



ARL-MR-0988 • SEP 2018



# Automated Postprocessing for the Weather Running Estimate-Nowcast (WRE-N) Model

by Leelinda P Dawson and John W Raby

Approved for public release; distribution unlimited.

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



# **Automated Postprocessing for the Weather Running Estimate-Nowcast (WRE-N) Model**

**by Leelinda P Dawson and John W Raby**  
*Computational and Information Sciences Directorate, ARL*

**REPORT DOCUMENTATION PAGE**

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> September 2018		<b>2. REPORT TYPE</b> Memorandum Report		<b>3. DATES COVERED (From - To)</b> October 2017–July 2018	
<b>4. TITLE AND SUBTITLE</b> Automated Postprocessing for the Weather Running Estimate-Nowcast (WRE-N) Model				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Leelinda P Dawson and John W Raby				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> US Army Research Laboratory ATTN: RDRL-CIE-M 2800 Powder Mill Road Adelphi, MD 20783-1138				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> ARL-MR-0988	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Approved for public release; distribution unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> Postprocessing is often required for the forecast data output produced by the US Army Research Laboratory's Weather Running Estimate-Nowcast (WRE-N) model as a precursor for various model assessment and verification processes that are used to improve the performance and accuracy of WRE-N forecasts. This report documents the design and implementation of the automation of WRE-N postprocessing using the Unified Post Processor tool to generate hourly GRIdded Binary files. This process allows multiple user configurations as well as produces a controlled input and output data structure that could be used during the evaluation of WRE-N model improvements and data analyses.					
<b>15. SUBJECT TERMS</b> nowcast model assessment, weather research and forecasting, WRF, Weather Running Estimate-Nowcast, WRE-N, Unified Post Processor, UPP					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b> UU	<b>18. NUMBER OF PAGES</b> 22	<b>19a. NAME OF RESPONSIBLE PERSON</b> Leelinda P Dawson
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			<b>19b. TELEPHONE NUMBER (include area code)</b> 301-394-5636

## Contents

---

---

<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>iv</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Development Environment</b>	<b>2</b>
<b>3. Script Requirements and Description</b>	<b>2</b>
<b>4. Data Structure</b>	<b>4</b>
<b>5. Configuration and Template Files</b>	<b>5</b>
<b>6. Forecast Data</b>	<b>7</b>
<b>7. Ensemble-Stat Configuration Option</b>	<b>8</b>
<b>8. Output Results</b>	<b>9</b>
<b>9. Log File</b>	<b>11</b>
<b>10. Conclusion</b>	<b>13</b>
<b>11. References</b>	<b>14</b>
<b>List of Symbols, Abbreviations, and Acronyms</b>	<b>15</b>
<b>Distribution List</b>	<b>16</b>

## List of Figures

---

Fig. 1	Standard data structure of the automated UPP script.....	4
Fig. 2	Example configuration file, <i>upp_auto_config</i> , for the automated UPP script .....	5
Fig. 3	Automated UPP script's data structure based on the example configuration file.....	6
Fig. 4	Two example ensemble member list files .....	8
Fig. 5	Example GRIB file depicting 2-m above ground level temperature in Celsius .....	10
Fig. 6	Example log file from the automated UPP script .....	12

## List of Tables

---

Table 1	Automated UPP script's configuration parameters in <i>upp_auto_config</i> .....	3
---------	---	---

## 1. Introduction

---

The Weather Running Estimate–Nowcast (WRE-N) model<sup>1</sup> was designed by US Army Research Laboratory (ARL) modelers to address the Army’s need to produce high-resolution weather forecasts and improve model forecast accuracy for mission planning and decision making by field deployed units. The core of WRE-N is the Weather Research and Forecasting (WRF)<sup>2</sup> model, which is an open-source model developed by the National Center for Atmospheric Research (NCAR), that produces global- and regional-scale forecasts. After a WRE-N model run has been successfully completed, the WRE-N forecast output data often require postprocessing as a precursor for various model assessment and verification processes that are implemented to improve performance and accuracy of WRE-N forecasts.

In this report, a process for the automation of the WRE-N postprocessing is implemented using the Unified Post Processor (UPP)<sup>3</sup> version 3.0 tool developed by the National Centers for Environmental Prediction. A script was developed to assist in the postprocessing of WRE-N forecast output data in a fast and more efficient manner. It takes the WRE-N forecast output data and generates hourly GRIdded Binary (GRIB) files through the automation of the UPP tool. ARL has implemented multiple tools from Model Evaluation Tools (MET)<sup>4</sup> developed by NCAR to characterize domain-level model performance. The UPP output can be ingested into various MET, such as the automated process discussed in Dawson et al. (2016) using the Point-Stat tool or the Ensemble-Stat tool, implementing various model assessment and verification techniques to generate error-statistics data.<sup>5</sup> The UPP output can also be used by researchers or modelers for other data analysis as needed.

