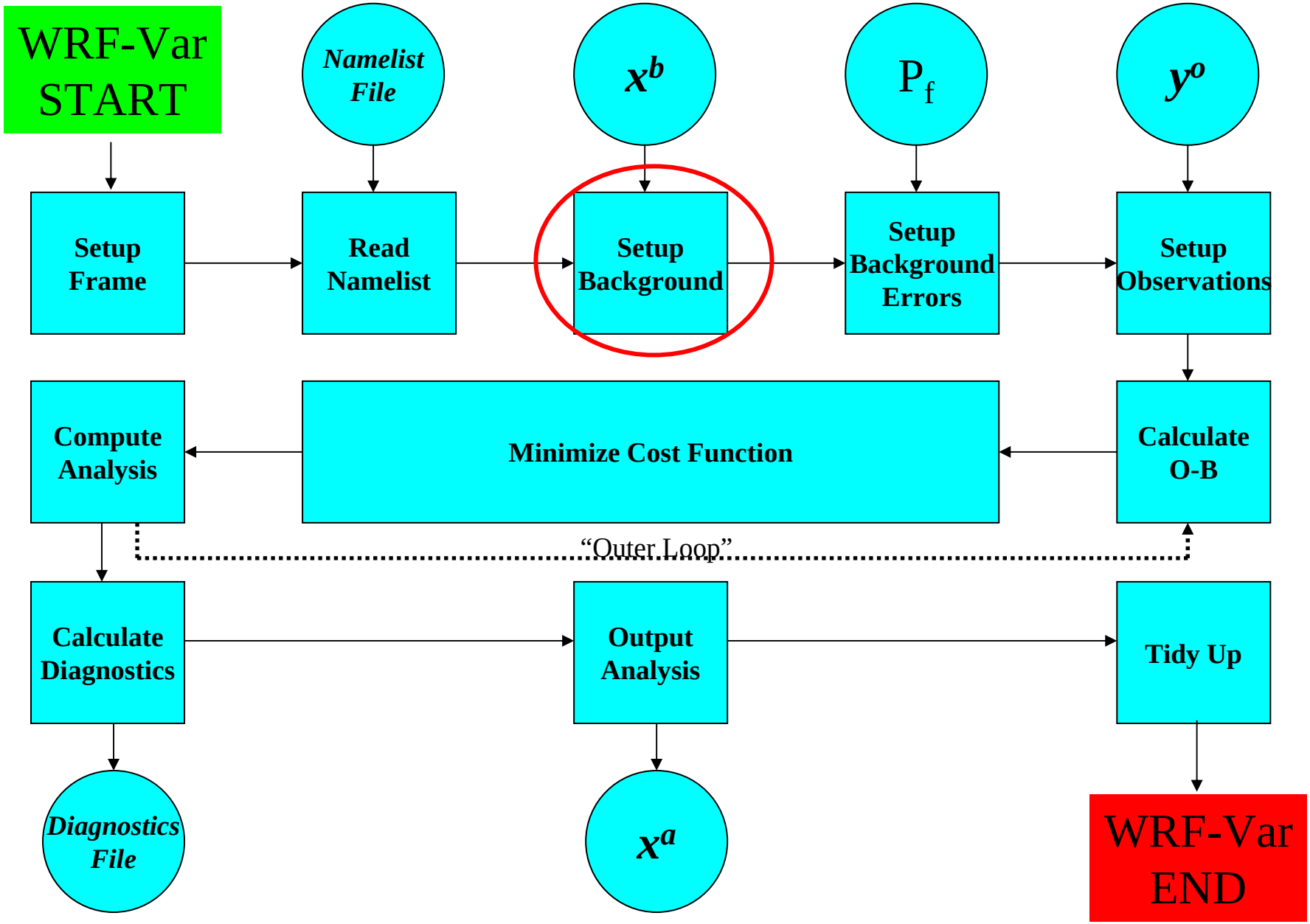
Read Namelist

- **Reads WRF-Var data assimilation options from "namelist.3dvar" file.**
- **"namelist.3dvar" file is created automatically at run-time by the DA_Run_WRF_Var.csh script in wrfvar/run.**
- **Performs consistency checks between namelist options.**

