

Mechanics of running HWRF

Mrinal K Biswas

Developmental Testbed Center
NCAR

2014 HWRF Tutorial, College Park, MD, January 2014

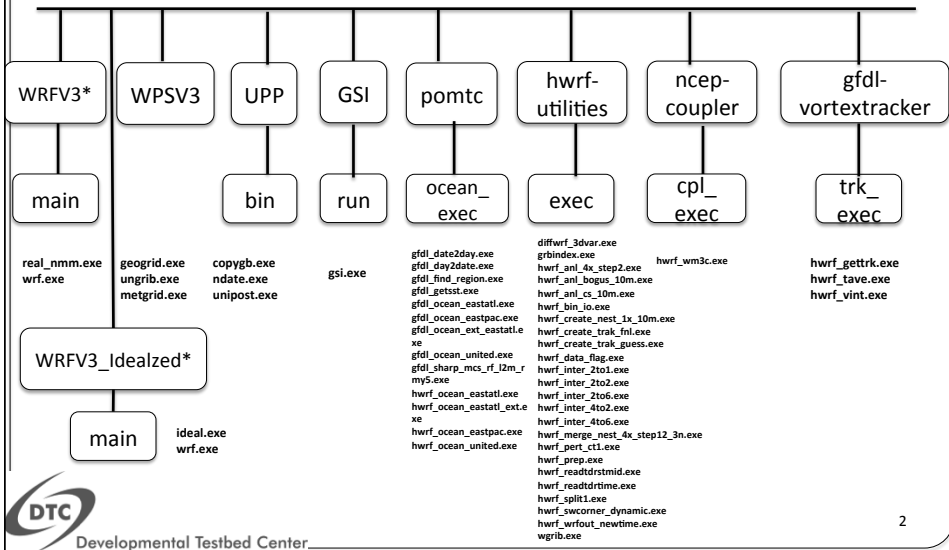


Developmental Testbed Center

1

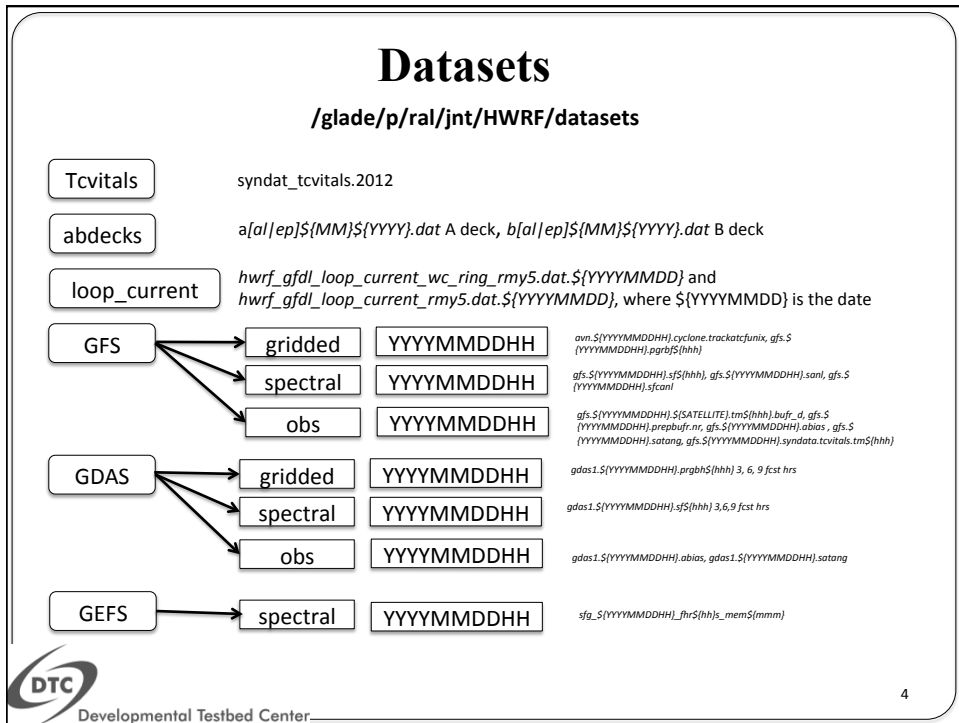
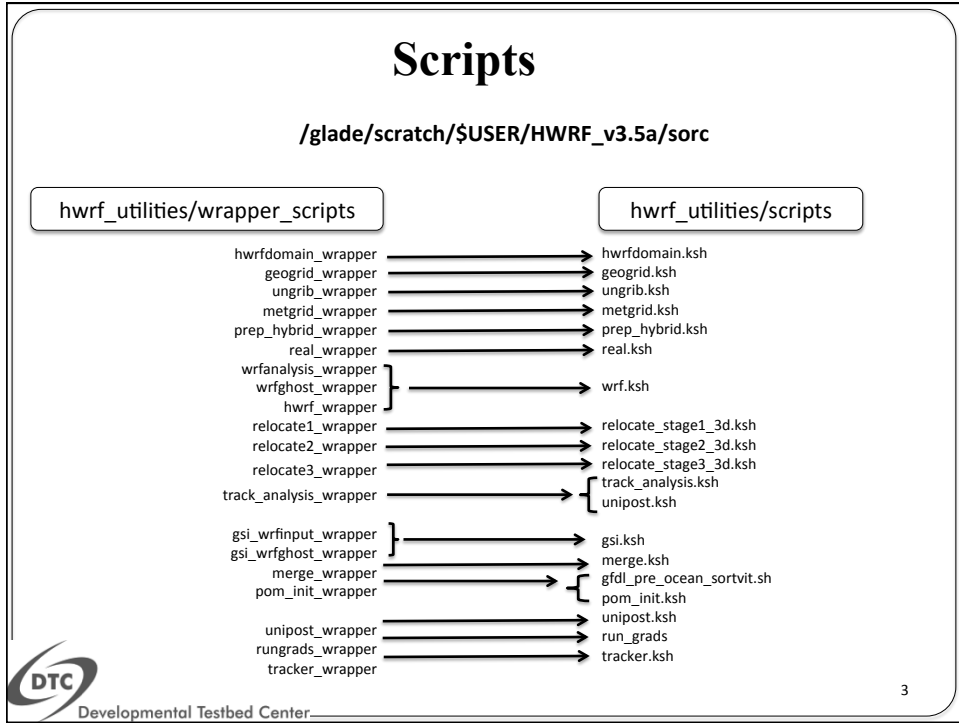
Source Structure and Executables

