

# WRF/WPS ユーザーガイド

## はじめに

これは WRF/WPS という気象解析予報モデルプログラム及びその初期化プログラムに関して、色々なトラブルに見舞われつつインストール&使用していった時の覚え書きです。

現在の利用環境は以下の通り。Fedora は Red Hat Enterprise Linux (RHEL) にフィードバックしているので、RHEL 互換の OS ではほぼ同様と考えて良いかと思えます。

- ・ Fedora 15 64bit
- ・ WRF 3.3
- ・ WPS 3.3

## 目次

### はじめに

- ・ インストールに関する基礎知識
- ・ 各種インストールの準備
- ・ MPICH2 設定

### WRF/WPS のインストール

- ・ WRF 3.3
- ・ WPS 3.3
- ・ ARWPost 3.1

### WRF/WPS の使用

- ・ WPS with GUI
- ・ WPS without GUI
- ・ WRF ラン
- ・ ARWPost ラン

### Appendix

- ・ WRF の修正パッチ
- ・ WPS の修正パッチ
- ・ ARWPost の修正パッチ
- ・ namelist.input の例

## はじめに

### インストールに関する基礎知識

Linux をインストール後、各種プログラムのインストールにはコマンドライン端末を使う。背景が灰色、インデント有り、コマンドラインのシンボル (\$、#) があれば、コマンドの実行を示す。端末の立ち上げは、Fedora では「アプリケーション」→「システムツール」→「端末」を実行する。

```
$ vi
```

上記のコマンドは、vi (汎用テキストエディタ) プログラムを実行する。他に、gedit、emacs などのテキストエディタがある。マニュアルでは vi コマンドの実行指示があるが、各自の使いやすいエディタへ適宜読み替えること。

```
# ← root 権限で実行するコマンド
$ ← ユーザ権限で実行するコマンド
```

上記の表記の違いは、プログラムを実行する権限の違いを示している。権限の一時的な切り替えには su コマンドを用いる。

```
$ su {User_Name}
$ exit
$ su -
#
```

コマンドオプションにハイフンを用いると、そのユーザでのログインを行う。用いない場合、権限のみの変更となる。またユーザ名を省略すると root でのログインを試す。ユーザの一時切り替えを終わるには、exit コマンドを用いる。

各プログラムをインストールするための圧縮ファイルは、/tmp にダウンロードされていると仮定する。他のディレクトリにダウンロードしてある場合は、mv コマンド等を用いて/tmp に移動しておくことを勧める。

### 各種インストール前準備

はじめに、WRF/WPS を使用するためのプログラムをインストールするために、開発ツールやライブラリが必要となる。以降のインストールにおける依存性を解決するソフトウェアをインストールしておく。

yum (Fedora におけるアップデートマネージャ) の groupinstall 機能で、開発ツール・ライブラリをまとめてインストールしておく。

```
# yum groupinstall "development tools"
# yum groupinstall "development libraries"
# yum groupinstall "x software development"
```

グループインストールされない開発ライブラリを yum の install 機能で個別に追加する。

```
# yum install jasper-devel
# yum install hdf5-devel hdf5-static
# yum install netcdf-devel netcdf-static
# yum install ncview
# yum install ncl-devel
# yum install mpich2-devel
```

## MPICH2 設定

WRF をマルチプロセッサで走らせるため、MPICH2 の設定を行う。ユーザ権限に移動し、環境設定を記述。

```
$ cd ~  
$ vi ~/.bashrc
```

```
// ----- Add to end of the file -----  
# MPICH2  
export MPICH2="/usr/lib64/mpich2"  
export PATH="${MPICH2}/bin:${PATH}"  
export LD_LIBRARY_PATH="${MPICH2}/lib:${LD_LIBRARY_PATH}"
```

編集が終わったら、環境設定を有効にする。

```
$ . ~/.bashrc
```

MPI デーモンの設定を書き出す。mpd.conf ファイルには MPD\_SECRETWORD を記述する。これは MPI 機能を利用する際のパスワードとなる。例では空白としている。mpd.hosts は利用するホスト名と、そのホストにおける CPU 数をコロンで区切り、記述する。例では localhost に CPU が 8 個あるとしている。その他には記述していないので、localhost のみで MPI ランが実行される。

```
$ echo "MPD_SECRETWORD=" > ~/.mpd.conf  
$ chmod 600 ~/.mpd.conf  
$ echo "localhost:8" > ~/.mpd.hosts
```

MPI 機能を利用する時は、MPI デーモンをバックグラウンドで起動し、ブート設定を行う。-n オプションでは使用するノード数を渡すことが出来る。CPU 数ではないので注意。例では localhost しかないので 1 とする。-f オプションでは設定したホストファイルを渡す。

```
$ mpd &  
$ mpdboot -n 1 -f ~/.mpd.hosts
```

## WRF/WPS のインストール

WRF: Weather Research and Forecasting Model (気象解析予報モデル)。気象シミュレーション解析プログラムの一種。

WPS: WRF Preprocessing System の略。WRF の前処理プログラム。

ARWpost: WRF で出力した NetCDF ファイルを可視化するために用いる。ここでは GrADS で読み込むことを想定している。NetCDF 出力のままでは充分であれば、インストールする必要はない。