namelist.3dvar

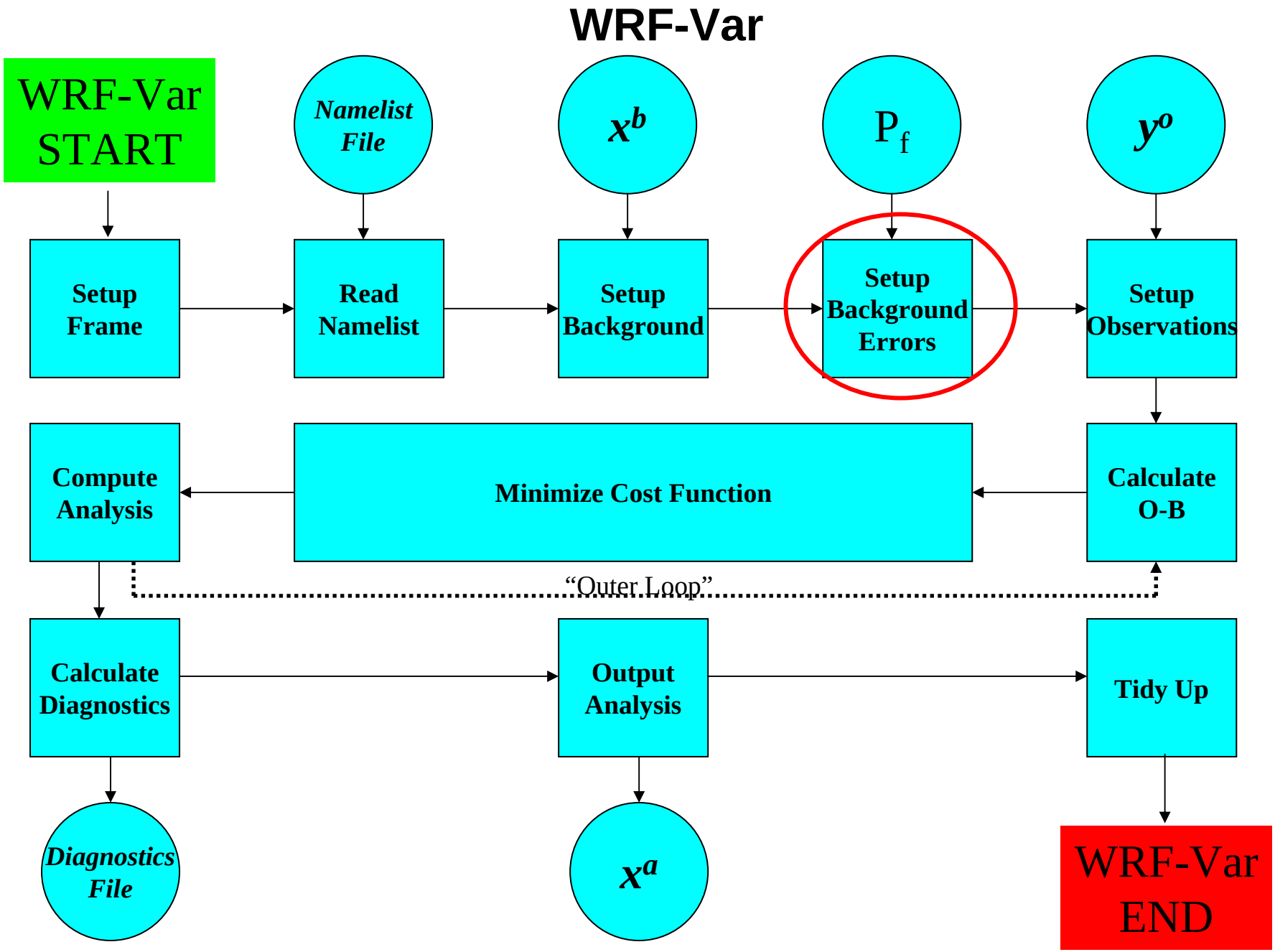
```
&record1
 MODEL_TYPE = 'WRF',
 WRITE_INCREMENTS = .FALSE. ,
 GLOBAL          = .FALSE.,
 PRINT_DETAIL   = 0 /

&record2
 ANALYSIS_TYPE = '3D-VAR',
 ANALYSIS_DATE = '2004-05-01_00:00:00.0000',
 ANALYSIS_ACCU =  900 /

&record3
 fg_format = 1,
 ob_format = 2,
 num_fgat_time = 1 /

&record4
 PROCESS_OBS    = 'YES',
 obs_qc_pointer = 0,
 Use_SynopObs   = .TRUE.,
 Use_ShipsObs   = .TRUE.,
 Use_MetarObs   = .TRUE.,
 Use_PilotObs   = .TRUE.,
 Use_SoundObs   = .TRUE.,
 Use_SatemObs   = .TRUE.,
 Use_GeoAMVObs  = .TRUE.,
 Use_PolarAMVObs = .TRUE.,
 Use_AirepObs   = .TRUE.,
 Use_GpspwObs   = .TRUE.,
 Use_GpsrefObs  = .TRUE.,
 Use_ProfilerObs = .TRUE.,
 Use_BuoyObs     = .TRUE.,
 Use_SsmiRetrievalObs = .FALSE.,
 Use_SsmiTbObs  = .FALSE.,
 use_ssmt1obs   = .FALSE.,
 use_ssmt2obs   = .FALSE.,
 use_qscatobs   = .TRUE.,
 use_radarobs   = .FALSE.,
 Use_Radar_rv   = .FALSE.,
 Use_Radar_rf   = .FALSE.,
 check_max_iv   = .FALSE.,
 use_obs_errfac = .FALSE.,
 put_rand_seed  = .FALSE.,
 omb_set_rand   = .FALSE.,
 omb_add_noise  = .FALSE. /

&record5
 TIME_WINDOW    = 3.,
 /

&record6
 max_ext_its    = 1,
 EPS          = 1.E-02, 1.E-02, 1.E-02, 1.E-02, 1.E-02, 1.E-02, 1.E-02,,
 NTMAX         = 100,
 NSAVE         = 4,
 WRITE_SWITCH   = .FALSE.,
 WRITE_INTERVAL = 5 /

&record7
 RF_PASSES      = 6,
 VAR_SCALING1   = 1.0,
 VAR_SCALING2   = 1.0,
 VAR_SCALING3   = 1.0,
 VAR_SCALING4   = 1.0,
 VAR_SCALING5   = 1.0,
 LEN_SCALING1   = 1.0,
 LEN_SCALING2   = 1.0,
 LEN_SCALING3   = 1.0,
 LEN_SCALING4   = 1.0,
 LEN_SCALING5   = 1.0 /

&record8
 def_sub_domain = .FALSE.,
 x_start_sub_domain = 55.0,
 y_start_sub_domain = 35.0,
 x_end_sub_domain   = 80.0,
 y_end_sub_domain   = 60.0 /

&record10
 Testing_3DVAR  = .FALSE.,
 Test_Transforms = .FALSE.,
 Test_Statistics = .FALSE.,
 Interpolate_Stats = .TRUE. /

&record11
 minimisation_option = 2,
 write_outer_loop    = .FALSE.,
 lat_stats_option    = .FALSE.,
 calculate_cg_cost_function = .FALSE.,
 cv_options     = 3,
 cv_options_hum = 3,
 check_rh       = 2,
 as1          = 0.25, 0.75, 1.5,
 as2          = 0.25, 0.75, 1.5,
 as3          = 0.25, 0.75, 1.5,
 as4          = 0.25, 0.75, 1.5,
 as5          = 0.25, 0.75, 1.5,
sfc_assi_options = 1,
 set_omb_rand_fac = 1.0,
 seed_array1    = 0,
 seed_array2    = 0 /

&record12
 balance_type   = 1 /

&record13
 vert_corr      = 2,
 vertical_ip    = 0,
 vert_evalue    = 1,
 max_vert_var1  = 99.0,
 max_vert_var2  = 99.0,
 max_vert_var3  = 99.0,
 max_vert_var4  = 99.0,
 max_vert_var5  = 99.0 /

&pseudo_ob_nl
 num_pseudo     = 0,
 pseudo_x       = 165.0,
 pseudo_y       = 65.0,
 pseudo_z       = 15.0,
 pseudo_val     = 1.0,
 pseudo_err     = 1.0,
 pseudo_var     = 'u' /
```

WRF-Var


WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

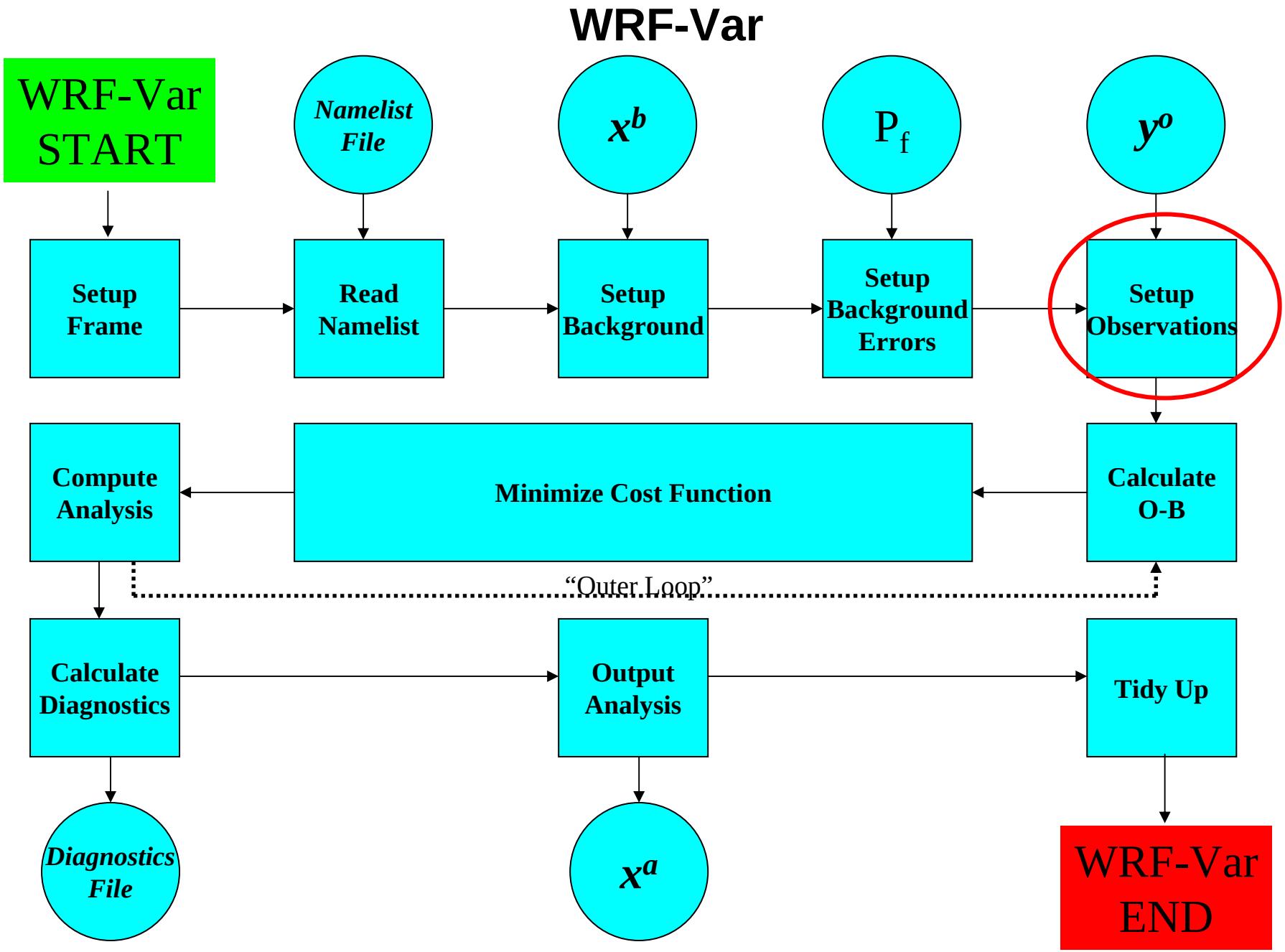
Setup Background (First-Guess)