/glade/scratch/\$USER/HWRF_v3.5a/sorc
on Yellowstone



Developmental Testbed Center

2



Datasets

/glade/p/ral/jnt/HWRF/datasets

TDR → **STORM** **YYYYMMDDHH** *gdas1.YYYYMMDDHH.tldplr.tm00.bufr_d*

fix → **ocean** *gfdl_ocean_topo_and_mask.eostat1.gfdl_ocean_topo_and_mask.eostat1_extrn.gfdl_H4eepps.eostat1_etc*

 → **gsi** **CRTM_Coefficients** ...

hwrf_basinscale_satinfo.txt , nam_errtable.r3dv, prepobs_errtable.hwrf etc

 → **upp** **HWRF_SpcCoeff** ...


HWRF_TauCoeff ...

hwrf_CRTM_AerosolCoeff.bin_goestb hwrf_CRTM_imgr_g11.SpcCoeff.bin_goestb
hwrf_CRTM_imgr_g12.TauCoeff.bin_goestb etc

wps_geog → **Landuse_2m**

 → **topo_2m**

:



Developmental Testbed Center

5

Set up global variables


Edit global_vars file

```

/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/global_vars.ksh
# 1. Storm Info (settings for SANDY 18L 2012 storm)
export START_TIME=2012102806
export START_TIME_MINUS6=2012102800
export FCST_LENGTH=12
export FCST_INTERVAL=6
export STORM_NAME=SANDY
export SID=18L
export BASIN=AL

# HWRF Options
export RUN_PREP_HYB=T
export UPP_PROD_SAT=T
export RUN_GSI=T
    
```

*Need Additional Steps. Please contact wrfhelp@ucar.edu

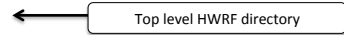


Developmental Testbed Center

6

Set up global variables..contd..

3. PATHS: SOURCE, SCRIPTS, DATA & OUTPUT DIRECTORY
export HWRP_SRC_DIR=/glade/scratch/\$USER/HWRP3.5a/sorc



DATA PATHS
export HWRP_DATA_DIR=/glade/p/ral/jnt/HWRP/datasets



export CYCLE_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME_MINUS6}



export DOMAIN_DATA=\${HWRP_OUTPUT_DIR}/\${SID}/\${START_TIME}



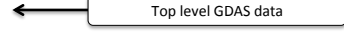
export GFS_DIR=\${HWRP_DATA_DIR}/GFS
export GFS_SPECTRAL_DIR=\${GFS_DIR}/spectral
export GFS_GRIDDED_DIR=\${GFS_DIR}/gridded
export GFS_OBS_DIR=\${GFS_DIR}/obs



export GEFS_DIR=\${HWRP_DATA_DIR}/GEFS
export GEFS_ENS_FCST_DIR=\${GEFS_DIR}/spectral



export GDAS_DIR=\${HWRP_DATA_DIR}/GDAS
export GDAS_SPECTRAL_DIR=\${GDAS_DIR}/spectral
export GDAS_GRIDDED_DIR=\${GDAS_DIR}/gridded
export GDAS_OBS_DIR=\${GDAS_DIR}/obs



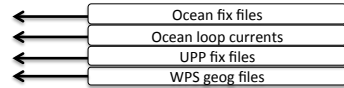
export TCVITALS=\${HWRP_DATA_DIR}/Tcvitals



Developmental Testbed Center

Set up global variables..contd..