### WRF 3.3

WRF Download ([http://www.mmm.ucar.edu/wrf/users/download/get\\_source.html](http://www.mmm.ucar.edu/wrf/users/download/get_source.html)) からダウンロード可能だが、ダウンロードにはユーザ登録が必要。

WRF はユーザ権限でインストールする。インストール設定のために、環境設定を記述しておく。

```
$ cd ~
$ vi ~/.bashrc
```

```
// ----- Add to end of the file -----
# Jasper
export JASPERLIB="/usr/lib64"
export JASPERINC="/usr/include/jasper"

# WRF
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
```

編集が終わったら、環境設定を有効にする。

```
$ . ~/.bashrc
```

mkdir で WRF 用のディレクトリを作成する。圧縮ファイルを解凍。

```
$ mkdir ~/Model
$ cd ~/Model
$ tar zxvf /tmp/WRFV3.3.TAR.gz
```

修正パッチを適用する。修正点は Appendix を参照。

```
$ patch -p0 < /tmp/WRFV3.patch
```

WRFV3 内に移動。NetCDF のシンボリックリンクを作成する。

```
$ cd WRFV3
$ mkdir netcdf_links
$ ln -s /usr/include/netcdf_links/include
$ ln -s /usr/lib64/netcdf_links/lib
```

configure を実行する。netcdf\_links 利用に y と入力。設定は 18 の gfortran with gcc (dm+sm) を採用。ネスティング設定は 1 の basic を採用。configure が終了したらログをとりつつ、コンパイル。

```
$ ./configure
$ ./compile em_real > log 2>&1
```

main ディレクトリに ndown.exe、real.exe、wrf.exe が作成されていればコンパイル成功。

### WPS 3.3

WRF Download のページから WPS をダウンロードする。また地理データとして geog.tar.gz もダウンロードすること。

```
$ cd ~/Model
$ tar zxvf /tmp/WPSV3.2.1.TAR.gz
$ tar zxvf /tmp/geog.tar.gz
$ mv geog GEOG
```

WRF 同様、修正パッチを適用する。修正点は Appendix を参照。

```
$ patch -p0 < /tmp/WPS.patch
```

WPS 内へ移動。NetCDF のシンボリックリンクを作成する。

```
$ cd WPS
$ mkdir netcdf_links
$ ln -s /usr/include netcdf_links/include
$ ln -s /usr/lib64 netcdf_links/lib
```

configure で設定。netcdf\_links 利用に y と入力。設定モードは 8 の gfortran compiler, DM PARALLEL を選択する。設定後、コンパイルを実行。

```
$ ./configure
$ ./compile > log 2>&1
```

geogrid.exe、ungrib.exe、metgrid.exe が出力されていればコンパイル完了。

### ARWpost 3.1

WRF Download から ARWpost をダウンロードする。解凍後、ディレクトリへ移動。

```
$ cd ~/Model
$ tar zxvf /tmp/ARWpost.tar.gz
```

修正パッチをあてる。修正点は Appendix を参照。

```
$ patch -p0 < /tmp/ARWpost.patch
```

ARWpost 内へ移動後、NetCDF のシンボリックリンクを作成する。

```
$ cd ARWpost
$ mkdir netcdf_links
$ ln -s /usr/include netcdf_links/include
$ ln -s /usr/lib64 netcdf_links/lib
```

configure で設定。netcdf\_links 利用に y と入力。設定モードは 3 の gfortran compiler を選択。設定後、コンパイルを実行。

```
$ ./configure
$ ./compile > log 2>&1
```

ARWpost.exe が出力されていればコンパイル完了。

## WRF/WPS の使用

WPS を使うには FNL データを必要とする。WRF Download にある「WRF Preprocessing System test data」をダウンロードし、適当なディレクトリを作成して解凍。

```
$ cd ~/Model
$ mkdir -p FNL/sample
$ tar zxvf /tmp/avn_data.tar.gz -C FNL/sample
```

ここではテスト用として配布されている FNL データを用いるが、UCAR のサイトで配布されている NCEP FNL Operational Model Global Tropospheric Analyses から実用的な FNL データをダウンロードできる。ダウンロードにはユーザ登録が必要である。

WPS は Java を用いた GUI による実行、端末画面のみでの実行の 2 種類がある。用途と環境に合わせ、適宜選択すること。

### WPS with GUI

WPS の設定を DomainWizard という WPS 用 GUI を用いて設定を行う。DomainWizard は Java を利用しているので、当然 Java がインストールされていなければならない。Fedora ではグループインストールにおいて Java をインストールすることが出来る。初期設定のインストールオプションでは Java もインストールされるが、もし入っていない場合は次のコマンドで Java を導入する。

