# Setting Up & Running the WRF Standard Initialization

Paula McCaslin

John Smart, Brent Shaw and Matt Pyle

NOAA / Earth Systems Research Laboratory

August 2006

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# Overview

The WRF modeling system includes 2 dynamic cores:

- Advanced Research WRF (ARW) developed by NCAR/MMM (formerly referred to as the Eulerian Mass core)
- Nonhydrostatic Mesoscale Model (NMM) developed by NOAA/NCEP

Each dynamic core currently has a separate SI package and graphical user interface (GUI)

- Options unique to each package are noted in this presentation

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# SI Capabilities

- Three steps in preparing data for WRF:

1. Define domain and nest domains
2. Create non-time-varying terrain and land state variables files for the domain grids (land use, soil type, etc)
3. Down-scale national and global models to this domain for data cases
    - Decode GRIB files to access meteorological data (temp, winds, soil layers, sea-surface temp)
    - Interpolate this data to your domain -- horizontally and vertically, including grid staggering

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# Source Code

- SI Source code
  - Available at http:/wrfsi.noaa.gov/release
  - ARW v2.1.2 - Feb 2006 (wrfsi_v2.1.2.tar.gz)
  - NMM v2.1.2 - Feb 2006 (wrfnmm_si_v2.1.2.tar.gz)
  - > 1 GB

# Source Code

- SI non-time-varying data files -- Geographical and surface characteristics data files
  - Available at http:/wrfsi.noaa.gov/release
  - 6 GB
  - Topography
  - Land use categories (wetland, water, forest)
  - Annual greenness fraction (min and max)
  - Soil temperature, adjusted mean annual
  - Soil type – top and bottom layer categories (silt, sand, clay, bedrock):
  - Albedo and Max Snow Albedo
  - Terrain slope index

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# System Requirements

- Unix or Linux operating system
  - ARW
    - Routinely built on IBM AIX, Intel-Linux, Alpha-Linux at FSL
    - Built on Alpha-True64, AMD Opteron64 and SGI-IRIX at NCAR
  - NMM
    - IBM AIX, Intel-Linux
- FORTRAN 90/95 Compiler
- C Compiler (gcc is preferred)
- make Utility
- NetCDF Libaries
- Perl (and Perl/Tk for GUI)
- NCAR Graphics & Command Language (NCL) – optional

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# Install Software

- Installation Overview

1. Choose directory location for wrfsi (and its subdirectories, if desired)
2. Check compiler options
3. Check for, or build, the NetCDF libraries
4. Run the install script
5. Check for success

# Install Software

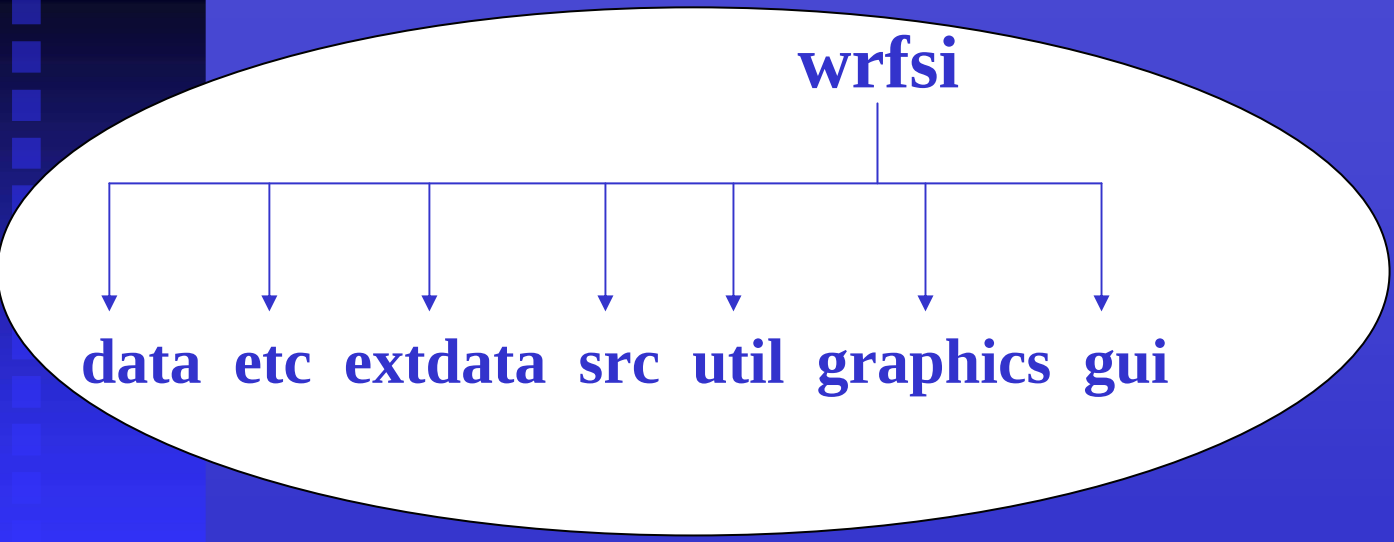
1. Choose directories

• *All* SI directory structures will be located within the wrfsi directory.

• But, any number of the SI directory structures can be located *anywhere* on your system if you simply define them for the installation process.

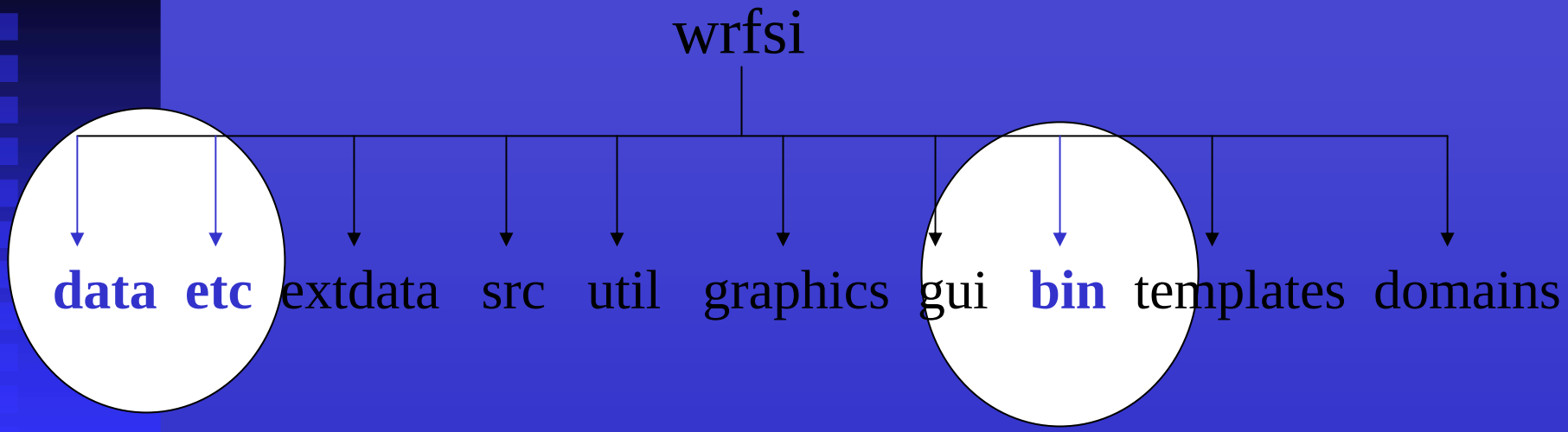
•For example:

- •Put source code and executables in different dir locations.
- •Write output dir other than to source code dir.
- •Write intermediate data files to a scratch partition dir.
- •Your computer resource has multiple users of the SI, but you only need one set of the 6 GB-sized geography data.



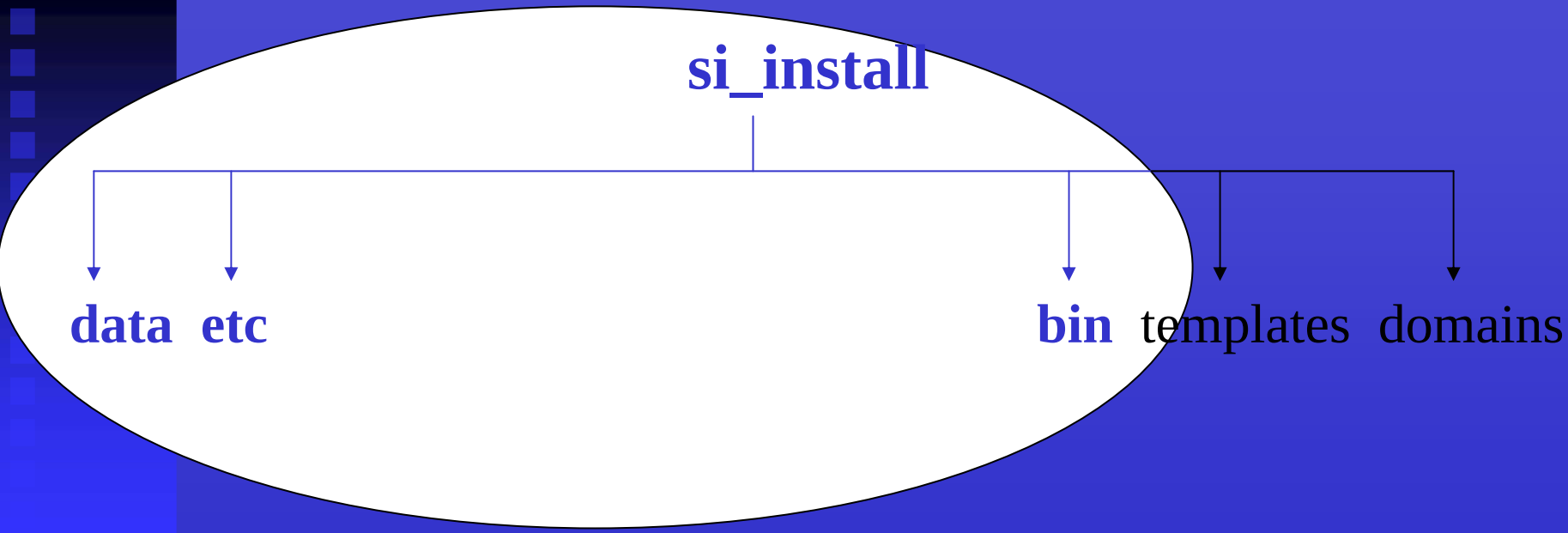

## ~ source code directory ~

SI structure when code is first extracted from the SI tar file, typically a subdirectory of WRF. Note that this dir is called wrfsi_nmm, for the NMM.