FIXED FILES
export OCEAN_FIXED_DIR=\${HWRP_DATA_DIR}/ocean
export LOOP_CURRENT_DIR=\${HWRP_DATA_DIR}/loop_current
export CRTM_FIXED_DIR=\${HWRP_DATA_DIR}/upp
export GEOG_DATA_PATH=\${HWRP_DATA_DIR}/wps_geog



GSI PATHS
export GSI_FIXED_DIR=\${HWRP_DATA_DIR}/gsi
export GSI_CRTM_FIXED_DIR=\${GSI_FIXED_DIR}/CRTM_Coefficients



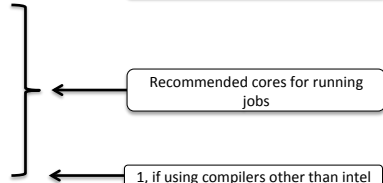
export TDR_OBS_DIR=\${HWRP_DATA_DIR}/TDR



export MPIRUN=mpirun.lsf



export WRF_ANAL_CORES=12
export WRF_GHOST_CORES=12
export HWRP_FCST_CORES=202
export GEOGRID_CORES=12
export METGRID_CORES=12
export REAL_CORES=1
export GSI_CORES=200
export UNI_CORES=12
export PREP_HYB_CORES=12



export ATCFNAME=HCOM



export GRADS_BIN=/glade/p/ral/jnt/HWRP/bin
export GADDIR=/glade/apps/opt/grads/2.0.2/data



Developmental Testbed Center

Namelist

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/parm

```

num_metgrid_soil_levels = @[GFS_SOURCE==GRIB_1x1_REDUCED?2:4],
/
&physics
num_soil_layers          = 4,
mp_physics                = 85,    85,    85,
ra_lw_physics            = 98,    98,    98,
ra_sw_physics            = 98,    98,    98,
sf_sfclay_physics        = 88,    88,    88,
sf_surface_physics        = 88,    88,    88,
bl_pbl_physics           = 3,     3,     3,
cu_physics                = 84,    84,    0,
mommix                   = 1.0,  1.0,  1.0,
var_ric                  = 1.0,
coef_ric_l               = 0.16,
coef_ric_s               = 0.25,
h_diff                   = 1.0,   1.0,   1.0,
gwd_opt                  = @[GWD==YES?2:0], 0,    0,
    
```

This will be filled in according to the options in global_vars.ksh
DO NOT CHANGE



Developmental Testbed Center

9

Scripts structure

Example: hwrfdomain.ksh

```

# Main
function main {
    typeset errno

    # Init the function library
    . ${HWRF_SCRIPTS}/funcs/init
    set -e

    # Check to see if all the variables are set
    check_vars DOMAIN_DATA START_TIME

    # Create a working directory (and cd into it)
    create_work_dir ${DOMAIN_DATA}/
    messages

    # Create the tcvital and tcvitals.as files
    lat_lon=( ${tcvitals} )

    # Create the storm centre file
    storm_centre ${lat_lon[@]}
    
```

functions

```

# Create the storm centre file
function storm_centre {
    typeset lat
    typeset lon

    lat=${ parse_ll $1 }
    lon=${ parse_ll $2 }

    printf "%lat\n$lon\n" > storm.center
}
    
```

External functions library (if used more than once)

```

#####
typeset hr=0
typeset f

check_vars -s FGAT
f=${FGAT}
if [ -n "$FGAT_INDEX" ]; then
    hr=${FGAT_INDEX}
fi
echo $hr
}
    
```



Developmental Testbed Center

10

How to submit jobs

Copy template file from /glade/p/ral/jnt/HWRF/bin

bsub_Yellowstone_wrapper.ksh

```
# LSBATCH: User input
#!/bin/ksh
#
# LSF batch script to run an MPI application
#
#BSUB -P NRAL0006 # project code
#BSUB -W 00:20 # wall-clock time (hrs:mins)
#BSUB -n $NPROCS # number of tasks in job
#BSUB -J myjob # job name
#BSUB -o myjob.%J.out # output file name in which %J is replaced by the job ID
#BSUB -e myjob.%J.err # error file name in which %J is replaced by the job ID
#BSUB -q regular # queue
#BSUB -U HWRF # reservation
#BSUB -K # do not return prompt
```

Rename the template files for each wrapper script and modify the \$NPROCS, wall clock time and wrapper script. Also the job name and std out and error file

We have used bsub_{\$wrapper_script} in our instructions

#run the wrapper

\$WRAPPER_NAME

bsub < bsub_{\$wrapper_script} for parallel jobs → For submitting wrappers

./\$wrapper_script for serial jobs



Developmental Testbed Center

11

Set up HWRF domain

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/hwrfdomain_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/hwrfdomain.ksh

- ✓ Check env vars
- ✓ Create working directory (\$\${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/messages)
- ✓ Create tcvital and tcvitals.as from Tcvitals file for the particular storm case
- ✓ Determine storm center
- ✓ Determine domain center
- ✓ Create GSI status file: if RUN_GSI=T -> go_gsi else no_gsi



Developmental Testbed Center

Details can be found in Section 3.2.1 in UG 12

Running hwrfdomain

Determine the domain center for the parent domain, as well as the storm center

In this lecture Sandy cold start case (2012102806) is taken as an example, however, when cycled with TDR is discussed 2012102812 case is considered

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

./hwrfdomain_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/messages

Following files will be generated:

domain.center
go_gsi (Or no_gsi)
storm.center
tcvital
tcvital.as



Developmental Testbed Center

13

WPS: Geogrid

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/geogrid_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/geogrid.ksh