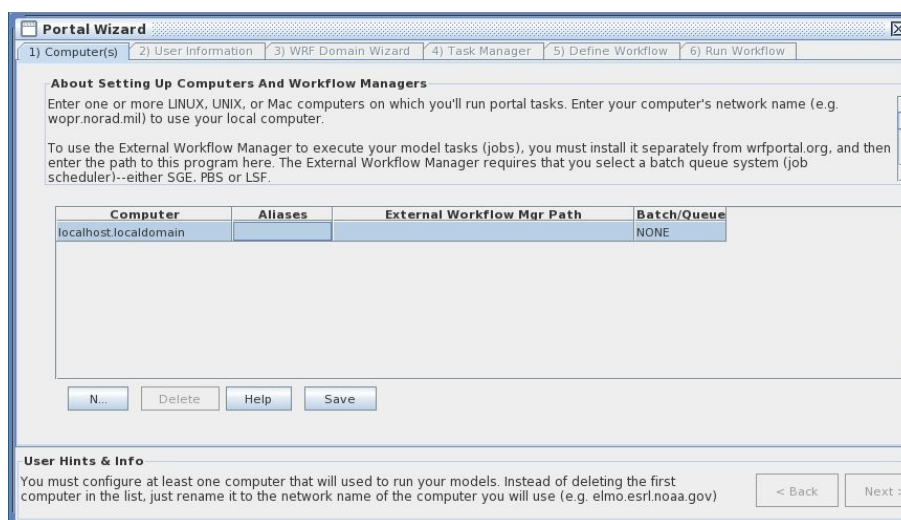
```
# yum groupinstall java
```

GUI で設定するために、WRF Portal を使う。WRF Portal のページ中程にあるリンクからダウンロードが可能である。wrf-portal.zip をダウンロード後、解凍。展開されたファイルを実行する。

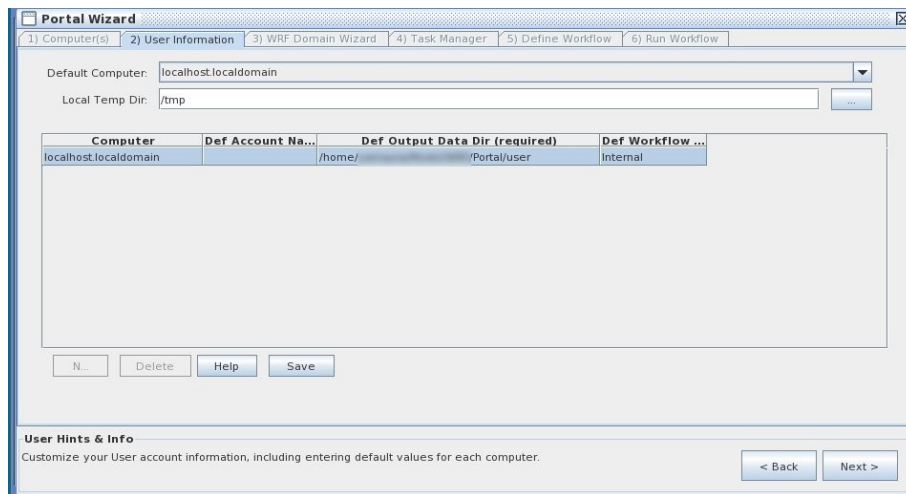
```
$ cd ~/Model
$ mkdir Domains
$ mkdir -p Portal/user
$ unzip /tmp/wrf-portal.zip -d Portal
$ cd Portal
$ sh runWRFPortal
```

Java による GUI が起動する。

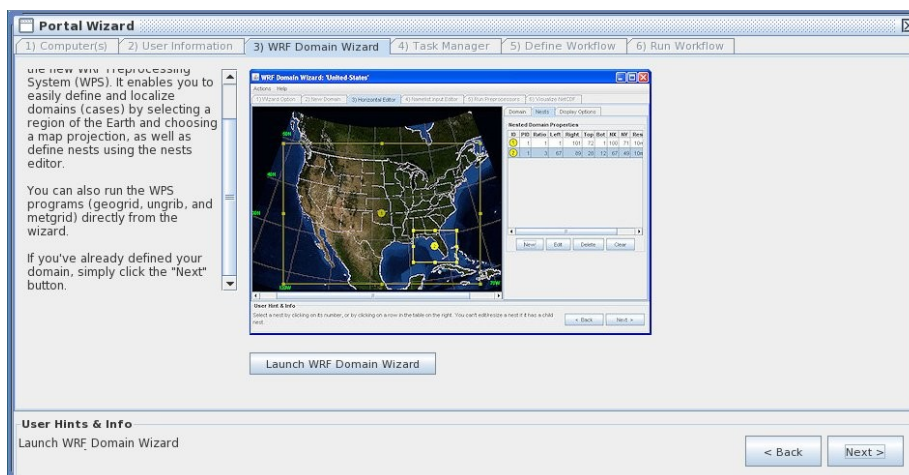
Portal の設定を行う。Computer に適当な名前を入力し、Save で保存。



