User's Guide for the NMM Core of the Weather Research and Forecast (WRF) Modeling System Version 2.1

Chapter 7: NCEP WRF Post-Processor v2.0

Table of Contents

Introduction

The NCEP WRF Post-Processor was designed to interpolate both WRF-NMM and WRF-ARW output from their native grids to National Weather Service (NWS) standard levels (pressure, height, etc.) and standard output grids (AWIPS, Lambert Conformal, polar-stereographic, etc.) in NWS and World Meteorological Organization (WMO) GRIB format. This package also provides an option to output fields on the model's native vertical levels.

The adaptation of the original WRF Post-Processor package and User's Guide (by Mike Baldwin of NSSL/CIMMS and Hui-Ya Chuang of NCEP/EMC) was done by Lígia Bernardet (NOAA/FSL/DTC) in collaboration with Dusan Jovic (NCEP/EMC), Robert Rozumalski (COMET), Wesley Ebisuzaki (NWS/HQTR), and Louisa Nance (NCAR/DTC). The WRF Post-Processor Version 2.0 was developed by Hui-Ya Chuang (NCEP/EMC) and adapted by Meral Demirtas (NCAR/DTC).

Functionalities

The WRF Post-Processor v2.0 is compatible with WRF v2.1 and above and can be used to post-process WRF-ARW and WRF-NMM forecasts. The WRF Post-Processor can ingest WRF history files (wrfout*) in two formats: netCDF and binary. The section **"Setting up the WRF model to interface with the WRF Post-Processor"** describes how to setup the WRF model to ensure compatibility with the WRF Post-Processor.

The WRF Post-Processor is divided into two parts:

1. ***Wrfpost***
   - Interpolates the forecasts from the model's native vertical coordinate to NWS standard output levels (pressure, height, etc.) and computes mean sea level pressure. If the requested parameter is already on a model's native level, then no vertical interpolation is performed. Computes diagnostic output quantities (e.g. convective available potential energy, helicity, radar reflectivity, etc.). A list of fields that can be generated by ***wrfpost*** is shown in Table 1.
   - Outputs the results in NWS and WMO standard GRIB1 format (for GRIB documentation, see http://www.nco.ncep.noaa.gov/pmb/docs/).
   - Destaggers the WRF-ARW forecasts (see next paragraph).
   - For WRF-NMM, outputs two navigation files, ***copygb_nav.txt*** and ***copygb_hwrf.txt***. These files can be used as input for ***copygb***.
     - ***copygb_nav.txt:*** This file contains the GRID GDS of a Lambert Conformal Grid similar in domain and grid spacing to the one used to run the WRF model. The Lambert Conformal map projection works well for mid-latitudes. .
     - ***copygb_hwrf.txt***: This file contains the GRID GDS of a Latitude-Longitude Grid similar in domain and grid spacing to the one used to run the WRF model. The latitude-longitude grid works well for tropics.
2. ***Copygb***
   - Destaggers the WRF-NMM forecasts (see next paragraph).
   - Interpolates the forecasts horizontally from their native grid to a standard AWIPS or user-defined grid (for information on AWIPS grids, see http://www.emc.ncep.noaa.gov/mmb/etagrids).
   - Outputs the results in NWS and WMO standard GRIB1 format (for GRIB documentation, see http://www.nco.ncep.noaa.gov/pmb/docs/).

When posting (step 1 above) forecasts computed by the WRF-ARW, ***wrfpost*** interpolates the velocity variables to the mass points of the WRF-ARW native C-grid, which is equivalent to an A-grid representation. Since the A-grid is a regular non-staggered grid, output from ***wrfpost*** for WRF-ARW can be displayed directly by most display codes without going through ***copygb*** (step 2 above). On the other hand, no de-staggering is applied when posting WRF-NMM forecasts. Therefore, the posted WRF-NMM output is still on the staggered native E-grid and must go through ***copygb*** to be interpolated to a regular non-staggered grid.

In addition to ***wrfpost*** and ***copygb***, a utility called ***ndate*** is distributed with the WRF Post-Processor tarfile. This utility is used to format the dates of the forecasts to be posted for ingestion by the codes.

Computational Aspects and Supported Platforms

The WRF Post-Processor v2.0 has been tested on IBM and LINUX platforms. For LINUX, the Portland Group (PG) compiler has been used. Only ***wrfpost*** (step 1) is parallelized because it requires several 3-dimensional arrays (the model's history variables) for the computations. When running ***wrfpost*** on more than one processor, the last processor will be designated as an I/O node, while the rest of the processors are designated as computational nodes. For example, if three processors are requested to run the ***wrfpost***, only the first two processors will be used for computation, while the third processor will be used to write output to GRIB files.

One limitation of the current version of the WRF Post-Processor is that only one forecast time can be processed per execution.