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory (\${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/geoprd)
- ✓ Create namelist
- ✓ Copy geogrid table
- ✓ Run geogrid



Developmental Testbed Center

Details can be found in Section 3.2.2 in UG 14

Run WPS: Geogrid

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_geogrid_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/geoprd

Following files will be generated:

geo_nmm.d01.nc, geo_nmm_nest.l01.nc, and geo_nmm_nest.l02.nc



WPS: Ungrib

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/ungrib_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/ungrib.ksh

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory ($\${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/ungribprd/\${YYYYMMDDHH}$)
 - If TDR present and cycled create directories for -3, 0 +3 hr of START_TIME*
- ✓ Create namelist
- ✓ Copy ungrib table
- ✓ Link grib files
- ✓ Run geogrid



Run WPS: Ungrib

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_ungrib_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/ungribprd/2012102806

Following files will be generated:

FILE:2012-10-28_06

Check output (when TDR present and cycled):

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/ungribprd/2012102809/
FILE:2012-10-28_09

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/ungribprd/2012102812/
FILE:2012-10-28_12

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/ungribprd/2012102815/
FILE:2012-10-28_15



Developmental Testbed Center

17

WPS: Metgrid

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/metgrid_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/metgrid.ksh

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory (\$HWRF_OUTPUT_DIR)/\${SID}/\${START_TIME}/metgridprd/\${YYYYMMDDHH})
 - If TDR present and cycled create directories for -3, 0, +3 hr of START TIME*
- ✓ Create namelist
- ✓ Copy metgrid table
- ✓ Copy geogrid output file
- ✓ Run metgrid



Developmental Testbed Center

Details can be found in Section 3.2.5 in UG 18

Run WPS: Metgrid

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_metgrid_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/metgridprd/2012102806

Following files will be generated (for non TDR):

met_nmm.d01.2012-10-28_06:00:00.nc

Check output (when TDR present and cycled):

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/metgridprd/2012102809/

met_nmm.d01.2012-10-28_09:00:00.nc

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/metgridprd/2012102812/

met_nmm.d01.2012-10-28_12:00:00.nc

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/metgridprd/2012102815/

met_nmm.d01.2012-10-28_15:00:00.nc



Developmental Testbed Center

19

Prep_hybrid

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/prep_hybrid_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/prep_hybrid.ksh

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/prep_hybrid/\${YYYYMMDDHH} (for ICs) & \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/prep_hybrid_GFS/\${YYYYMMDDHH} (for LBCs)
 - if TDR present and cycled create directories for -3, 0, +3 hr for Ics of START TIME*
- ✓ Make sure there is large enough stack
- ✓ Copy input
- ✓ Run prep_hybrid
- ✓ Link output



Developmental Testbed Center

Details can be found in Section 3.2.3 in UG

20

Running prep_hybrid

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_prep_hybrid_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/prep_hybrid/2012102806

Following files will be generated:

hwrfinit_00, hwrfbcs00_00, hwrfbcs00_01

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/prep_hybrid_GFS/
2012102806

Following files will be generated:

hwrfinit_00, hwrfbcs00_\${bc_index} bc_index=0,1,2,...21

Directory when TDR present and cycled:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/prep_hybrid/2012102809
/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/prep_hybrid/2012102812
/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102812/prep_hybrid/2012102815



Developmental Testbed Center

21

Real

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/real_wrapper



/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/real.ksh

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/realprd/\${YYYYMMDDHH} (for ICs) & \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/realprd_GFS (for LBCs)

If TDR present and cycled create directories for -3, 0, +3 hr for ICs of START_TIME

- ✓ Link input and fix files
- ✓ Run hwrf_swcorner_dynamic.exe to calculate the nest domain location
- ✓ Generate namelist
- ✓ Make sure there is large enough stack (> 2 GB)
- ✓ Run real



Developmental Testbed Center

Details can be found in Section 4.3.1 in UG

22

Running Real

Check if the following variables are correctly defined:

```
export BKG_MODE=GFS
export FGAT_INDEX=1
```

```
export BKG_MODE=GDAS
export FCST_LENGTH=3
export FCST_INTERVAL=3
```



In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory
bsub < bsub_real_wrapper

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/realprd/
2012102606
```

Following files will be generated:

```
wrfinput_d01, wrfbdy_d01, fort.65
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/realprd_GFS
```

Following files will be generated

```
wrfinput_d01, wrfbdy_d01, fort.65
```



Developmental Testbed Center

GSI: gsi_wrfinput

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/gsi_wrfinput_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/gsi.ksh
```

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory `/${HWRF_OUTPUT_DIR}/${SID}/${START_TIME}/gsiprd/wrfinput`
- ✓ Copy fix files and background analysis
- ✓ Flag the obs data near storm center so that it is not assimilated
- ✓ Generate namelist
- ✓ Copy ensemble data
- ✓ Run gsi



Developmental Testbed Center

Details can be found in Section 4.3.2 in UG

Running gsi_wrfinput

Check if the following variables are correctly defined:

```
In gsi_wrfinput_wrapper
export DOMAIN=wrfinput
```