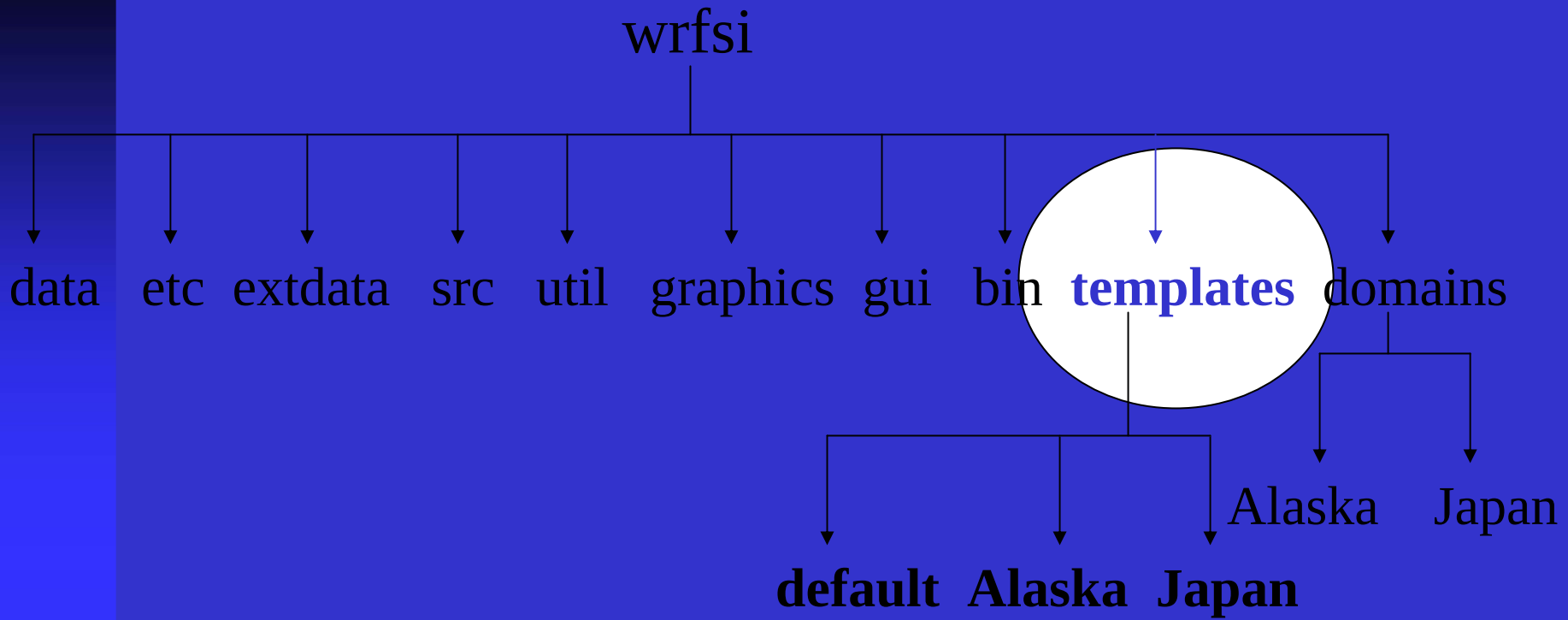

## ~ installation directories ~

SI structure where the compiled binary executables and scripts will be found after running install_wrfsi.pl to build and install.



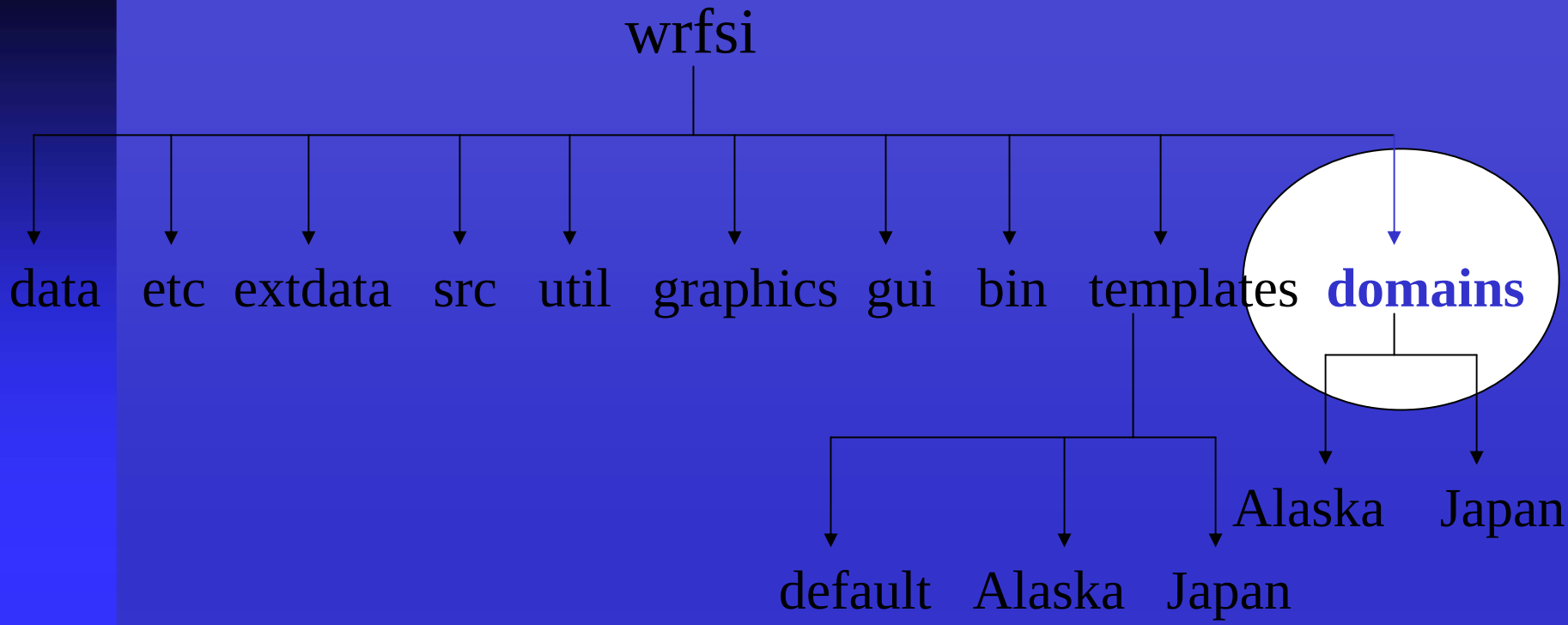

## ~ when installation dir differs from wrfsi dir ~

SI structure after installation process where install_wrfsi.pl command-line option --installroot="directory location" other than wrfsi. A user might want to create two builds from a common wrfsi source, for example named si_install_aix and si_install_linux64.



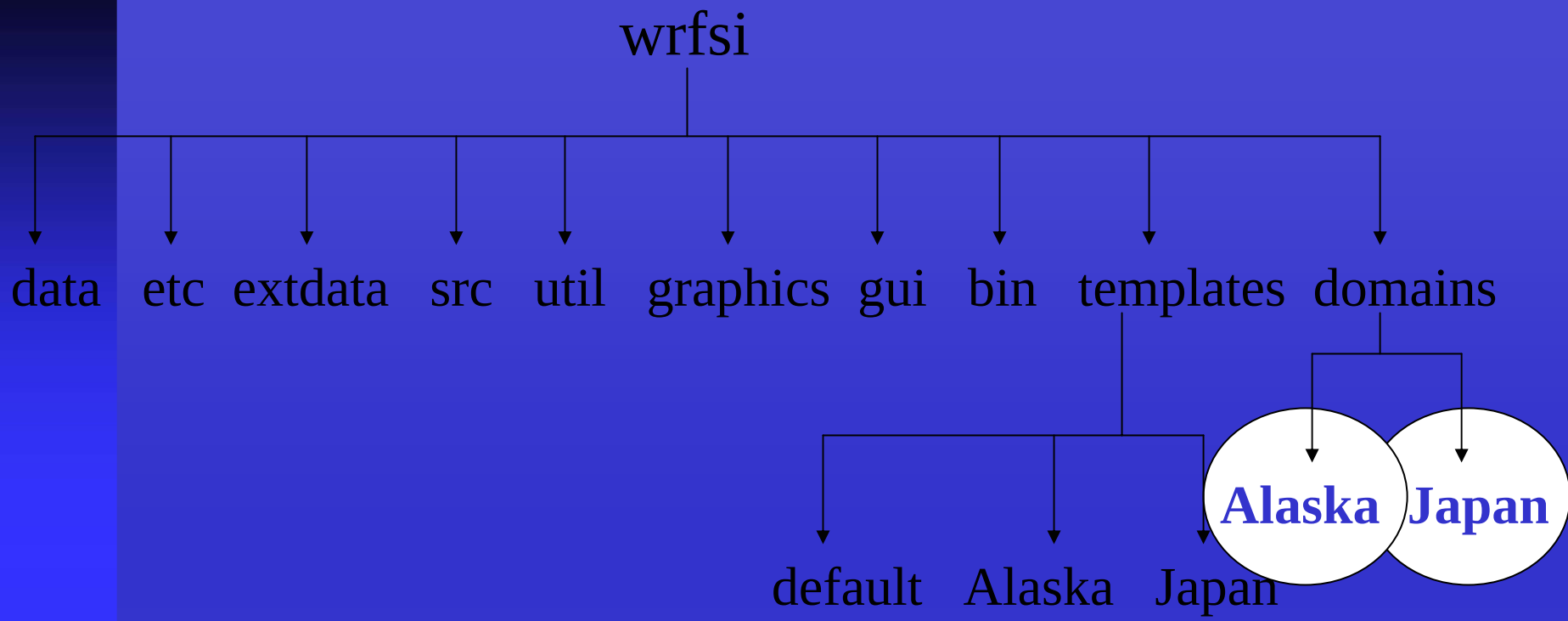

## ~ templates directory ~

SI structure with subdirectories that have a unique case name (like “Alaska”) that also includes a custom-edited wrfsi.nl containing projection information. Can use install_wrfsi.pl command-line option --templates=“directory” to define this location.




## ~ domains directory ~

Directory for a collection of domain subdirectories, called MOAD_DATAROOT. These will contain the domain's data files. Can use install_wrfsi.pl command-line option --dataroot="directory" to define this location.

