# METplus Version 2.0.4

Automation for the Model Evaluation Tools

**Developmental Testbed Center**
**Boulder, Colorado**


*Daniel Adriaansen[1], Minna Win-Gildenmeister[1,4], James Frimel[2,4], Julie Prestopnik[1,4], Mallory Row[3], John Halley Gotway[1,4], George McCabe[1,4], Tara Jensen[1,4], Jonathan Vigh[1,4], Christina Kalb[1], and Hank Fisher[1]*

*[1] National Center for Atmospheric Research,*
*Research Applications Laboratory*
*[2] Cooperative Institute for Research in the Atmosphere at*
*National Oceanic and Atmospheric Administration (NOAA)*
*Earth System Research Laboratory*
*[3] I.M. Systems Group at*
*NOAA Environmental Modeling Center*
*[4] Developmental Testbed Center*



**February 2019**

# Contents

# Foreword: A note to METplus users

This User's Guide is provided as an aid to users of the Model Evaluation Tools (MET) and it's companion package METplus. MET is a set of verification tools developed and supported to community via the Developmental Testbed Center (DTC) for use by the numerical weather prediction community. METplus is intended to be a suite of Python wrappers and ancillary scripts to enhance the user's ability to quickly set-up and run MET. Over the next few years, METplus will become the authoritative repository for verification of the Unified Forecast System.

It is important to note here that METplus is an evolving software package. Previous releases of METplus have occurred since 2017. This documentation describes the 2.0 release in September 2018. Intermediate releases may include bug fixes. METplus is also be able to accept new modules contributed by the community. If you have code you would like to contribute, we will gladly consider your contribution. While we are setting up our community contribution protocol, please send email to: met_help@ucar.edu and inform us of your desired contribution. We will then determine the maturity of new verification method and coordinate the inclusion of the new module in a future version.

This User's Guide was prepared by the developers of the METplus, including Dan Adriaansen, Minna Win-Gildenmeister, Julie Prestopnik, Jim Frimel, Mallory Row, John Halley Gotway, George McCabe, Paul Prestopnik, Christana Kalb, Hank Fisher, Jonathan Vigh, Lisa Goodrich, Tara Jensen, Tatiana Burek,and Bonny Strong.

## New for METplus v2.0

**METplus v2.0.4 Release Notes:**

Configuration:

- Updated config files to match sample data directory structure

General:

- Moved large mask files from repository to sample data tarballs

- Improved logging message clarity
- List METplus version number in final configuration file and logging output

**METplus v2.0.3 Release Notes:**

Configuration:

- Added DO_NOT_RUN_EXE config variable to prevent applications from actually running
- Added LOG_TIMESTAMP_USE_RUNTIME config variable to use data time in log file names instead of run time
- METPLUS_BASE config variable is automatically set to the location METplus is being run
- Added automatically generated CLOCK_TIME config variable to keep track of time METplus was run

Wrapper specific:

- mode_wrapper
  - new python wrapper for MET tool mode
- mtd_wrapper
  - new python wrapper for MET tool mtd (mode time domain)
- pcp_combine_wrapper
  - Threshold values specified in the config files now require a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le Previously _THRESH values were assumed to use >= by pcp_combine
- grid_stat_wrapper
  - grid_stat will now process all name/level/threshold combinations in a single run if desired (some cases require splitting up calls to grid_stat, such as processing probabilistic forecasts or precip accumulations)
  - Added probability threshold configs for grid_stat probabilistic forecast evaluation

General:

- Compressed input files with certain file extensions (gz, zip, bz2) will be automatically uncompressed and placed into a staging area for use in METplus (with option to scrub staging directory after run) - Gempak files now can automatically be converted to NetCDF for use in METplus (See [FCST/OBS]_[METAPP]_DATATYPE)

- NetCDF field levels can now be specified in config files, i.e. (0,0,*,*). NOTE: Quotes around these items are required
- Updated MET config files to use MET 8.0
- Cleanup of plotting scripts

**METplus v2.0.2 Release Notes:**

Wrapper specific:

- grid_stat_wrapper
  - Forecast lead time set in environment as FCST_TIME to be read by grid_stat MET config file

General:

- Users can define custom environment variables in METplus config files to be used in MET config files. (See section 3.5 User Defined Config)

**METplus v2.0.1 Release Notes:**

Configuration:

- OBS_WINDOW_BEG in point_stat_wrapper, grid_to_obs.conf changed to OBS_WINDOW_BEGIN

Wrapper specific:

- pcp_combine_wrapper:
  - fixed bug selecting accumulation files.
  - sum method and file template matching.

General:

- Fixed typo in variable name in getraw_interp function.

**METplus v2.0 Release Notes:**

Wrapper specific:

- tc_stat_wrapper

  - can now be run stand-alone

- tc_pairs_wrapper
  - can now read ATCF input file formats
  - support for numerous input file naming conventions
  - support for input data organized into one directory or subdirectories with date information in the name

- cyclone_plotter_wrapper
  - replaced the dependency on Basemap toolkits (which are unstable on some platforms) with Cartopy for map rendering

- tcmpr_plotter_wrapper
  - now supports whitespace in plot title, sub-title, and legend

- pb2nc_wrapper
  - new python wrapper for the MET tool pb2nc

- point_stat_wrapper
  - new python wrapper for the MET tool point_stat

# TERMS OF USE

## IMPORTANT!

USE OF THIS SOFTWARE IS SUBJECT TO THE FOLLOWING TERMS AND CONDITIONS:

1. **License**. Subject to these terms and conditions, University Corporation for Atmospheric Research (UCAR) grants you a non-exclusive, royalty-free license to use, create derivative works, publish, distribute, disseminate, transfer, modify, revise and copy the Model Evaluation Tools (MET) software, in both object and source code (the "Software").

You shall not sell, license or transfer for a fee the Software, or any work that in any manner contains the Software.

2. **Disclaimer of Warranty on Software.** Use of the Software is at your sole risk. The Software is provided "AS IS" and without warranty of any kind and UCAR EXPRESSLY DISCLAIMS ALL WARRANTIES AND/OR CONDITIONS OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT OF A THIRD PARTY'S INTELLECTUAL PROPERTY, MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE. THE PARTIES EXPRESSLY DISCLAIM THAT THE UNIFORM COMPUTER INFORMATION TRANSACTIONS ACT (UCITA) APPLIES TO OR GOVERNS THIS AGREEMENT. No oral or written information or advice given by UCAR or a UCAR authorized representative shall create a warranty or in any way increase the scope of this warranty. Should the Software prove defective, you (and neither UCAR nor any UCAR representative) assume the cost of all necessary correction.

3. **Limitation of Liability.** UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL UCAR BE LIABLE FOR ANY DIRECT, INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES INCLUDING LOST REVENUE, PROFIT OR DATA, WHETHER IN AN ACTION IN CONTRACT OR TORT ARISING OUT OF OR RELATING TO THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF UCAR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

4. **Compliance with Law.** All Software and any technical data delivered under this Agreement are subject to U.S. export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all applicable laws and regulations in connection with use and distribution of the Software, including export control laws, and you acknowledge that you have responsibility to obtain any required license to export, re-export, or import as may be required.

5. **No Endorsement/No Support**. The names UCAR/NCAR, National Center for Atmospheric Research and the University Corporation for Atmospheric Research may not be used in any advertising or publicity to endorse or promote any products or commercial entity unless specific written permission is obtained from UCAR. The Software is provided without any support or maintenance, and without any obligation to provide you with modifications, improvements, enhancements, or updates of the Software.

6. **Controlling Law and Severability**. This Agreement shall be governed by the laws of the United States and the State of Colorado. If for any reason a court of competent jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this Agreement shall continue in full force and effect. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is hereby expressly excluded.

7. **Termination.** Your rights under this Agreement will terminate automatically without notice from UCAR if you fail to comply with any term(s) of this Agreement. You may terminate this Agreement at any time by destroying the Software and any related documentation and any complete or partial copies thereof. Upon termination, all rights granted under this Agreement shall terminate. The following provisions shall survive termination: Sections 2, 3, 6 and 9.

8. **Complete Agreement**. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this Agreement will be binding unless in writing and signed by UCAR.

9. **Notices and Additional Terms**. Copyright in Software is held by UCAR. You must include, with each copy of the Software and associated documentation, a copy of this Agreement and the following notice:

"The source of this material is the Research Applications Laboratory at the National Center for Atmospheric Research, a program of the University Corporation for Atmospheric Research (UCAR) pursuant to a Cooperative Agreement with the National Science Foundation; ©2007-2017 University Corporation for Atmospheric Research. All Rights Reserved."

**The following notice shall be displayed on any scholarly works associated with, related to or derived from the Software:**

*"Model Evaluation Tools (MET) and METplus were developed at the National Center for Atmospheric Research (NCAR) through grants from the National Science Foundation (NSF), the National Oceanic and Atmospheric Administration (NOAA), and the United States Air Force (USAF). NCAR is sponsored by the United States National Science Foundation."*

**By using or downloading the Software, you agree to be bound by the terms and conditions of this Agreement.**

The citation for this User's Guide should be:

Adriaansen, D., M. Win-Gildenmeister, J. Frimel, J. Prestopnik, J. Halley Gotway,
T. Jensen, J. Vigh, C. Kalb, G. McCabe, and H. Fisher, 2018:
The METplus Version 2.0 User's Guide. Developmental Testbed Center.
Available at: https://github.com/NCAR/METplus/releases. 85 pp.

# Acknowledgments

We thank the the National Science Foundation (NSF) along with three organizations within the National Oceanic and Atmospheric Administration (NOAA): 1) Office of Atmospheric Research (OAR); 2) Next Generation Global Predition System project (NGGPS); and 3) United State Weather Research Program (USWRP) for their support of this work. Thanks also go to the staff at the Developmental Testbed Center for their help, advice, and many types of support. We released METplus Alpha in February 2017 and would not have made a decade of cutting-edge verification support without those who participated in DTC planning workshops and the NGGPS United Forecast System Strategic Implementation Plan Working Groups (NGGPS UFS SIP WGs).

The DTC is sponsored by the National Oceanic and Atmospheric Administration (NOAA), the United States Air Force, and the National Science Foundation (NSF). NCAR is sponsored by the National Science Foundation (NSF).