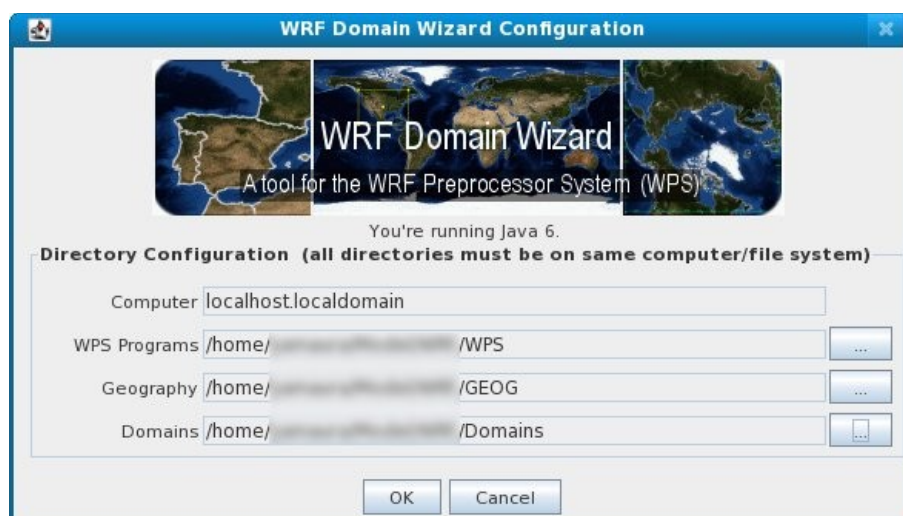
Portal のユーザ情報を入力。Default Output Directory に /home/{User Name}/Model/Portal/user を入力。  
Local Temp Directory に /tmp を入力。



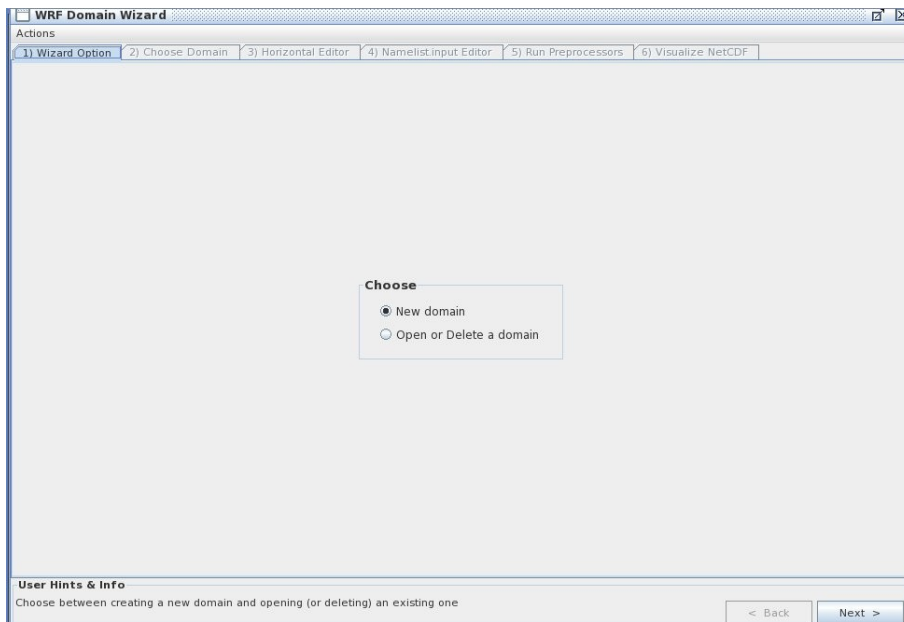
Launch WRF Domain Wizard で、WPS の GUI ツールを起動。