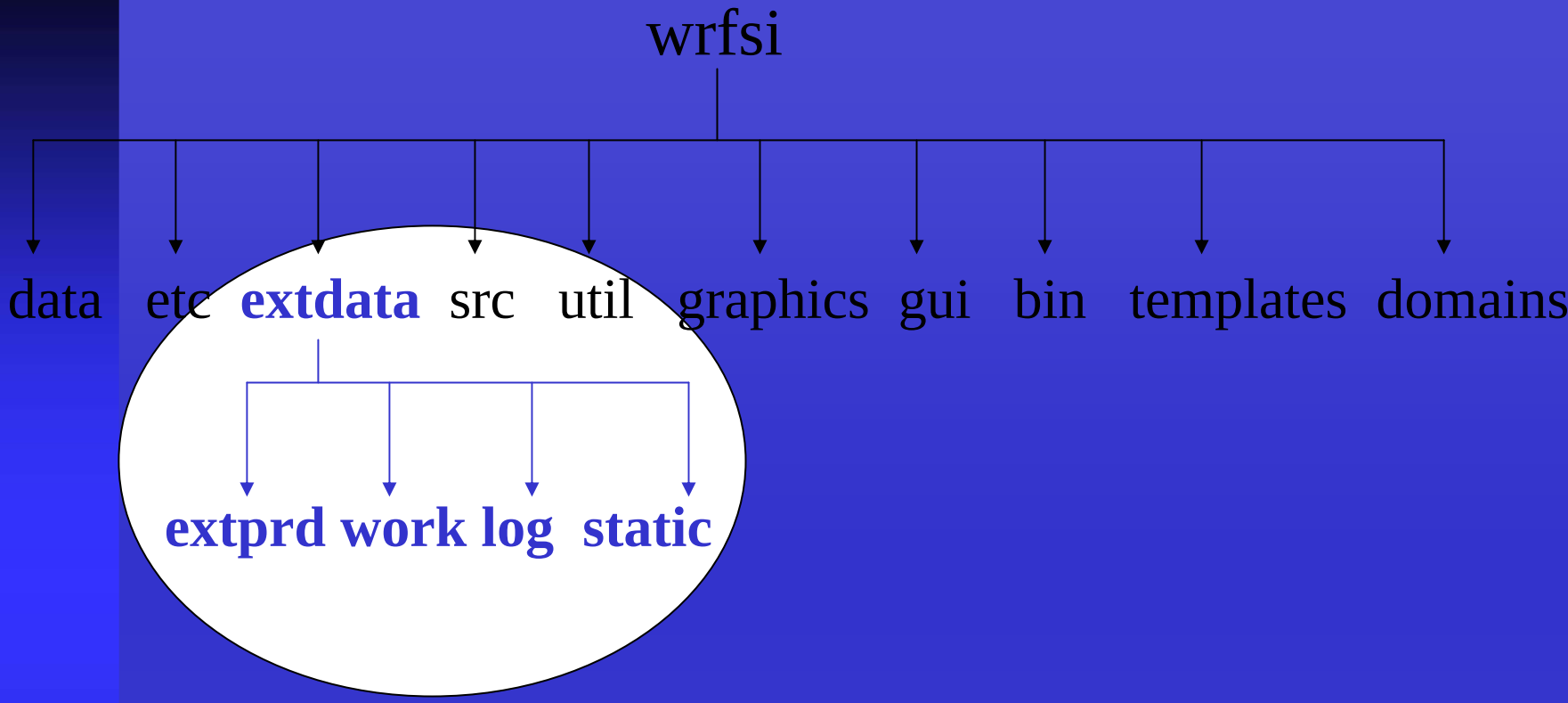



# MOAD_DATAROOT

Directory where a domain's definition files and the wrf input data will be written. You can have multiple MOAD_DATAROOTS but **only one is set at a time**.

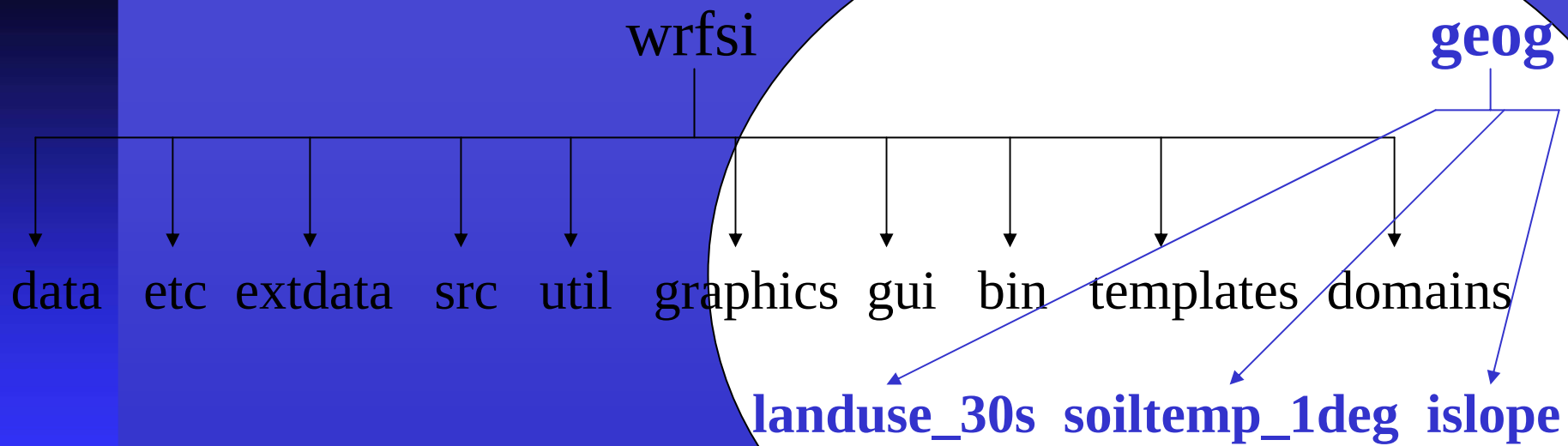
setenv MOAD_DATAROOT /wrf/wrfsi/domains/Alaska

setenv MOAD_DATAROOT /wrf/wrfsi/domains/Japan




## ~ external output data directory ~

SI structure with subdirectories created that will contain misc output from degribbing process using grib_prep. A location for the intermediate decoded and time-interpolated GRIB files. This directory area supports multiple MOAD_DATAROOTs. Allow at least 2 GB.




## ~ geography data directories ~

SI structure containing geography data subdirectories, landuse_30s, etc. Can use install_wrfsi.pl command-line option --geog_dataroot="directory" to define this location.

# Install Software

2. Check compiler options
   - Look for makefile_*.inc.in in dir **wrfsi**/src/include (where * is the machine on which you are building)
     - makefile_ibm.inc.in, makefile_pc.inc.in, etc
     - script evaluates `uname –m` to find a makefile
   - If a makefile does not exist for your machine (OS or compiler),
     - create a new makefile by copying an existing makefile then editing it to your meet your needs
     - Or, you can use install_wrfsi.pl command-line option --machine =“pc” for ia64linux, or x86_64linux
   - Check the makefile’s selected compiler, compiler options and flag settings for your machine and its OS

# Install Software

3. Check or build the NetCDF libraries and locate Perl

   - Locate path to NetCDF
   - Set environment var **NETCDF** = “Directory path to NetCDF”

   - Locate your systems’ Perl executable (e.g. /usr/bin/perl)
   - Can set environment var **PATH_TO_PERL** = “Directory path to Perl”, or use the Perl found on system

   - Substitution values *before* then *after* running install_wrfsi.pl

     wrfprep.pl.in: **#!@PERL@**
     wrfprep.pl.in: **"@NETCDF@/bin";**

     wrfprep.pl: **#!/bin/perl**
     wrfprep.pl: **"/usr/local/apps/netcdf/bin";**

# Install Software

4. Run the install script
   - cd "wrfsi"
   - Option A) Run install_wrfsi.pl. Let script automatically configure all the SI directory structures.
   - Option B) Run install_wrfsi.pl with command line options to configure some (or all) of the SI directory structures:

     perl install_wrfsi.pl --installroot=/home/WRFV2/si_aix
     --geog_dataroot=/wrf/geog
     --path_to_netcdf=/usr/local/netcdf
     --machine=ibm

     (machine corresponds to the makefile_{MACH}.inc.in you want to use)
   - Option C) Run install_wrfsi.pl after you have set some (or all) of the SI environment variable directory structures.

     setenv GEOG_DATAROOT /data/geog

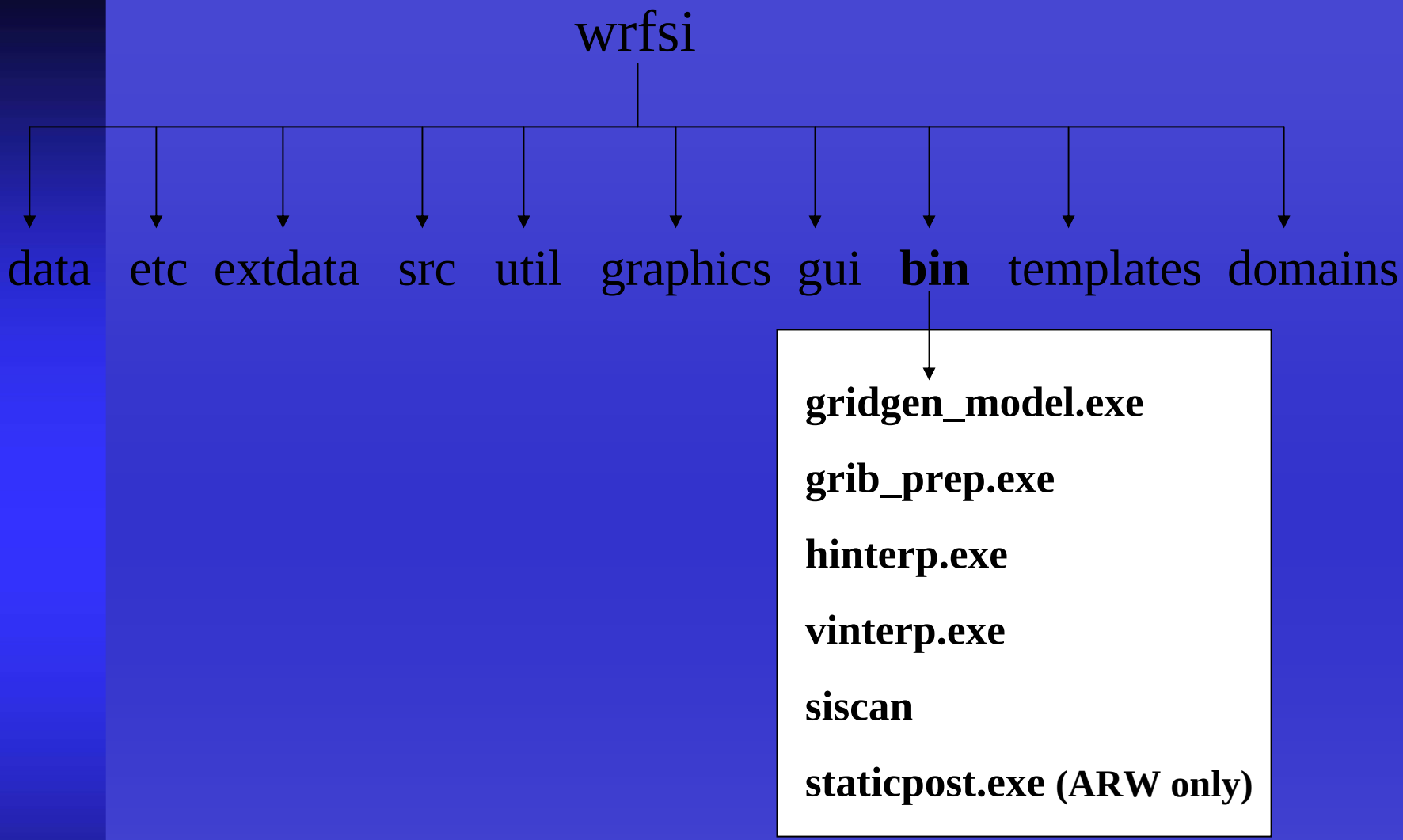
# Install Software

4. Run the install script (con't)

- Various ways to define your SI directory structure:

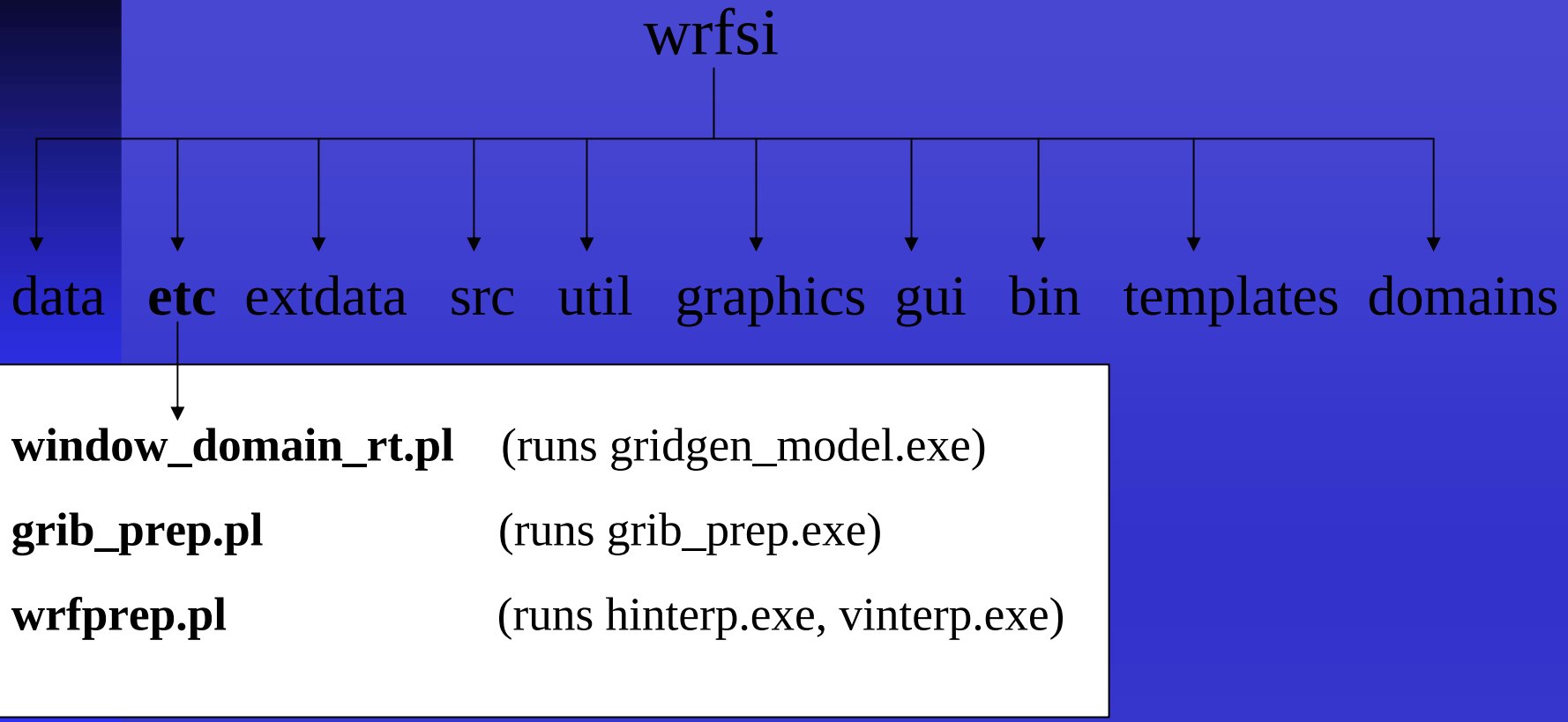
| Option A (default) | Option B (command line) | Option C (env var) |
|---|---|---|
| wrfsi/ | --source_root= | env SOURCE_ROOT |
| wrfsi/ | --installroot= | env INSTALLROOT |
| wrfsi/templates/ | --templates= | env TEMPLATES |
| wrfsi/domains/ | --dataroot= | env DATAROOT |
| wrfsi/extdata/ | --ext_dataroot= | env EXT_DATAROOT |
| wrfsi/extdata/GEOG/ | --geog_dataroot= | env GEOG_DATAROOT |

- **config_paths**
  - Lists all the SI directory structures
  - Can be used in a later session to redefine SI directory for use with the GUI




# Check for Fortran executables

to indicate a successful wrfsi installation after running install_wrfsi.pl.



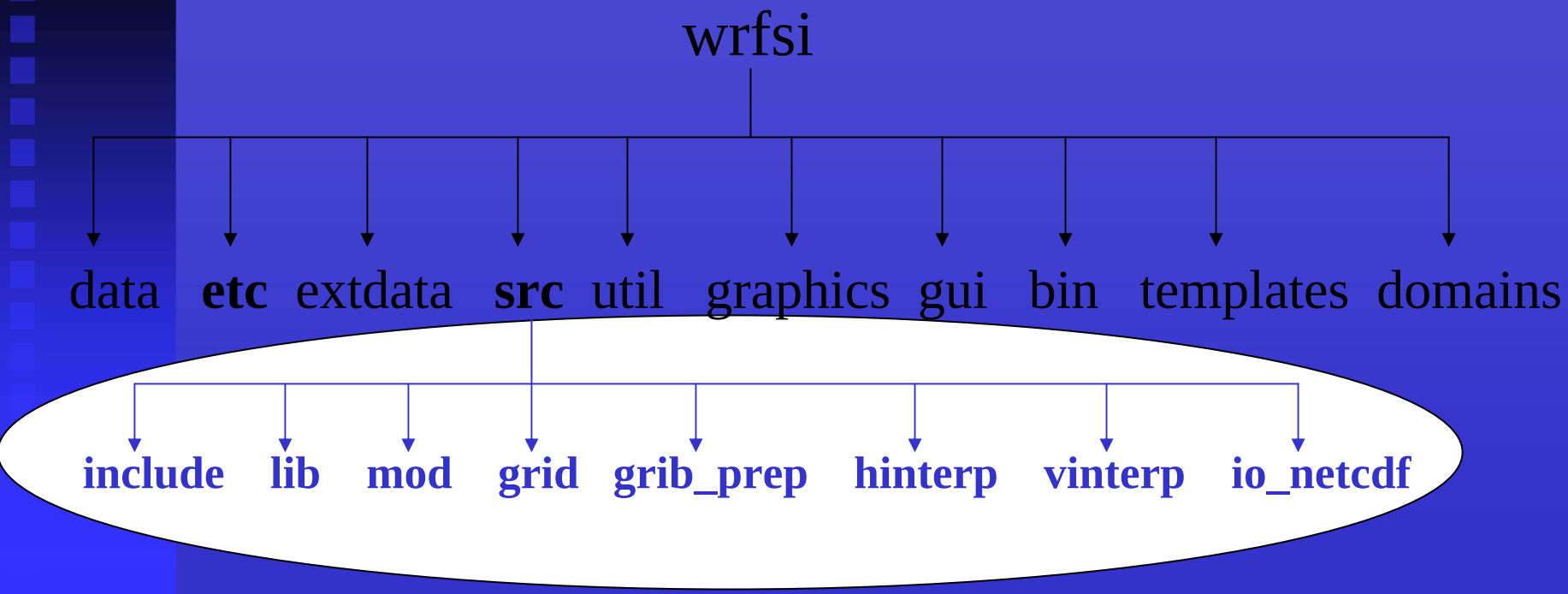

## Check for Perl scripts

to indicate a successful wrfsi installation after running install_wrfsi.pl.

# Install Software

5. Check for a success

   - During build status information is written to the screen and to "wrfsi"/make_install.log
   - If an executable is missing from bin/ there is a problem
   - If NetCDF is not built with the same complier used to build SI, you need to rebuild NetCDF. E.g. if using pgf90 for SI, then NetCDF must be built with the same version of pgf90.
   - If compile errors are found try to debug the software for your system, or ask for help from wrfhelp@ucar.edu. Also forward any problems and their solutions, if possible, to wrfhelp@ucar.edu.




# Look for src directory

It's used to build an executable in the event that an executable is not found in bin/.

(E.g. cd grid for gridgen_model.exe, run 'make' to manually build the desired executable. On success, run 'make install' to move the exe to bin/.)

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# Configure Domains

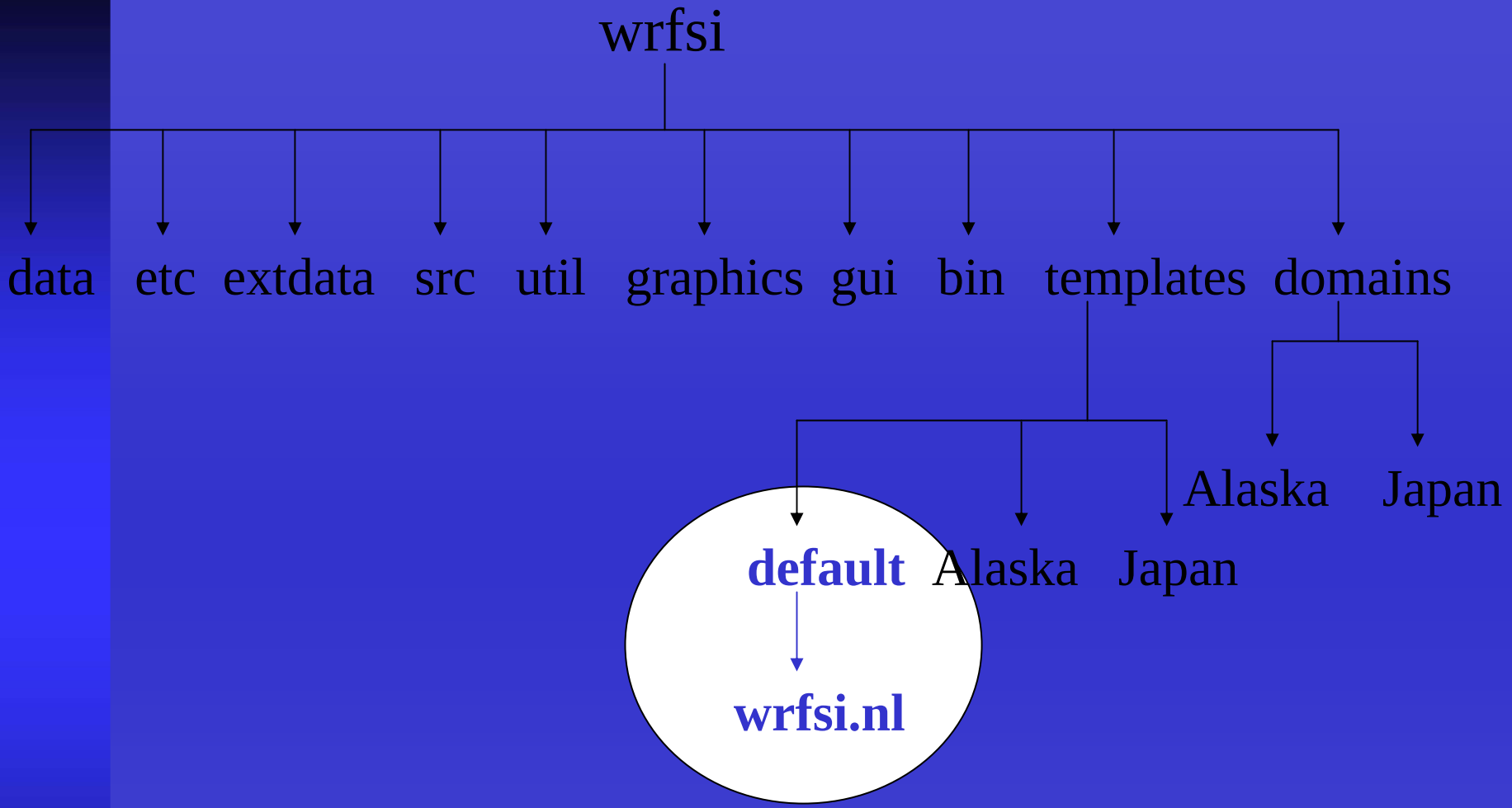
- Configuring a Domain Overview

  To Define and localize a domain

1. Create both a template and domain directory for your domain
2. Copy and edit wrfsi.nl
3. Run window_domain_rt.pl
4. Check for success

# Configure Domains

1. Create a template and domain directory for your domain
   - Each domain needs a “my-case” subdirectory under templates/ and domains/
     - mkdir TEMPLATES/my-case
     - mkdir DATAROOT/my-case




# Principal copy of wrfsi.nl

After installation processes, location of the principal copy of the SI Fortran namelist, wrfsi.nl, will be found in templates/default/wrfsi.nl (and wrfsi/data/static).