- **Reads in the first-guess field.**
- **Format depends on namelist option “fg_format” – 1 = WRF, 2 = MM5, etc.**
- **Extracts necessary fields.**
- **Creates background FORTRAN 90 derived data type “xb” e.g. xb % mix, xb % u(:,:,:), ….**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

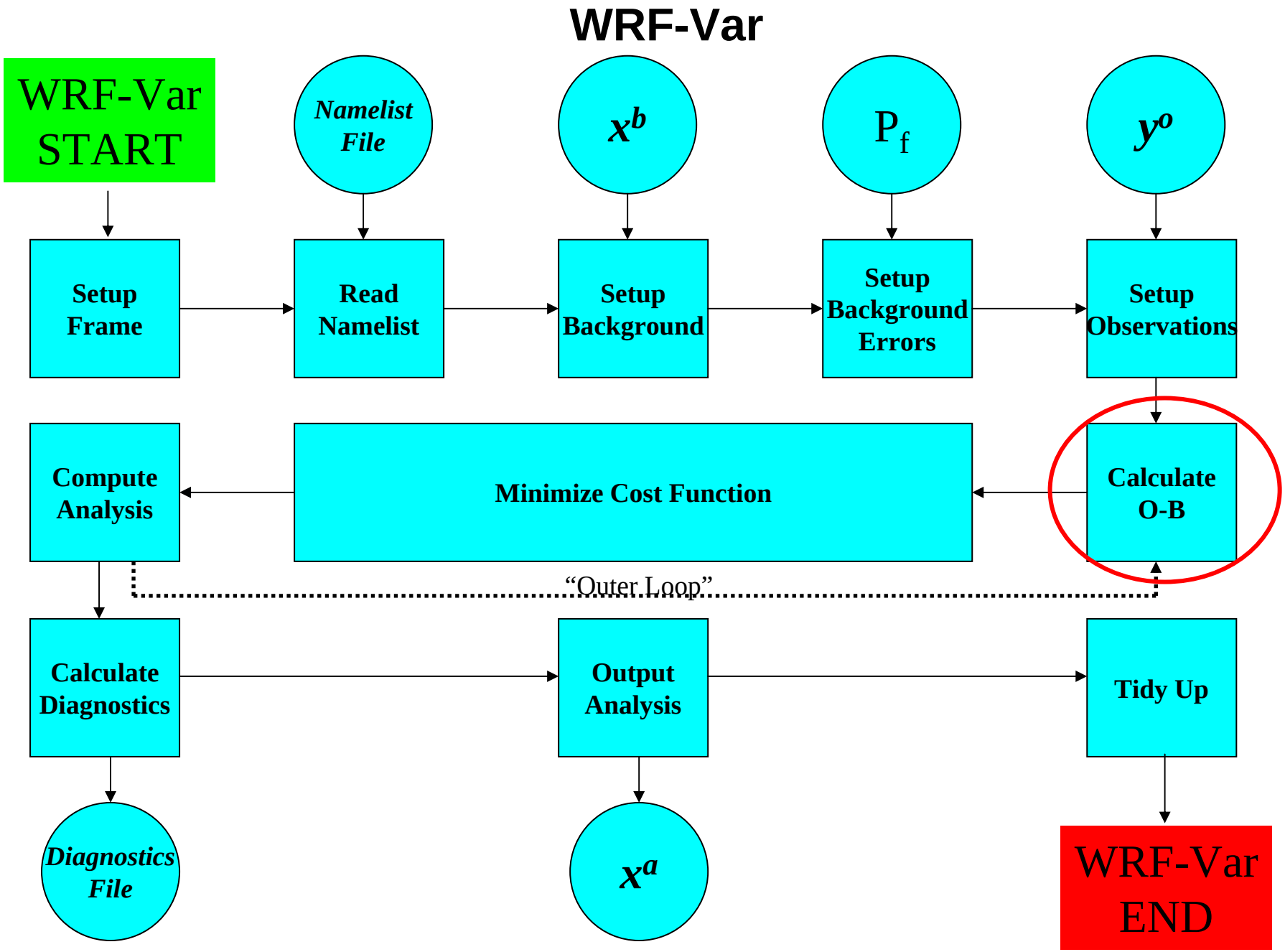
Setup Background Errors

- **Reads in background error statistics.**
- **Format depends on namelist option “cv_options” – 2=MM5, 3 = GFS-based, 4=Global, 5=WRF regional.**
- **Extracts necessary quantities – eigenvectors, eigenvalues, lengthscales, regression coefficients, etc (see gen_be talk).**
- **Creates background error FORTRAN 90 derived data type “be” e.g. be % v1 % evec(:,:), be % v2 % eval(:), etc, ….**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Setup Observations

- **Reads in observations.**
- **Format depends on namelist variable “ob_format” – 1 = BUFR, 2 = ASCII “WRF-Var” format.**
- **Creates observation FORTRAN 90 derived data type “ob” e.g. ob % num_gpspw, ob % metar(:), ob % sound(:) % u(:), etc, ….**
- **Identifies Obs outside/inside the domain**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Calculate O-B
Minimize Cost Function
Compute Analysis
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

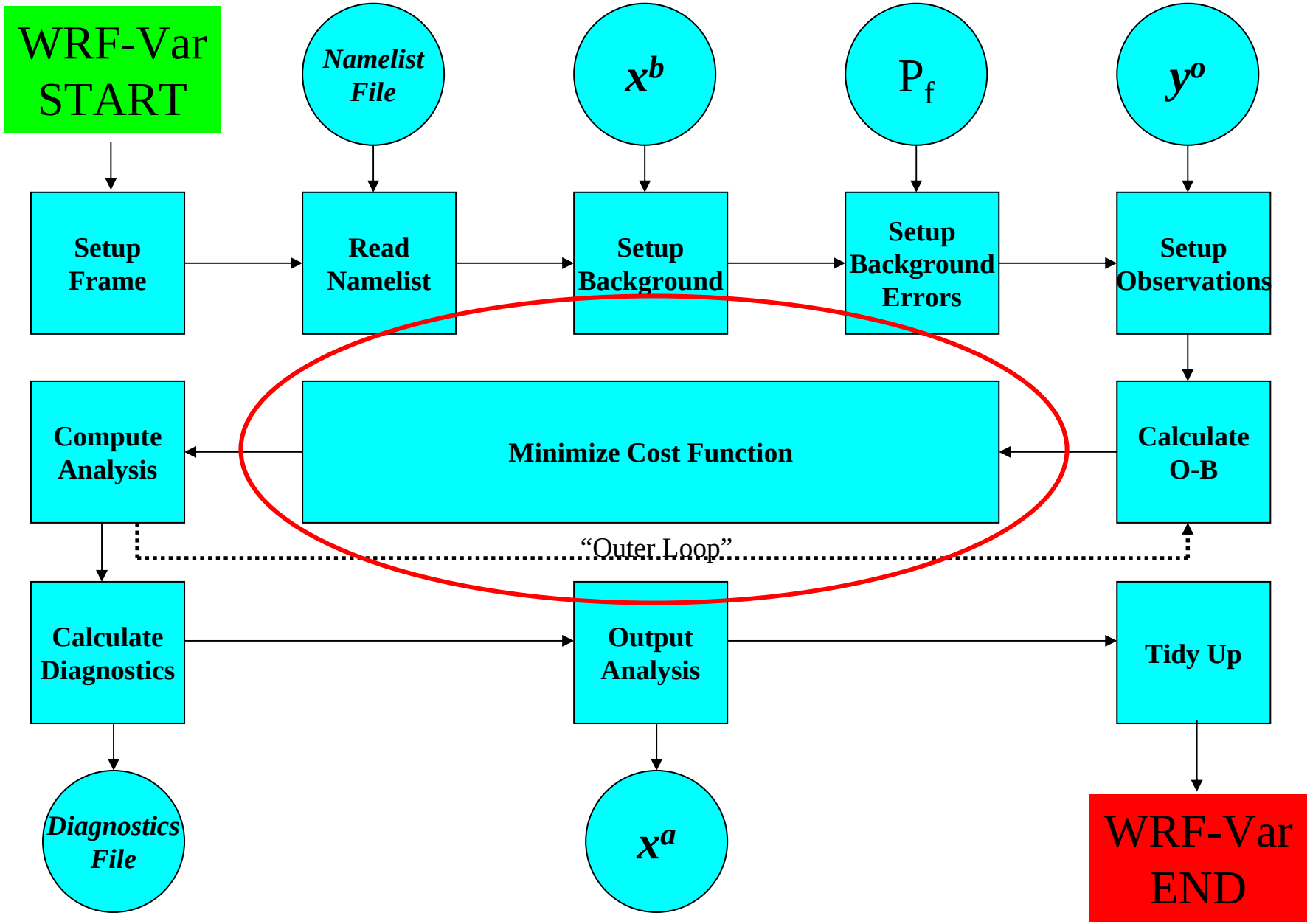
Calculate Innovation Vector (O-B)