# Chapter 1

# Overview of METplus

## 1.1 Purpose and organization of the User's Guide

The goal of this User's Guide is to equip users with the information needed to use the Model Evaluation Tools (MET) and it's companion package METplus. MET is a set of verification tools developed and supported to community via the Developmental Testbed Center (DTC) for use by the numerical weather prediction community. METplus is a suite of Python wrappers and ancillary scripts to enhance the user's ability to quickly set-up and run MET. Over the next few years, METplus will become the authoritative repository for verification of the Unified Forecast System.

The METplus User's Guide is organized as follows. Chapter 1 provides an overview of METplus. Chapter 2 contains basic information about how to get started with METplus - including system requirements, required software , and how to download METplus. Chapter 4

## 1.2 The Developmental Testbed Center (DTC)

METplus has been developed, and will be maintained and enhanced, by the Developmental Testbed Center (DTC; `http://www.dtcenter.org/` ). The main goal of the DTC is to serve as a bridge between operations and research, to facilitate the activities of these two important components of the numerical weather prediction (NWP) community. The DTC provides an environment that is functionally equivalent to the operational environment in which the research community can test model enhancements; the operational community benefits from DTC testing and evaluation of models before new models are implemented operationally. METplus serves both the research and operational communities in this way - offering capabilities for researchers to test their own enhancements to models and providing a capability for the DTC to evaluate the strengths and weaknesses of advances in NWP prior to operational implementation.

METplus will also be available to DTC visitors and to the WRF modeling community for testing and evaluation of new model capabilities, applications in new environments, and so on. The METplus release

schedule is coincident with the MET release schedule and the METplus major release number is six less than the MET major release number (e.g. MET 8.0 is released with METplus 2.0).

## 1.3 METplus goals and design philosophy

METplus is a Python scripting infrastructure for the MET tools. The primary goal of METplus development is to provide MET users with a highly configurable and simple means to perform model verification using the MET tools. Prior to the availability of METplus, users who had more complex verifications that required the use of more than one MET tool were faced with setting up multiple MET config files and creating some automation scripts to perform the verification. METplus provides the user with the infrastructure to modularly create the necessary steps to perform such verifications.

METplus has been designed to be modular and adaptable. This is accomplished through wrapping the MET tools with Python and the use of hierarchical configuration files to enable users to readily customize their verification environments. Wrappers can be run individually, or as a group of wrappers that represent a sequence of MET processes. New wrappers can readily be added to the METplus package due to this modular design. Currently, METplus can easily be applied by any user on their own computer platform that supports Python 2.7.

The METplus code and documentation is maintained by the DTC in Boulder, Colorado. METplus is freely available to the modeling, verification, and operational communities, including universities, governments, the private sector, and operational modeling and prediction centers through a publicly accessible GitHub repository. Users simply need access to a web browser to download the source code and any other relevant documentation and data samples.

## 1.4 METplus components

The major components of METplus package are METplus Python wrappers to the MET tools, MET configuration files and a hierarchy of METplus configuration files. Some Python wrappers do not correspond to a particular MET tool, but wrap utilities to extend METplus functionality.

## 1.5 Future development plans

METplus is an evolving application. New capabilities are planned in controlled, successive version releases that are synchronized with MET releases. Bug fixes and user-identified problems will be addressed as they are found and posted to the known issues section of the METplus Users web page (`www.dtcenter.org/met/users/support`). Future METplus development plans are based on several contributing factors, including the needs of both the operational and research community. Issues that are in the development queue detailed in the "Issues" section of the GitHub repository. Please send questions to `met_help@ucar.edu`.

## 1.6 Code support

METplus support is provided through a MET-help e-mail address: met_help@ucar.edu. We will endeavor to respond to requests for help in a timely fashion. In addition, information about METplus and tools that can be used with MET are provided on the MET Users web page (`http://www.dtcenter.org/met/users/`).

We welcome comments and suggestions for improvements to METplus, especially information regarding errors. Comments may be submitted using the MET Feedback form available on the MET website. In addition, comments on this document would be greatly appreciated. While we cannot promise to incorporate all suggested changes, we will certainly take all suggestions into consideration.

METplus is a "living" set of wrappers and configuration files. Our goal is to continually enhance it and add to its capabilities. Because our time, resources, and talents are limited, we welcome contributed code for future versions of METplus. These contributions may represent new use cases or new plotting functions. For more information on contributing code to METplus, please contact `met_help@ucar.edu`.

# Chapter 2

# Software Installation/Getting Started

## 2.1 Introduction

This chapter describes how to download and set up METplus. METplus has been developed and tested on the Debian Linux operating system.

## 2.2 Supported architectures

METplus was developed on Debian Linux and is supported on this platform.

## 2.3 Programming/scripting languages

METplus is written in Python 2.7. METplus is intended to be a tool for the modeling community to use and adapt. As users make upgrades and improvements to the tools, they are encouraged to offer those upgrades to the broader community by offering feedback to the developers or coordinating for a GitHub pull. For more information on contributing code to METplus, please contact `met_help@ucar.edu`.

## 2.4 Pre-requisites

The following software is required to run METplus:

- Python 2.7

- R version 3.2.5 [1]
- nco (netCDF operators)
- MET version 6.1 or above
- Basic familiarity with MET
- GitHub account (if you plan on contributing code to METplus)

## 2.5 METplus directory structure

Once you have cloned the METplus from the GitHub repository at https://github.com/NCAR/METplus to a location on your host, change directories to the METplus directory. You should have the following directory structure:

```
METplus
    doc
    internal_tests
    parm
    sorc
    ush
    README.md
```

The top-level METplus directory consists of a README.md file and several subdirectories.

The doc/ directory contains documentation for users (PDF) and Doxygen files that are used to create the developer documentaton. The Doxygen documentation can be created and viewed via web browser if the developer has Doxygen installed on the host.

The internal_tests/ directory contains unit test scripts that are only relevant to METplus developers and contributors.

The parm/ directory contains all the configuration files for MET and METplus.

The sorc/ directory contains Doxygen executables to generate documentation for developers.

The src/ directory contains the source code for each of the wrappers in METplus.

The ush/ directory contains the Python wrappers to the MET tools.

[1] R version 3.2.5 is required when the tcmpr_plotter_wrapper.py wraps the plot_tcmpr.R script. Please refer to Chapter 21 Plotting and Graphics Support for more information about plot_tcmpr.R.

## 2.6 Getting the METplus source code

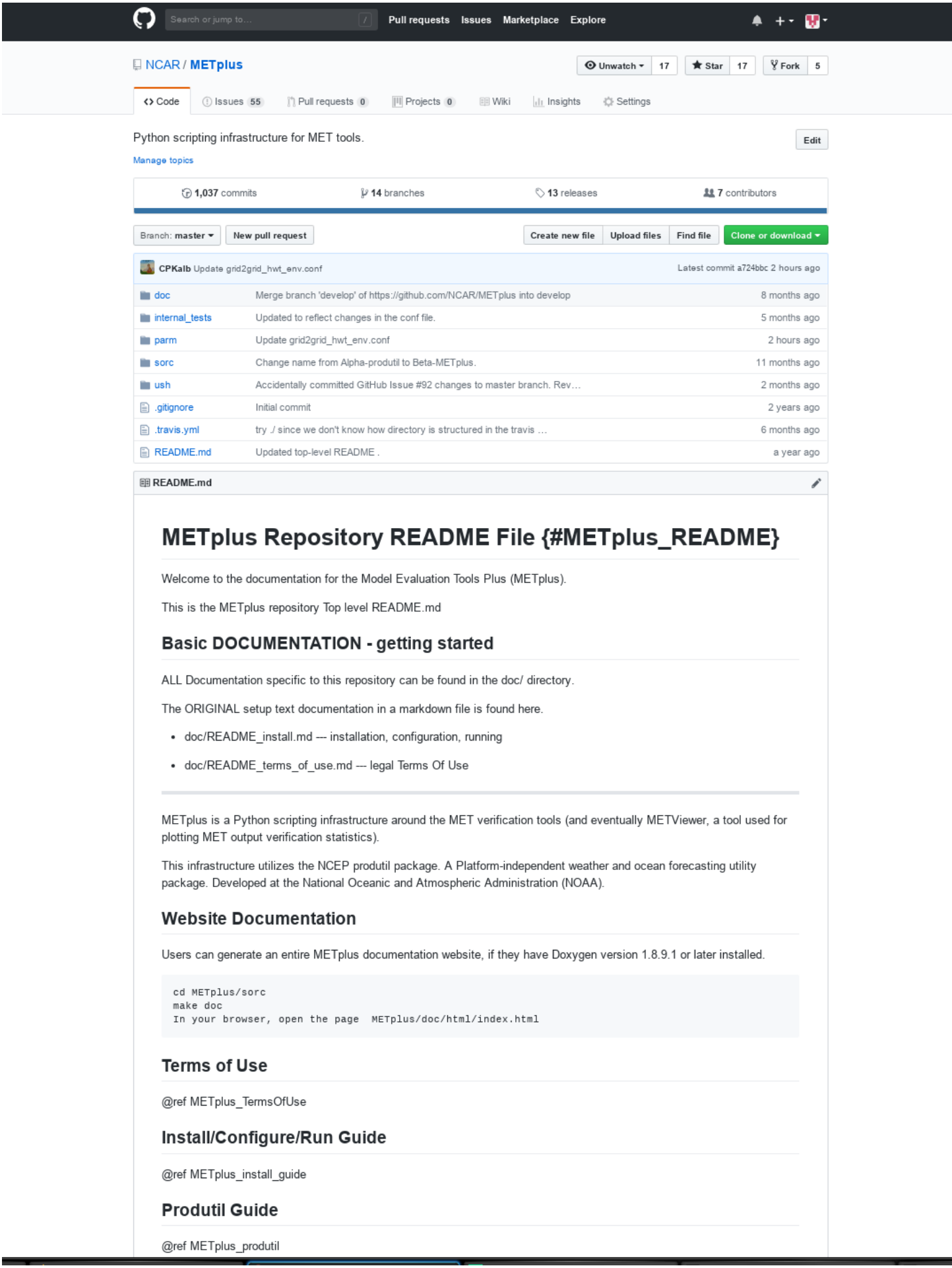
The METplus source code is available for download from a public GitHub repository. You can retrieve the source code through your web browser or the command line.