The previous methods for running the UPP tool to perform WRE-N postprocessing consistently overwrote the data of a previous run. The automated process of implementing the UPP tool applies a standard naming convention as well as a controlled input and output data structure, which allows it to be easily ingested into MET tools or used for further data analysis. Each data set is uniquely identified, which eliminates the issue of the data being overwritten. In addition, this process can automatically generate files necessary for aspects of the model assessment process, such as the ensemble member lists required as input data for the MET Ensemble-Stat tool. This can dramatically save time during the model verification processes when this tool is being used to produce statistics that quantify ensemble spread and accuracy.

## 2. Development Environment

---

The script for the automated UPP process was developed using Python 3.5.<sup>6</sup> The Weather Running Estimate-Nowcast-Real Time (WREN\_RT) system is also implemented in Python. WREN\_RT will be a system that will automate real-time WRE-N model simulations, collect and quality control check weather observations for assimilation and verification, and produce graphical output of weather forecasts with verification statistics. The ultimate goal will be to integrate WREN\_RT with the automated UPP process and its script, described in this report along with the automated model assessment process using the MET Point-Stat tool detailed in Dawson et al. (2016), so that error statistics can be generated “on the fly” after each completed WRE-N model run. Developing all the predecessor and postprocesses of WREN\_RT in Python will bring consistency to the whole automation process and make integration in the future easier.

## 3. Script Requirements and Description

---

The Python script called *run\_upp.py* was developed for the automation of the UPP process. This script handles the reading and parsing of each configuration parameter defined in its configuration file, *upp\_auto\_config*. The script also handles the process of automatically running the UPP tool based on the configuration values. Before running the script, Python 2.6 or higher and UPP version 3.0 are required to be successfully installed and running on the user’s system.

Before executing the script, the user must define the configuration values in the *upp\_auto\_config* file to satisfy his or her unique requirements (Table 1). Then, the user can execute the following command to automatically generate hourly GRIB files:

```
python run_upp.py
```

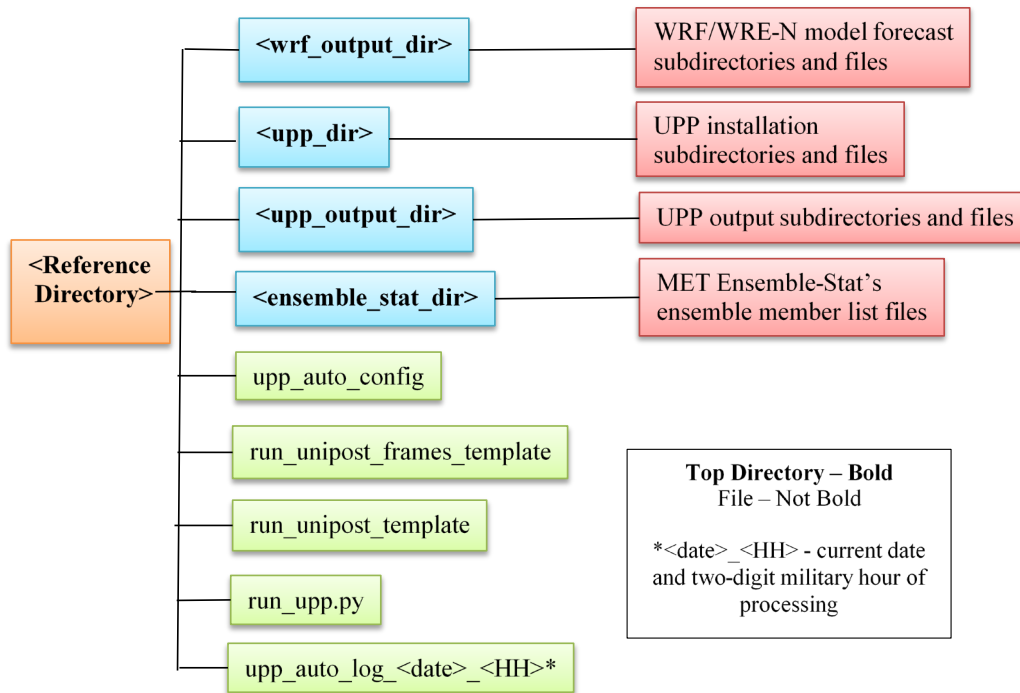
During the script’s execution, the script makes repetitive command calls to the UPP tool, which generates hourly GRIB files based on the user’s configuration specified in the *upp\_auto\_config* file.