- **Calculates “model equivalent” B of observation O through interpolation and change of variable.**

- **Computes observation minus first guess (O-B) value.**

- **Creates innovation vector FORTRAN 90 derived data type “iv” e.g. iv % metar(:), iv % qscat(:) % u, etc, ….**

WRF-Var


WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Minimize Cost Function

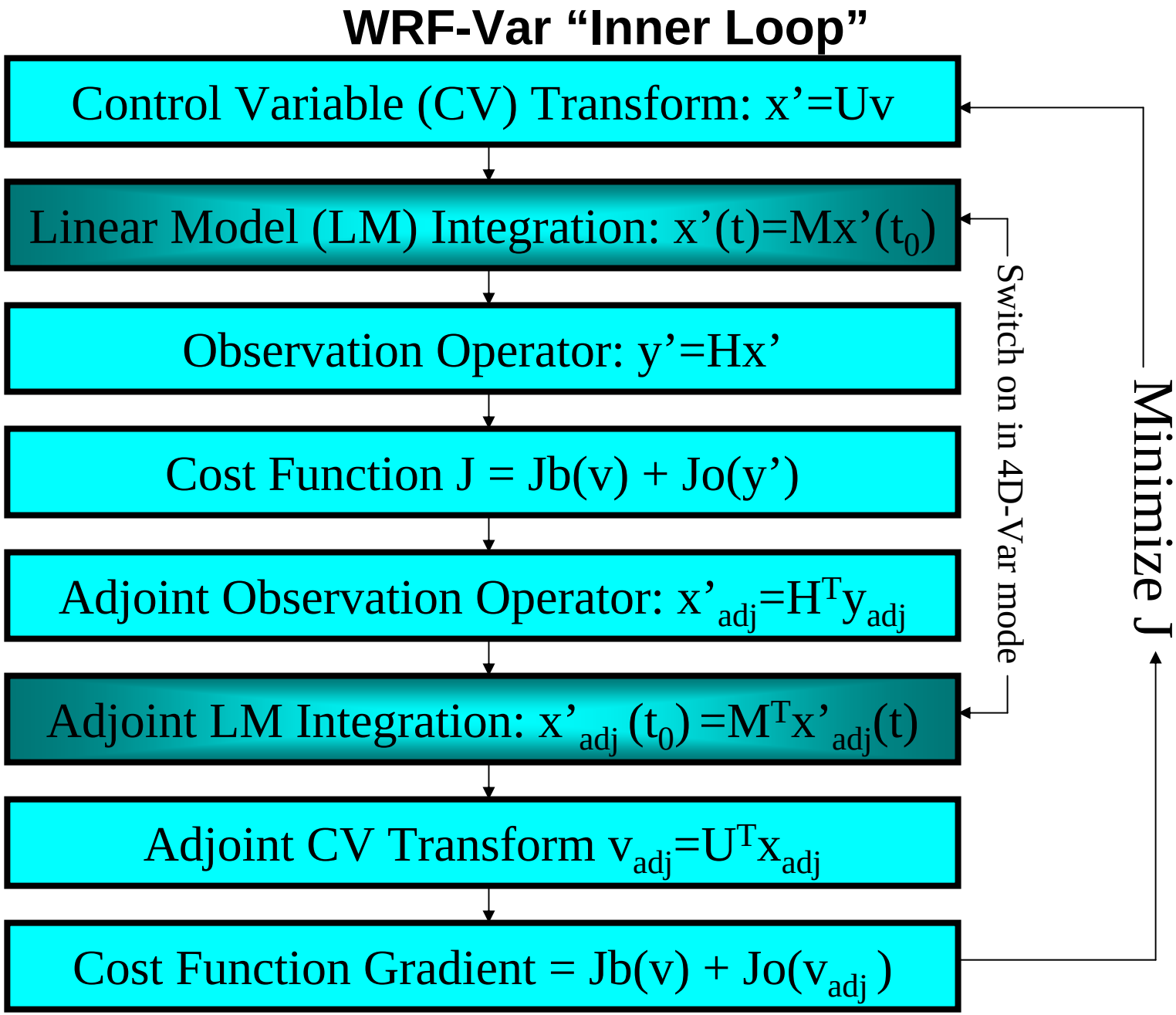
a) Initializes analysis increments to zero.

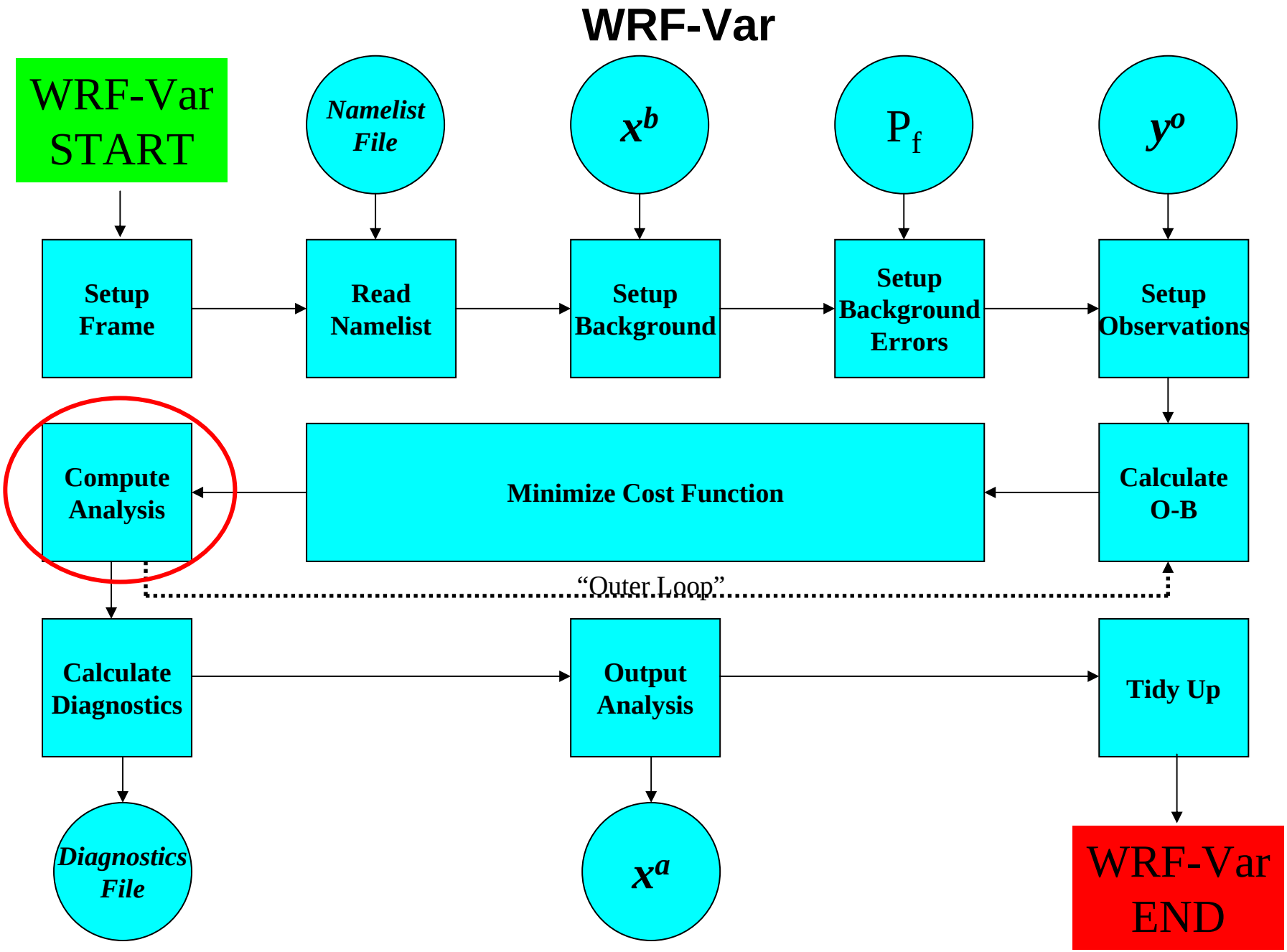
b) Computes cost function.