### 2.6.1 Get the source code via your Web Browser

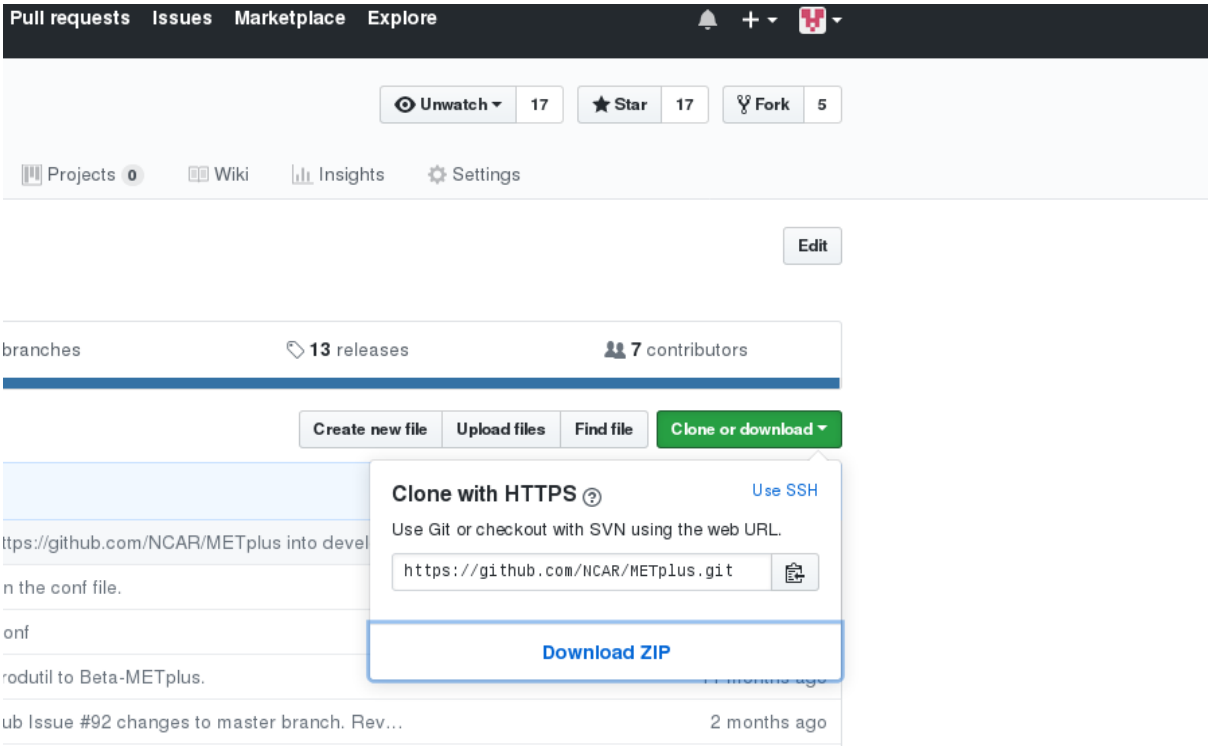
#### 2.6.1.1 Source code only:

If you wish to retrieve only the source code, then the following steps will illustrate how to quickly access the METplus source code and relevant documentation:

- On your local host (or wherever you wish to install the METplus code) create a directory where you want the code to reside
- Open the browser of your choice and navigate to https://github.com/NCAR/METplus. You will see something like the following:


Search or jump to...
Pull requests
Issues
Marketplace
Explore
NCAR / METplus
Unwatch 17
Star 17
Fork 5
Code
Issues 55
Pull requests 0
Projects 0
Wiki
Insights
Settings
Python scripting infrastructure for MET tools.
Edit
Manage topics
1,037 commits
14 branches
13 releases
7 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
CPKalb Update grid2grid_hwt_env.conf
Latest commit a724bbc 2 hours ago
doc Merge branch 'develop' of https://github.com/NCAR/METplus into develop 8 months ago
internal_tests Updated to reflect changes in the conf file. 5 months ago
parm Update grid2grid_hwt_env.conf 2 hours ago
sorc Change name from Alpha-produtil to Beta-METplus. 11 months ago
ush Accidentally committed GitHub Issue #92 changes to master branch. Rev... 2 months ago
.gitignore Initial commit 2 years ago
.travis.yml try ./ since we don't know how directory is structured in the travis ... 6 months ago
README.md Updated top-level README . a year ago
README.md
METplus Repository README File {#METplus_README}
Welcome to the documentation for the Model Evaluation Tools Plus (METplus).
This is the METplus repository Top level README.md
Basic DOCUMENTATION - getting started
ALL Documentation specific to this repository can be found in the doc/ directory.
The ORIGINAL setup text documentation in a markdown file is found here.
doc/README_install.md --- installation, configuration, running
doc/README_terms_of_use.md --- legal Terms Of Use
METplus is a Python scripting infrastructure around the MET verification tools (and eventually METViewer, a tool used for plotting MET output verification statistics).
This infrastructure utilizes the NCEP produtil package. A Platform-independent weather and ocean forecasting utility package. Developed at the National Oceanic and Atmospheric Administration (NOAA).
Website Documentation
Users can generate an entire METplus documentation website, if they have Doxygen version 1.8.9.1 or later installed.
cd METplus/sorc
make doc
In your browser, open the page METplus/doc/html/index.html
Terms of Use
@ref METplus_TermsOfUse
Install/Configure/Run Guide
@ref METplus_install_guide
Produtil Guide
@ref METplus_produtil

- You should be directed to the 'master' branch, verify this by looking at the button labelled 'Branch' in the upper left corner of your window, directly beneath the solid blue horizontal line.
- Click on the green "Clone or download" button near the top right of the page.
- A box appears with "Clone with HTTPS" label
- Click on the blue text: "Download Zip" :

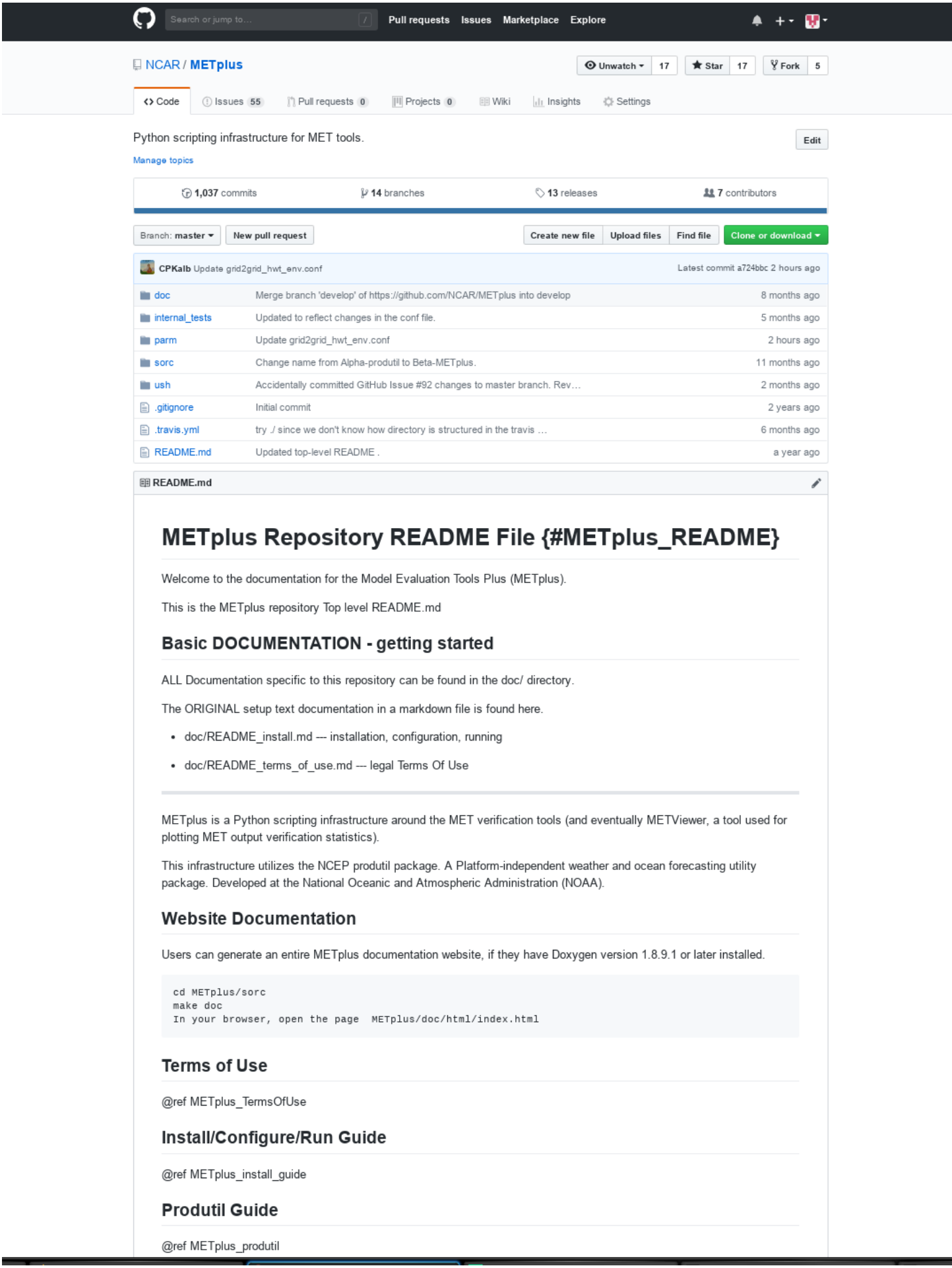



- Your browser should prompt you on what to do with this file. Save it to the directory you created above
- cd to the directory where you saved the code. You should see the file METplus-master.zip
- Uncompress the file:
  - Linux/Unix:
  - unzip METplus-master.zip
  - You should now have a METplus-master directory
    - If you downloaded the code via the command line, you will get a METplus directory rather than METplus-master.
    - GitHub appends the '-master' to the name to emphasize that is is from the master branch
    - To avoid clutter and confusion, you can now remove the METplus-master.zip (optional)