Tarfile Overview and Directory Structure

Expanding "***wrfpost_v2.0.tar***" creates a main directory ***wrfpostproc_v2*** and five subdirectories:

- **sorc**: contains source codes for ***wrfpost***, ***ndate***, and ***copygb***.
- **scripts**: contains sample running scripts
  - **run_wrfpost**: run ***wrfpost*** and ***copygb***.
  - **run_wrfpostandgempak**: run ***wrfpost***, ***copygb***, and GEMPAK to plot various fields.
  - **run_wrfpostandgrads**: run ***wrfpost***, ***copygb***, and GrADS to plot various fields.
- **lib:** contains source code subdirectories for the WRF Post-Processor libraries and is the directory where the WRF Post-Processor compiled libraries will reside.
  - **w3lib**: Library for coding and decoding data in GRIB format
    Note: The version of this library included in this package is Endian-independent and can be used on LINUX and IBM systems.
  - **iplib**: General interpolation library (see ***lib/iplib/iplib.doc***)
  - **splib**: Spectral transform library (see ***lib/splib/splib.doc***)
  - **wrfmpi_stubs**: Contains some ***C*** and ***FORTRAN*** codes to genereate ***libmpi.a*** library. It supports MPI implementation for LINUX applications.
- **parm**: Contains the parameter files, which can be modified by the user to control how the post processing is performed.
- **exec**: Location of executables after compilation.

Required Compilers, Scripting Languages and Libraries

The WRF Post-Processor is written mainly in FORTRAN with some additional files in C. Software requirements are:

- FORTRAN 90 or 95
- C compiler

Two libraries that are not part of the wrfpost_v2.0 distribution are needed to build the WRF Post-Processor:

WRF IO API library: The WRF IO API library is included with the WRF model tar file. The WRF Post-Processor must be linked to WRF v2.1 or later.
NETCDF library: The NETCDF library should be pre-installed on the system

How to Install WRF Post-Processor

The WRF Post-Processor package can be downloaded from:
http://www.dtcenter.org/wrf-nmm/users

A summary of the installation procedure is provided below.

Installation steps:

- Download the package tar file, and type '***tar –zxvf wrfpost_v2.0.tar.gz***' to unzip and untar the file. This command will create a directory called ***wrfpostproc_v2*** (see description above).
- cd to ***wrfpostproc_v2*** directory
- Type ***configure***, and provide the required info. For example:

./configure

```
Please select from the following supported platforms.
1. LINUX (PC)
2. AIX (IBM)

Enter selection [1-2]:
1
Enter your NETCDF path:
/usr/local/netcdf-pgi
Enter your WRF model source code path:
/home/user/WRFV2
"YOU HAVE SELECTED YOUR PLATFORM TO BE:" LINUX
```

- To modify the default compiler options, edit the appropriate platform specific makefile (i.e. makefile_linux or makefile_ibm) and repeat the configure process.
- For LINUX systems, check ***sorc/wrfpost*** for a link to ***lib/wrfmpi_stubs/mpif.h***. If this link is not there, type the following:

cd sorc/wrfpost
ln –fs ../../lib/wrfmpi_stubs/mpif.h

- From the ***wrfpostproc_v2*** directory, type ***make***. This command should create three WRF Post-Processor libraries (***libmpi.a***, ***libsp.a***, ***libip.a***, and ***libw3.a***) and three WRF Post-Processor executables (***wrfpost.exe***, ***ndate.exe***, and ***copygb.exe***).

Note: The ***makefile*** included in the tar file currently only contains the setup for single processor compilation of ***wrfpost*** for LINUX. Those users wanting to implement the parallel capability of this portion of the package will need to modify the compile options for ***wrfpost*** in the ***makefile***.

Control File Overview

The user interacts with ***wrfpost*** through the control file, ***wrf_cntrl.parm***. The control file is composed of a header and a body. The header allows the user to specify the name of the output file, whereas the body allows the user to select which fields to post, the type of level, and which levels to post the fields to.

The header of the ***wrf_cntrl.parm*** file contains the following variables:

- **KGTYPE**: defines output grid type, which should always be 255.
- **IMDLTY:** identifies the process ID for AWIPS.
- **DATSET**: defines the prefix used for the output file name. Currently set to "***WRFPRS"***.

The body of the ***wrf_cntrl.parm*** file is composed of a series of line pairs similar to the following:

```
(PRESS ON MDL SFCS   ) SCAL=( 3.0)
L=(11000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
00000 00000 00000)
```

The top line specifies the variable to be posted, the type of level to use for posting, and the degree of accuracy to be retained in the GRIB output. The second line specifies the levels on which the variable is to be posted. A list of all possible output fields for ***wrfpost*** is provided in Table 1. This table provides the full name of the variable in the first column and an abbreviated name in the second column. The abbreviated names are used in the control file. Note that the variable names also contain the type of level on which they are output. For instance, temperature is available on "model surface" and "pressure surface".

Controlling which variables *wrfpost* outputs

To output a field, the body of the control file needs to contain an entry for the appropriate variable and output for this variable must be turned on for at least one level (see discussion below). If an entry for a particular field is not yet available in the control file, two lines may be added to the control file with the appropriate entries for that field.

Controlling which levels *wrfpost* outputs

The second line of each pair determines which levels ***wrfpost*** will output. Output on a given level is turned off by a "0" or turned on by a "1".