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

```
bsub < bsub_wrfinput_wrapper
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/gsiprd/wrfinput
```

Following files will be generated:

```
stdout, wrf_inout
```



Developmental Testbed Center

25

WRF Analysis

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/wrfanalysis_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/wrf.ksh
```

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/wrfanalysisprd/YYYYMMDDHH
 - If TDR present and cycled create directories for -3, 0, +3 hr of START_TIME*
- ✓ Link fix data, initial conditions and boundary conditions
- ✓ Run hwrf_swcorner_dynamic.exe to calculate istart and jstart values for middle nest domain grid
- ✓ Generate namelist
- ✓ Run wrfanalysis



Developmental Testbed Center

Details can be found in Section 4.3.3 in UG 26

Running wrfanalysis

Check if the following variables are correctly defined:

```
In wrfanalysis_wrapper
export WRF_MODE=analysis
```

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

```
bsub < bsub_wrfanalysis_wrapper
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/wrfanalysisprd/
2012102806
```

Following files will be generated:

```
wrfanl_d02*, wrfanl_d03*
```



Developmental Testbed Center

27

WRF Ghost

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/wrfghost_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/wrf.ksh
```

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/wrfghostprd/YYYYMMDDHH
 - If TDR present and cycled create directories for -3, 0, +3 hr of START_TIME*
- ✓ Link fix data, initial conditions and boundary conditions
- ✓ Run hwrf_swcorner_dynamic.exe to calculate istart and jstart values for middle nest domain grid
- ✓ Generate namelist
- ✓ Run wrfghost



Developmental Testbed Center

Details can be found in Section 4.3.4 in UG ²⁸

Running wrfghost

Check if the following variables are correctly defined:

In wrfanalysis_wrapper
export WRF_MODE=ghost

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_wrfghost_wrapper

Check output:

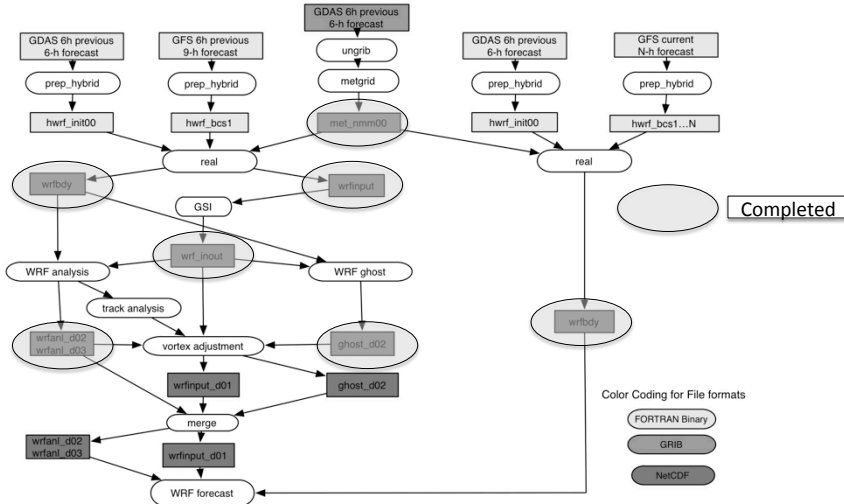
/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/wrfghostprd/2012102806

Following files will be generated:

wrfanl_d02*, wrfanl_d03*



HWRF v3.5a Atmospheric Initialization without inner core data



Track Analysis

`/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/track_analysis_wrapper`



`/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/track_analysis.ksh`

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory `${HWRF_OUTPUT_DIR}/${SID}/${START_TIME}/trkanalysisprd/YYYYMMDDHH`
- ✓ Copy UPP fix files and wrfout_d01 from wrfanalysis 90-s run
- ✓ Run `hwrf_wrfout_newtime.exe` to change the timestamp in the 90-s wrfout_d01 to time 0 for GFDL tracker to start from zero hour
- ✓ Run `unipost` to post-process wrfout_d01
- ✓ Run `copygb` to de-stagger the output to regular lat/lon
- ✓ Run GFDL vortex tracker



Developmental Testbed Center

Details can be found in Section 4.3.5 in UG ³¹

Running Track Analysis

In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory

`bsub < bsub_track_analysis_wrapper`

Check output:

`/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/trkanalysisprd/2012102806`

Following files will be generated:

`gfs-anl-fix.atcfunix`



Developmental Testbed Center

32

Relocate Stage 1

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/relocate1_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/relocate_stage1_3d.ksh
```

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory `${HWRF_OUTPUT_DIR}/${SID}/${START_TIME}/relocateprd/${YYYYMMDDHH}/stage1`
If cycled and TDR create directories for -3, 0, +3 hr of START_TIME
- ✓ Check if previous cycle exist and the storm intensity is greater than 16m/s; if not, exit
- ✓ Run `merge_nest_4x_step12_3n.exe` to merge `wrfout_d01`, `wrfout_d02` and `wrfout_d03` onto 3X domain
- ✓ Run `hwrf_create_trak_guess.exe` to produce a guess track (0,3,6,9 hour) for the current forecast using previous cycle forecast track.
- ✓ Run `wrf_split1.exe` to separate `data_4x_hwrf` into two parts: an environment field (`wrf_env`) and a storm vortex (`storm_pert`).



Run `hwrf_pert_ct1.exe` to do adjustments to `storm_pert`.

Details can be found in Section 4.3.6 in UG ³³

Developmental Testbed Center

Running Relocate Stage 1

Check if the following variables are correctly defined