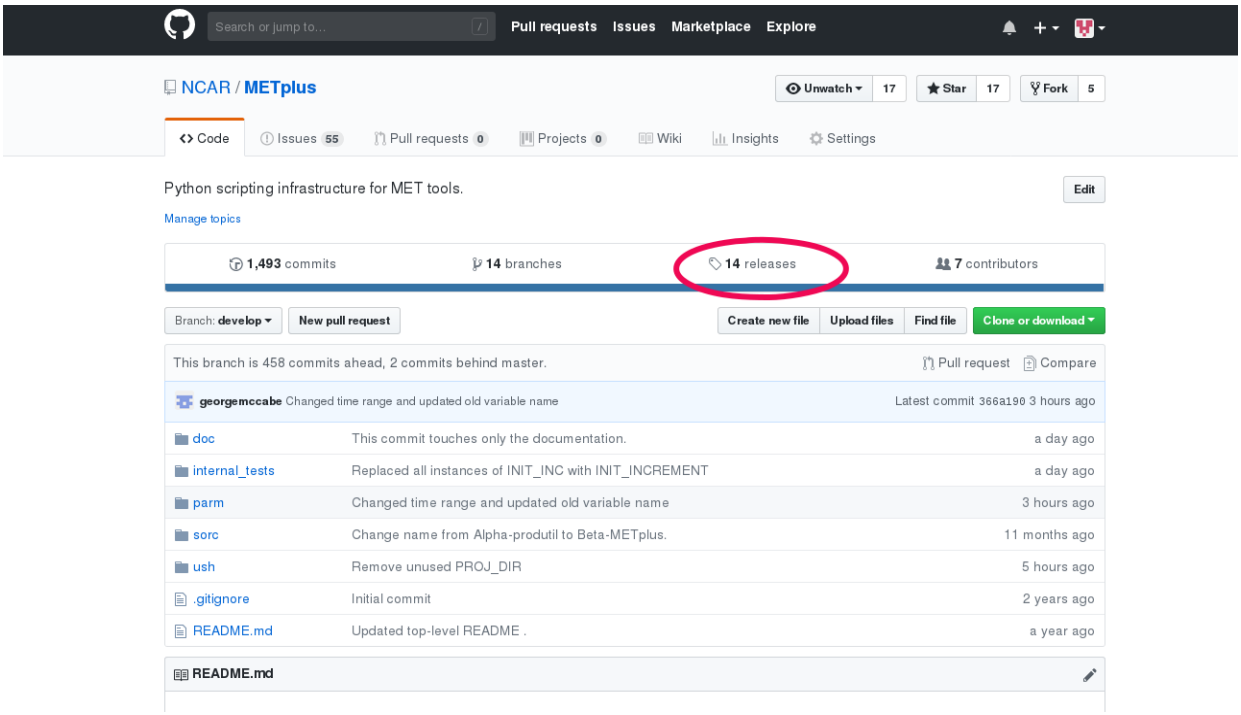
#### 2.6.1.2 Source code, additional documentation, and sample data

If you are a new METplus user and would like to experiment with the use cases, you will want to follow these instructions to retrieve the source code, additional documentation and sample data that accompanies the use cases:

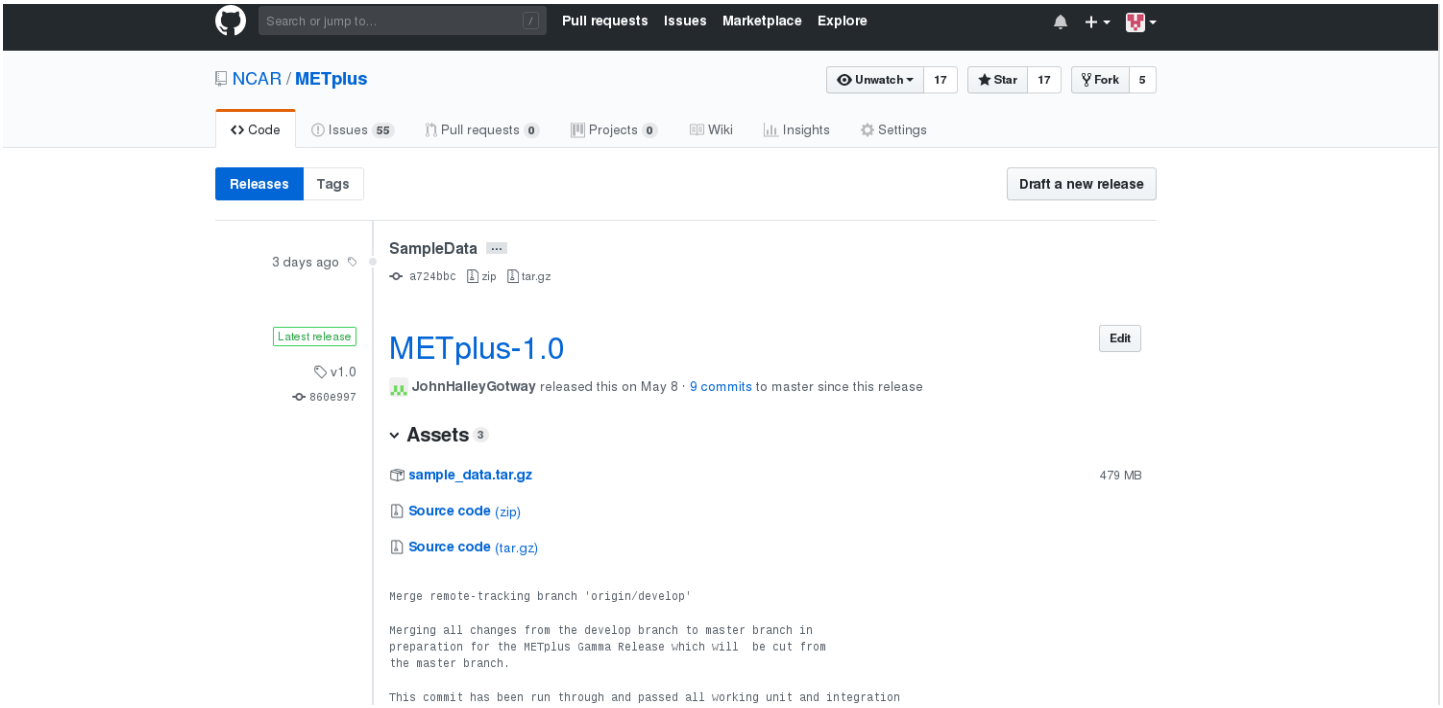
- On your local host (or wherever you wish to install the METplus code) create a directory where you want the code to reside
- Open the browser of your choice and navigate to https://github.com/NCAR/METplus. You will see something like the following:


Search or jump to...
Pull requests
Issues
Marketplace
Explore
NCAR / METplus
Unwatch 17
Star 17
Fork 5
Code
Issues 55
Pull requests 0
Projects 0
Wiki
Insights
Settings
Python scripting infrastructure for MET tools.
Edit
Manage topics
1,037 commits
14 branches
13 releases
7 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
CPKalb Update grid2grid_hwt_env.conf
Latest commit a724bbc 2 hours ago
doc Merge branch 'develop' of https://github.com/NCAR/METplus into develop 8 months ago
internal_tests Updated to reflect changes in the conf file. 5 months ago
parm Update grid2grid_hwt_env.conf 2 hours ago
sorc Change name from Alpha-produtil to Beta-METplus. 11 months ago
ush Accidentally committed GitHub Issue #92 changes to master branch. Rev... 2 months ago
.gitignore Initial commit 2 years ago
.travis.yml try ./ since we don't know how directory is structured in the travis ... 6 months ago
README.md Updated top-level README . a year ago
README.md
METplus Repository README File {#METplus_README}
Welcome to the documentation for the Model Evaluation Tools Plus (METplus).
This is the METplus repository Top level README.md
Basic DOCUMENTATION - getting started
ALL Documentation specific to this repository can be found in the doc/ directory.
The ORIGINAL setup text documentation in a markdown file is found here.
doc/README_install.md --- installation, configuration, running
doc/README_terms_of_use.md --- legal Terms Of Use
METplus is a Python scripting infrastructure around the MET verification tools (and eventually METViewer, a tool used for plotting MET output verification statistics).
This infrastructure utilizes the NCEP produtil package. A Platform-independent weather and ocean forecasting utility package. Developed at the National Oceanic and Atmospheric Administration (NOAA).
Website Documentation
Users can generate an entire METplus documentation website, if they have Doxygen version 1.8.9.1 or later installed.
cd METplus/sorc
make doc
In your browser, open the page METplus/doc/html/index.html
Terms of Use
@ref METplus_TermsOfUse
Install/Configure/Run Guide
@ref METplus_install_guide
Produtil Guide
@ref METplus_produtil

- Click on the 'releases' link, highlighted by a red circle in the diagram below:




- You will be redirected to another screen. The latest available release appears at the top of the screen:




- Click on the 'Source code' link (either the *zip* or *tar.gz*) and when prompted, save it to the directory you created.

- Uncompress the source code (on Linux/Unix*: gunzip* for zip file or *tar xvfz* for the tar.gz file)
- Create a directory for the sample data directory
- Click on the *sample_ data.tar.gz* link and when prompted, save the file to the directory you created above

### 2.6.2 Get the source code via Command line

- On your local host (or wherever you wish to install the METplus code) create a directory where you want the code to reside
- cd to the directory you just created.
- On the command line, enter the following:
  - *git clone https://github.com/NCAR/METplus*
  - The source code should appear under the METplus directory
- To update your copy, cd to your METplus install directory: */path/to/*METplus and enter *git pull* at the command line

## 2.7 Set up your environment

Environment variables need to be set to allow the METplus application to be run from any directory and for locating the necessary Python modules. There is an option to set the JLOGFILE environment variable, which indicates where JLOGS will be saved. JLOGS provide information pertinent to the configuration-file framework. If this environment is unset, then output from the configuration framework will be directed to stdout (your display).

Add the following information to your .cshrc (C shell) or .bashrc (Bash shell):

**.cshrc:**

- Open your .cshrc file and do the following:
- To your PATH, add: *full-path-to-METplus*/ush
- Optional: add JLOGFILE variable and set to *full-path-to-save-jlog-files*
- Close your .cshrc file and run `source ~/.cshrc`
- *e.g.*
  set path = (other_path_entries ~/METplus/ush
  # optional
  setenv JLOGFILE ~/jlog_out

**.bashrc:**

- Open your .bashrc file and do the following:
- To your PATH, add : *full-path-to-METplus*/ush
- Optional: add a JLOGFILE environment variable and set it to the directory where you want the logs to reside
- Close your .bashrc file and run `source ~/.bashrc`
- *e.g.*
  export PATH=~/METplus/ush:$PATH
  #optional
  export JLOGFILE=~/

## 2.8 Set up METplus Configuration files

There are four METplus configuration files that must be defined prior to running METplus. These configuration files reside in the METplus_INSTALL_DIRECTORY/METplus/parm/metplus_config

The following configuration files are automatically loaded during a METplus run and do not need to be invoked on the command line.