- For isobaric output, 47 levels are possible, from 50 to 1000 hPa, every 25 hPa.
- For model-level output, all model levels are possible, from the highest to the lowest.
- For soil layers, the levels are 0-10 cm, 10-40 cm, 40-100 cm, and 100-200 cm.
- For PBL layer averages, the levels correspond to 6 layers with a thickness of 30 hPa each.
- For flight level, the levels are 914.E0,1524.E0,1829.E0, 2134.E0, 2743.E0, 3658.E0, and 6000.E0 in m
- For AGL RADAR Reflectivity, the levels are 4000 and 1000 m.
- For surface or shelter-level output, only the first position of the line can be turned on.

For example, the sample control file ***parm/wrf_cntrl.parm*** has the following entry for surface dew point temperature:

```
(SURFACE DEWPOINT    ) SCAL=(-4.0)
L=(00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
00000 00000)
```

Based on this entry, surface dew point temperature will not be output by ***wrfpost***. To add this field to the output, modify the entry to read:

```
(SURFACE DEWPOINT    ) SCAL=(-4.0)
L=(10000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000
00000 00000)
```

Running the WRF Post-Processor

Three scripts for running the WRF Post-Processor package are included in the tar file:
run_wrfpost (runs ***wrfpost*** and ***copygb***)
run_wrfpostandgrads (runs ***wrfpost***, ***copygb*** and GrADS)
run_wrfpostandgempak (runs ***wrfpost*** and ***copygb*** and GEMPAK)

The WRF Post-Processor should be run with one of these scripts. After selecting one of the scripts, it is necessary to edit the script to customize the following variables:

- ***TOP_DIR***: top level directory for source codes (WRF-POSTPROC and WRFV2).
- ***DOMAINPATH***: top level directory of WRF model run.
  NOTE: The scripts are configured such that ***wrfpost*** expects the WRF history files (***wrfout****** files) to be in subdirectory ***wrfprd***, the *wrf_cntrl_parm* file to be in the subdirectory ***parm*** and the Post-Processor working directory to be a subdirectory ***postprd*** under ***DOMAINPATH***.

- ***startdate***: YYYYMMDDHH of forecast cycle to be post-processed.
- ***fhr***: first forecast hour to be post-processed.
- ***lastfhr***: last forecast hour to be post-processed.
- ***incrementhr***: increment (in hours) between forecast files.
- ***gridno***: grid to interpolate WRF model output to (this variable is only used if the forecast will be interpolated onto a standard AWIPS grid).
- ***Copygb*** command: The scripts are setup such that ***copygb*** can be run in 3 ways. Uncomment the preferable one.
    - ***copygb*** is run with a pre-defined AWIPS grid (variable ***gridno***, see above).
    - ***copygb*** ingests a kgds definition on the command line.
    - ***copygb*** ingests the contents of file ***copygb_gridnav.txt*** or ***copygb_hwrf.txt*** through variable ***nav***.

- ***GEMEXEC:*** Defines a path name for GEMPAK related executables, if ***run_wrfpostandgempak*** is used.

Before running any of the above listed scripts, perform the following instructions:

cd to your ***DOMAINPATH*** directory.

Make a directory to put the WRF POST-PROCESSOR results.

mkdir postprd

Make a directory to put a copy of ***wrf_cntrl.parm*** file.

mkdir parm
cd parm

Copy over default ***wrfpostproc_v2/parm/wrf_cntrl.parm*** to customize ***wrfpost***.

Edit ***wrf_cntrl.parm*** file to reflect the fields and levels the user wants ***wrfpost*** to output.

Once these directories are set up and the edits outlined above are completed, the scripts can be run interactively from the scripts directory by simply typing the script name on the command line.

Overview of the steps in script *run_wrfpost* to run the WRF Post-Processor

Note: It is recommended that the user refer to the script while reading this overview.
1. Set up environmental variables ***TOP_DIR*** and ***DOMAINPATH***.
2. Define location of the post-processor executables.
3. ***cd*** to working directory for post-processor (sample script assumes the working directory is the subdirectory "***postprd***" of ***DOMAINPATH***).

4. Link control file ***../parm/wrf_control.parm*** and microphysical table ***${WRFPATH}/test/nmm_real/ETAMP_DATA*** to working directory.
5. Set up time parameters of forecast to be processed.
6. Set up grid to be posted to (see note below).
7. Set up loop for forecast hours to be post-processed.
8. Create namelist ***itag*** that will be read in by ***wrfpost.exe*** from stdin (unit 5). This namelist contains 4 lines:
   i. Name of the WRF output file to be posted.
   ii. Format of WRF model output (netcdf or binary).
   iii. Forecast valid time (not model start time) in WRF format.
   iv. Model name (NMM or NCAR, where NCAR refers to the WRF-ARW core).

9. Run ***wrfpost*** and check for errors. The execution command in the distributed scripts is for a single processor ***wrfpost.exe < itag > outpost***. To run ***wrfpost*** on multiple processors, the command line should be:
   - LINUX-MPI systems: ***mpirun -np N wrfpost.exe < itag > outpost***
   - IBM: ***poe wrfpost.exe < itag > outpost***

10. Run ***copygb*** and check for errors.

    copygb.exe –xg"grid [kgds]" input_file output_file

    where ***grid*** refers to the output grid to which the native forecast is being interpolated.