c) Computes gradient of cost function.

d) Uses cost function and gradient to calculate new value of control variable v

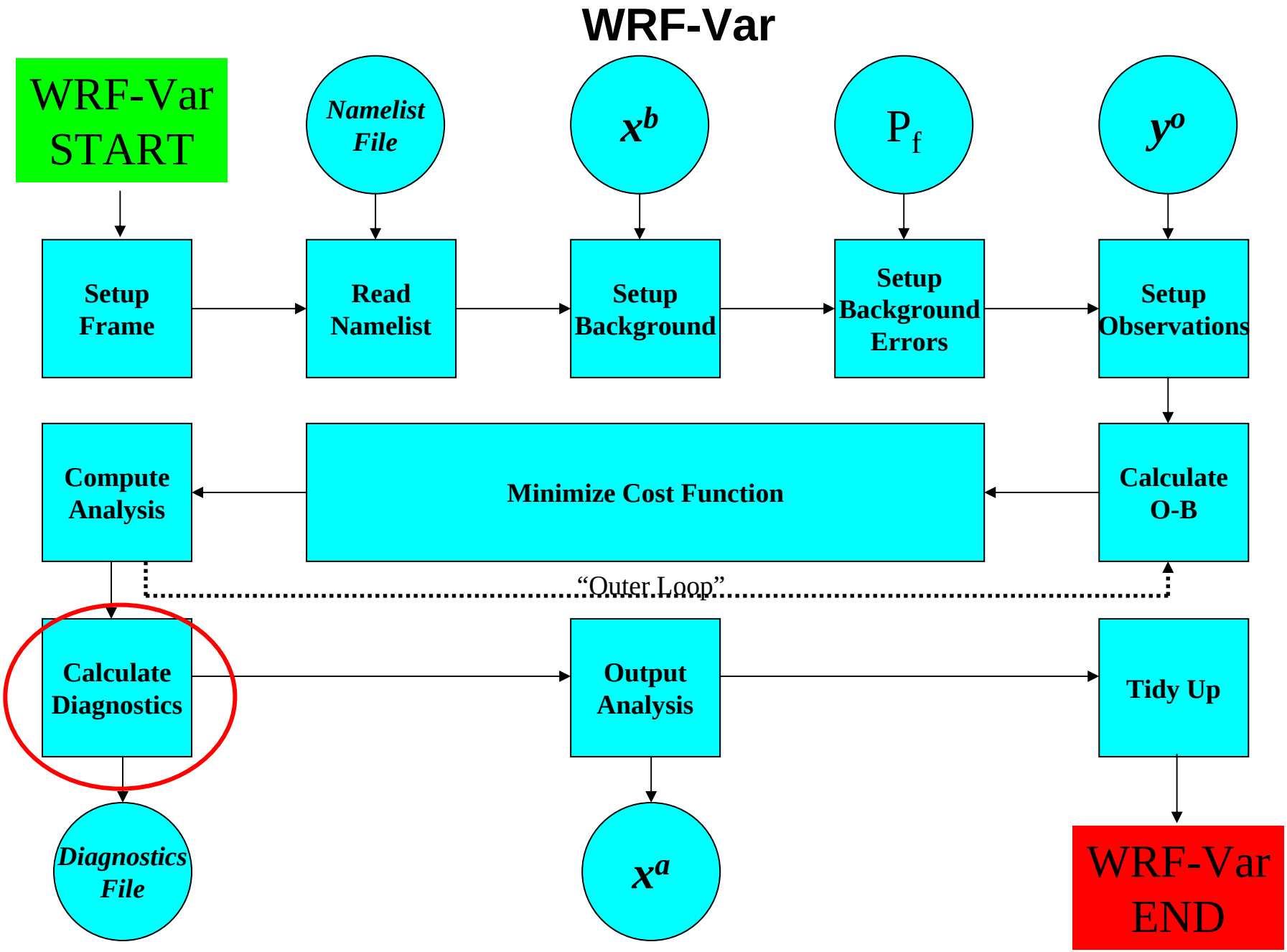
e) Iterate b) to d).


WRF-Var "Inner Loop"
Control Variable (CV) Transform: x'=Uv
Linear Model (LM) Integration: x'(t)=Mx'(t0)
Observation Operator: y'=Hx'
Cost Function J = Jb(v) + Jo(y')
Adjoint Observation Operator: x'adj=HTyadj
Adjoint LM Integration: x'adj (t0) =MTx'adj(t)
Adjoint CV Transform vadj=UTxadj
Cost Function Gradient = Jb(v) + Jo(vadj)
Switch on in 4D-Var mode
Minimize J