- metplus_data.conf
  - data-relevant settings:
    - filename templates
    - regular expressions for input or output filenames
    - directories where input data are located
- metplus_logging.conf
  - set logging levels for METplus and MET output
  - turn on/off logging to stdout (screen) or log files
- metplus_runtime.conf
  - runtime-related settings:
    - location of METplus master_metplus.conf file (the 'master' conf file that is a collection of all the final METplus configuration files)
- metplus_system.conf
  - system-related settings:
    - location of METplus source code

* location of MET source and build
* location of other non-MET executables/binaries
* location of METplus parm directory

They must be fully defined by replacing all variables with */path/to's* with valid path names, or have those variables defined in a down-stream config file. If configuring METplus in a common location for multiple users, it is recommended that the these four configuration files are fully defined. Individual users have the option to make customizations by over-riding any of these values in their own configuration files.

## 2.9 Running METplus

Running METplus involves invoking the Python script master_metplus.py from any directory followed by a list of configuration files (file path relative to the *path_to_METplus_install_dir*/METplus/parm directory).

Example: Using a "default"configuration

create your own config file and under the [config] header/family section, add the following:

```
//This is a comment, comments are defined with a // at the beginning of the line.
// Setting the PROCESS_LIST to Usage indicates that we want usage information [config]
PROCESS_LIST = Usage
// Set the MET_INSTALL_DIR to any real directory. We just need to override the /path/to placeholder
set in the // metplus_system.conf file [dir] MET_INSTALL_DIR = /home/minnawin/latest/METplus
// Set these to any valid directory to override the /path/to placeholder set in the
TMP_DIR = /tmp
PROJ_DIR = /tmp
OUTPUT_BASE = /tmp
```

>master_metplus.py -c ./my_user_config.conf

or

>master_metplus.py -c /username/my_user_config.conf

if you saved your default config in a directory other than where you are running master_metplus.py

A usage message appears, indicating that other config files are required to perform useful tasks and a list of currently supported wrappers:

```
USAGE: This is a default process, please indicate more specific processes in the PROCESS_LIST variable
in one or more of the following configuration files:
-parm/metplus_config/metplus_runtime.conf
-parm/metplus_use_cases/<usecase_name>/<usecase_name>.conf
-parm/metplus_use_cases/<usecase_name>/examples/<example_name>.conf Currently available
processes are:
- TcPairs
- ExtractTiles
- SeriesByInit
- SeriesByLead
- PcpCombine
- RegridDataPlane
- GridStat
- Mode
- MTD
- RegridDataPlane
- CyclonePlotter
- TCMPRPlotter
- PB2NC
- PointStat
```

Example: Using a use-case configuration
>master_metplus.py -c use_cases/feature_relative/feature_relative.conf
Runs METplus using the defaults set in the three config files found in parm/metplus_config. Any variables defined in these three config files can be over-ridden in the parm/use_cases/feature_relative/feature_relative.conf file. METplus will run using the values specified in the feature_relative.conf file.

Example: Using example configuration to perform specific evaluation (e.g. Model 1 vs. Obs1, Model 1 vs Obs 2, Model 2 vs. Obs 1, etc.)
>master_metplus.py -c use_cases/feature_relative/feature_relative.conf \
-c use_cases/feature_relative/example/series_by_lead_all_fhrs.conf
This runs METplus using the defaults set in the three config files found in parm/metplus_config, where variables can be over-ridden by parm/use_cases/feature_relative/feature_relative.conf or in parm/use_cases/feature_relative/example/series_by_lead_all_fhrs.conf. The order in which conf files are called is important. Variables that are defined in intermediate conf files will be over-ridden by the same variables set in the conf file following it, or the last conf file.

# Chapter 3

# METplus Python Wrappers

This chapter provides a description of each supported Python wrapper in METplus. A METplus wrapper is a Python script that encapsulates the behavior of a corresponding MET tool.

## 3.1 compare_ensemble_wrapper

## 3.2 compare_gridded_wrapper

## 3.3 cyclone_plotter_wrapper

## 3.4 ensemble_stat_wrapper

## 3.5 extract_tiles_wrapper

### 3.5.1 Description

The extract_tiles_wrapper.py script is used to regrid and extract subregions from paired tropical cyclone tracks that are created by the tc_pairs_wrapper. Unlike the other METplus wrappers, the extract_tiles_wrapper does not correspond to a specific MET tool. It invokes the tc_stat_wrapper, which in turn calls the MET tc_stat tool to determine the lat/lon positions of the paired track data. This information is then used to create tiles of subregions. The extract_tiles_wrapper creates a 2n degree x 2m degree grid/tile with each storm located at the center.

### 3.5.1 Configuration

The following should be set in the METplus configuration file to define the dimensions and density of the tiles comprising the subregion:

LON_ADJ - set to a value in degrees, found under the [config] header/family. This defines the 2n portion of the 2n x 2m subregion tile.

LAT_ADJ - set to a value in degrees, found under the [config] header/family. This defines the 2m portion of the 2n x 2m subregion tile.

NLAT - set to a whole number, found under the [config] header/family. This defines the number of latitude points to incorporate into the subregion (density).

NLON - set to a whole number, found under the [config] header/family. This defines the number of longitude points to incorporate into the subregion (density).

DLON - set to the value that defines the resolution of the data (in decimal degrees). Found under the [config] header/family section of the METplus config file.

DLAT - set to the value that defines the resolution of the data (in decimal degrees). Found under the [config] header/family section of the METplus config file.

EXTRACT_TILES_FILTER_OPTS - Additional filtering by summary (via the MET tc_stat tool). Please refer to Chapter 20 in the MET Users Guide (TC-STAT Tools) for all the available options for filtering by summary method in tc-stat. If no additional filtering is required, simply leave the value to EXTRACT_TILES_FILTER_OPTS blank/empty in the METplus configuration file. This is located in the [config] header/family section of the METplus config file.

PROCESS_LIST - set this to ExtractTiles to ensure that the extract_tiles_wrapper.py script is run.

## 3.6 grid_stat_wrapper

## 3.7 mode_wrapper

## 3.8 pb2nc_wrapper

### 3.8.1 3.8.1 Description

The pb2nc_wrapper is a Python script that encapsulates the behavior of the MET pb2nc tool to convert prepBUFR files into netCDF.

### 3.8.2 Configuration

The following values must be defined in the METplus configuration file:

PREPBUFR_MODEL_DIR_NAME - set this to the name of the model data (e.g. NAM, GFS, nam, gfs). This is found under the [dir] section/family. The output directory will be given this name.

PREPBUFR_DATA_DIR - The full path to the prepBUFR input data. This variable is found under the [dir] header/family section of the METplus config file.

PB2NC_CONFIG_FILE - The full path to the MET pb2nc configuration file. This variable is found under the [config] section/family section of the METplus config file.

PB2NC_OUTPUT_DIR - The full path to where the netCDF output will be saved. This variable is located under the [config] header/family section of the METplus config file.

PREPBUFR_DIR_REGEX - The regular expression that describes the input prepBUFR directory name. This variable is located under the [regex_pattern] header/family of the METplus config file.

PREPBUFR_FILE_REGEX - The regular expression that describes the input prepBUFR file name. This variable is located under the [regex_patter] section/family of the METplus config file.

NC_FILE_TMPL - The file template that describes the format of the output netCDF files that will be created by MET pb2nc. This variable is located under the [filename_templates] header/family section of the METplus config file.

PROCESS_LIST - set this to PB2NC to ensure that the pb2nc_wrapper.py script is invoked.

## 3.9 pcp_combine_wrapper

## 3.10 point_stat_wrapper

### 3.10.1 Description

The point_stat_wrapper is a Python script that encapsulates the MET point_stat tool. It provides the infrastructure to read in gridded model data and netCDF point observation data to perform grid-to-point (grid-to-obs) verification.

### 3.10.2 Configuration

FCST_INPUT_DIR - the location (full path) to the input point observation data. This variable is located under the [config] header/family section of the METplus config file.

PROCESS_LIST - set to PointStat (to ensure that the point_stat_wrapper is run). This variable is located under the [config] header/family section of the METplus config file.

POINT_STAT_OUTPUT_DIR - full path to the output files, based on the OUTPUT_BASE value (e.g.: POINT_STAT_OUTPUT_DIR = {OUTPUT_BASE}/path/to). This variable is located in the [dir] header/family section of the METplus config file.

START_HOUR - The beginning hour defining the time window of interest in HH format (e.g. 00). This variable is located under the [config] header/family of the METplus config file.

END_HOUR - The ending hour defining the time window of interest in HH format (e.g. 23). This variable is located under the [config] header/family section of the METplus config file.

BEG_TIME - The year, month, day (YYYYMMD) defining the start of the time window of interest. This is located under the [config] header/family section of the METplus config file.

FCST_HR_START - The starting hour of the forecast hour of interest. This variable is located under the [config] header/family section in the METplus config file.

FCST_HR_END - The ending hour of the forecast hour of interest. This variable is located under the [config] header/family section in the METplus config file.

FCST_HR_INTERVAL - The step size/interval in hours used to define all the forecast hours of interest. This variable is located in the [config] header/family section of the METplus config file.

POINT_STAT_POLY - the verification masking region, corrseponding to the poly mask dictionary in the MET point-stat config file. This variable is found under the [config] header/family section of the METplus config file.

POINT_STAT_STATION_ID - corresponds to the sid mask dictionary used for the verification masking regions in the MET point-stat config file. This variable is found under the [config] header/family section of the METplus config file.

The following variables represent the field names, levels, and options in the fcst dictionary of the MET point-stat config file. The FCST_VARn_NAME corresponds to the fcst and obs dictionary:

```
fcst = {
field = [ { name = "${NAME}"; level = [ "${LEVEL}" ]; } ];
}

obs = {
field = [ { name = "${NAME}"; level = [ "${LEVEL}" ]; } ];
}
```

Options can be cat_thresholds or GRIB_lvl_val = ###, where ### represents a numerical grib level value.

FCST_VAR1_NAME

FCST_VAR1_LEVELS

FCST_VAR1_OPTIONS

...

FCST_VARn_NAME

FCST_VARn_LEVELS

FCST_VARn_OPTIONS

## 3.11 reformat_gridded_wrapper

## 3.12 regrid_data_plane_wrapper

## 3.13 series_by_init_wrapper

### 3.13.1 Description

The series_by_init_wrapper provides the infrastructure needed to perform a series analysis on tropical cyclone data, based on initialization times. The series_by_init_wrapper creates numerous plots that represent the field, level, and statistic for each initialization time.

### 3.13.2 Configuration

ADECK_TRACK_DATA_DIR - The full filepath to the Adeck data files. This variable is found under the [dir] header/family section in the METplus config file.