# Configure Domains

2. Copy and edit wrfsi.nl

   - cp templates/default/wrfsi.nl to templates/my-case
   - chmod –R u+w templates/my_case
   - cd templates/my-case

# Configure Domains

2. Copy and edit wrfsi.nl (con't)
   - Edit wrfsi.nl
     - &project_id section (simulation_name and user_desc)
     - &hgridspec section (sets up horizontal domain)
       - XDIM, YDIM = dimension in E-W and N-S direction, respectively
       - MAP_PROJ_NAME =
         - 'lambert', 'mercator', or 'polar' (ARW)
         - 'rotlat' (NMM)
       - MOAD_KNOWN_LAT/LON = center latitude (+ N), longitude (+ E)
       - MOAD_STAND_LATS = true latitude1 (lambert and polar), true latitude for lambert (set to +/- 90 for polar)
       - MOAD_STAND_LONS = orientation longitude
       - MOAD_DELTA_X and Y = grid spacing in the x and y directions
         - in meters (ARW)
         - in radian degrees (NMM)
   - We recommend removing all namelist sections and vars that do not differ from those found in templates/default/wrfsi.nl

# Configure Domains

3. Run window_domain_rt.pl

   - Script *runs* gridgen_model.exe
   - Check that SI environment vars are configured
   - Run INSTALLROOT/etc/**window_domain_rt.pl** with:
     - -t templates/my-case
     - -w wrfsi (ARW), wrfsi.rotlat (NMM)
     - Optional:
       - -s, -i, -d flags all override the environment vars, if necessary
       - -c removes the directories log, siprd, static, and cdl. (When omitting -c only directories static and cdl are rewritten.)
   - Example command line:

     perl window_domain_rt.pl -w wrfsi -t templates/"my-case"

# Configure Domains

4. Check for success

- 'window_domain_rt.pl' will result in either

  *success*:

  "window_domain_rt complete"

  *failure*:

  "Lines with error found in localize_domain.log"

  "→ localization incomplete ←"

  "window_domain_rt incomplete"

- Diagnostic print statements are written to log file MOAD_DATAROOT/log/**localize_domain.log**

# Configure Domains

4. Check for success (con't)

- Common Error conditions:
  - Geography path is not set properly
  - Not enough geog tiles to cover your domain (esp. true if you get only one or two of the quarter sphere geog tar files)
  - Values in wrfsi.nl may not be set properly
  - Note: on an error "window_domain_rt.pl" will create MOAD_DATAROOT/**static_err** for you to evaluate

- The existence of a static file indicates success:
  - Check for MOAD_DATAROOT/static/**static.wrfsi.d01** (ARW)
  - Check for MOAD_DATAROOT/static/**static.wrfsi.rotlat** (NMM)
  - Both are netCDF files that can be viewed with ncdump or ncBrowse utility

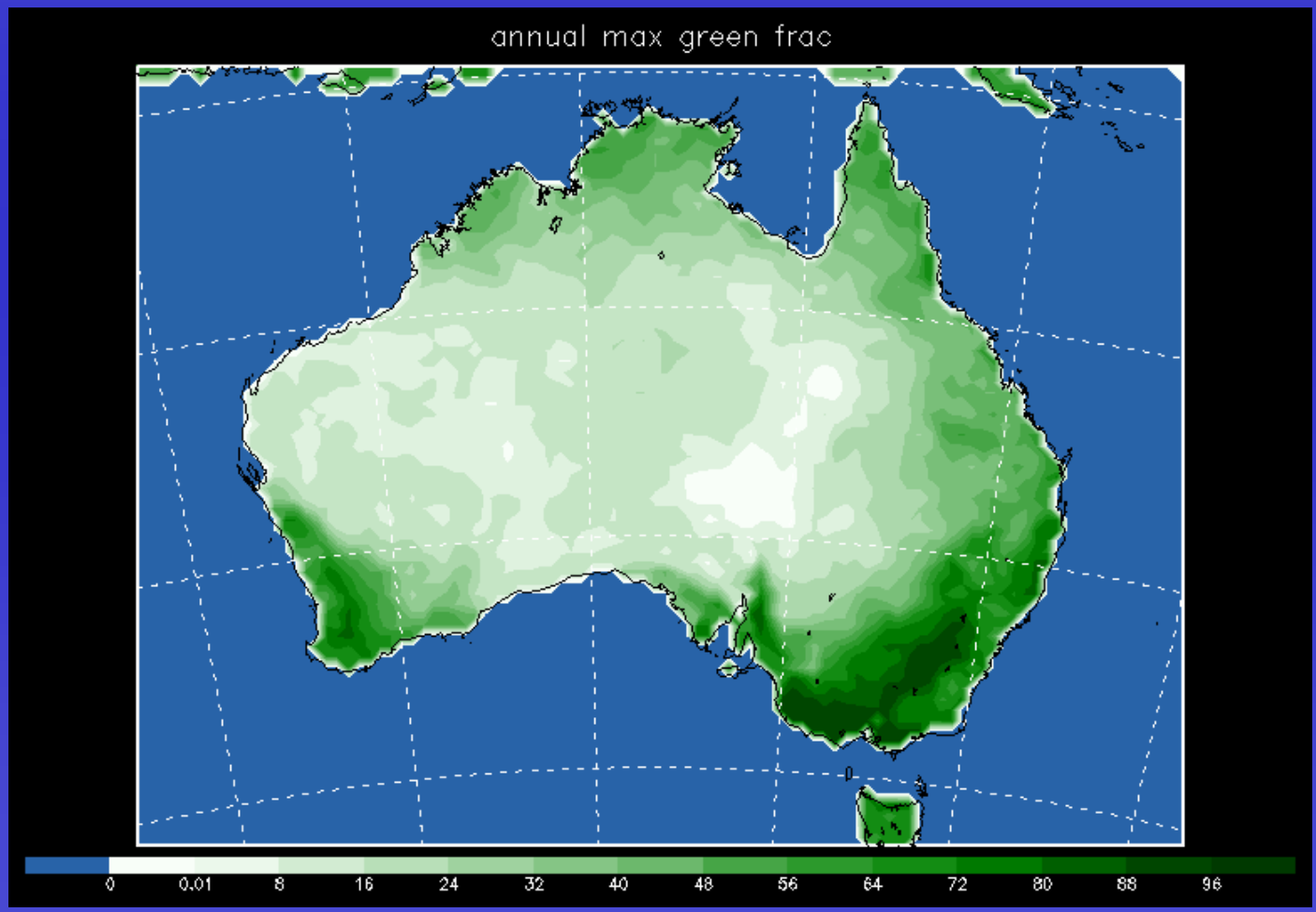
# Configure Domains

4. Check for success (con't)

- (Optional) generate NCL gmeta images
  - setenv NCARG_ROOT /usr/local/ncarg
  - setenv NCL_COMMAND $NCARG_ROOT/bin/ncl
    - Or, see wrfsi.noaa.gov/gui/faq_ncl
  - cd INSTALLROOT/graphics/ncl
  - generate_images.pl -domain=/wrfsi/domains/Alaska
  - idt /wrfsi/domains/Alaska/static/meta.d01.ncgm

# Configure Domains

4. Check for success (con't)



# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install the Software
- Configure Domains
- Prepare Data
- Interpolate Data
- Initialize the Model
- Summary