    The output grid can be specified in three ways:

    i. As the grid id of a pre-defined AWIPS grid:

       copygb.exe -g${gridno} -x input_file output_file

       For example, using grid 218:

       copygb.exe -xg"218" WRFPRS${fhr}.tm00 wrfnmm${fhr}.tm00

    ii. As a user defined standard grid, such as for grid 255:

       copygb.exe –xg"255" input_file output_file

       where the user defined grid is specified by a full set of kgds parameters determining a GRIB GDS (grid description section) in the ***W3FI63*** format. Details on how to specify the kgds parameters are documented in file ***lib/w3lib/w3fi71.f***. For example:

       copygb.exe -xg" 255 3 109 91 37719 -77645 8 -71000 10433 9966 0 64 42000 42000" input_file output_file

    iii. Specifiying output grid as a file: When WRF-NMM output in netCDF format is processed by ***wrfpost***, two text files ***copygb_gridnav.txt*** and ***copygb_hwrf.txt*** are created. These files contain the GRID GDS of a Lambert Conformal Grid (file ***copygb_gridnav.txt***) or lat/lon grid

(***copygb_hwrf.txt***) similar in domain and grid spacing to the one used to run the WRF-NMM model. The contents of one of these files are read into variable ***nav*** and can be used as input to ***copygb.exe***. For example:
copygb.exe -xg"$nav" input_file output_file

Such as using "***copygb_gridnav.txt"*** for an application:
read nav < 'copygb_gridnav.txt'
export nav
copygb.exe -xg"${nav}" WRFPRS${fhr}.tm00 wrfnmm${fhr}.tm00

If scripts ***run_wrfpostandgrads*** or ***run_wrfpostandgempak*** are used, additional steps are taken to create image files (see **Visualization** section below).

Upon a successful run, ***wrfpost*** and ***copygb*** will generate output files ***WRFPRShh.tm00*** and ***wrfnmmhh.tm00***, respectively, in the post-processor working directory, where ***hh*** denotes the forecast hour. In addition, script ***run_wrfpostandgrads*** will produce a suite of gif images named ***variablehh_GrADS.gif***, and script ***run_wrfpostandgempak*** will produce a suite of gif images named ***variablehh.gif***. An additional file containing native grid navigation information (***griddef.out***), which is currently not used, will be produced.

It should be noted that ***copygb*** is a flexible program that can accept several command line options specifying details of how the horizontal interpolation from the native grid to the output grid should be performed. Complete documentation of ***copygb*** can be found in ***wrfpostproc/sorc/copygb.doc***.

Fields produced by the WRF Post-Processor

Table 1 lists basic and derived fields that are currently produced by ***wrfpost***. The abbreviated names listed in the second column describe how the fields should be entered in the control file (***wrf_cntrl.parm***).

Table 1: *Fields produced by **wrfpost** (column 1), abbreviated names used in **wrfpost** control file (column 2), corresponding GRIB identification number for the field (column 3), and corresponding GRIB identification number for the vertical coordinate (column 4).*

Field name	Name in control file	Grib ID	Vertical level
Radar reflectivity on model surface	RADAR REFL MDL SFCS	211	109
Pressure on model surface	PRESS ON MDL SFCS	1	109
Height on model surface	HEIGHT ON MDL SFCS	7	109
Temperature on model surface	TEMP ON MDL SFCS	11	109
Potential temperature on model surface	POT TEMP ON MDL SFCS	13	109
Dew point temperature on model surface	DWPT TEMP ON MDL SFC	17	109

Specific humidity on model surface	SPEC HUM ON MDL SFCS	51	109
Relative humidity on model surface	REL HUM ON MDL SFCS	52	109
Moisture convergence on model surface	MST CNVG ON MDL SFCS	135	109
U component wind on model surface	U WIND ON MDL SFCS	33	109
V component wind on model surface	V WIND ON MDL SFCS	34	109
Cloud water on model surface	CLD WTR ON MDL SFCS	153	109
Cloud ice on model surface	CLD ICE ON MDL SFCS	58	109
Rain on model surface	RAIN ON MDL SFCS	170	109
Snow on model surface	SNOW ON MDL SFCS	171	109
Cloud fraction on model surface	CLD FRAC ON MDL SFCS	71	109
Omega on model surface	OMEGA ON MDL SFCS	39	109
Absolute vorticity on model surface	ABS VORT ON MDL SFCS	41	109
Geostrophic streamfunction on model surface	STRMFUNC ON MDL SFCS	35	109
Turbulent kinetic energy on model surface	TRBLNT KE ON MDL SFC	158	109
Richardson number on model surface	RCHDSN NO ON MDL SFC	254	109
Master length scale on model surface	MASTER LENGTH SCALE	226	109
Asymptotic length scale on model surface	ASYMPT MSTR LEN SCL	227	109
Radar reflectivity on pressure surface	RADAR REFL ON P SFCS	211	100
Height on pressure surface	HEIGHT OF PRESS SFCS	7	100
Temperature on pressure surface	TEMP ON PRESS SFCS	11	100
Potential temperature on pressure surface	POT TEMP ON P SFCS	13	100
Dew point temperature on pressure surface	DWPT TEMP ON P SFCS	17	100
Specific humidity on pressure surface	SPEC HUM ON P SFCS	51	100
Relative humidity on pressure surface	REL HUMID ON P SFCS	52	100
Moisture convergence on pressure surface	MST CNVG ON P SFCS	135	100
U component wind on pressure surface	U WIND ON PRESS SFCS	33	100
V component wind on pressure	V WIND ON PRESS SFCS	34	100