BDECK_TRACK_DATA_DIR - The full filepath to the Bdeck (Best track) data files. This variable is found under the [dir] header/family section of the METplus config file.

INIT_BEG - set this to the starting YYYYMMDD of the time window of interest (e.g. 20180601). This variable is found under the [config] header/family section of the METplus config file.

INIT_END - set this to the ending YYYYMMDD of the time window of interest (e.g. 20180615). This variable is found under the [config] header/family section of the METplus config file.

INIT_INCREMENT - set this to the step size/increment (in seconds) between your beginning and ending init times of interest (eg. 21600, which corresponds to 6 hours). Tjis variable is located under the [config] header/family section of the METplus config file.

INIT_HOUR_END - This is the ending hour to of the init hour time window (e.g. INIT_HOUR_END = 23 to end your ini time at 23Z). This variable is located under the [config] header/family section of the METplus config file.

INIT_INCLUDE - This is a list of dates in YYYYMMDD_hh format data that might reside outside your specified time window of interest but want to include in your verification (e.g. INIT_INCLUDE = 20180616_12). This variable is located under the [config] header/family section of the METplus config file.

INIT_EXCLUDE - This is a list of dates in YYYYMMDD_hh format of data that you wish to exclude from your verification. (e.g. If, for some reason, you wish to exclude the 20180611_12, 20180613_06 data and include the 20180616_12 data, you would specify it in the following manner: INIT_EXCLUDE = 20180611_12, 20180613_06). This variable is located under the [config] header/family section of the METplus config file.

SERIES_ANALYSIS_FILTER_OPTS - Apply tc_stat syntax for performing any additional filtering on your input data, which is done via tc_stat_wrapper/tc_stat tool (e.g. SERIES_ANALYSIS_FILTER_OPTS = -init_beg {INIT_BEG} -init_end {INIT_END} . This limits results that lie within the INIT_BEG and INIT_END times that were specified in your METplus configuration file, Refer to Chapter 20 of the MET User's Guide for the syntax to use for performing filtering via the MET tc_stat tool).

## 3.14 series_by_lead_wrapper

### 3.14.1 Description

The series_by_lead_wrapper provides the infrastructure needed to perform a series analysis on tropical cyclone data, based on lead (forecast hour) times. The series_by_lead_wrapper creates numerous plots that represent the field, level, and statistic for each lead (forecast) time. The series_by_lead can be done in one of two ways: by all forecast hours or by forecast hour groupings. Performing a series analysis by valid time with forecast hour groupings can be useful when analyzing storm tracks based on time "bins" such as by days (eg. day 1, day 2, day 3, etc.).

### 3.14.2 Configuration

The input track and model data files are defined in any one of the user's METplus configuration files. If creating a final configuration file that overrides all other config files, it is customary to define the

MODEL_DATA_DIR, pointing to the directory where all model data resides. The full file path to the INIT_INCLUDE and INIT_EXCLUDE are used to list the times in YYYYMMDD_HH format to include or exclude from your time window. If these values are undefined (i.e. no value is set for the variable), then all available times in your time window will be considered. For example, if your data is available every 6 hours and you are interested in creating a series analysis from init time 20180601 to 20180615 for all available times, from 00z to 23z, you would set the following:

INIT_BEG = set this to the starting YYYYMMDD of the time window of interest (e.g. 20180601). This variable is found under the [config] header/family section of the METplus config file.

INIT_END = set this to the ending YYYYMMDD of the time window of interest (e.g. 20180615). This variable is found under the [config] header/family section of the METplus config file.

INIT_INCREMENT = set this to the step size/increment (in seconds) between your beginning and ending init times of interest (eg. 21600, which corresponds to 6 hours)

INIT_HOUR_END = 23

FHR_BEG - The beginning forecast hour of interest.

FHR_END - The ending forecast hour of interest.

FHR_INC - The increment/time step in hours.

The following should be undefined if you are performing a series analysis for all available forecast hours. Otherwise, define these if you wish to perform a series analysis by lead time (to create your own forecast hour groupings and associated labels).

FHR_GROUP_BEG - This is a list of forecast hours that define each "block" or "grouping" of forecast hours. The number of beginning forecast hours must match the number of ending forecast hours. This variable is found under the [config] header/family section of the METplus config file.

FHR_GROUP_END - This is a list of forecast hours that define the end of each "block" or "grouping" of forecast hours. The number of ending forecast hours must match the number of beginning forecast hours. This variable is found under the [config] header/family section of the METplus config file.

FHR_GROUP_LABELS - This is a list of the labels to be applied for each "block" or "grouping" of forecast hours. The number of labels must equal the number of forecast hour begin and end items. This variable is found under the [config] header/family section of the METplus config file.

INIT_INCLUDE - This is a list of any data you wish to include that might lie outside your specified time window (e.g. INIT_INCLUDE=20180616_12 would include any data corresponding to 20180616_12). This is found under the [config] header/family section of the METplus config file.

INIT_EXCLUDE - A list of any data that you wish to exclude from your verification. If, for some reason, you wish to exclude the 20180611_12, 20180613_06 data and include the 20180616_12 data, you would specify

it in the following manner: INIT_EXCLUDE = 20180611_12, 20180613_06. This variable is located under the [config] header/family section of the METplus config file.

SERIES_ANALYSIS_FILTER_OPT - Perform additional filtering on your input data, via the tc_stat_wrapper/MET tc_stat tool. Refer to Chapter 20 of the MET User's Guide for the syntax to use for performing filtering via the MET tc_stat tool.

## 3.15 stat_analysis_wrapper

## 3.16 tc_pairs_wrapper

### 3.16.1 Description

The tc_pairs_wrapper encapsulates the behavior of the the MET tc_pairs tool. The wrapper accepts Adeck and Bdeck (Best track) cyclone track data in extra tropical cyclone format (such as the data used by sample data provided in the METplus tutorial), or ATCF formatted track data. If data is in an extra tropical cyclone (non-ATCF) format, the data is reformatted into an ATCF format that is recognized by MET.

### 3.16.2 Configuration

TC_PAIRS_CONFIG_FILE - The full path to the MET tc_pairs config file. This variable is located under the [config] header/family section of the METplus config file.

INIT_BEG - The start of the initialization time window of interest, in YYYYMMDD format. This variable is located under the [config] header/family section of the METplus config file.

INIT_END - The end of the initialization time window of interest in YYYYMMDD format. This variable is located under the [config] header/family section of the METplus config file.

INIT_INCREMENT - The time step/increment in seconds to be used to determine which data files will be used/considered in the verification. (e.g. INIT_INCREMENT = 21600 sets the time step size to 6 hours, so data that will be considered will be INIT_BEG, INIT_BEG + INIT_INCREMENT, until the INIT_END is reached). This variable is found under the [config] header/family section of the METplus config.

INIT_HOUR_END - The cutoff for the last date to be considered in the time window (eg. if INIT_HOUR_END=18 and INIT_END=20190121, then the last date to be included in the time window is 20190121_18). This variable is located in the [config] header/family section of the METplus config file.

INIT_INCLUDE - The list of initialization times in YYYYMMDD_hh format to include in the verification (e.g. INIT_INCLUDE = 20170601_00, 20170601_12, 20170602_6). This variable is located in the [config] header/family of the METplus config file.

INIT_EXCLUDE - The list of initialization times in YYYYMMDD_hh format to be excluded from the verification (e.g. INIT_EXCLUDE = 20190121_06, 20181231_23). This variable is located in the [config] header/family section of the METplus config file.

TOP_LEVEL_DIRS - This defines how to run the MET tc-pairs tool. If set to 'yes', then the MET tc-pairs tools will search the input file directory and all its subdirectories for input data. If set to 'no', then

MODEL - The list of models to include in the verification. If left empty/unassigned, then all models in the track files will be considered. This variable corresponds to the model dictionary value in the MET config file. This variable is located under the [config] header/family section of the METplus config file.

STORM_ID - The list of storm ids to include in the verification. If left empty/unassigned (ie STORM_ID = ), then all storm ids in the track files will be included in the verification. This variable is located under the [config] header/family section of the METplus config file.

BASIN - A list of basins of interest. If left empty/undefined, then all basins are included in the verification. This variable is found under the [config] header/family section of the METplus config file.

CYCLONE - A list of cyclones to consider in the verification. If this is left empty/unassigned, then all cyclones will be considered in the verification. This variable is found in the [config] header/family section of the METplus config file.

STORM_NAME - A list of storm names to include in the verification. If this is left empty/unassigned, then all storm names are included in the verification. This variable is found in the [config] header/family section of the METplus config file.

DLAND_FILE - The full path of the file that has the gridded representation of the minimum distance from land. This variable is found under the [config] header/family section of the METplus config file.

TRACK_TYPE - This is used to determine whether the Adeck and Bdeck data are in ATCF format or not. If left empty/unassigned, then the input data is in ATCF format and the wrapper does not attempt to perform any reformatting. If set to 'extra_tropical_cyclone' (this is the non-ATCF format of data like that provided in the sample data in the tutorial tar ball) then the wrapper will reformat it to an ATCF format (which the MET tc_pairs tools requires). This variable is found in the [config] header/family section of the METplus config file.

ADECK_FILE_PREFIX - This is the prefix to the Adeck files if the input Adeck file is the same format as the data included in the sample data tutorial tar ball. Leave this empty/undefined when dealing with ATCF formatted data. This variable is found under the [config] header/family section of the METplus config file.

BDECK_FILE_PREFIX - This is the prefix that describes the Bdeck files when the input Bdeck file is the same format as the data included in the sample data tutorial tar ball. Leave this empty/undefined if your data is ATCF formatted. This variable is found in the [config] header/family section of the METplus config file.

MISSING_VAL_TO_REPLACE - This is applicable only if using the sample data in the tutorial tar ball, which is the extra tropical cyclone data (i.e. data that is in a non-ATCF format). This is the value used to replace the value of the data's original missing value.

MISSING_VAL - For non-ATCF, extra tropical cyclone data (such as the data supplied in the feature_relative tutorial tar ball), this is the value used to define missing values.

## 3.17 tc_stat_wrapper

## 3.18 tcmpr_plotter_wrapper

## 3.19 wavelet_stat_wrapper

# Chapter 4

# METplus System Configuration

This chapter is a guide on configuring METplus.