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

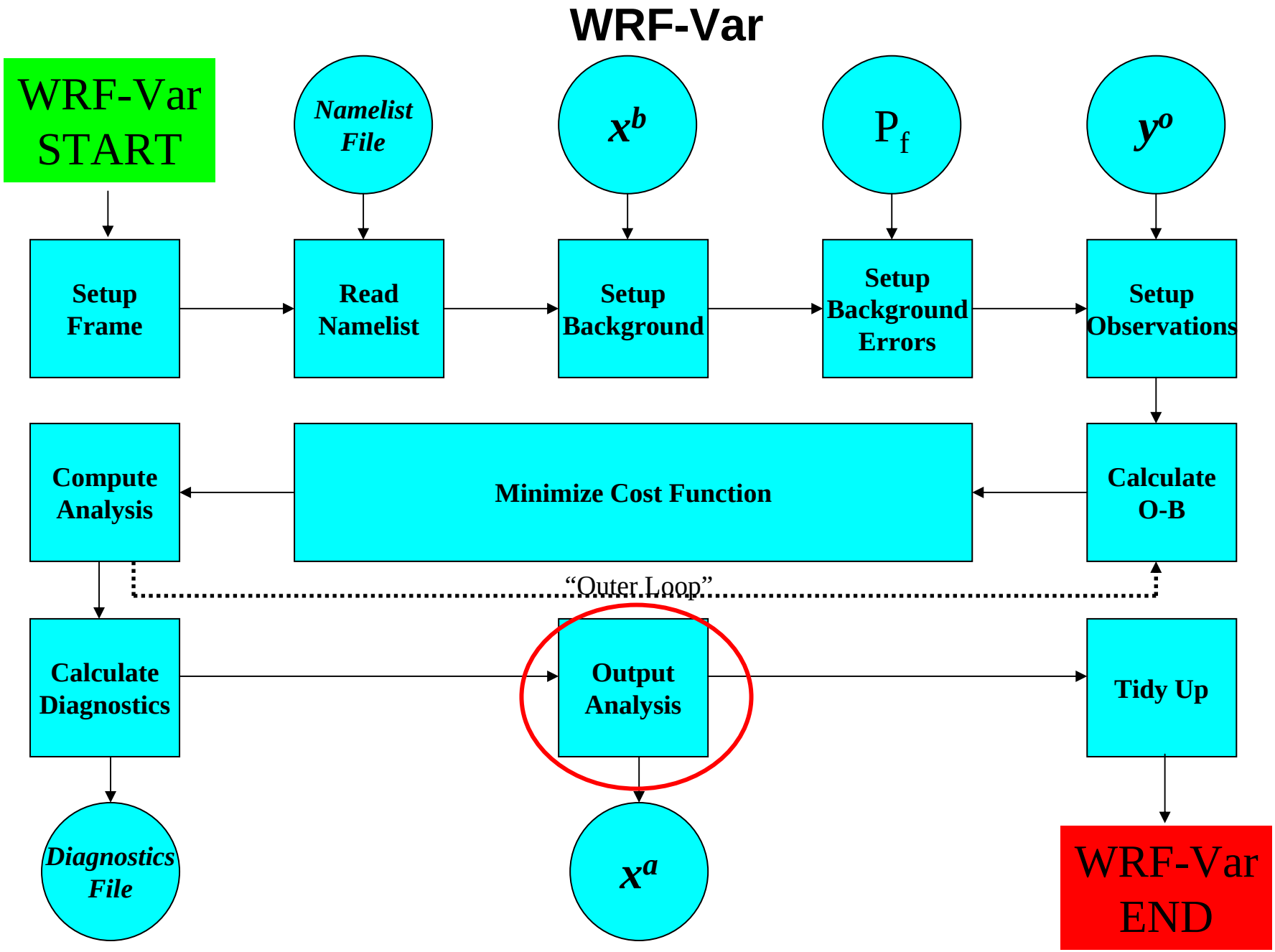
Compute Analysis

- **Once WRF-Var has found a converged control variable, convert to model space analysis increments.**
- **Calculate analysis = first-guess + analysis increment.**
- **Performs consistency checks e.g. remove negative humidities from analysis.**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Compute Diagnostics

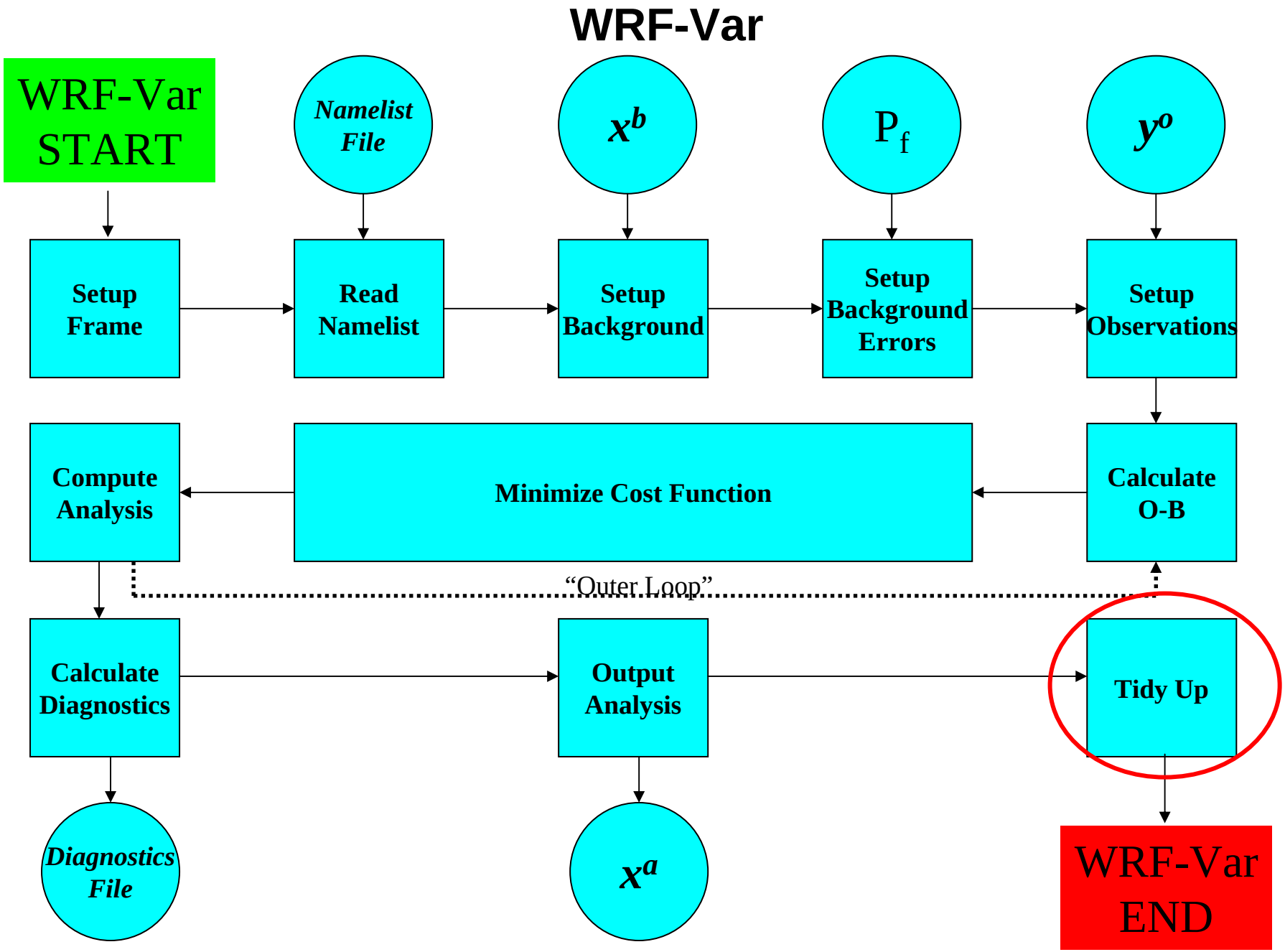
- **Compute O-B, O-A statistics for all observation types and variables.**
- **Compute A-B (analysis increment) statistics for all model variables and levels.**
- **Statistics include minimum, maximum (and their locations), mean and standard deviation.**
- **Also compute “specialist diagnostics” for error tuning (fort.45, fort.46, fort.47, fort.50 etc.).**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

Output Analysis

- **Outputs analysis in native model format. Choice made by namelist option fg_format – 1 = WRF, 2 = MM5, etc.**

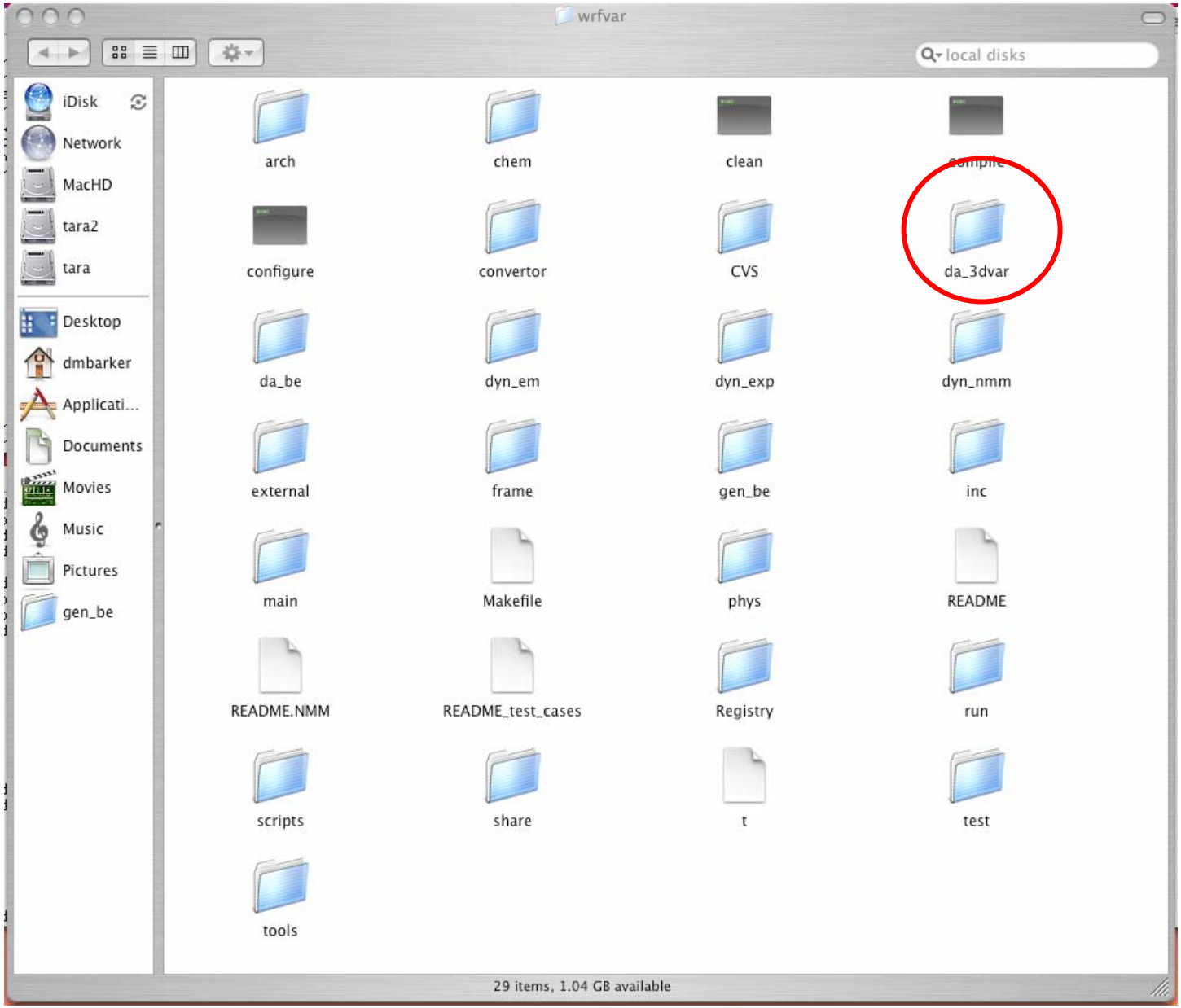
- **Also output analysis increments (for diagnostic purposes) in native model format. Switch off by setting DA_WRITE_INCREMENTS = .FALSE. in namelist.3dvar.**