surface			
Omega on pressure surface	OMEGA ON PRESS SFCS	39	100
Absolute vorticity on pressure surface	ABS VORT ON P SFCS	41	100
Geostrophic streamfunction on pressure surface	STRMFUNC ON P SFCS	35	100
Turbulent kinetic energy on pressure surface	TRBLNT KE ON P SFCS	158	100
Cloud water on pressure surface	CLOUD WATR ON P SFCS	153	100
Cloud ice on pressure surface	CLOUD ICE ON P SFCS	58	100
Rain on pressure surface	RAIN ON P SFCS	170	100
Snow water on pressure surface	SNOW ON P SFCS	171	100
Total condensate on pressure surface	CONDENSATE ON P SFCS	135	100
Mesinger (Membrane) sea level pressure	MESINGER MEAN SLP	130	102
Shuell sea level pressure	SHUELL MEAN SLP	2	102
2 M pressure	SHELTER PRESSURE	1	105
2 M temperature	SHELTER TEMPERATURE	11	105
2 M specific humidity	SHELTER SPEC HUMID	51	105
2 M dew point temperature	SHELTER DEWPOINT	17	105
2 M RH	SHELTER REL HUMID	52	105
10 M u component wind	U WIND AT ANEMOM HT	33	105
10 M v component wind	V WIND AT ANEMOM HT	34	105
10 M potential temperature	POT TEMP AT 10 M	13	105
10 M specific humidity	SPEC HUM AT 10 M	51	105
Surface pressure	SURFACE PRESSURE	1	1
Terrain height	SURFACE HEIGHT	7	1
Skin potential temperature	SURFACE POT TEMP	13	1
Skin specific humidity	SURFACE SPEC HUMID	51	1
Skin dew point temperature	SURFACE DEWPOINT	17	1
Skin Relative humidity	SURFACE REL HUMID	52	1
Skin temperature	SFC (SKIN) TEMPRATUR	11	1
Soil temperature at the bottom of soil layers	BOTTOM SOIL TEMP	85	111
Soil temperature in between each of soil layers	SOIL TEMPERATURE	85	112
Soil moisture in between each of soil layers	SOIL MOISTURE	144	112
Snow water equivalent	SNOW WATER EQUIVALNT	65	1
Snow cover in percentage	PERCENT SNOW COVER	238	1
Heat exchange coeff at surface	SFC EXCHANGE COEF	208	1
Vegetation cover	GREEN VEG COVER	87	1

Soil moisture availability	SOIL MOISTURE AVAIL	207	112
Ground heat flux - instantaneous	INST GROUND HEAT FLX	155	1
Lifted index—surface based	LIFTED INDEX—SURFCE	131	101
Lifted index—best	LIFTED INDEX—BEST	132	116
Lifted index—from boundary layer	LIFTED INDEX—BNDLYR	24	116
CAPE	CNVCT AVBL POT ENRGY	157	1
CIN	CNVCT INHIBITION	156	1
Column integrated precipitable water	PRECIPITABLE WATER	54	200
Column integrated cloud water	TOTAL COLUMN CLD WTR	136	200
Column integrated cloud ice	TOTAL COLUMN CLD ICE	137	200
Column integrated rain	TOTAL COLUMN RAIN	138	200
Column integrated snow	TOTAL COLUMN SNOW	139	200
Column integrated total condensate	TOTAL COL CONDENSATE	140	200
Helicity	STORM REL HELICITY	190	106
U component storm motion	U COMP STORM MOTION	196	106
V component storm motion	V COMP STORM MOTION	197	106
Accumulated total precipitation	ACM TOTAL PRECIP	61	1
Accumulated convective precipitation	ACM CONVCTIVE PRECIP	63	1
Accumulated grid-scale precipitation	ACM GRD SCALE PRECIP	62	1
Accumulated snowfall	ACM SNOWFALL	65	1
Accumulated total snow melt	ACM SNOW TOTAL MELT	99	1
Precipitation type (4 types) - instantaneous	INSTANT PRECIP TYPE	140	1
Precipitation rate - instantaneous	INSTANT PRECIP RATE	59	1
Composite radar reflectivity	COMPOSITE RADAR REFL	212	200
Low level cloud fraction	LOW CLOUD FRACTION	73	214
Mid level cloud fraction	MID CLOUD FRACTION	74	224
High level cloud fraction	HIGH CLOUD FRACTION	75	234
Total cloud fraction	TOTAL CLD FRACTION	71	200
Time-averaged total cloud fraction	AVG TOTAL CLD FRAC	71	200
Time-averaged stratospheric cloud fraction	AVG STRAT CLD FRAC	213	200
Time-averaged convective cloud fraction	AVG CNVCT CLD FRAC	72	200