**Table 1 Automated UPP script’s configuration parameters in *upp\_auto\_config***

<b>Configuration parameter</b>	<b>Description</b>
<i>wrf_output_dir</i>	<p><b>Full directory path where the WRF/WRE-N forecast data are stored.</b></p> <p>The script searches this directory and its subdirectories for the forecast data. There can be multiple subdirectories containing data. The script uses the lowermost directory names to append to its UPP output and to generate the ensemble member lists, so it is recommended to organize each WRF/WRE-N model run using lowermost directory names, such as run1 or case1.</p>
<i>wrf_output_hrtype</i>	<p><b>single (default) or multiple</b></p> <p>Specify the hour type of the WRF/WRE-N forecast data. For single hourly files, this should be set to “single” and for multi-hour files, it should be set to “multiple”. By default, the script sets this parameter to “single”.</p>
<i>upp_dir</i>	<p><b>Full directory path of the local UPP installation</b></p> <p>The script uses this directory to run the UPP tool and assumes it to be running correctly according to its user guide.</p>
<i>upp_output_dir</i>	<p><b>Full directory path to store the generated UPP output</b></p> <p>The script uses this directory to store the hourly GRIB forecast files generated by the UPP tool.</p>
<i>firstfhr</i>	<p><b>HH - First forecast lead hour</b></p> <p>The first two-digit forecast lead hour for the script to postprocess the WRF/WRE-N forecast data. This is typically set to 00 in the case of 00 h lead time processing.</p>
<i>lastfhr</i>	<p><b>HH - Last forecast lead hour</b></p> <p>The last two-digit forecast lead hour for the script to postprocess the WRF/WRE-N forecast data. This is typically set to the same value as firstfhr when processing one WRF/WRE-N forecast file only.</p>
<i>ensemble_stat</i>	<p><b>yes (default) or no</b></p> <p>If this parameter is set to “yes”, then the script generates ensemble member lists to be later ingested into the MET Ensemble-Stat tool. If it is set to “no”, then ensemble member lists are not generated.</p>
<i>ensemble_stat_dir</i>	<p><b>Full directory path where the generated ensemble member lists is stored.</b></p> <p>If the parameter, ensemble_stat, is set to “yes”, then the ensemble member lists will be stored in this directory. Otherwise, it will be ignored if the parameter, ensemble_stat, is set to “no”.</p>
<i>ens_list_base_dir</i>	<p><b>Base directory path on a different system where the generated UPP output files can be found for the ensemble member lists (optional).</b></p> <p>If the MET Ensemble-Stat tool will be run on the same system as the automated UPP script, then this parameter should be LEFT BLANK since the base path directory will be identical to the value set in the parameter, <i>upp_output_dir</i>. Otherwise, if the generated UPP output files are transferred to a different system and the MET Ensemble-Stat tool will be run on this system, then this parameter should be set to the base path directory, so the ensemble member lists can be found correctly while running the tool.</p>

## 4. Data Structure

The script for the automated UPP process uses input and output data that are organized into a controlled data structure. An example standard data structure for the automated UPP process is depicted in Fig. 1. All the directories highlighted in blue on Fig. 1 can vary based on what is set by the user in the automated UPP script's configuration file, *upp\_auto\_config*. The input data required by the script or the UPP tool are the script's configuration file, *upp\_auto\_config* (Section 5), UPP's template file (Section 5), WRF/WRE-N forecast data (Section 6), and the location of the local UPP installation.



**Fig. 1 Standard data structure of the automated UPP script**

As shown in Figs. 1 and 2, the automated UPP script's configuration file, *upp\_auto\_config*, allows the user to specify the hour type for the WRF/WRE-N input forecast data files (i.e., single hourly files or multi-hour files), first and last forecast lead hours for postprocessing, and the option to generate ensemble member lists for running the MET Ensemble-Stat tool. The location of the WRE-N forecast data specified in *<wrf\_output\_dir>* is required by the UPP tool as depicted in Fig. 1. Similarly, the location of the local UPP installation specified in *<upp\_dir>* is required by the UPP tool. One of the two template files provided along with the script, *run\_unipost\_template* or *run\_unipost\_frames\_template*, is also required by the UPP tool, as shown in Fig. 1, based on the user's data needs.

```
#Example configuration file, upp_auto_config, for the automated UPP script
wrf_output_dir=/home/jdoe/wrf-output_test
wrf_output_hrtype=multiple
upp_dir=/home/jdoe/UPPV3.0
upp_output_dir=/home/jdoe/UPP_output/test
firstfhr=00
lastfhr=24
ensemble_stat=yes
ensemble_stat_dir=/home/jdoe/test_ens/
ens_list_base_dir=/h/jdoe/UPP_output/test
```