# Prepare Data

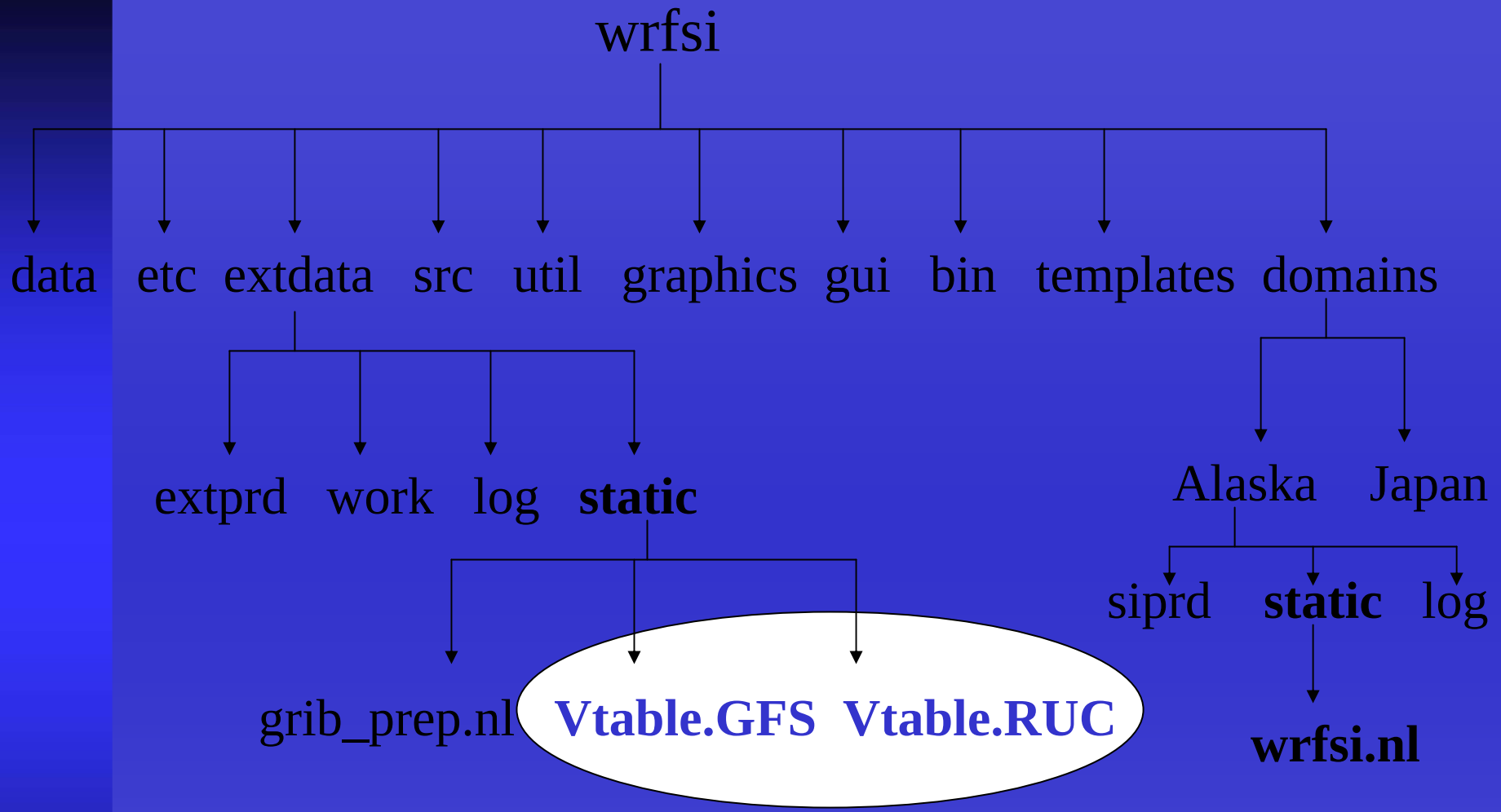
- Prepare Data Overview

  Decode GRIB files to acquire meteorological data

1. Locate GRIB data and Vtables
2. Configure grib_prep
3. Run grib_prep
4. Did code run properly?

# Prepare Data

1. Locate GRIB data and Vtables
   - Data
     - Acquire time-varying meteorological data for WRF's initial and lateral boundary conditions
     - Ensure GRIB data is available in the specified directory, for requested time intervals
   - Recognized GRIB file naming conventions are:
     - FSL: yyjjjhhmmffff
     - NCEP: ???.ThhZ.??????ff.????
     - NCEP: ???_yymmdd_hh_ff
     - And others
   - If your files do not conform to these naming conventions, use -f with the first several characters of the filename (-f avn2004)
   - Otherwise, the grib_prep script will assume every file is a possible match



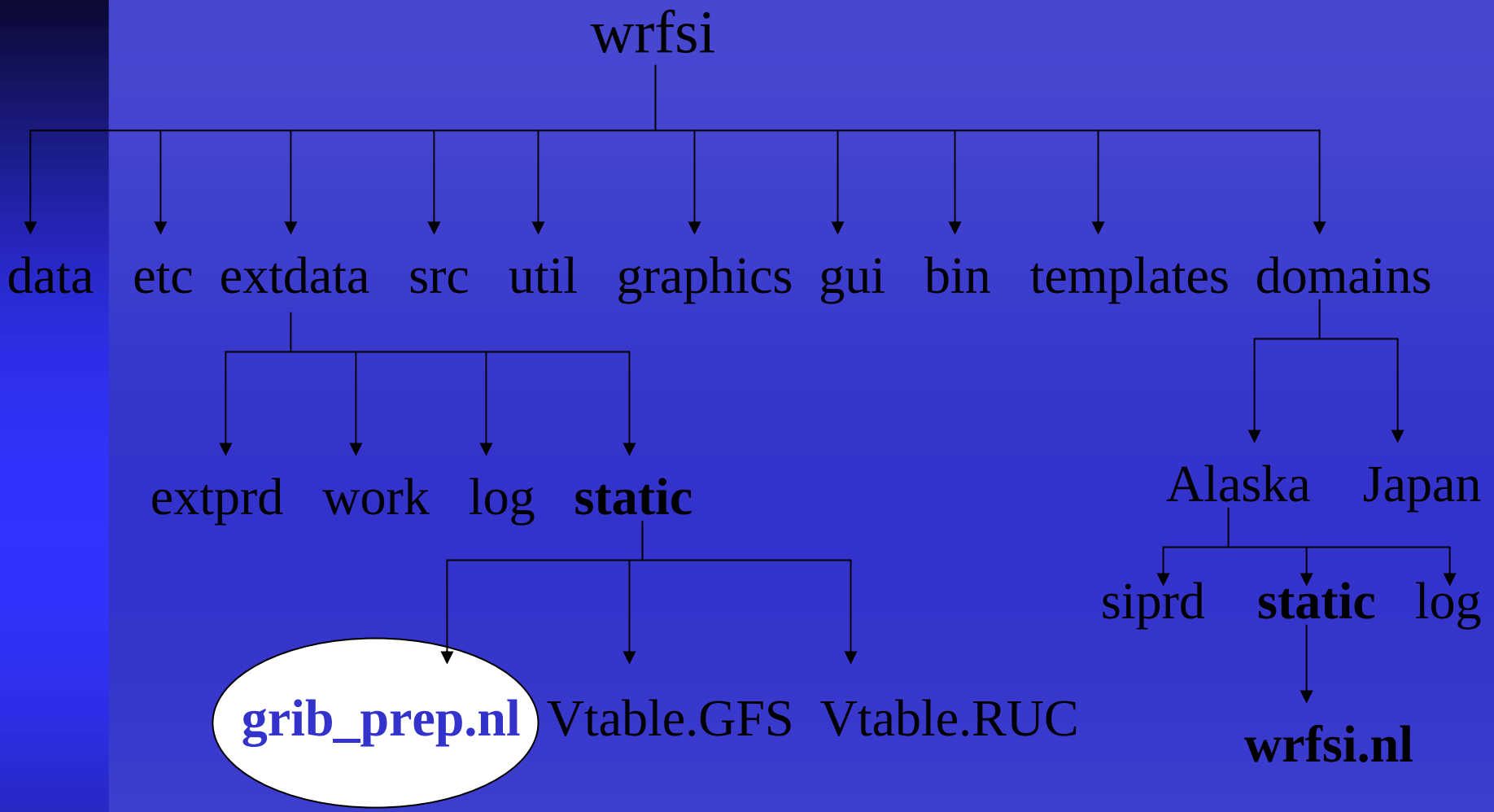

# GRIB Vtable

A variable table file containing information to extract model data from a GRIB file to initialize WRF.

# Prepare Data

1. Locate GRIB data and Vtables (con't)

- Decoding GRIB
  - A Vtable (variable table) file contains information to extract model data from a GRIB file to initialize WRF
    - Vtable.GFS, Vtable.ETA, etc. are provided
    - Contains a list of variables and their levels based on standard GRIB codes; e.g. 11 is temp, 1 is surface level, thus parameter is skin temperature, TSK
  - Confirm that there is a Vtable for your selected GRIB data file available in extdata/static/Vtable.{model} where model is, e.g. GFS or RUC40
  - Easy to create a Vtable."newGRIBsource" (depending on model) by copying an existing Vtable




# grib_prep namelist

Edit grib_prep.nl for **grib_prep.pl** to locate and decode GRIB files.

# Prepare Data

2. grib_prep configuration

- Edit EXT_DATAROOT/static/**grib_prep.nl**
  - &filetimespec
    - Set start and stop times
    - For real-time runs these values would be set automatically
  - &gpinput_defs
    - Set Vtable name, frequency of grid receipt time, typical time delay (in hours) after cycle time

SRCNAME = 'GFS', ‘ETA‘
SRCVTAB = 'GFS', ‘ETA’
SRCPATH ='/data/grib/gfs/0p5deg’, (1.0 or 0.5 deg res, FNL anal data)
'/data/grib/40km_eta‘
SRCCYCLE = 3, 3
SRCDELAY = 1, 0

# Prepare Data

3. Run grib_prep.pl

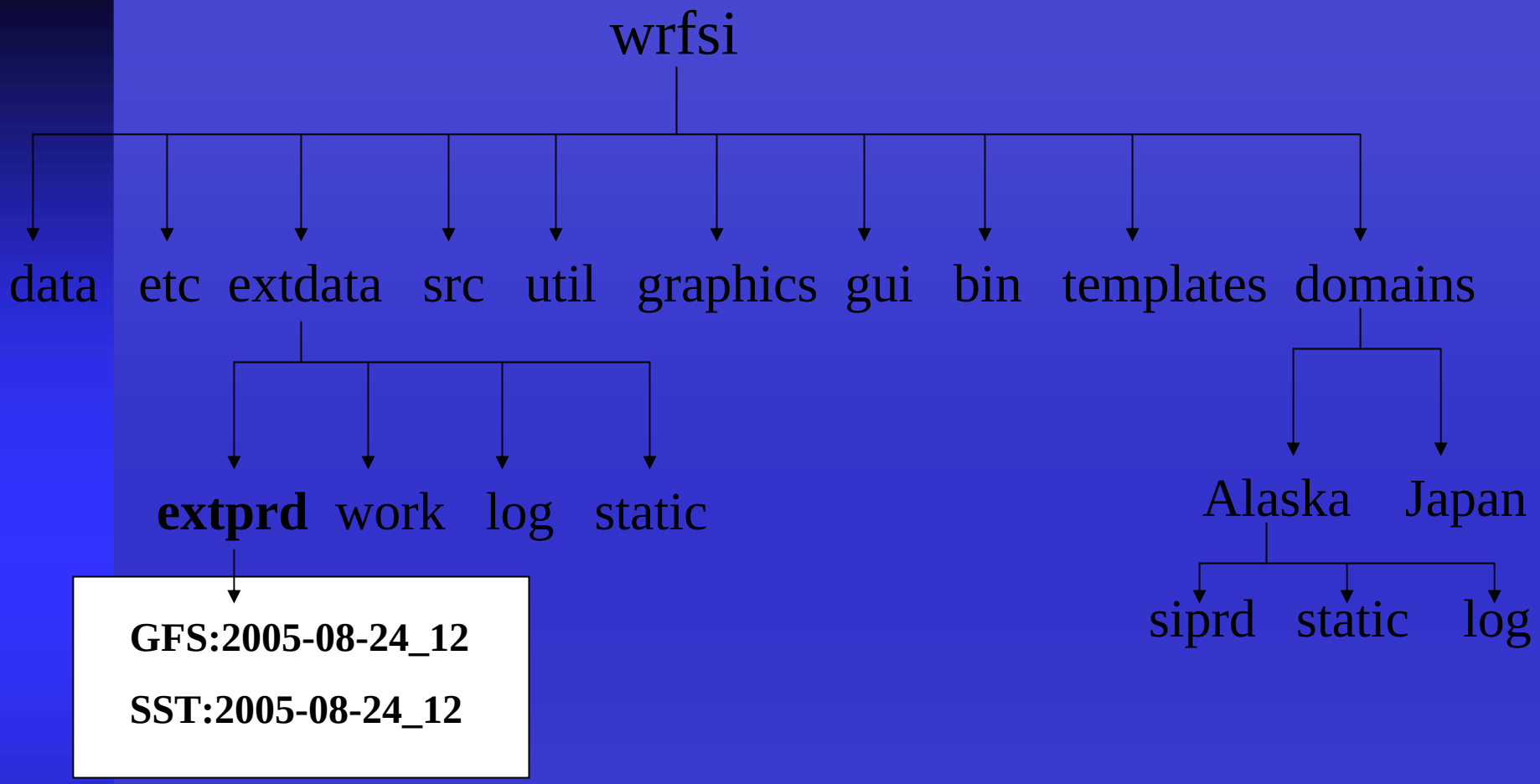
   - grib_prep.pl *manages* the time-series of data to process
   - grib_prep.pl *runs* the Fortran executable grib_prep.exe – which loads the Fortran namelist EXT_DATAROOT/static/grib_prep.nl setting all the necessary filenames and values
   - grib_prep.exe *decodes*, *extracts* and *time interpolates* GRIB model parameters