Cloud bottom pressure	CLOUD BOT PRESSURE	1	2
Cloud top pressure	CLOUD TOP PRESSURE	1	3
Cloud bottom height (above MSL)	CLOUD BOTTOM HEIGHT	7	2
Cloud top height (above MSL)	CLOUD TOP HEIGHT	7	3
Convective cloud bottom pressure	CONV CLOUD BOT PRESS	1	242
Convective cloud top pressure	CONV CLOUD TOP PRESS	1	243
Shallow convective cloud bottom pressure	SHAL CU CLD BOT PRES	1	248
Shallow convective cloud top pressure	SHAL CU CLD TOP PRES	1	249
Deep convective cloud bottom pressure	DEEP CU CLD BOT PRES	1	251
Deep convective cloud top pressure	DEEP CU CLD TOP PRES	1	252
Grid scale cloud bottom pressure	GRID CLOUD BOT PRESS	1	206
Grid scale cloud top pressure	GRID CLOUD TOP PRESS	1	207
Convective cloud fraction	CONV CLOUD FRACTION	72	200
Convective cloud efficiency	CU CLOUD EFFICIENCY	134	200
Above-ground height of LCL	LCL AGL HEIGHT	7	5
Pressure of LCL	LCL PRESSURE	1	5
Cloud top temperature	CLOUD TOP TEMPS	11	3
Temperature tendency from radiative fluxes	RADFLX CNVG TMP TNDY	216	109
Temperature tendency from shortwave radiative flux	SW RAD TEMP TNDY	250	109
Temperature tendency from longwave radiative flux	LW RAD TEMP TNDY	251	109
Outgoing surface shortwave radiation - instantaneous	INSTN OUT SFC SW RAD	211	1
Outgoing surface longwave radiation - instantaneous	INSTN OUT SFC LW RAD	212	1
Incoming surface shortwave radiation - time-averaged	AVE INCMG SFC SW RAD	204	1
Incoming surface longwave radiation - time-averaged	AVE INCMG SFC LW RAD	205	1
Outgoing surface shortwave radiation - time-averaged	AVE OUTGO SFC SW RAD	211	1
Outgoing surface longwave radiation - time-averaged	AVE OUTGO SFC LW RAD	212	1
Outgoing model top shortwave radiation - time-averaged	AVE OUTGO TOA SW RAD	211	8

Outgoing model top longwave radiation - time-averaged	AVE OUTGO TOA LW RAD	212	8
Incoming surface shortwave radiation - instantaneous	INSTN INC SFC SW RAD	204	1
Incoming surface longwave radiation - instantaneous	INSTN INC SFC LW RAD	205	1
Roughness length	ROUGHNESS LENGTH	83	1
Friction velocity	FRICTION VELOCITY	253	1
Surface drag coefficient	SFC DRAG COEFFICIENT	252	1
Surface u wind stress	SFC U WIND STRESS	124	1
Surface v wind stress	SFC V WIND STRESS	125	1
Surface sensible heat flux - time-averaged	AVE SFC SENHEAT FX	122	1
Ground heat flux - time-averaged	AVE GROUND HEAT FX	155	1
Surface latent heat flux - time-averaged	AVE SFC LATHEAT FX	121	1
Surface momentum flux - time-averaged	AVE SFC MOMENTUM FX	172	1
Accumulated surface evaporation	ACC SFC EVAPORATION	57	1
Surface sensible heat flux - instantaneous	INST SFC SENHEAT FX	122	1
Surface latent heat flux - instantaneous	INST SFC LATHEAT FX	121	1
Latitude	LATITUDE	176	1
Longitude	LONGITUDE	177	1
Land sea mask (land=1, sea=0)	LAND SEA MASK	81	1
Sea ice mask	SEA ICE MASK	91	1
Surface midday albedo	SFC MIDDAY ALBEDO	84	1
Sea surface temperature	SEA SFC TEMPERATURE	80	1
Press at tropopause	PRESS AT TROPOPAUSE	1	7
Temperature at tropopause	TEMP AT TROPOPAUSE	11	7
Potential temperature at tropopause	POTENTL TEMP AT TROP	13	7
U wind at tropopause	U WIND AT TROPOPAUSE	33	7
V wind at tropopause	V WIND AT TROPOPAUSE	34	7
Wind shear at tropopause	SHEAR AT TROPOPAUSE	136	7
Height at tropopause	HEIGHT AT TROPOPAUSE	7	7
Temperature at flight levels	TEMP AT FD HEIGHTS	11	103
U wind at flight levels	U WIND AT FD HEIGHTS	33	103
V wind at flight levels	V WIND AT FD HEIGHTS	34	103
Freezing level height (above mean sea level)	HEIGHT OF FRZ LVL	7	4