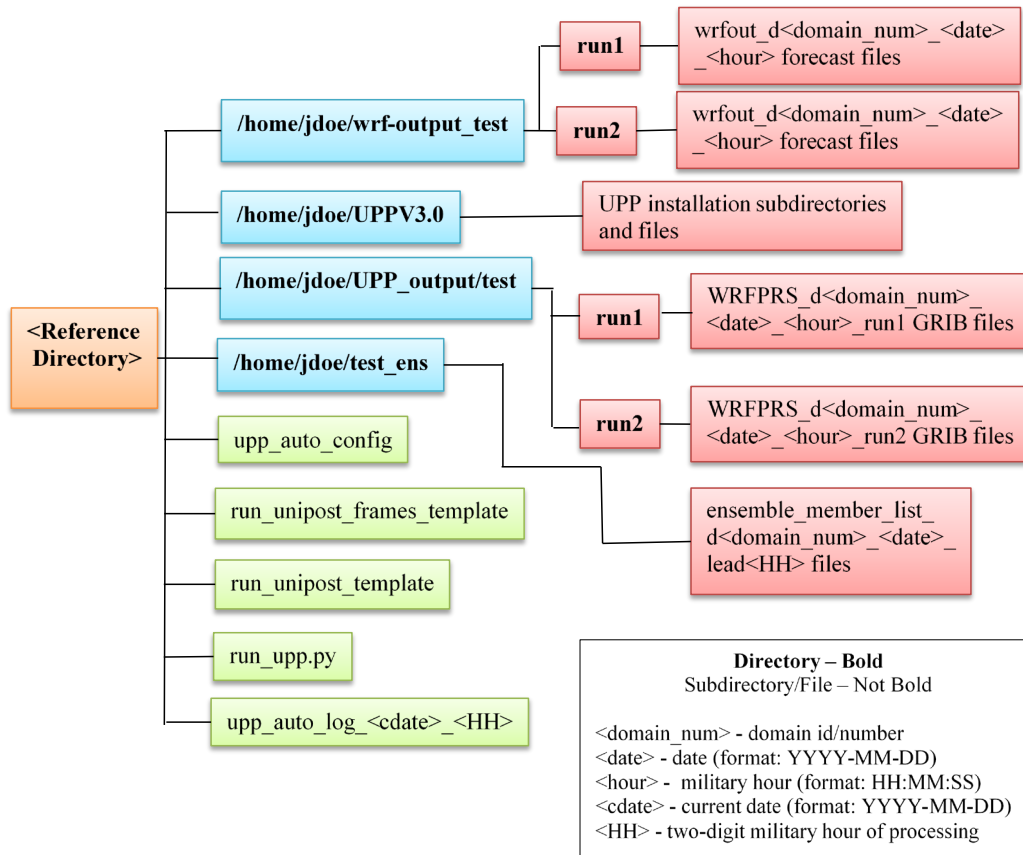
**Fig. 2** Example configuration file, *upp\_auto\_config*, for the automated UPP script

The output data consists of hourly GRIB forecast files generated from the UPP tool, and optionally, the MET Ensemble-Stat’s ensemble member list files, which can be stored if the user needs to generate them as a precursor for running the MET Ensemble-Stat tool. The hourly GRIB output files of the automated UPP script will be located in the directory specified in *<upp\_output\_dir>*, as depicted in Fig. 1. Likewise, the generated ensemble member list files will be located in the directory specified in *<ensemble\_stat\_dir>*.

## 5. Configuration and Template Files

---

The configuration file, *upp\_auto\_config*, controls the process and output of the automated UPP script. All the values set in this configuration file determine how the script uses the UPP tool to automatically produce hourly GRIB files. A description of all these configuration values is in Table 1. A template for *upp\_auto\_config* is provided along with the automated UPP script, which can be modified by the user to match the user’s unique requirements. An example *upp\_auto\_config* file and its corresponding data structure are shown in Figs. 2 and 3, respectively. The ultimate goal is to autogenerate or transfer the configuration file once the WREN\_RT system integration is complete, so that the hourly GRIB files can be developed “on-the-fly” after each WRF/WRE-N model run.



**Fig. 3 Automated UPP script’s data structure based on the example configuration file**

In addition, there are two template files, *run\_unipost\_template* and *run\_unipost\_frames\_template*, that are provided along with the automated UPP script, as shown in Fig. 1. These template files are required for the script to run the UPP tool, which reads multiple parameter fields from the WRF/WRE-N forecast files. The script automatically copies the appropriate template file to the local UPP installation subdirectory, *postprd*, and replaces the contents of UPP’s *run\_unipost* script with the contents of the template file. The subdirectory *postprd* is created, if it does not already exist. More importantly, the template file used during the script’s execution is determined by the parameter value, *wrf\_output\_hrtype*, set in the configuration file *upp\_auto\_config* (see Table 1). The script automatically uses *run\_unipost\_template* when the input WRF/WRE-N data consists of single hourly WRF/WRE-N forecast files, while it uses the *run\_unipost\_frames\_template* for WRF/WRE-N data consisting of multi-hour WRF/WRE-N forecast files. The template files also provide the script’s functionality to modify the model’s start date based on the input WRF/WRE-N forecast data as well as the first lead forecast hour and last forecast hour during a UPP run using the parameter values *firstfhr* and *lastfhr*. Furthermore, the user should note if there are any additional changes needed

to the basic path variables set in the default UPP's *run\_unipost* script (located in `$UNIPOST_HOME/postprd`) that are not included in the script's configuration file, then the corresponding template files provided with the script will also need to be modified accordingly to produce accurate results. For example, if the parameter, `$STOP_DIR`, in the default UPP's *run\_unipost* script has been changed, then the corresponding *run\_unipost\_template* will also need to be modified. In other words, the *run\_unipost\_template* and *run\_unipost\_frames\_template* need to be modified for single- and multi-hour WRF/WRE-N output, respectively, to match the parameters changed in the *run\_unipost* script.