   - Output is written to EXT_DATAROOT/extprd
   - Log files are written to EXT_DATAROOT/log
   - While files are being created they are written to EXT_DATAROOT/work/GFS (or other source name)

   - Run

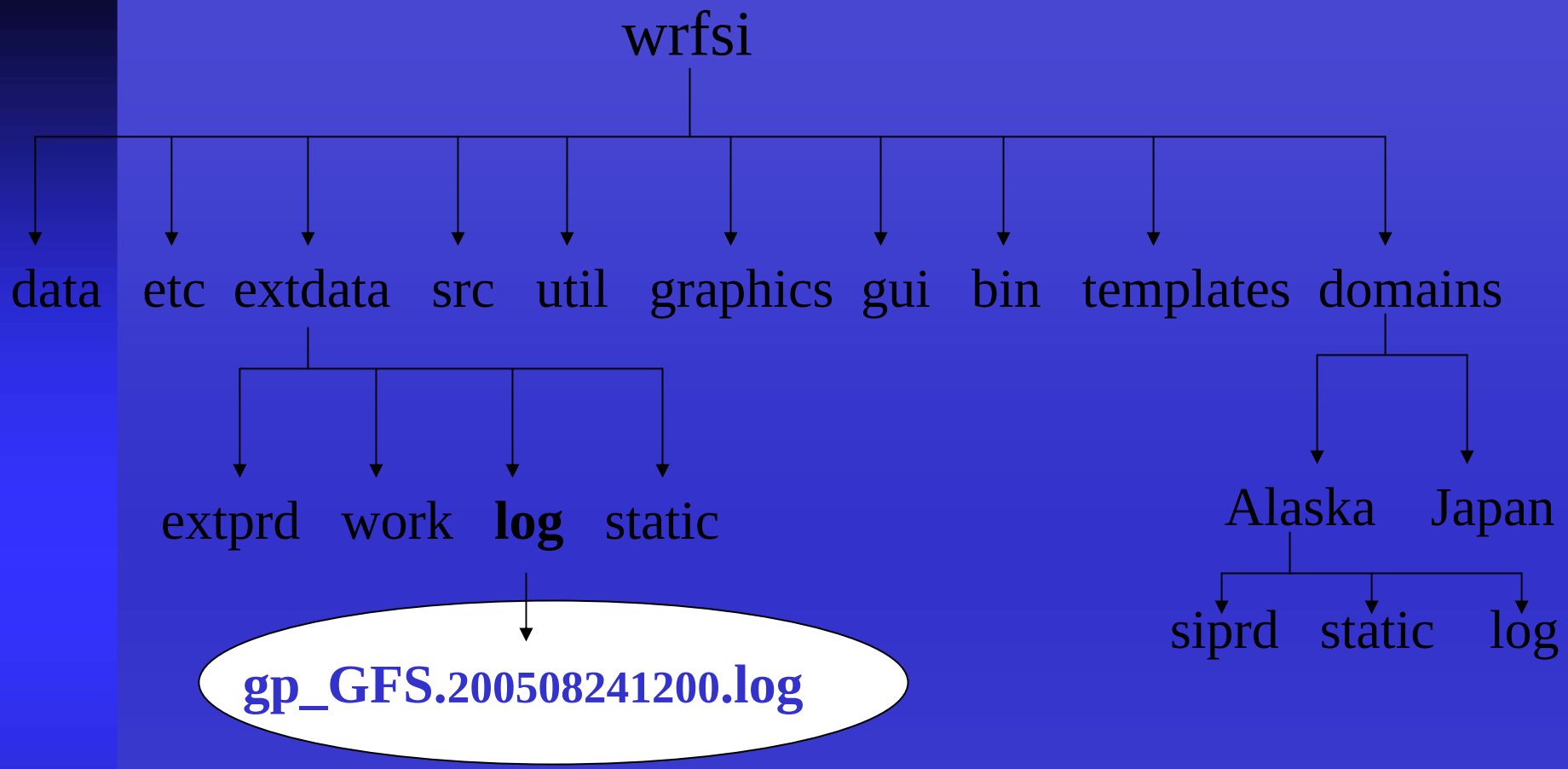
   INSTALLROOT/etc/grib_prep.pl -s 200508241200 -l 12 -t 6 GFS

   and INSTALLROOT/etc/grib_prep.pl -help (for command options)




# Look for intermediate data files

in extdata/extprd to indicate successfully running grib_prep.pl.




# Diagnostic log files

Evaluate grib_prep's gp_* log file to help determine what was needed for GRIB data.

# Overview

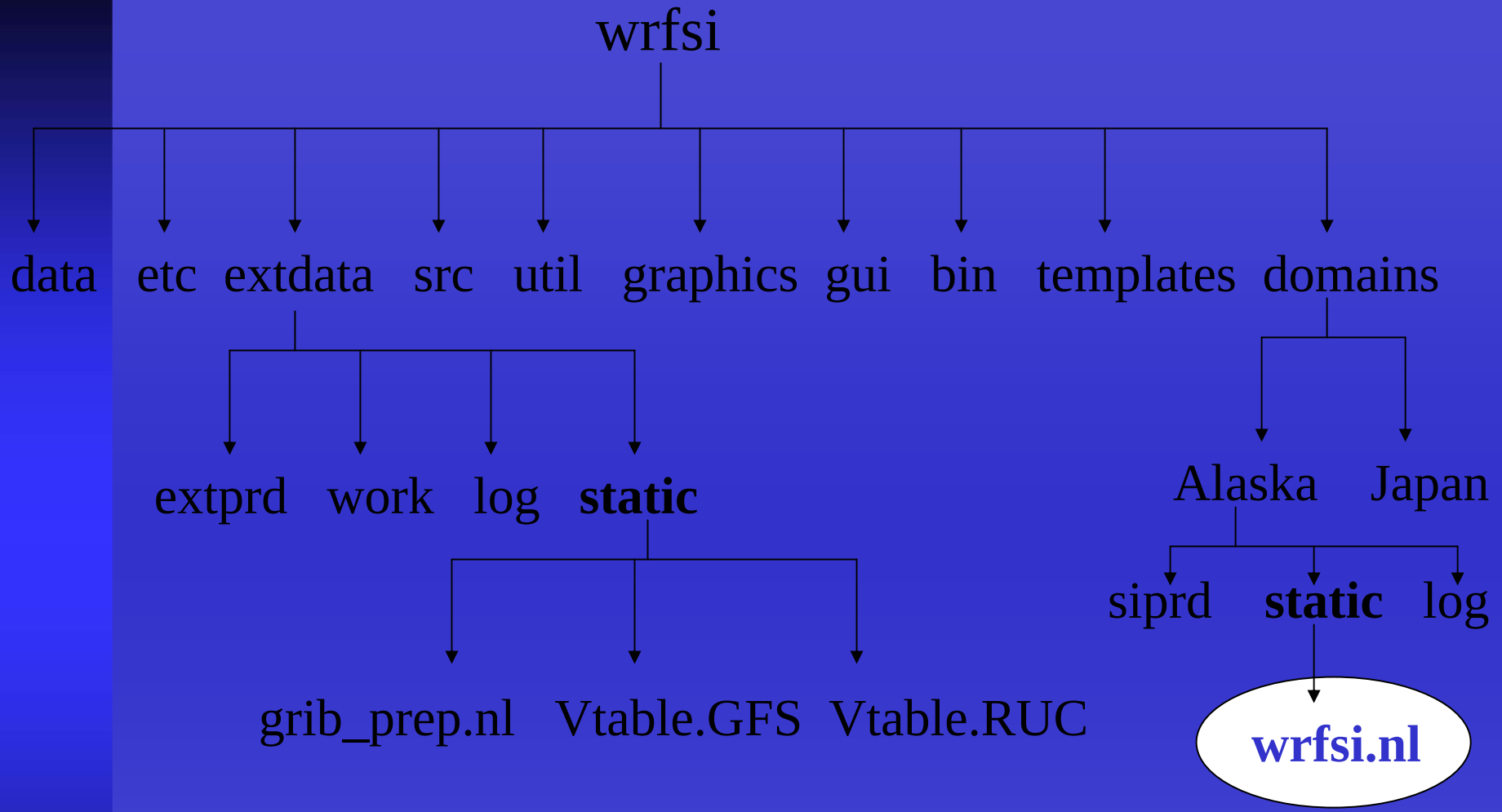
- SI Capabilities
- Source Code
- System Requirements
- Install the Software
- Configure Domains
- Prepare Data
- Interpolate Data
- Initialize the Model
- Summary

# Interpolate Data

- Interpolate Data Overview

  Interpolate data to your domain

1. Configure wrfprep
2. Confirm set up
3. Run wrfprep
4. Did code run properly?
5. Check the output




# Principal wrfprep namelist

Edit wrfsi.nl for **wrfprep.pl** to locate and process your extracted GRIB data.

# Interpolate Data

1. wrfprep configuration
   - Edit MOAD_DATAROOT/my-case/static/**wrfsi.nl**
   - &interp_control
     - PTOP_PA:  Top pressure level to consider from input data
     - HINTERP_METHOD and LSM_HINTERP_METHOD
       - 0 = nearest neighbor (not recommended for HINTERP_METHOD)
       - 1 = 4-point linear
       - 2 = 16-point quadratic
     - INIT_ROOT,
     - LBC_ROOT, and
     - LSM_ROOT: Model source prefixes (e.g. ETA:) of binary data files in EXT_DATAROOT/extprd to use for dynamic data

       (E.g. to run with ETA model and SST for LSM set
       - INIT_ROOT='ETA', LBC_ROOT='ETA' and LSM_ROOT='SST')

# Interpolate Data

1. wrfprep configuration (con't)
   - LEVELS: List of “full” vertical levels
     - bottom to top of atmosphere
     - range from 1.0 - 0.0
   - OUTPUT_COORD: vertical coordinate
     - ‘ETAP’ - mass version (ARW)
     - ‘NMMH’ - hybrid vertical coordinate (NMM)
   - ARW nests
     - NUM_ACTIVE_SUBNESTS=3 (default=0, range 0 to N nests)
     - ACTIVE_SUBNESTS=2,3,5 (list of subnests to process)
   - Its recommended to leave all other settings as they are

# Interpolate Data

2. Confirm set up

   - Ensure domain configuration step has been successfully completed
   - Ensure wrfsi.nl has the correct data path values
   - Are INSTALLROOT and MOAD_DATAROOT path values set