```
In relocate1_wrapper
export INNER_CORE_DA=1
```

In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory

```
bsub < bsub_relocate1_wrapper
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/relocateprd/2012102806/
stage1
```

Following files will be generated:

```
analysis.namelist.input, ghost.namelist.input
```

Following files will be generated: (if cycled and storm intensity gt 16 m/s)

```
storm_size_p, storm_pert_new, storm_sym, storm_radius, wrf_env
```



Developmental Testbed Center

34

Relocate Stage 2

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/relocate2_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/relocate_stage2_3d.ksh
```

- ✓ Check env vars and executables
- ✓ Create working directory `/${HWRP_OUTPUT_DIR}/${SID}/${START_TIME}/relocateprd/${YYYYMMDDHH}/stage2`
If cycled and TDR create directories for -3, 0, +3 hr of START_TIME
- ✓ Run `diffwrf_3dvar.exe` to convert `wrfinput_d01`, `wrfanl_d02`, `wrfanl_d03` and `wrfghost_d02` (copied from `wrfghost_d03`) into binary files
- ✓ Run `hwrf_create_nest_1x_10m.exe` to rebalance the inner nest domain data.
- ✓ Run `hwrf_create_trak_fnl.exe` to create `trak_fnl.all_gfs`, a guess track file from `atcfunix`.
- ✓ Run `hwrf_merge_nest_4x_step12_3n.exe` to merge inner domain (`new_gfs_d03`), middle domain (`new_gfs_d02`), and outer domain (`new_gfs_d01`) onto the 3X domain.
- ✓ Run `hwrf_split1.exe` to separate the `data_4x_gfs` into environment data (`gfs_env`) and storm vortex (`storm_pert_gfs`).



Developmental Testbed Center

Details can be found in Section 4.3.7 in UG ³⁵

Running Relocate Stage 2

In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory

bsub < bsub_relocate2_wrapper

Check output:

`/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/relocateprd/2012102806/stage2`

Following files will be generated:

`gfs_env`, `roughness2`, `storm_pert_gfs`, `storm_radius_gfs`



Developmental Testbed Center

36

Relocate Stage 3

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/relocate3_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/relocate_stage3_3d.ksh
```

- ✓ Check env vars and executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/relocateprd/\${YYYYMMDDHH}/stage3

If cycled and TDR create directories for -3, 0, +3 hr of START_TIME

- ✓ If cold start, run hwrf_anl_bogus_10m.exe
- ✓ For cycled, run hwrf_anl_4x_step2.exe and if max wind is less than observed run hwrf_anl_cs_10m.exe to get new_data_4x
- ✓ Run hwrf_inter_4to6.exe to interpolate new_data_4x from 3x domain onto outer domain
- ✓ Run *hwrf_inter_4to2.exe* to interpolate new_data_4x from 3x onto ghost domain.
- ✓ Run diffwrf_3dvar.exe to convert data_merge_d01 to netcdf wrfinput_d01.
- ✓ Run diffwrf_3dvar.exe to convert data_merge_2x to netcdf wrffghost_d02



Developmental Testbed Center

Details can be found in Section 4.3.8 in URG

Running Relocate Stage 3

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_relocate3_wrapper

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/relocateprd/2012102806/  
stage3
```

Following files will be generated:

```
wrfinput_d01, wrffghost_d02
```



Developmental Testbed Center

38

GSI: gsi_wrfghost

Use only when TDR present and cycled

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/gsi_wrfghost_wrapper
```

↓

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/gsi.ksh
```

- ✓ Check env vars
- ✓ Check executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/gsiprd/wrfghost
- ✓ Link fix files
- ✓ Copy wrfghost_d02 file
- ✓ Copy TDR bufr file
- ✓ Generate namelist
- ✓ Run gsi



Details can be found in Section 4.3.2 in UG

39

Running gsi_wrfghost

Check if the following variables are correctly defined:

```
In gsi_wrfinput_wrapper
export DOMAIN=wrfghost
```

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

```
bsub < bsub_wrfghost_wrapper
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/gsiprd/wrfghost
```

Following files will be generated:

```
Ghost_d02*, ghost_d03*, two "analysis" files for HWRF middle and "ghost" domains
```


40

Merge

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/merge_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/merge.ksh
```

- ✓ Check env vars and executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/mergeprd
- ✓ Copy the input analysis file
- ✓ Copy namelist
- ✓ Run diffwrf_3dvar.exe to convert wrfinput_d01, wrfinput_d02, wrfinput_d03 and, wrfghost_d02 to unformatted data files
- ✓ Run hwrf_inter_2to6.exe
- ✓ Run hwrf_inter_2to1.exe
- ✓ Run hwrf_inter_2to2.exe
- ✓ Run diffwrf_3dvar.exe
- ✓ Rename files



Developmental Testbed Center

Details can be found in Section 4.3.10 in UG ⁴¹

Running Merge

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_merge_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/mergeprd

Following files will be generated:

wrfinput_d01, wrfanl_d02_\${YYYY-MM-DD_HH}:00:00;, wrfanl_d03_\${YYYY-MM-DD_HH}:00:00



Developmental Testbed Center

42

POM-TC Initialization

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/pom_init_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/pom_init.ksh
```