## 6. Forecast Data

---

The main data inputs for the automated UPP script are gridded WRF/WRE-N forecast files, which are required for the UPP tool. The forecast data files should be WRF/WRE-N native output, which is commonly in the netCDF<sup>7</sup> format. As stated in Section 5, the top directory's location of the WRF/WRE-N forecast files that needs to be postprocessed and its hour type should be set in the *upp\_auto\_config* file by using the configuration parameters, *wrf\_output\_dir* and *wrf\_output\_hrtype*, respectively before executing the automated UPP script. The script accepts the two hour types for the WRF/WRE-N forecast files: single hourly files and multi-hour files. The user can specify the hour type through setting the *wrf\_output\_hrtype* parameter in the *upp\_auto\_config* file to "single" for single hourly WRF/WRE-N forecast files or "multiple" for multi-hour WRF/WRE-N forecast files, as described in Table 1. The number of forecast hours that need to be postprocessed by the user are set through both the configuration parameters, *firstfhr* and *lastfhr*, in the *upp\_auto\_config* file. The first forecast lead hour for the WRF/WRE-N output files is denoted in the configuration parameter, *firstfhr*, which is typically set to "00" to represent the first lead hour or 0 lead time hour that will be postprocessed. Similarly, the last forecast lead hour for the WRF/WRE-N output is denoted in the configuration parameter, *lastfhr*. For example, as shown in Fig. 2, if a WRF/WRE-N forecast file containing 24 hourly Coordinated Universal Time (UTC) forecasts for the time period of 1200 UTC to 1200 UTC next day, then *firstfhr* should be set to "00" to represent the first forecast lead hour and *lastfhr* should be set to "24" to represent the last forecast lead hour. If the user has only one single-hour WRF/WRE-N output file to postprocess, then the configuration parameter, *lastfhr*, should be set to "00".

In addition, it is recommended that the WRF/WRE-N forecast files for each model run have its own lowermost directory, especially if the user has different data sets from multiple, distinct WRF/WRE-N model runs. For example, as shown in Fig. 3, there are two subdirectories, *run1* and *run2*, under *wrf\_output\_dir* that contain

WRF/WRE-N forecast files to be postprocessed by the automated UPP script. However, there can be several subdirectories as needed by the user (such as *run1*, *run2*, *run3*, *run4*, etc.), which contain forecast files from multiple, distinct WRF/WRE-N model runs. This allows for the model postprocessing to be completed quickly for multiple data sets when the user needs to process all the data synchronously.

## 7. Ensemble-Stat Configuration Option

---

The automated UPP script has the option to generate all the ensemble member lists for the user, which could be used with the MET Ensemble-Stat tool. In the case where this option is enabled, the script will run the UPP tool as well as automatically produce both the UPP output and ensemble member lists to ingest into the MET Ensemble-Stat tool. Then, the MET Ensemble-Stat tool can be used to create ensemble forecasts (i.e., ensemble spread graphics and verification statistics) from a set of several WRF/WRE-N forecast files. The set of forecast files are ingested into the tool using ensemble member lists. Each ensemble member list is an American Standard Code for Information Interchange file containing a list of the ensemble member filenames and their paths for each lead hour specified between the *firstfhr* and *lastfhr* configuration parameters. For example, if these parameters were set as depicted in Fig. 2, there would be 25 ensemble member list files for each domain and each lead hour from 00 through 24. If the lead hours, “00” and “01”, for a group of these WRF/WRE-N forecast files are 12:00:00 UTC and 13:00:00 UTC, respectively, on 9 Feb 2012 with two different model runs (i.e., *run1* and *run2* based on Fig. 3), then their ensemble member list files, *ensemble\_member\_list\_d01\_2012-02-09\_lead00* and *ensemble\_member\_list\_d01\_2012-02-09\_lead01*, would be similar to the files shown in Fig. 4.

```
#ensemble_member_list_d01_2012-02-09_lead00
/home/jdoe/wrf-output_test/run1/WRFPRS_d01_2012-02-09_12:00:00_run1
/home/jdoe/wrf-output_test/run2/WRFPRS_d01_2012-02-09_12:00:00_run2

#ensemble_member_list_d01_2012-02-09_lead01
/home/jdoe/wrf-output_test/run1/WRFPRS_d01_2012-02-09_13:00:00_run1
/home/jdoe/wrf-output_test/run2/WRFPRS_d01_2012-02-09_13:00:00_run2
```

**Fig. 4** Two example ensemble member list files

In addition, if the user needs to implement the option for the automated UPP script to generate ensemble member lists, then the configuration parameter, *ensemble\_stat*, should be set to “yes” in the configuration file, *upp\_auto\_config*, as

described in Table 1. Similarly, the full directory path location where the ensemble member list files should be stored by the automated UPP script is specified in the *upp\_auto\_config* file by the user through the parameter, *ensemble\_stat\_dir*, as described in Table 1. The automated UPP script uses the following standard naming convention when creating the filenames of the ensemble member list files:

`ensemble_member_list_d<domain_num>_<model_date>_lead<model_hour>`

*domain\_num* is the two-digit domain number of the model run, *model\_date* is the date of the model forecast in the format of YYYY-MM-DD, and *model\_hour* is the two-digit model forecast hour.