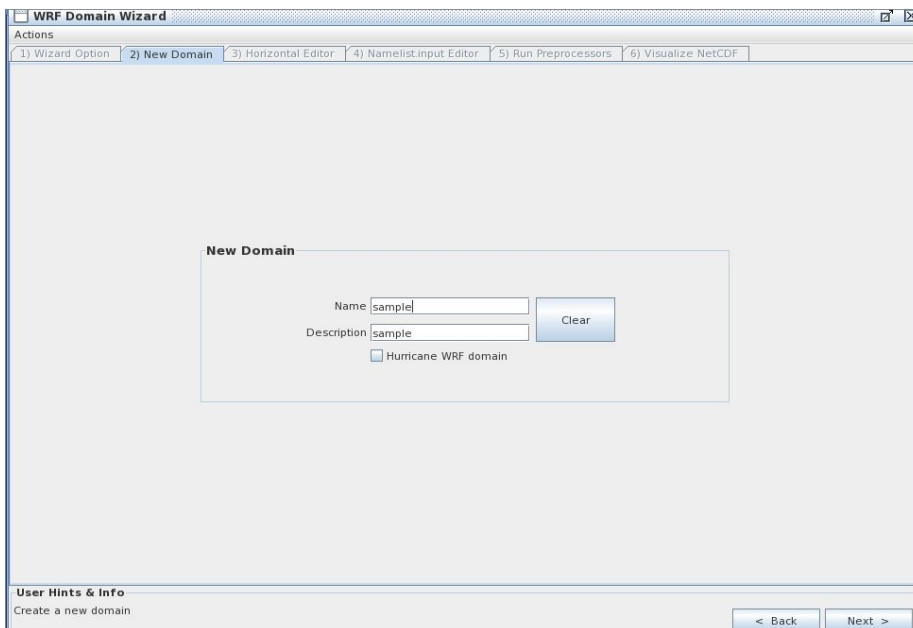
ディレクトリの初期設定を行う。



ドメインの設定を行う。新しく作る場合は New Domain を選択し、Next ボタンをクリック。



ドメインの名前と説明の設定は適当。ここでは Name/Description とともに sample と入力する。

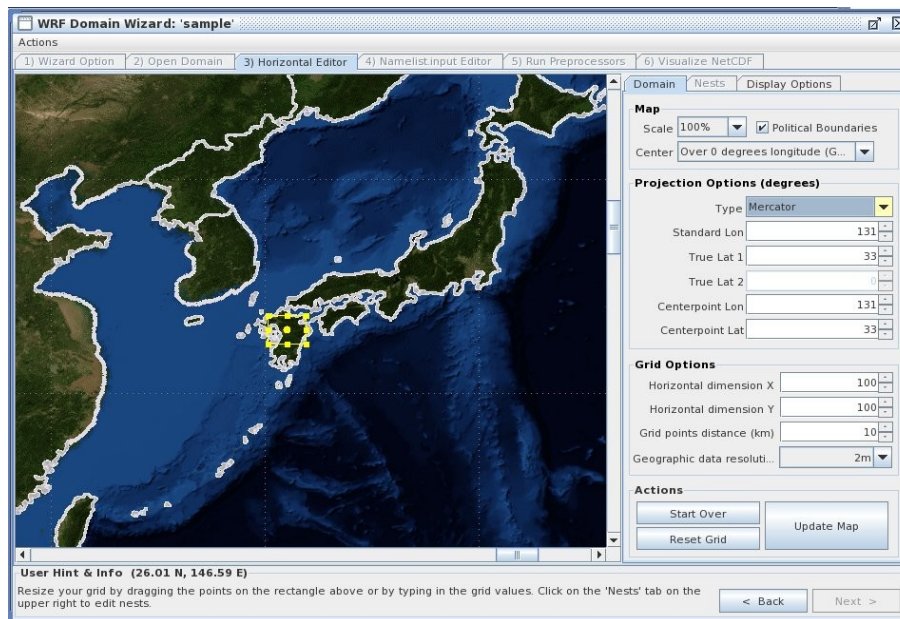


ドメインを指定する世界地図が表示される。地図サイズが大きすぎる場合は、右上の Map 設定から Scale: 100%とする。テストのため、マウスを用いて日本の九州地方付近を短形で囲む。緯度・経度は 33°N、131°E を中心とする。

Projection Options で Type: Mercator を選択し、Actions で Update Map を行う。すると、短形選択付近の地図だけが抜き出された形で表示される。Grid Options を次のように指定する。

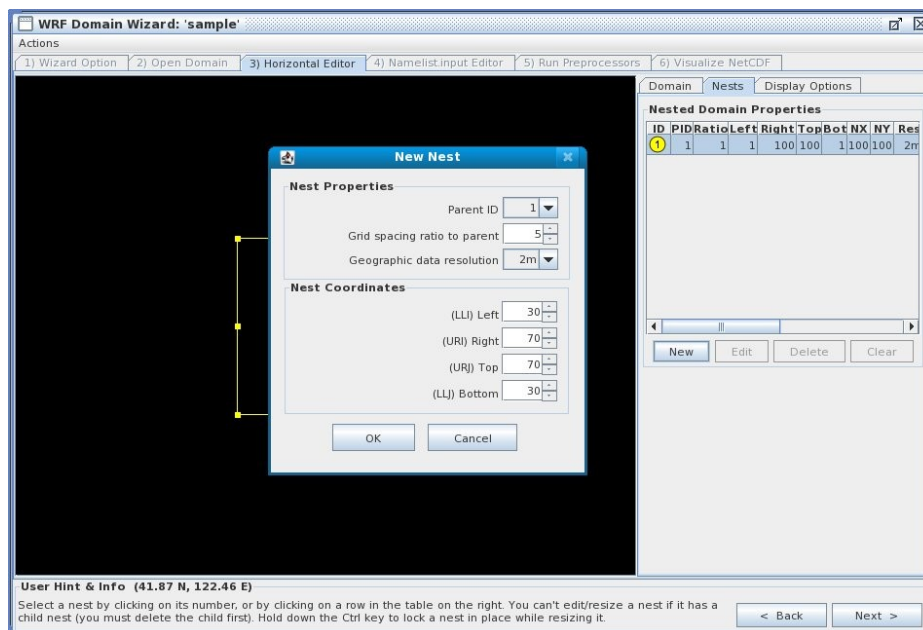
- ・ Horizontal dimension X: 100
- ・ Horizontal dimension Y: 100

- Grid point distance (km): 10
- Geographic data resolution: 2m



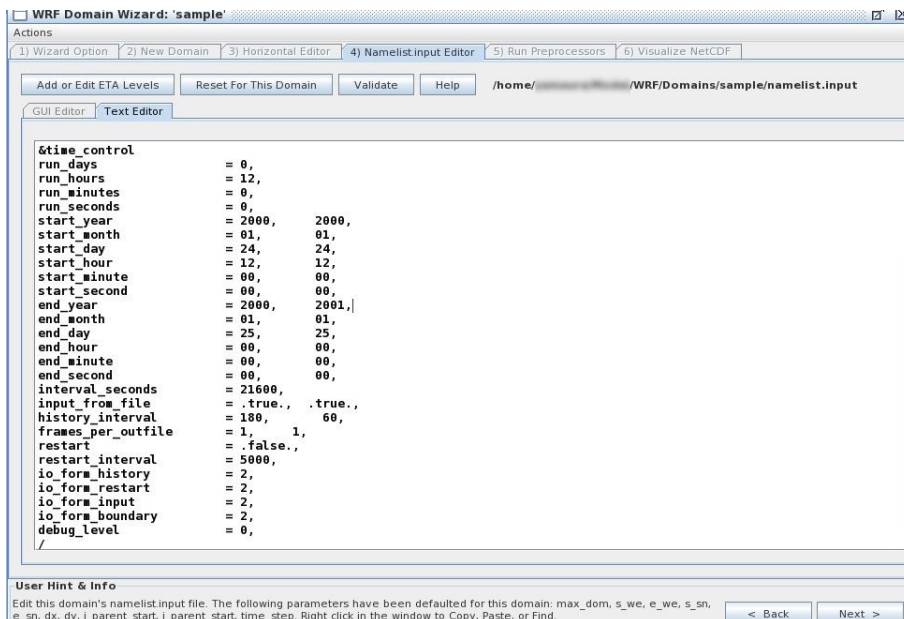
右上の Nest タブをクリックし、Nested Domain Properties から New をクリック。New Nest の設定を次のように指定する。