# Running

3. Run wrfprep.pl

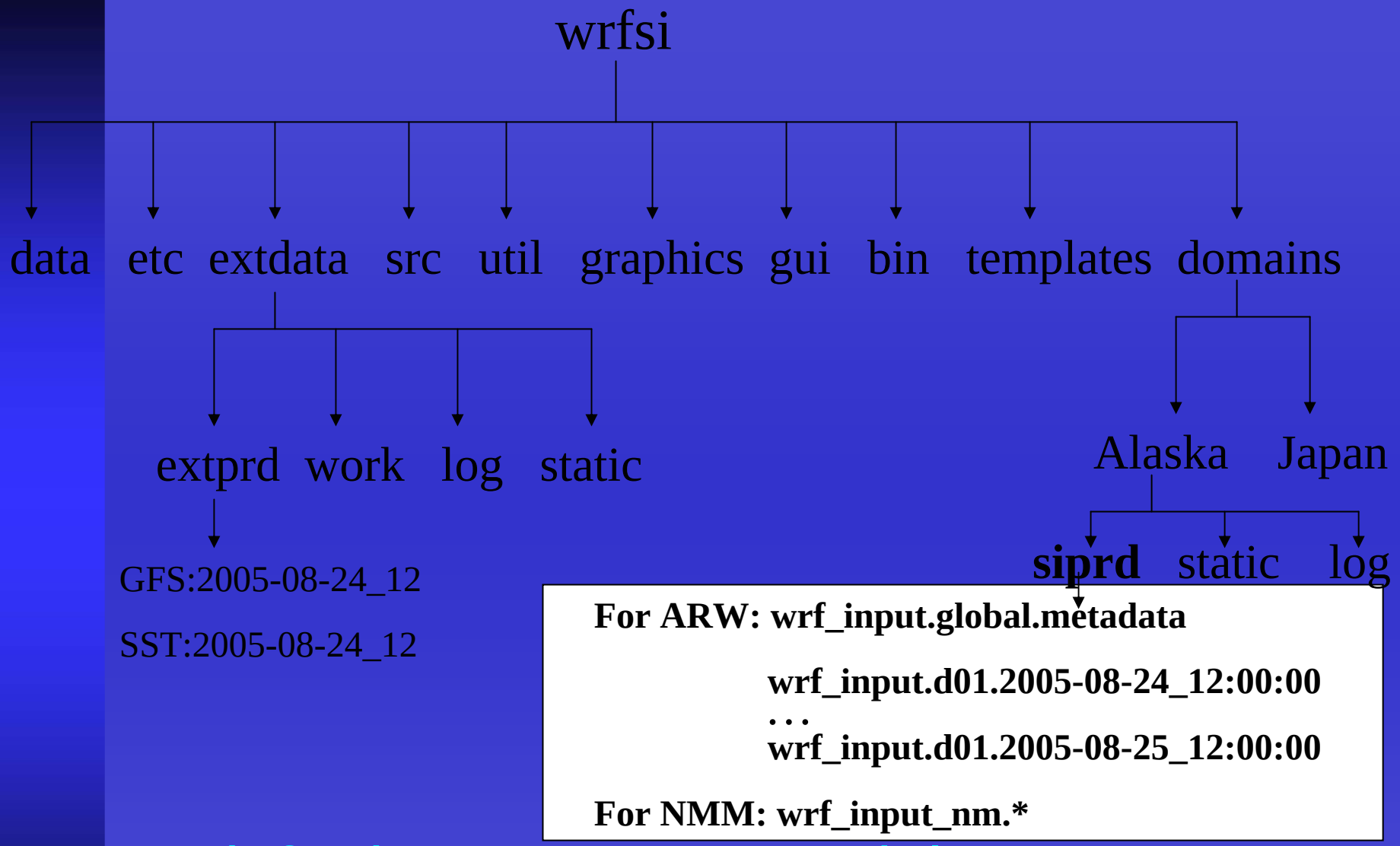
   - wrfprep.pl *uses* values set in the wrfsi.nl *&interp_controls* to locate and process data
   - wrfprep.pl *runs* the Fortran executables: hinterp.exe and vinterp.exe
   - hinterp.exe and vinterp.exe *interpolate* the extracted GRIB model parameters to fit the user defined domain

   - Output files are written to MOAD_DATAROOT/siprd
   - Log files are written to MOAD_DATAROOT/log for the hinterp and vinterp runs

   - Run

   INSTALLROOT/etc/wrfprep.pl -s 200508241200 -f 12

   and INSTALLROOT/etc/wrfprep.pl -help (for command options)



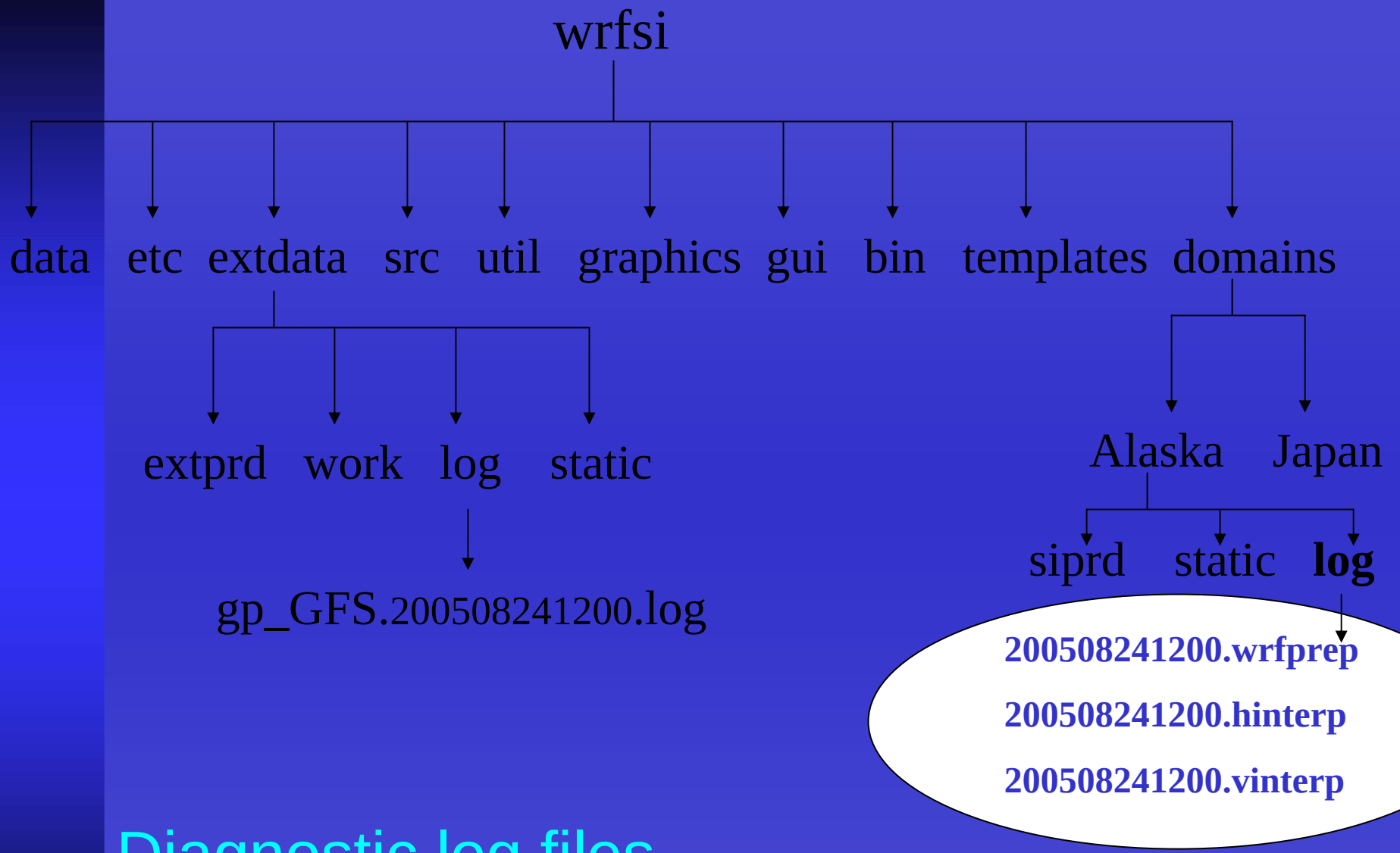

# Look for input to WRF model

in MOAD_DATAROOT/siprd to indicate successfully running hinterp and vinterp, via wrfprep.pl.

# Interpolate Data

4. Did code run properly?

   - Frequency of output files is set in &filetimespec in wrfsi.nl
   - If files are not present, check the diagnostic logs to determine what is needed. Then, consider rerunning wrfprep.pl with time resolution flags.
     - E.g. -o -2 (offset time of previous 2 hrs for a real-time process)




# Diagnostic log files

Evaluate wrfprep's log files to generate WRF input data.

# Interpolate Data

5. Checking the output

- Use bin/siscan to dump the file contents
  - siscan {file}
- "wrfsi"/util IDL routines read the hinterp and vinterp output (ARW)

- Other Notes
  - Executables can be run without the use of any scripts by simply setting MOAD_DATAROOT environment var and ensuring the namelists are correct
  - Consider using the WRFSI GUI to accomplish all of the SI processes
  - See wrfsi.noaa.gov for additional SI and GUI information

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare Data
- Interpolate Data
- Initialize the WRF Model
- Summary

# Initialize the WRF Model

- Edit the WRF *namelist.input* file to make it consistent with the WRFSI domain configuration
  - ztop, dx, dy, io_form, etc
  - Or, use the -r option with wrfprep.pl to automatically edit namelist.input and run real.exe, only if wrfsi is a subdirectory of WRF
- Run the WRF real routine using the WRFSI output files
  - Run real.exe using the wrf_input.* as input (ARW)
  - Run real_nmm.exe using wrf_input_nmm.* (NMM)
- You can then run the model!
- See *Initialization of Real* presentation for more details.

# Overview

- SI Capabilities
- Source Code
- System Requirements
- Install Software
- Configure Domains
- Prepare and Interpolate Data
- Initialize the WRF Model
- Summary

# Summary

- Setting up and running the WRFSI is done by:
- 1) Building the software
- 2) Configuring your domain
  - Domain configuration is easy with the use of templates and localization scripts
- 3) Running the grib_prep.pl and wrfprep.pl scripts
- The SI system is flexible enough to run each component separately

- NOAA development of SI is complete and operational.
- NCAR development will optimize and parallelize several components of the SI package with long-term plans to unify the SI package for the two dynamic cores.
- Bug fixes and minor enhancements will be done as resources permit
- wrfhelp@ucar.edu welcomes feedback, bug reports, etc.

# Setting up and Running the SI GUI

- Overview

1. Its recommend that your sys admin install Perl/Tk on your system
2. To build the SI GUI: when installing SI answer "**Yes**" at the command prompt "Do you want to install the SI GUI?"
   - If Perl/Tk is not found, then the install script tries to build this
3. To run the GUI use INSTALLROOT/**wrf_tools**
4. A GUI User's Guide is available, see wrfsi.noaa.gov/gui
5. A log file is written to /tmp/wrf_tools.log (by default)
6. For help with the GUI, see wrfsi.noaa.gov/gui/faq or contact wrfhelp@ucar.edu