Freezing level RH	REL HUMID AT FRZ LVL	52	4
Highest freezing level height	HIGHEST FREEZE LVL	7	204
Pressure in boundary layer (30 mb mean)	PRESS IN BNDRY LYR	1	116
Temperature in boundary layer (30 mb mean)	TEMP IN BNDRY LYR	11	116
Potential temperature in boundary layers (30 mb mean)	POT TMP IN BNDRY LYR	13	116
Dew point temperature in boundary layer (30 mb mean)	DWPT IN BNDRY LYR	17	116
Specific humidity in boundary layer (30 mb mean)	SPC HUM IN BNDRY LYR	51	116
RH in boundary layer (30 mb mean)	REL HUM IN BNDRY LYR	52	116
Moisture convergence in boundary layer (30 mb mean)	MST CNV IN BNDRY LYR	135	116
Precipitable water in boundary layer (30 mb mean)	P WATER IN BNDRY LYR	54	116
U wind in boundary layer (30 mb mean)	U WIND IN BNDRY LYR	33	116
V wind in boundary layer (30 mb mean)	V WIND IN BNDRY LYR	34	116
Omega in boundary layer (30 mb mean)	OMEGA IN BNDRY LYR	39	116
Visibility	VISIBILITY	20	1
Vegetation type	VEGETATION TYPE	225	1
Soil type	SOIL TYPE	224	1
Canopy conductance	CANOPY CONDUCTANCE	181	1
PBL height	PBL HEIGHT	221	1
Slope type	SLOPE TYPE	222	1
Snow depth	SNOW DEPTH	66	1
Liquid soil moisture	LIQUID SOIL MOISTURE	160	112
Snow free albedo	SNOW FREE ALBEDO	170	1
Maximum snow albedo	MAXIMUM SNOW ALBEDO	159	1
Canopy water evaporation	CANOPY WATER EVAP	200	1
Direct soil evaporation	DIRECT SOIL EVAP	199	1
Plant transpiration	PLANT TRANSPIRATION	210	1
Snow sublimation	SNOW SUBLIMATION	198	1
Air dry soil moisture	AIR DRY SOIL MOIST	231	1
Soil moist porosity	SOIL MOIST POROSITY	240	1
Minimum stomatal resistance	MIN STOMATAL RESIST	203	1
Number of root layers	NO OF ROOT LAYERS	171	1
Soil moist wilting point	SOIL MOIST WILT PT	219	1

Soil moist reference	SOIL MOIST REFERENCE	230	1
Canopy conductance - solar component	CANOPY COND SOLAR	246	1
Canopy conductance - temperature component	CANOPY COND TEMP	247	1
Canopy conductance - humidity component	CANOPY COND HUMID	248	1
Canopy conductance - soil component	CANOPY COND SOILM	249	1
Potential evaporation	POTENTIAL EVAP	145	1
Heat diffusivity on sigma surface	DIFFUSION H RATE S S	182	107
Surface wind gust	SFC WIND GUST	180	1
Convective precipitation rate	CONV PRECIP RATE	214	1
Radar reflectivity at certain above ground heights	RADAR REFL AGL	211	105

Setting up the WRF model to interface with the WRF Post-Processor

The ***wrfpost*** program is currently set up to read a large number of fields from the WRF model history files. This configuration stems from NCEP's need to generate all of its required operational products. Lists of the fields that are currently read in by ***wrfpost*** for WRF-NMM and WRF-ARW are provided in Tables 2 and 3, respectively. This program is configured such that is will run successfully if an expected input field is missing from the WRF history file as long as this field is not required to produce a requested output field. If the pre-requisites for a requested output field are missing from the WRF history file, ***wrfpost*** will abort at run time. For example, if isobaric state fields are requested, but the pressure fields on model interfaces (PINT for WRF-NMM and PB and P for WRF-ARW) are not available in the history file, ***wrfpost*** will abort at run time. The fields written to the WRF history file are controlled by the settings in the Registry file (see the ***Registry.EM*** and ***Registry.NMM*** files in the ***Registry*** subdirectory of the main ***WRFV2*** directory). Note that it is necessary to re-compile the WRF model source code after modifying the appropriate Registry file.

Table 2.*List of all possible fields read in by* ***wrfpost*** *for the WRF-NMM:*

T	SFCEXC	NRDSW
U	VEGFRC	ARDSW
V	ACSNOW	ALWIN
Q	ACSNOM	ALWOUT
CWM	CMC	NRDLW
F_ICE	SST	ARDLW
F_RAIN	EXCH_H	ALWTOA
F_RIMEF	EL_MYJ	ASWTOA
W	THZ0	TGROUND

PINT	QZ0	SOILTB
PT	UZ0	TWBS
PDTOP	VZ0	SFCSHX
FIS	QS	NSRFC
SMC	Z0	ASRFC
SH2O	PBLH	QWBS
STC	USTAR	SFCLHX
CFRACH	AKHS_OUT	GRNFLX
CFRACL	AKMS_OUT	SUBSHX
CFRACM	THS	POTEVP
SLDPTH	PREC	WEASD
U10	CUPREC	SNO
V10	ACPREC	SI
TH10	CUPPT	PCTSNO
Q10	LSPA	IVGTYP
TSHLTR	CLDEFI	ISLTYP
QSHLTR	HTOP	ISLOPE
PSHLTR	HBOT	SM
SMSTAV	HTOPD	SICE
SMSTOT	HBOTD	ALBEDO
ACFRCV	HTOPS	ALBASE
ACFRST	HBOTS	GLAT
RLWTT	SR	XLONG
RSWTT	RSWIN	GLON
AVRAIN	CZEN	DX_NMM
AVCNVC	CZMEAN	NPHS0
TCUCN	RSWOUT	NCLOD
TRAIN	RLWIN	NPREC
NCFRCV	SIGT4	NHEAT
NCFRST	RADOT	
SFROFF	ASWIN	
UDROFF	ASWOUT	
SFCEVP		