- Parent ID: 1
- Grid spacing ratio to parent: 5
- Geographic data resolution: 2m
- (LLI) Left: 30
- (URI) Right: 70
- (URJ) Top: 70
- (LLJ) Bottom: 30



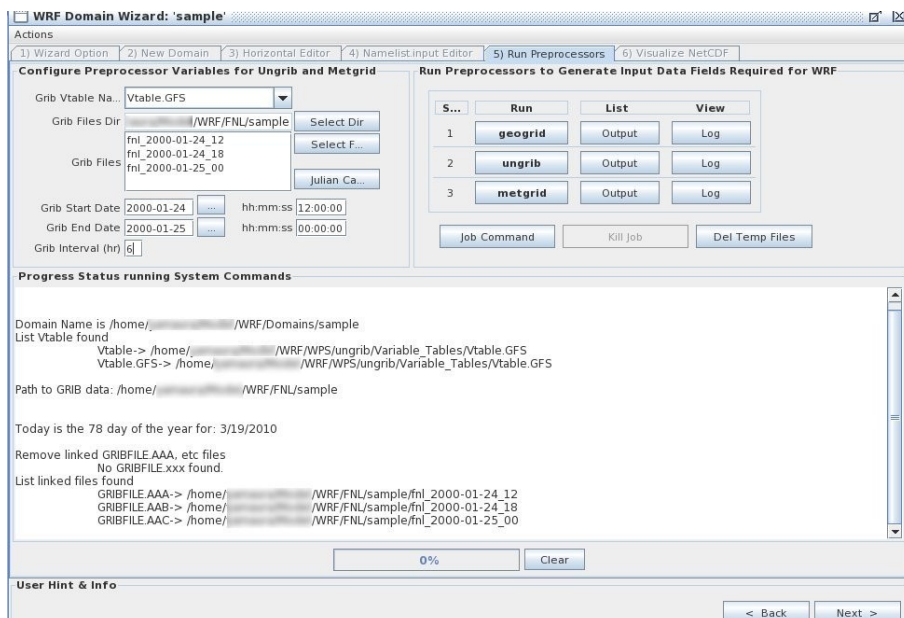
設定が終わったら OK をクリックし、第 2 ネストが表示されることを確認する。

namelist.input の設定を行う。GUI Editor と Text Editor の二つをタブで切り替えて編集できる。追加行については Text Editor で追加できる。Appendix に namelist.input の例を載せる。



編集が終わったら Next ボタンをクリック。

WPS を実際に実行するプロセスに入る。Grib Table Name は Vtable.GFS を選択し、Grib Files Dir は FNL データを置いたディレクトリを指定。Grib Files でサンプルの FNL データを全て加える。開始時間と終了時間は、2000-01-24-12:00:00 から 2000-01-25-00:00:00 で、時間間隔は 6(hr)である。指定したら、右の Run Preprocessors から geogrid、ungrib、metgrid の順にクリックする。全てが成功したら、Next ボタンをクリック。



生成されたファイルの確認画面。初めは初期化に時間を要する。geo\_em.\*.nc が 2 つ、met\_em.\*.nc が 6 つ生成されていることを確認したら、Exit ボタンを押して終了。



WRF Domain Wizard が終了したら、Portal もそのまま閉じる。生成した namelist.input 及び nc ファイルを WRFV3/test/em\_real ディレクトリ下に置く。

```
$ cd ~/Model/Domains/sample
$ mv namelist.input geo_em.*.nc met_em.*.nc ~/Model/WRFV3/test/em_real
```

### WPS without GUI

WPS は前述の GUI を使用せずに実行することも出来る。X Window 環境がない場合や端末のみで実行しなければならない場合など、GUI を使えない環境での WPS の使用法を説明する。扱いに習熟すれば、こちらのほうが操作が早く終わるかもしれない。

WPS ディレクトリへ移動。

```
$ cd ~/Model/WPS
```

FNL データへのリンク、及び Vtable へのリンクを作成する。

```
$ ./link_grib.csh ../FNL/sample/*
$ ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable
```