- ✓ Check env vars and executables
- ✓ Alias the executables and scripts
- ✓ Create working directory $\${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/oceanprd$
- ✓ Set stack size
- ✓ Get the existing storm track information using `get_tracks`
- ✓ Find the ocean region using function `get_region` and then set it accordingly.
- ✓ Get the GFS SST using function `get_sst`
- ✓ Run the feature-based sharpening program using function `sharpen`.
- ✓ Run POM-TC phase 1 (a.k.a. phase 3) using function `phase_3`.
- ✓ Run POM-TC phase 2 (a.k.a. phase 4) using function `phase_4`.

Run POM-TC for AL and EP basin **ONLY**



Developmental Testbed Center

Details can be found in Section 5.2 in UG ⁴³

Running POM-TC

In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory

bsub < bsub_pom_init_wrapper

Check output:

`/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/oceanprd`

Following files will be generated:

`phase4/RST.final`, `track`, `getsst/sst.gfs.dat`, `getsst/mask.sst.dat`, `getsst/lonlat.gfs`, `sharpn/gfdl_initdata.united.10`



Developmental Testbed Center

44


WRF

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/hwrf_wrapper

↓

/glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/wrf.ksh

- ✓ Check env vars and executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/wrfprd
- ✓ Link input files required by WRF, incl. fix files, initial and boundary condition files and geog data files
- ✓ Run hwrf_swcorner)dynamic.exe to calculate the location of middle nest and generate WRF namelist
- ✓ For AL and EP basin
 - ✓ Link input files required by POM-TC
 - ✓ Generate namelist for POM-TC
 - ✓ Generate namelist for coupler
- ✓ Run wrf, coupler and ocean (for coupled runs)
- ✓ Run wrf (for uncoupled runs)



Details can be found in Section 6.2 in UG

45

Developmental Testbed Center

Running WRF

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_hwrf_wrapper


Check output:
/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/wrfprd

Following files will be generated:
*wrfout_d01_yyyy-mm-dd_hh:mm:ss, wrfout_d02_yyyy-mm-dd_hh:mm:ss,
 wrfout_d03_yyyy-mm-dd_hh:mm:ss*

auxhist1_d01_yyyy-mm-dd_hh:mm:ss, auxhist1_d02_yyyy-mm-dd_hh:mm:ss, auxhist1_d02_yyyy-mm-dd_hh:mm:ss

hifreq_d03.htcf

If coupled:
*GRADS.yymmddhh, EL.yymmddhh, MLD.yymmddhh, OHC.yymmddhh, T.yymmddhh,
 TXY.yymmddhh, U.yymmddhh, V.yymmddhh, WTSW.yymmddhh*


Developmental Testbed Center
46

UPP

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/unipost_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/unipost.ksh
```

- ✓ Check env vars and executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/postprd/\${FHR}
- ✓ Copy fix file and control file required by UPP
- ✓ If UPP_PROD_SAT=F no satellite variables will be produced, if T, satellite variables are produced
- ✓ Run *unipost.exe* for each forecast valid time for parent, middle and inner nest domains. A namelist itag is produced for each forecast valid time and domain then read in by *unipost.exe*
- ✓ Run *copygb* to horizontally interpolate the native UPP output variables to a variety of regular lat/lon grids
- ✓ Create merged UPP output files for input in the GFDL vortex tracker



Developmental Testbed Center

Details can be found in Section 7.2 in UG 47

Running Unipost

In /glade/scratch/\$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts directory

bsub < bsub_unipost_wrapper

Check output:

/glade/scratch/\$USER/HWRF_v3.5a/results/18L/2012102806/postprd

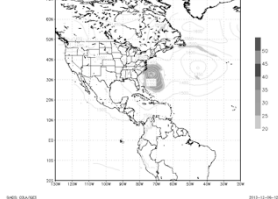
Following files will be generated:

- d01_d01_010_all.S(fhr)
- d01_d01_010_sel.S(fhr)
- d01_d01_025_all.S(fhr)
- d01_d01_025_sel.S(fhr)
- d01_d02p_003_all.S(fhr)
- d01_d02p_003_sel.S(fhr)
- d02_d01_010_all.S(fhr)
- d02_d01_010_sel.S(fhr)
- d02_d02_010_all.S(fhr)
- d02_d02_010_sel.S(fhr)
- d02_d02p_003_all.S(fhr)
- d02_d02p_003_sel.S(fhr)
- d03_d01_010_all.S(fhr)
- d03_d01_010_sel.S(fhr)
- d03_d02p_003_all.S(fhr)
- d03_d02p_003_sel.S(fhr)
- d03_d03_003_all.S(fhr)
- d03_d03_003_sel.S(fhr)
- merged_d01d02d03_d01_010_sel.S(fhr)
- merged_d01d02d03_d02p_003_sel.S(fhr)

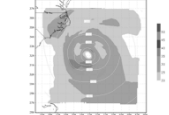
d01_d01_010_all.006 is the 6-hour HWRF parent domain forecast output that has been interpolated to a regular lat/lon grid covering an area similar to the one of the parent domain, with a horizontal resolution of 0.1 degree, containing all the variables present in the *unipost.exe* output file.

If "variables"="all", all the variables from the *WRFPFS* files are included in the interpolated GRIB file. When only those variables required by the GFDL vortex tracker are retained, "variable"="sel".

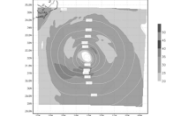
d01: Speed (m/s) & geo height (contour-m) at 850 hPa



d02: Speed (m/s) & geo height (contour-m) at 850 hPa



d03: Speed (m/s) & geo height (contour-m) at 850 hPa



Developmental Testbed Center

48

GFDL Vortex Tracker

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/tracker_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/tracker.ksh
```