Furthermore, as discussed in Table 1, if the user moves the UPP output and needs to run the MET Ensemble-Stat tool on a different system other than the current system, then the user should set the configuration parameter, *ens\_list\_base\_dir*, in the *upp\_auto\_config* file to the base directory path where the UPP output files are moved and stored on the other system in order for them to be found correctly by the automated UPP script when the ensemble member lists are generated. Otherwise, the user should leave this parameter as blank when the MET Ensemble-Stat tool will be run on the same system as the automated UPP script, where in this case, the base directory path will be the same as stated in the configuration parameter, *upp\_output\_dir*. For example, as shown in Fig. 2, the *ens\_list\_base\_dir* configuration parameter is set to a different directory path than the parameter, *upp\_output\_dir*, which means that the MET Ensemble-Stat will be run on a different system from the current system. Therefore, the ensemble member list files will be generated using the UPP output files set for *ens\_list\_base\_dir* parameter, located at `/h/jdoe/UPP_output/test`, instead of these files being set for the *upp\_output\_dir* parameter, located at `/home/jdoe/UPP_output/test`.

## 8. Output Results

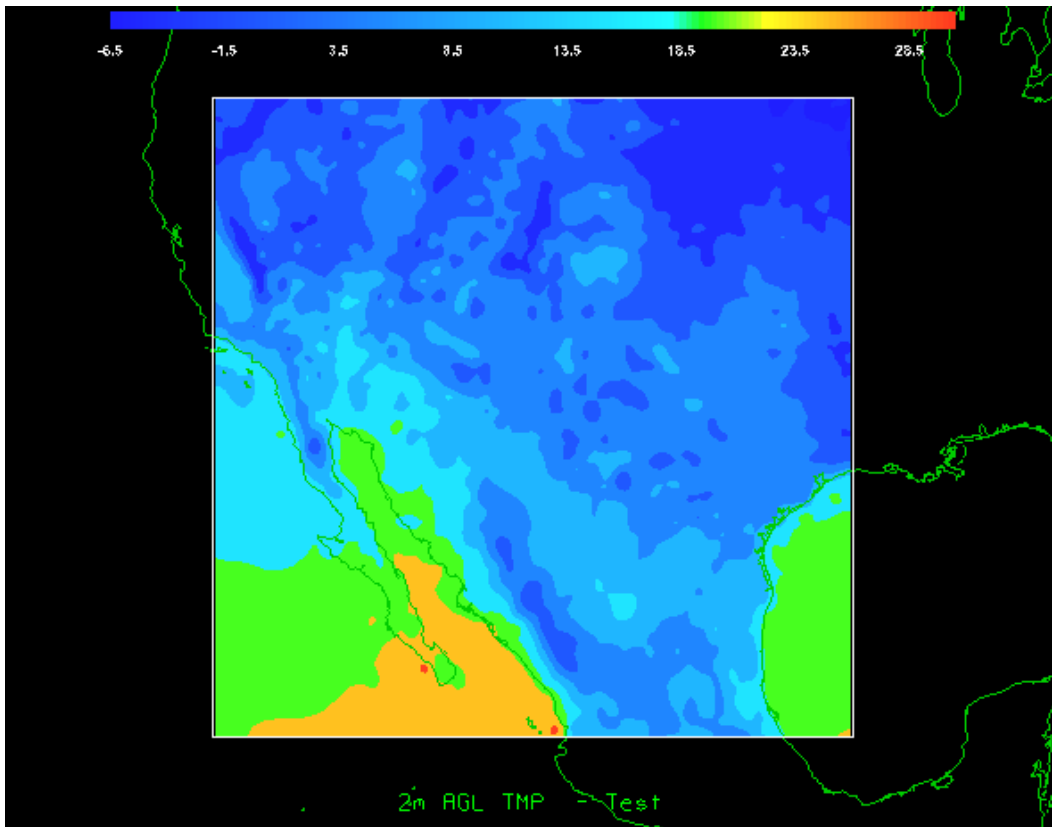
---

---

After the execution of the automated UPP script, hourly GRIB files are automatically produced and stored in the output directory specified by the user's configuration parameter, *upp\_output\_dir*, in the *upp\_auto\_config* file, as described in Table 1. After the script successfully completes running UPP, it takes the generated UPP output files for each forecast hour and automatically renames the files using a specific naming convention. The original UPP output files are in the format of `WRFPRS_d<domain_num>.<hh>`,<sup>4</sup> where *domain\_num* refers to the domain id/number and *hh* denotes the forecast hour. During the automated UPP script's execution, the contents of each original UPP output file are copied and renamed to a different filename using the following standard naming convention:

WRFPRS\_d<domain\_num>\_<date>\_<hour>\_<run/caseID>

*domain\_num* refers to the domain id/number, *date* denotes the forecast date in the format of YYYY-MM-DD, *hour* refers to the forecast military hour in the format of HH:MM:SS, and *run/caseID* refers to the run or case ID retrieved from the name of the lowermost directory of the WRF/WRE-N forecast files. Then, the renamed UPP output files are moved to the directory location specified in the *upp\_output\_dir* parameter, as described in Table 1. The process of renaming and organizing the UPP output files in a controlled data structure prevents them from being overwritten if the script is executed multiple times with similar configuration parameters on the same system, thus making it easier to identify the output files when ingesting them into MET, such as Point-Stat or Ensemble-Stat. The output files could also be easily used by modelers and researchers for further data analysis via various GRIB viewers, such as NCAR's Integrated Data Viewer,<sup>8</sup> as shown in Fig. 5, which displays an example GRIB file depicting 2-m above ground level temperature at 0900 UTC around the White Sands Missile Range area on 21 Nov 2016. Possibly, these GRIB files could also be ingested into various weather-decision aid tools as needed.



**Fig. 5** Example GRIB file depicting 2-m above ground level temperature in Celsius

## 9. Log File

---

After the execution of the automated UPP script, a log file is automatically created that contains all the log and debug information relating to the current run. An example log file, *upp\_auto\_log*, is provided in Fig. 6, where an *upp\_auto\_log* file was created after a successful run of the automated UPP script for one domain, two model runs (i.e., run1 and run2), and two forecast hours (i.e., 12:00:00 UTC, 13:00:00 UTC) on the date of 2012-02-09. The standard naming convention of the log file for the automated UPP script is as follows:

`upp_auto_log_<YYYYMMDD>_<HH>`

<YYYYMMDD> is the current date of processing and <HH> is the current military hour of processing. The log file is created and stored in the same directory where the automated script, *run\_upp.py*, is executed, as shown in Fig. 1.

The log file includes information, such as the number of WRF/WRE-N directories and the start date of the WRF/WRE-N model run found during execution, and new filenames of the renamed UPP output files. If there are any error messages from the execution of the automated UPP script, the *upp\_auto\_log* file will include them. If any errors or issues occur relating to the UPP tool, the tool's log file, `unipost_d<domain_num>.<forecast_hour>.out`,<sup>4</sup> will include them, where *domain\_num* is the domain number and *forecast\_hour* is the two-digit forecast hour. This file is typically located in the subdirectory, *postprd*, of the local UPP installation directory specified in the *upp\_dir* parameter.

```

Reading UPP auto configuration file, upp_auto_config...
Found 2 directory/directories containing WRF/WRE-N output in /home/jdoe/wrf-output_test

Running UPP...

Removing previous WRF/WRE-N output in UPP post_processing directories:
/home/jdoe/UPPV3.0/wrfprd (if any)
Copying WRE-N output from /home/jdoe/wrf-output_test/run1 to UPP WRF post-processing
directory, /home/jdoe/UPPV3.0/wrfprd
Searching WRF output file,wrfout_d01_2012-02-09_12:00:00,for model start date
Model START_DATE found: 2012020912
Finished running UPP

Renaming UPP output files for run/case ID, run1...
Copying UPP post-processed file, WRFPRS_d01.00 to renamed file, WRFPRS_d01_2012-
02-09_12:00:00_run1
Removing UPP post-processed file, WRFPRS_d01.00 and its symbolic link
Copying renamed UPP file,WRFPRS_d01_2012-02-09_12:00:00_run1, to directory,
/home/jdoe/UPP_output/test/run1
Writing full path of renamed UPP file,WRFPRS_d01_2012-02-09_12:00:00_run1, to
ensemble_member_list_d01_2012-02-09_lead00, for Ensemble-Stat processing
Copying UPP post-processed file, WRFPRS_d01.01 to renamed file, WRFPRS_d01_2012-
02-09_13:00:00_run1
Removing UPP post-processed file, WRFPRS_d01.01 and its symbolic link
Copying renamed UPP file,WRFPRS_d01_2012-02-09_13:00:00_run1, to directory,
/home/jdoe/UPP_output/test/run1
Writing full path of renamed UPP file,WRFPRS_d01_2012-02-09_13:00:00_run1, to
ensemble_member_list_d01_2012-02-09_lead01, for Ensemble-Stat processing

Copying WRE-N output from /home/jdoe/wrf-output_test/run2 to UPP WRF post-processing
directory, /home/jdoe/UPPV3.0/wrfprd
Searching WRF output file,wrfout_d01_2012-02-09_12:00:00,for model start date
Model START_DATE found: 2012020912
Finished running UPP

Renaming UPP output files for run/case ID, run2...
Copying UPP post-processed file, WRFPRS_d01.00 to renamed file, WRFPRS_d01_2012-
02-09_12:00:00_run2
Removing UPP post-processed file, WRFPRS_d01.00 and its symbolic link
Copying renamed UPP file,WRFPRS_d01_2012-02-09_12:00:00_run2, to directory,
/home/jdoe/UPP_output/test/run2
Writing full path of renamed UPP file,WRFPRS_d01_2012-02-09_12:00:00_run2, to
ensemble_member_list_d01_2012-02-09_lead00, for Ensemble-Stat processing
Copying UPP post-processed file, WRFPRS_d01.01 to renamed file, WRFPRS_d01_2012-
02-09_13:00:00_run2
Removing UPP post-processed file, WRFPRS_d01.01 and its symbolic link
Copying renamed UPP file,WRFPRS_d01_2012-02-09_13:00:00_run2, to directory,
/home/jdoe/UPP_output/test/run2
Writing full path of renamed UPP file,WRFPRS_d01_2012-02-09_13:00:00_run2, to
ensemble_member_list_d01_2012-02-09_lead01, for Ensemble-Stat processing