WPS の各プログラムを実行する際に必要となる設定ファイルを記述する。

```
$ vi namelist.wps
```

```
// ----- Edit the file -----
&share
wrf_core           = 'ARW',
max_dom            = 2,
start_date         = '2000-01-24_12:00:00', '2000-01-24_12:00:00',
end_date           = '2000-01-25_00:00:00', '2000-01-25_00:00:00',
interval_seconds   = 21600,
io_form_geogrid    = 2,
/
```

```

&geogrid
parent_id           = 1,1,
parent_grid_ratio   = 1,5,
i_parent_start      = 1,30,
j_parent_start      = 1,30,
e_we                = 100,201,
e_sn                = 100,201,
geog_data_res       = '10m','2m',
dx                  = 10000,
dy                  = 10000,
map_proj            = 'mercator',
ref_lat             = 33,
ref_lon             = 131,
truelat1            = 33,
truelat2            = 0,
stand_lon           = 131,
geog_data_path      = './GEOG',
ref_x                = 50.0,
ref_y                = 50.0,
/

&ungrib
out_format          = 'WPS',
prefix              = 'FILE',
/

&metgrid
fg_name             = 'FILE',
io_form_metgrid     = 2,
/

```

geogrid.exe、ungrib.exe、metgrid.exe の順にプログラムを実行する。

```

$ ./geogrid.exe
$ ./ungrib.exe
$ ./metgrid.exe

```

上記3つのプログラムが成功すると、複数のncファイルが生成される。namelist.inputと一緒にWRFのテストランを実行するディレクトリへ移動させる。

```

$ mv namelist.input geo_em.*.nc met_em.*.nc ~/Model/WRFV3/test/em_real

```

さらに実行ディレクトリにあるnamelist.inputも編集する。namelist.inputについてはAppendixを参照。

```

$ vi ~/Model/WRFV3/test/em_real/namelist.input

```

## WRF ラン

WPSで作成したテストデータを置いたディレクトリへ移動。

```

$ cd ~/Model/WRFV3/test/em_real

```

初期値・境界値データを作成する。MPIデーモンを起動、ブート設定を行っておく。Fedoraではスタックサイズに10240KB制限があるので、real.exeを実行する前にスタックサイズを無制限にしておく。

```

$ mpirun &

```

```
$ mpdboot -n 1 -f ~/.mpd.hosts
$ ulimit -s unlimited
$ mpirun ./real.exe
```

初期値・境界値データが wrfbdy、wrfinput というファイルで作成される。MPI を使ったランではエラーやメッセージ等の出力は同階層に rsl.\*ファイルに出力されているので、確認したい場合はそれらのファイルを参照すること。この後、wrf.exe を実行してシミュレーションを行う。

```
$ mpirun ./wrf.exe
```

WRF の実行は非常に時間がかかる。CPU プロセスに余裕をもてる時に実行するのが望ましい。計算範囲やタイムステップによって実行時間は大きく変化する。シミュレーションが終了すると、wrfout ファイルが作成される。

### ARWpost ラン

WRF で作成されたファイルは NetCDF のファイル形式をとっている。この出力ファイルを GrADS で表示できるように変換する。WRF で作成したファイルを ARWpost ディレクトリに移動する。

```
$ cd ~/Model/ARWpost
$ mv ~/Model/WRFV3/test/em_real/wrfout* .
```

GrADS 変換プログラムを走らせるために、namelist.ARWpost を編集する。

```
$ vi namelist.ARWpost
```

```
// ----- Edit the file -----
&datetime
start_date      = '2000-01-24_12:00:00',
end_date        = '2000-01-25_00:00:00',
interval_seconds = 3600,
tacc            = 0,
debug_level     = 0,
/

&io
io_form_input   = 2
input_root_name = './wrfout_d02*'
output_root_name = './sample'
plot            = 'all_list'
fields          = 'height,pressure,tk,tc'
output_type     = 'grads'
mercator_defs   = .false.
/

&interp
interp_method   = 0,
/
```