WRF-Var
WRF-Var START
Namelist File
x^b
P_f
y^o
Setup Frame
Read Namelist
Setup Background
Setup Background Errors
Setup Observations
Compute Analysis
Minimize Cost Function
Calculate O-B
"Outer Loop"
Calculate Diagnostics
Output Analysis
Tidy Up
Diagnostics File
x^a
WRF-Var END

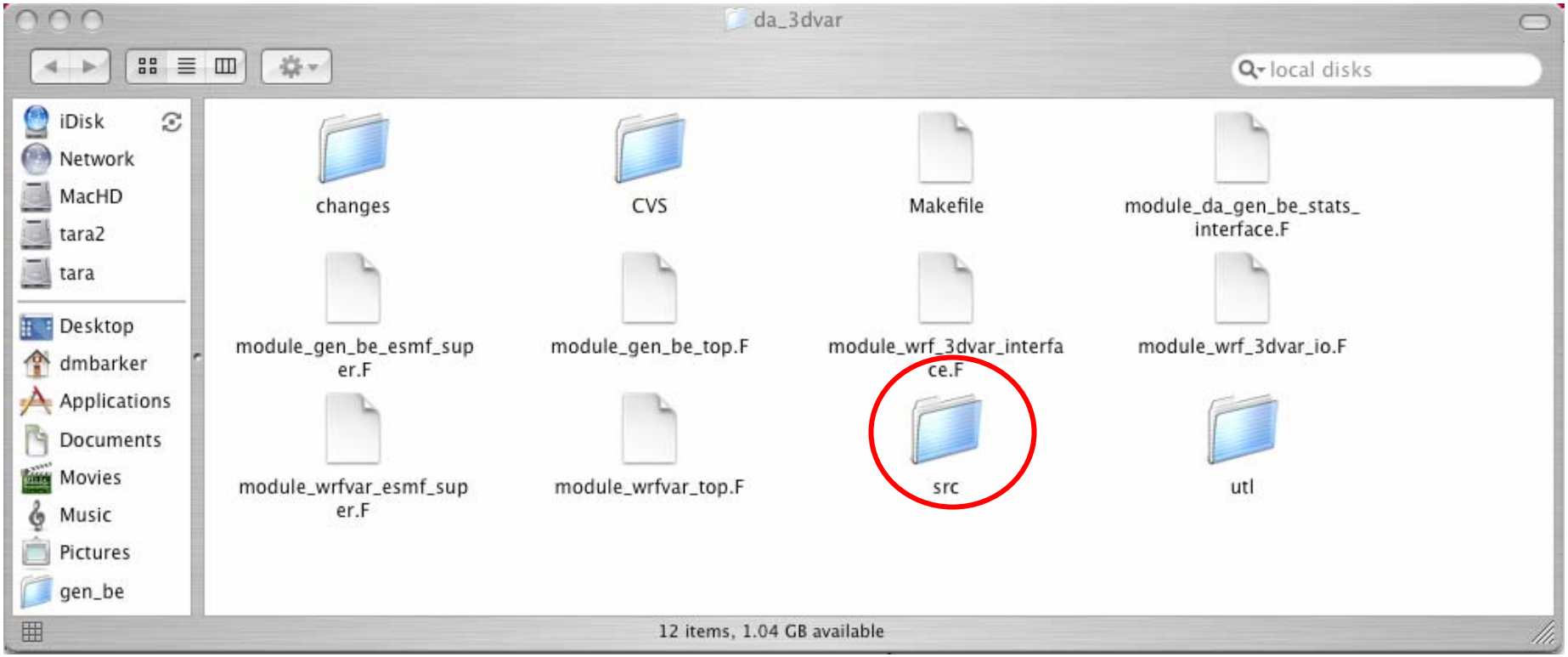
Tidy Up

- **Deallocate dynamically-allocated arrays, structures, etc.**
- **Timing information.**
- **Clean end to WRF-Var.**