## 4.1 Config Best Practices

Below is a list of Best Practices:

1. Set your log level to an appropriate level.
   (a) Debug is the most verbose and is useful for developers and when you are troubleshooting problems
   (b) Info is the less verbose than Debug and is the recommended level to initially set your log level
   (c) Warning - only logs warnings, error or critical events
   (d) Error - only logs errors or critical events
   (e) Critical is the least verbose
2. Direct your logging either to stdout or to a log file.
3. Review your log file to verify that all your processes ran cleanly.
4. The order in which you list your METplus config files matter. The last config file on the command line will over-ride any key-values defined in an earlier config file.
5. Check the master_metplus.conf file, as it contains all the key-values based on what you have specified. This will help you determine whether you forgot to replace any */path/to* with valid paths or to verify that you have defined things as you expected.

## 4.2 Config File Structure

METplus employs a hierarchy of configuration files employed in METplus. At the lowest level are the "set-and-forget" type configuration files that reside in the *METplus_installation_dir*/parm/metplus_configl At the next level are the configuration files that pertain to a user's specific needs in the *METplus_installation_dir*/parm/use_cases/*specific_use_case*

- Four configuration files are required for METplus to be fully configured (i.e. all keywords are defined by either whitespace or a valid value):
  - metplus_system
  - metplus_data
  - metplus_logging
  - metplus_runtime

  By default, key-values that require the user's input are set to */path/to*. Make sure to replace these with the appropriate directory for your project.
- Additional configuration files are optional and the key-values defined there will over-ride any values defined in the four mandatory METplus configuration files. These additional configuration files enables users to use a common set of configuration files and to create customized environments for their verification tasks.

## 4.3 Config Quick Start Example

**Track and Intensity Use case with sample data**

- Create a directory where you wish to store the sample data
- Retrieve the sample data from the GitHub repository:
  - In your browser, navigate to https://github.com/NCAR/METplus/releases
  - locate the latest release and click on the sample_data.tar.gz link associated with that release
  - save it to the directory you created above, hereafter referred to as INPUT_DATA_DIRECTORY
  - cd to your $INPUT_DATA_DIRECTORY and uncompress the tarball: *tar xvfz sample_data.tar.gz*
  - when you perform a listing of the sample_data directory, the INPUT_DATA_DIRECTORY/sample_data/GFS contains the data you will need for this use case
- Set up the configuration file:
  - Your METplus install directory will hereafter be referred to as METplus_INSTALL

  - Verify that all the *path/to* values are replaced with valid paths in the METplus_INSTALL/parm/metplus_conf/met and METplus_INSTALL/parm/metplus_conf/metplus_system.conf files
  - Two configuration files are used in this use case, track_and_intensity.conf file and tcmp_mean_median.conf to take cyclone track data, and using tc_pairs_wrapper.py which wraps the MET TC-Pairs tool (to match ADeck and BDeck cyclone tracks to generate matched pairs and error statistics). The tcmpr_plotter_wrapper.py is then used (wraps the MET tool plot_tcmpr.R) to generate a mean and median plots for these matched pairs.
  - In your editor, open the METplus_INSTALL/METplus/parm/use_cases/track_and_intensity.conf file:
    * You will replace any */path/to* with actual paths by setting the following:
    * PARM_BASE to the path to where you installed METplus, appended with with '*parm*': METplus_INSTALL/all_users/METplus/*parm*
    * OUTPUT_BASE to where you wish to save the output:
      · ADECK_TRACK_DATA_DIR to INPUT_DATA_DIRECTORY/sample_data/GFS/track_data
    * save your changes and exit your editor
    * In your editor, open the METplus_INSTALL/METplus/parm/use_cases/track_and_intensity/examples/tcm
    * Verify that PROCESS_LIST is set to TcPairs, TCMPRPlotter. This instructs METplus to run the TcPairs wrapper first (TC-Pairs) followed by the TCMPR plotter wrapper (plot_TCMPR.R).

- Run the use case:
  - Make sure you have set the following environment in your .cshrc (C shell) or .bashrc (Bash):
    * csh: setenv RSCRIPTS_BASE $MET_BASE/scripts/Rscripts
    * bash: export RSCIPTS_BASE $MET_BASE/scripts/Rscripts
    * Refer to Section 2.7 for the full instructions on setting up the rest of your environment
    * on your command line, run:
      · master_metplus.py -c use_cases/track_and_intensity/track_and_intensity.conf -c use_cases/track_and
    * When complete, you will have a log file in the output directory you specified, and under the tc_pairs directory you will see .tcst files under the 201412 subdirectory. These are the matched pairs created by the MET tool Tc-pairs and can be viewed in any text editor.
    * Plots are generated under the tcmpr_plots subdirectory, in .png format. You should have the following plots which can be viewed by any graphics viewers such as 'display' on Linux/Unix hosts:
      · AMAX_WIND-BMAX_WIND_mean.png
      · AMAX_WIND-BMAX_WIND_median.png
      · AMSLP-BMSLP_mean.png
      · AMSLP-BMSLP_median.png
      · TK_ERR_mean.png
      · TK_ERR_median.png

## 4.4 A-Z Config Glossary

This glossary was created from the two commands:
$ cat METplus/parm/metplus_config/*.conf METplus/parm/use_cases/*/*.conf METplus/parm/use_cases/*/*/*.conf > allopts.conf
$ grep = allopts.conf | grep -v \# | sort | uniq > uniqueopts.conf

General form of glossary entry:

**CONFIG_NAME_HERE**
...Some description here...

***Used by:*** Which METplus utility is this used by?
***Family:*** Which family? [dir], [config], [filename_templates], [exe], [regex_pattern], etc...
***Default:*** If it makes sense to include a default value (or value shipped in a release), do it here

---

### 4.4.1 A

---

**ADECK_FILE_PREFIX**
Prefix of the files in ATCF format containing tropical cyclone forecast data ("adeck" matched pairs).

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**ADECK_TRACK_DATA_DIR**
Directory that contains the ATCF formatted files containing tropical cyclone forecast data ("adeck" matched pairs).

***Used by:*** tc_pairs_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**AMODEL**
The model name of the ADeck model data

***Used by:*** cyclone_plotter_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:***

---

**ANLY_ASCII_REGEX_LEAD**

The regular expression describing the analysis (obs) file name (in ASCII format) of the intermediate file generated when running a series by lead case.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [regex_pattern]
***Default:***

---

**ANLY_NC_TILE_REGEX**
The regular expression used to search the input files that are in netCDF format and used in the series by analysis task.

***Used by:*** series_by_lead_wrapper.py, series_by_init_wrapper.py
***Family:*** [regex_pattern]
***Default:***

---

**ANLY_TILE_PREFIX**
The prefix to the filename for the analysis file that is created as part of a series analysis.

***Used by:*** feature_util.py
***Family:*** [regex_pattern]
***Default:***

---

**ANLY_TILE_REGEX**
The regular expression for the analysis input file the file is in GRIB2.

***Used by:*** series_by_lead_wrapper.py, series_by_init_wrapper.py
***Family:*** [regex_pattern]
***Default:***

### 4.4.2 B

---

**BACKGROUND_MAP**
Control whether or not a background map shows up for series analysis plots. Set to 'yes' if background map desired.

***Used by:*** series_by_lead_wrapper.py, series_by_init_wrapper.py
***Family:*** [config]
***Default:*** no

---

**BASIN**
Control what basins are desired for tropical cyclone analysis.

Per the MET users' guide, acceptable basin ID's are:
WP = Western Northern Pacific
IO = Northern Indian Ocean
SH = Southern Hemisphere
CP = Central Northern Pacific
EP = Eastern Northern Pacific
AL = Northern Atlantic
SL = Southern Atlantic

***Used by:*** cyclone_plotter_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**BDECK_FILE_PREFIX**

Relevant for non-ATCF tropical cyclone data. The filename prefix for the BDeck data.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**BDECK_TRACK_DATA_DIR**
The input directory where the BDeck track data resides.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**BEG_TIME**
Beginning time for analysis in YYYYMMDD format.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**BMODEL**
The model name of the BDeck model data.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:***

---

### 4.4.3 C

---

**CIRCLE_MARKER_SIZE**
Control the size of the circle marker in the cyclone plotter.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** 41

---

**CLOCK_TIME**
Automatically set by METplus with the time that METplus was started. Setting this variable has no effect as it will be overwritten. Can be used for reference in metplus_final.conf or used with other config variables.

***Used by:*** All
***Family:*** [config]
***Default:*** Set automatically to current clock time in %Y%m%d%H%M%S format

---

**CONFIG_DIR**
Directory containing config files relevant to MET tools.

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**CONFIG_FILE**
Specific configuration file name to use for MET tools.

***Used by:*** grid_stat_wrapper.py, mode_wrapper.py, tcmpr_plotter_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**CONVERT_EXE**
Path to the ImageMagick "convert" executable.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**CROSS_MARKER_SIZE**
Control the size of the cross marker in the cyclone plotter.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** 51

---

**CUT_EXE**
Path to the Linux "cut" executable.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**CYCLONE**
Specify which cyclone numbers to include in the tropical cyclone analysis. Per the MET users' guide, this can be any number 01-99 (HH format). Use a space or comma separated list, or leave unset if all cyclones are desired.

***Used by:*** tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**CYCLONE_INIT_DATE**
Initialization date for the cyclone forecasts in YYYYMMDD format.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**CYCLONE_INIT_HR**
Initialization hour for the cyclone forecasts in HH format.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**CYCLONE_INPUT_DIR**
Input directory for the cyclone plotter. This should be the output directory for the MET TC Pairs utility.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**CYCLONE_MODEL**
Define the model being used for the tropical cyclone forecasts.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**CYCLONE_OUT_DIR**
Specify the directory where the output from the cyclone plotter should go.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**CYCLONE_PLOT_TITLE**
Title string for the cyclone plotter.

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.4 D

---

**DEMO_YR**
The demo year. This is an optional value used by the plot_TCMPR.R script, (which is wrapped by tcmpr_plotter_wrapper.py). Please refer to Chapter 21 in the MET User's Guide for more details.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**DEP_VARS**
Corresponds to the optional flag -dep in the plot_TCMPR.R, which is wrapped by tcmpr_plotter_wrapper.py. The value to this flag is a comma-separated list of dependent variable columns to plot. Please refer to Chapter 21 in the MET User's Guide for more details.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**DLAND_FILE**
The file generated by the MET tool tc_dland, containing the gridded representation of the minimum distance to land. Please refer to Chapter 18 of the MET User's Guide for more information about the tc_dland tool.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**DLAT**
The latitude value, in degrees.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** 0.5

---

**DLON**
The longitude value, in degrees.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** 0.5

---

**DO_NOT_RUN_EXE**
True/False. If True, applications will not run and will only output command that would have been called.