編集が終了したら、変換プログラムを走らせる。

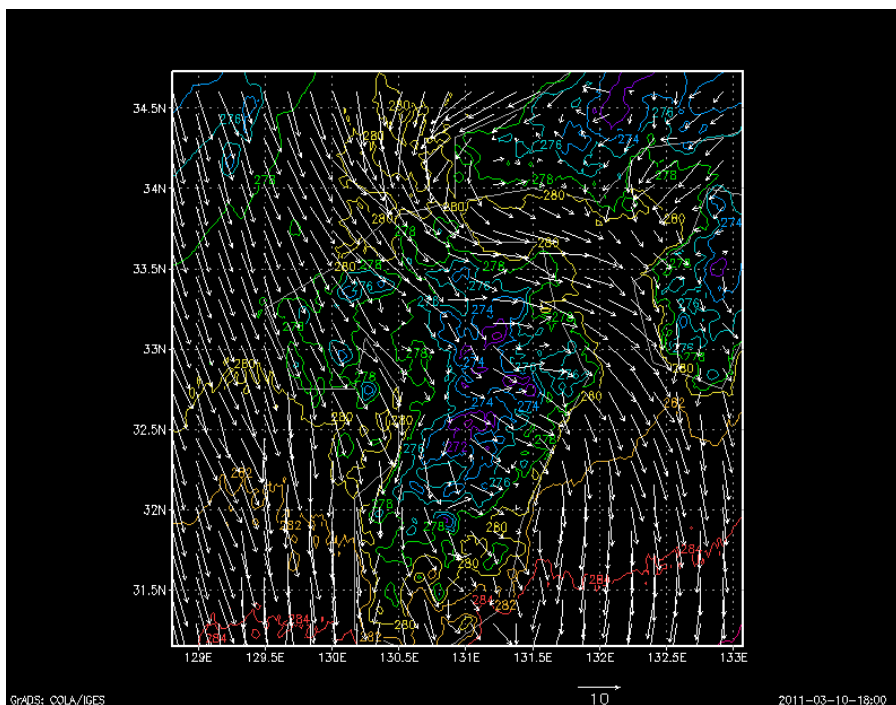
```
$ ./ARWpost.exe
```

出力が完了すると、GrADS 形式の sample.ctl と sample.dat が作成される。GrADS を起動し、出力を確認する。

```
$ grads
```

```
ga-> open sample.ctl
ga-> set t 10
ga-> d tk
ga-> d skip(u10,10,10);v10
```

設定通りに行うと、以下のような画像が出力される。



## Appendix

コンパイル途中、diff コマンドで差分をとって作成したパッチをあてている。パッチの修正内容は本文に書くと長くなるので、Appendixとして付記する。diff でとった修正内容は、前半に修正前、後半に修正後を書くという記述法をとっている。詳しくは man コマンドにて diff を参照すること。

### WRF / パッチ

パッチの要旨は次の通り。

- ・ Problem with Program real.exe (posted 5/27/11) 取り込み
- ・ Problem with New Configure Script (posted 4/8/11) 取り込み
- ・ Bug in SKEB code (posted 4/6/11) 取り込み
- ・ configure コンパイラテスト修正
- ・ arch/noopt\_exceptions\_f モジュール設定追加
- ・ phys/module\_cu\_g3.F バグ修正

### WPS / パッチ

パッチの要旨は次の通り。

- ・ geogrid/src/Makefile ライブラリ設定修正
- ・ metgrid/src/Makefile ライブラリ設定修正
- ・ ungrib/src/ngl/Makefile リンカ設定修正

### ARWpost / パッチ

パッチの要旨は次の通り。

- ・ src/Makefile ライブラリ設定修正。

### namelist.input の例

編集した部分を紫、追加した部分を赤で記載している。DomainWizard 側が自動で編集している部分もあるので、GUI Editor での表示をよく読んで書き換えること。

```
&time_control
run_days           = 0,
run_hours          = 12,
run_minutes        = 0,
run_seconds        = 0,
start_year         = 2000,      2000,
start_month        = 01,       01,
start_day          = 24,       24,
start_hour         = 12,       12,
start_minute       = 00,       00,
start_second       = 00,       00,
end_year           = 2000,      2000,
end_month          = 01,       01,
end_day            = 25,       25,
end_hour           = 00,       00,
end_minute         = 00,       00,
end_second         = 00,       00,
interval_seconds   = 21600,
input_from_file    = .true.,   .true.,
history_interval   = 180,      60,
frames_per_outfile = 1000,     1000,
```

```

restart                = .false.,
restart_interval      = 5000,
io_form_history       = 2,
io_form_restart      = 2,
io_form_input        = 2,
io_form_boundary     = 2,
debug_level          = 0,
/

&domains
time_step             = 60,
time_step_fract_num  = 0,
time_step_fract_den  = 1,
max_dom              = 2,
e_we                 = 100,          201,
e_sn                 = 100,          201,
e_vert               = 28,           28,
p_top_requested      = 5000,
num_metgrid_levels   = 27,
num_metgrid_soil_levels = 2,
dx                   = 10000,        2000,
dy                   = 10000,        2000,
grid_id              = 1,            2,
parent_id            = 1,            1,
i_parent_start       = 1,            30,
j_parent_start       = 1,            30,
parent_grid_ratio    = 1,            5,
parent_time_step_ratio = 1,          5,
feedback             = 1,
smooth_option        = 0,
/

&physics
mp_physics           = 3,            3,
ra_lw_physics        = 1,            1,
ra_sw_physics        = 1,            1,
radt                 = 30,           30,
sf_sfclay_physics    = 2,            2,
sf_surface_physics   = 2,            2,
bl_pbl_physics       = 2,            2,
bldt                 = 0,            0,
cu_physics           = 1,            0,
cudt                 = 10,           10,
isfflx               = 1,
ifsnow               = 0,
icloud               = 1,
surface_input_source = 1,
num_soil_layers       = 4,
sf_urban_physics     = 0,            0,
mp_zero_out          = 2,
mp_zero_out_thresh   = 1.e-8,
/

&fdda
/

&dynamics
w_damping             = 0,
diff_opt              = 0,

```

```

km_opt                = 1,
diff_6th_opt         = 0,          0,
diff_6th_factor      = 0.12,      0.12,
base_temp            = 290.,
damp_opt            = 0,
zdamp               = 5000.,      5000.,
dampcoef            = 0.2,        0.2,
khdif               = 0,          0,
kvdif               = 0,          0,
non_hydrostatic     = .false., .true.,
moist_adv_opt       = 1,          1,
scalar_adv_opt      = 1,          1,
/

&bdy_control
spec_bdy_width      = 5,
spec_zone           = 1,
relax_zone          = 4,
specified           = .true., .false.,
nested              = .false., .true.,
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups          = 1,
/

```