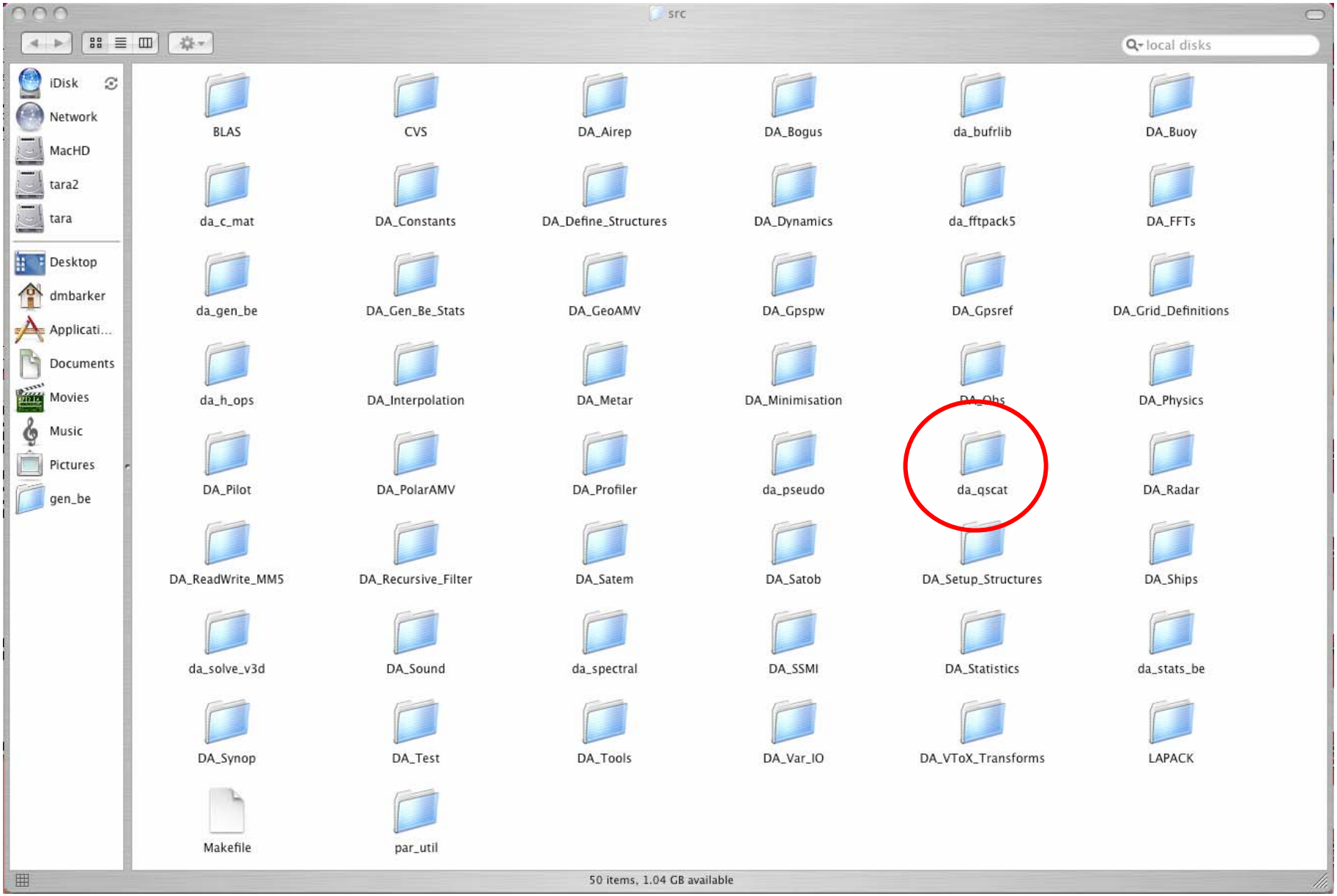
Source Code 1: *wrfvar*



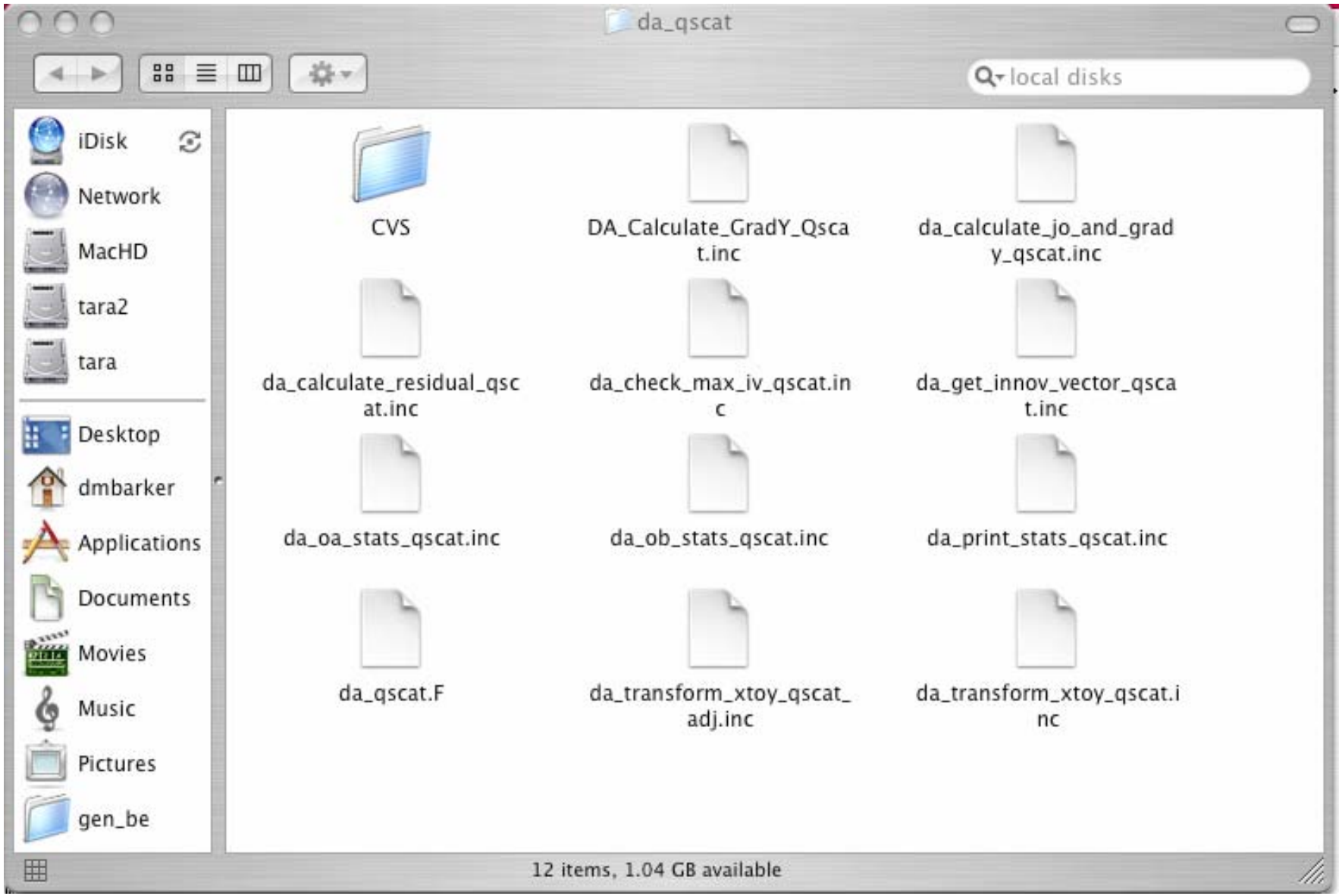
Source Code 2: *wrfvar/da_3dvar*



Source Code 3: *wrfvar/da_3dvar/src*



Source Code 4: *wrfvar/da_3dvar/src/da_qscat*



Source Code 5: *wrfvar/da_3dvar/src/da_qscat/da_qscat.F*

da_qscat.F

```
module da_qscat

   USE DA_Constants
   USE DA_Define_Structures
   USE DA_Interpolation
   USE DA_Statistics
   USE DA_Tools
   USE PAR_UTIL

! The "stats_qscat_type" is ONLY used locally in DA_Qscat:

   TYPE residual_qscat1_type
      REAL          :: u                        ! u-wind.
      REAL          :: v                        ! v-wind.
   END TYPE residual_qscat1_type

   TYPE maxmin_qscat_stats_type
        TYPE (maxmin_type)         :: u, v
   END TYPE maxmin_qscat_stats_type

   TYPE stats_qscat_type
        TYPE (maxmin_qscat_stats_type)  :: maximum, minimum
        TYPE (residual_qscat1_type)     :: average, rms_err
   END TYPE stats_qscat_type

CONTAINS

#include "da_calculate_jo_and_grady_qscat.inc"
#include "da_calculate_residual_qscat.inc"
#include "da_check_max_iv_qscat.inc"
#include "da_get_innov_vector_qscat.inc"
#include "da_oa_stats_qscat.inc"
#include "da_ob_stats_qscat.inc"
#include "da_print_stats_qscat.inc"
#include "da_transform_xtoy_qscat.inc"
#include "da_transform_xtoy_qscat_adj.inc"
#include "DA_Calculate_GradY_Qscat.inc"


end module da_qscat
```

Procedure for adding new Observations

- **Edit DA_Define_Structure.F to add new data type.**
- **Make new observation sub-directory under “src”.**
- **Develop desired programs like getting innovation vector, forward observation operator, tangent linear & its adjoint, gradient & cost function etc. in this new sub-directory.**
- **Input observation (update DA_Obs).**
- **Sometimes it might be needed to add certain grid arrays in Registry.**
- **Link into minimization package (DA_Minimisation)**

DA_Run_WRF-Var.csh (summary)

 # USER: Define non-default job via environment variables:

e.g.: setenv START_DATE 2004050200 overrides the default.

##
#USER: DO NOT MAKE CHANGES BELOW (if you do, you're on your own!)
##

 # [1.0] Specify default environment variables:

e.g. if (! $?START_DATE) setenv START_DATE 2004050100 # Analysis date.

 # [2.0] Perform sanity checks:

e.g. check input observation file exists

 # [3.0] Prepare for assimilation:

Create WRF-Var V2.1 namelist file (namelist.3dvar).

Create WRF V2.1 namelist file (namelist.input).

#[4.0] Run WRF-Var:

e.g. mpirun -v -np 16 -nolocal -machinefile hosts ./wrfvar.exe >&! /dev/null

Learning To Use WRF-Var

- **Consult the documentation at:**

  http://www.mmm.ucar.edu/wrf/WG4

- **Run through the Online WRF-Var Tutorial available at**

  http://www.mmm.ucar.edu/wrf/WG4/wrfvar.htm

- **If still confused, ask questions - wrfhelp@ucar.edu**