```

**Fig. 6 Example log file from the automated UPP script**

## 10. Conclusion

---

The automated UPP script was developed to easily generate hourly GRIB files of model output using the UPP tool as a precursor for many subsequent model assessment and verification processes, such as the generation of domain-level error statistics from WRE-N model output data using MET, as described in Dawson et al. (2016). The automated UPP script provides an efficient, controlled data process and structure that can be easily used by ARL modelers or researchers, thus increasing the performance for further data analysis and evaluation of WRE-N model improvements. The long-term goal for the automated UPP script is to link it with other scripts to further automate the WRF/WRE-N model assessment and verification process, such as integrating it with the automated MET Point-Stat tool script described in Dawson et al. (2016). Furthermore, additional parameters can be included in the script's configuration file, *upp\_auto\_config*, to enhance the entire UPP automation process.

## 11. References

---

1. Dumais R, Kirby S, Flanigan, R. Implementation of the WRF four-dimensional data assimilation method of observation nudging for use as an ARL weather running estimate-nowcast. White Sands Missile Range (NM): Army Research Laboratory (US); 2013. Report No.: ARL-TR-6485.
2. Skamarock WC, Klemp JB, Dudhia J, Gill DO, Barker DM, Duda M, Huang X-Y, Wang W, Powers JG. A description of the advanced research WRF version 3, NCAR Tech Note. Boulder (CO): National Center of Atmospheric Research; 2008. [accessed 2018 July]. [http://www2.mmm.ucar.edu/wrf/users/docs/arw\\_v3.pdf](http://www2.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf).
3. User's guide for the NMM core of the Weather Research and Forecast (WRF) modeling system version 3: chapter 7: post processing utilities–NCEP Unified Post Processor. Boulder (CO): Developmental Testbed Center; 2014. [accessed 2018 July]. [https://dtcenter.org/wrf-nmm/users/docs/user\\_guide/V3/users\\_guide\\_nmm\\_chap7.pdf](https://dtcenter.org/wrf-nmm/users/docs/user_guide/V3/users_guide_nmm_chap7.pdf).
4. Model evaluation tools version 5.2 (METv5.2) User's guide 5.2. Boulder (CO): Developmental Testbed Center; 2016. [accessed 2018 July]. [https://dtcenter.org/met/users/docs/users\\_guide/MET\\_Users\\_Guide\\_v5.2.pdf](https://dtcenter.org/met/users/docs/users_guide/MET_Users_Guide_v5.2.pdf).
5. Dawson, L, Raby, J, Smith, J. The automation of nowcast model assessment processes. Adelphi (MD): Army Research Laboratory (US); 2016. Report No.: ARL-MR-0940.
6. Python. Wilmington (DE): Python Software Foundation; 2018. [accessed 2018 July]. <http://www.python.org>.
7. Network Common Data Form (NetCDF). Boulder (CO): University Corporation for Atmospheric Research; 2018. [accessed 2018 July]. <https://www.unidata.ucar.edu/software/netcdf/>.
8. Integrated Data Viewer (IDV). Boulder (CO): University Corporation for Atmospheric Research; 2018. [accessed 2018 July]. <https://www.unidata.ucar.edu/software/idv/>.

## List of Symbols, Abbreviations, and Acronyms

---

ARL	US Army Research Laboratory
GRIB	GRIdded Binary
ID	identification
MET	Model Evaluation Tools
MSA	Meteorological Sensor Array
NCAR	National Center for Atmospheric Research
WRE-N	Weather Running Estimate-Nowcast
WREN_RT	Weather Running Estimate-Nowcast_Real Time
WRF	Weather Research and Forecasting
UPP	Unified Post Processor
UTC	Coordinated Universal Time

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

2 DIR ARL  
(PDF) IMAL HRA  
RECORDS MGMT  
RDRL DCL  
TECH LIB

1 GOVT PRINTG OFC  
(PDF) A MALHOTRA

4 ARL  
(PDF) RDRL CIE M  
L DAWSON  
J RABY  
R DUMAIS  
B MACCALL