- ✓ Check env vars and executables
- ✓ Create working directory \${HWRF_OUTPUT_DIR}/\${SID}/\${START_TIME}/gvtprd
- ✓ Create tracker namelist file
- ✓ Concatenate the UPP output files for all forecast times into one GRIB file
- ✓ Run *grbindex*
- ✓ Create a file, *fcst_minutes*, which contains the forecast lead times the tracker will process
- ✓ Link input files
- ✓ Run tracker executable *hwrf_gettrk.exe*

To generate Phase Space Diagnostics and how to run tracker in cyclogenesis mode refer to Section 8 in the Users Guide



Developmental Testbed Center

Details can be found in Section 8.2 in UG

49

Running GFDL Vortex Tracker

In `/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts` directory

bsub < bsub_tracker_wrapper

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/gvtprd
```

Following files will be generated:

combined/126hr/6hr/fort.64, fort.69

parent/126hr/6hr/fort.64, fort.69

combined/012hr/6hr/fort.64, fort.69

A sample of the vortex tracker output fort.69:

```
AL, 18, 2012102806, 03, HCOM, 00000, 315N, 737W, 65, 949, XX, 34, NEQ, 0351, 0106, 0141, 0329, 0, 0, 36
AL, 18, 2012102806, 03, HCOM, 00000, 315N, 737W, 65, 949, XX, 50, NEQ, 0058, 0058, 0070, 0070, 0, 0, 36
AL, 18, 2012102806, 03, HCOM, 00000, 315N, 737W, 65, 949, XX, 64, NEQ, 0000, 0000, 0039, 0037, 0, 0, 36
```

A sample of the vortex tracker output fort.64 is listed below:

```
AL, 18, 2012102806, 03, HCOM, 000, 315N, 737W, 65, 949, XX, 34, NEQ, 0351, 0106, 0141, 0329, 0, 0, 36, 0, 0, , 0,
0, , , , 0, 0, 0, THERMO PARAMS, -9999, -9999, -9999, U, 10, DT, -999
AL, 18, 2012102806, 03, HCOM, 000, 315N, 737W, 65, 949, XX, 50, NEQ, 0058, 0058, 0070, 0070, 0, 0, 36, 0, 0, , 0, , , ,
0, 0, 0, THERMO PARAMS, -9999, -9999, -9999, U, 10, DT, -999
AL, 18, 2012102806, 03, HCOM, 000, 315N, 737W, 65, 949, XX, 64, NEQ, 0000, 0000, 0039, 0037, 0, 0, 36, 0, 0, , 0, , , ,
0, 0, 0, THERMO PARAMS, -9999, -9999, -9999, U, 10, DT, -999
```

Details of the ATCF Format can be found in Section 8.6 (page 123) of the Users Guide



Developmental Testbed Center

50

Plotting the model output

- **WRF native output**
 - Ncview
- **UPP output**
 - GrADS (examples in the next slide)
 - NCL
 - Software of your choice
- **POM-TC output**
 - GrADS
 - `/glade/scratch/${USER}/HWRFv3.5a/sorc/pomtc/ocean_plot/pom-tc-united-grads.sh`
- **GFDL Vortex tracker output**
 - GrADS
 - `/glade/scratch/${USER}/HWRFv3.5a/sorc/gfdl-vortextracker/trk_plot_plottrack/atcfplot.sh`



Developmental Testbed Center

51

Plot UPP Output (Using GrADS)

```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/wrapper_scripts/rungrads_wrapper
```



```
/glade/scratch/$USER/HWRF_v3.5a/sorc/hwrf-utilities/scripts/run_grads
```

- ✓ Make sure `grib2ctl.pl` and `gribmap` is available in `$(GRADS_BIN)`

```
./rungrads_wrapper
```

Check output:

```
/glade/scratch/$USER/HWRF_v3.5a/results/18L/2012102806/postprd
```

Following files will be generated:

Once you are done creating plots with *rungrads_wrapper*, the image files you created may be viewed with the *display* command (or any command that can display the graphics).

Note: *display* is a utility in the *ImageMagick* program



Developmental Testbed Center

52