Table 3.*List of all possible fields read in by **wrfpost** for the WRF-ARW:*

T	MUB	SFROFF
U	P_TOP	UDROFF
V	PHB	SFCEVP
QVAPOR	PH	SFCEXC
QCLOUD	SMOIS	VEGFRA
QICE	TSLB	ACSNOW
QRAIN	CLDFRA	ACSNOM
QSNOW	U10	CANWAT
QGRAUP	V10	SST

W	TH2	THZ0
PB	Q2	QZ0
P	SMSTAV	UZ0
MU	SMSTOT	VZ0
QSFC	HGT	ISLTYP
Z0	ALBEDO	ISLOPE
UST	GSW	XLAND
AKHS	GLW	XLAT
AKMS	TMN	XLONG
TSK	HFX	MAPFAC_M
RAINC	LH	STEPBL
RAINNC	GRDFLX	HTOP
RAINCV	SNOW	HBOT
RAINNCV	SNOWC	

Note: For WRF-ARW, the accumulated precipitation fields (***RAINC*** and ***RAINNC***) are run total accumulations, whereas the WRF-NMM the accumulated precipitation fields (***CUPREC*** and ***ACPREC***) are zeroed every 6 hours. Hence, this field in the ***wrfout*** file represents an accumulation over the time period ***6*INT[(fhr-Σ)/6]*** to ***fhr***, where ***fhr*** represents the forecast hour and ***Σ*** is a small number. The ***grib*** file output by ***wrfpost*** and by ***copygb*** contains fields with the name of the accumulation period.

Visualization

GEMPAK

The GEMPAK utility ***nagrib*** is able to decode GRIB files whose navigation is on any non-staggered grid. Hence, GEMPAK is able to decode GRIB files generated by the WRF Post-Processing package and plot horizontal fields or vertical cross sections.
A sample script named ***run_wrfpostandgempak***, which is included in the ***scripts*** directory of the tar file, can be used to run ***wrfpost***, ***copygb***, and plot the following fields using GEMPAK:

- ***Sfcmap.gif:*** mean SLP and 6 hourly precipitation
- ***PrecipType.gif:*** precipitation type (just snow and rain)
- ***850mbRH.gif:*** 850 mb relative humidity
- ***850mbTempandWind.gif:*** 850 mb temperature and wind vectors
- ***500mbHandVort.gif:*** 500 mb geopotential height and vorticity
- ***250mbWindandH.gif:*** 250 mb wind speed isotacs and geopotential height

This script can be modified to customize fields for output. GEMPAK has an online users guide at http://my.unidata.ucar.edu/content/software/gempak/index.html

GrADS

The GrADS utilities ***grib2ctl.pl*** and ***gribmap*** are able to decode GRIB files whose navigation is on any non-staggered grid. These utilities and instructions on how to use them to generate GrADS control files are available from:
http://www.cpc.ncep.noaa.gov/products/wesley/grib2ctl.html.
The GrADS package is available from: http://grads.iges.org/grads/grads.html.
GrADS has an online Users' Guide at: http://grads.iges.org/grads/gadoc/.
A list of basic commands for GrADS can be found at:
http://www.egs.uct.ac.za/workshops/mm5/sheet.html.

A sample script named ***run_wrfpostandgrads***, which is included in the ***scripts*** directory of the WRF Post-Processing package, can be used to run ***wrfpost***, ***copygb***, and plot the following fields using GrADS:

- ***Sfcmaphh.gif***: mean SLP and 6-hour accumulated precipitation.
- ***850mbRH.gif***: 850 mb relative humidity
- ***850mbTempandWind.gif***: 850 mb temperature and wind vectors
- ***500mbHandVort.gif***: 500 mb geopotential heights and absolute vorticity
- ***250mbWindandH.gif***: 250 mb wind speed isotacs and geopotential heights

In order to use script ***run_wrfpostandgrads***, it is necessary to:

1. Set environmental variable ***GADDIR*** to the path of the GrADS fonts and auxiliary files. For example,
   setenv GADDIR /usr/local/grads/data

2. Add the location of the GrADS executables to the ***PATH***. For example
   setenv PATH /usr/local/grads/bin:$PATH

3. Link script ***cbar.gs*** to the post-processor working directory. (This scripts is provided in WRF-POSTPROC package, and ***run_wrfpostandgrads*** script makes a link from "scripts" directory to the run directory.) To generate the plots above, GrADS script cbar.gs is invoked. This script can also be obtained from the GrADS library of scripts at http://grads.iges.org/grads/gadoc/library.html.