***Used by:*** command_runner.py
***Family:*** [config]
***Default:*** False

---

### 4.4.5 E

---

**EGREP_EXE**
Path to the Linux "egrep" executable.

***Used by:*** feature_util.py, pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**END_DATE**
Ending time/date string for analysis with format YYYYMMDDHH.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**END_HOUR**
Ending hour for analysis with format HH.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**END_TIME**
Ending date string for analysis with format YYYYMMDD.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**EXTRACT_OUT_DIR**
Set the output directory for the METplus extract_tiles utility.

***Used by:*** extract_tiles_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**EXTRACT_TILES_FILTER_OPTS**
Control what options are passed to the METplus extract_tiles utility.

***Used by:*** extract_tiles_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**EXTRACT_TILES_VAR_LIST**
Control what variables the METplus extract_tiles utility runs on.

***Used by:*** feature_util.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.6 F

---

**FCST_EXACT_VALID_TIME**
Used when setting FCST_* variables to process observation data for comparisons. See OBS_EXACT_VALID_TIME. Rarely used.

***Used by:*** grid_stat_wrapper.py mode_wrapper.py, mtd_wrapper.py
***Family:*** [config]
***Default:*** False

---

**FCST_[N]_FIELD_NAME**
This variable is used to define a [N] hour accumulation NetCDF field in the forecast dataset used in the MET tool pcp_combine. [N] must be an integer >= 1.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_ASCII_REGEX_LEAD**
Regular expression used to find the forecast file (ASCII format) generated as an intermediate step in the series by lead use case.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**[deprecated] FCST_GEMPAK_INPUT_DIR**
Input directory for GEMPAK formatted forecast files. Use GEMPAKTOCF_INPUT_DIR if GempakToCF is in the PROCESS_LIST.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**[deprecated] FCST_GEMPAK_TEMPLATE**
Template used to specify input filenames for GEMPAK formatted forecast files. Use GEMPAKTOCF_INPUT_TEMPLATE if GempakToCF is in the PROCESS_LIST.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_GRID_STAT_INPUT_DATATYPE**
Specify the data type of the input directory for forecast files used with the MET grid_stat tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_GRID_STAT_INPUT_DIR**
Input directory for forecast files to use with the MET tool grid_stat.

***Used by:*** grid_stat_wrapper.py
***Family:*** [dir]

***Default:*** Varies

---

**FCST_GRID_STAT_INPUT_TEMPLATE**
Template used to specify forecast input filenames for the MET tool grid_stat.

***Used by:*** grid_stat_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_GRID_STAT_PROB_THRESH**
Threshold values to be used for probabilistic data in grid_stat. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le).

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** ==0.1

---

**FCST_HR_END**
Specify the maximum forecast hour to use.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_HR_INTERVAL**
Specify the stride for forecast lead times.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_HR_START**
Specify the starting forecast hour to use.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**[deprecated] FCST_INIT_INTERVAL**
Specify the stride for forecast initializations.

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_INPUT_DIR_REGEX**
Specify the regular expression used for searching for forecast file input directories.

***Used by:*** point_stat_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**[deprecated] FCST_INPUT_DIR**
Specify the input directory for the forecast files. Use FCST_[MET-APP]_INPUT_DIR instead, i.e. FCST_GRID_STAT_IN

***Used by:*** compare_gridded_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py, point_stat_wrapper.py, pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**FCST_INPUT_FILE_REGEX**
Regular expression to use when identifying which forecast file to use.

***Used by:*** point_stat_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**FCST_INPUT_FILE_TMPL**
Specify the filename template for input forecast files.

***Used by:*** point_stat_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_IS_DAILY_FILE**
Specify whether the forecast file is a daily file or not.

Acceptable values: true/false

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** False

---

**FCST_IS_PROB**
Specify whether the forecast data are probabilistic or not.

Acceptable values: true/false

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py
***Family:*** [config]

***Default:*** False

---

**FCST_LEVEL**
Specify what accumulation level should be used from the forecast data for the analysis. Used only when running pcp_combine with SUBTRACT mode set or processing accumulation files that do not have the accumulation specified in the filename template.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**[deprecated] FCST_MAX_FORECAST**
Specify the maximum forecast lead time to use for the analysis.

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_MODE_INPUT_DATATYPE**
Specify the data type of the input directory for forecast files used with the MET mode tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_MODE_INPUT_DIR**
Input directory for forecast files to use with the MET tool mode.

***Used by:*** mode_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**FCST_MODE_INPUT_TEMPLATE**
Template used to specify forecast input filenames for the MET tool mode.

***Used by:*** mode_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_MTD_INPUT_DATATYPE**
Specify the data type of the input directory for forecast files used with the MET mode-TD tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_MTD_INPUT_DIR**
Input directory for forecast files to use with the MET tool mode-TD.

***Used by:*** mtd_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**FCST_MTD_INPUT_TEMPLATE**
Template used to specify forecast input filenames for the MET tool mode-TD.

***Used by:*** mtd_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**[deprecated] FCST_NATIVE_DATA_TYPE**
Specify the data format of the forecast data. Use FCST_PCP_COMBINE_INPUT_DATATYPE instead

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_NC_TILE_REGEX**
Define the regular expression for input forecast files that are in netCDF.

***Used by:*** series_by_lead_wrapper.py, series_by_init_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**FCST_PCP_COMBINE_INPUT_DATATYPE**
Specify the data type of the input directory for forecast files used with the MET pcp_combine tool. Currently valid options are NETCDF, GRIB, and GEMPAK. Required by pcp_combine if FCST_PCP_COMBINE_RUN is True. Replaces deprecated variable FCST_NATIVE_DATA_TYPE.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_PCP_COMBINE_INPUT_DIR**
Specify the input directory for forecast files used with the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]

***Default:*** Varies

---

**FCST_PCP_COMBINE_INPUT_TEMPLATE**
Template used to specify input filenames for forecast files used by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_PCP_COMBINE_OUTPUT_DIR**
Specify the output directory for forecast files generated by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**FCST_PCP_COMBINE_OUTPUT_TEMPLATE**
Template used to specify output filenames for forecast files generated by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_PCP_COMBINE_RUN**
Specify whether to run the MET pcp_combine tool on forecast data or not.

Acceptable values: true/false

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_INPUT_DATATYPE**
Specify the data type of the input directory for forecast files used with the MET regrid_data_plane tool. Currently valid options are NETCDF, GRIB, and GEMPAK. Required by pcp_combine.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_INPUT_DIR**
Specify the input directory for forecast files used with the MET regrid_data_plane tool.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [dir]

***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE**
Template used to specify input filenames for forecast data used by the MET regrid_data_plane tool. It not set, METplus will use FCST_REGRID_DATA_PLANE_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_OUTPUT_TEMPLATE**
Template used to specify output filenames for forecast data used by the MET regrid_data_plane tool. It not set, METplus will use FCST_REGRID_DATA_PLANE_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_TEMPLATE**
Template used to specify filenames for forecast data used by the MET regrid_data_plane tool. To specify different templates for input and output files , use FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE and FCST_REGRID_DATA_PLANE_OUTPUT_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FCST_REGRID_DATA_PLANE_OUTPUT_DIR**
Specify the output directory for forecast files used with the MET regrid_data_plane tool.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**FCST_TILE_PREFIX**
Prefix for forecast tile files. Used to create filename of intermediate files that are created while performing a series analysis.

***Used by:*** feature_util.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**FCST_TILE_REGEX**
Regular expression for forecast input files that are in GRIB2.

***Used by:*** series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**[deprecated] FCST_VAR**

Define the name of the forecast variable to be used in the analysis. See FCST_VAR[N]_NAME, FCST_VAR[N]_LEVELS, FCST_VAR[N]_THRESH, and FCST_VAR[N]_OPTIONS where [N] = integer >= 1.

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, make_plots_wrapper.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_VAR[N]_LEVELS**

Define the levels for the [N]th forecast variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items. You can define NetCDF levels, such as (0,*,*), but you will need to surround these values with quotation marks so that the commas in the item are not interpreted as an item delimeter. Some examples:

FCST_VAR1_LEVELS = A06, P500
FCST_VAR2_LEVELS = "(0,*,*)", "(1,*,*)"

If FCST_VAR[N]_LEVELS is not set but OBS_VAR[N]_LEVELS is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_LEVELS
FCST_VAR2_LEVELS
...
FCST_VAR[N]_LEVELS

***Used by:*** make_plots_wrapper.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_VAR[N]_NAME**

Define the name for the [N]th forecast variable to be used in the analysis where [N] is an integer >= 1. If FCST_VAR[N]_NAME is not set but OBS_VAR[N]_NAME is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_NAME
FCST_VAR2_NAME
...
FCST_VAR[N]_NAME

***Used by:*** make_plots_wrapper.py, met_util.py

***Family:*** [config]
***Default:*** Varies

---

**FCST_VAR[N]_OPTIONS**
Define the options for the [N]th forecast variable to be used in the analysis where [N] is an integer >= 1. These addition options will be applied to every name/level/threshold combination for VAR[N]. If FCST_VAR[N]_OPTIONS is not set but OBS_VAR[N]_OPTIONS is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_OPTIONS
FCST_VAR2_OPTIONS
...
FCST_VAR[N]_OPTIONS

***Used by:*** make_plots_wrapper.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**FCST_VAR[N]_THRESH**
Define the threshold(s) for the [N]th forecast variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le). If FCST_VAR[N]_THRESH is not set but OBS_VAR[N]_THRESH is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_THRESH
FCST_VAR2_THRESH
...
FCST_VAR[N]_THRESH

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_BEG**
Specify the first forecast lead time to use in the analysis. Use in combination with FHR_END and FHR_INC.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_END**
Specify the last forecast lead time to use in the analysis. Use in combination with FHR_BEG and FHR_INC.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_GROUP_BEG**
Define which forecast lead time should be first in a group of forecast leads to use in the analysis. Use in combination with FHR_GROUP_END and FHR_INC.
Example:
FHR_GROUP_BEG = 24
FHR_GROUP_END = 42
FHR_INC = 6

List of forecast leads processed: [24, 30, 36, 42]

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_GROUP_END**
Define which forecast lead time should be the last in a group of forecast leads to use in the analysis. Use in combination with FHR_GROUP_BEG and FHR_INC.
Example:
FHR_GROUP_BEG = 24
FHR_GROUP_END = 42
FHR_INC = 6

List of forecast leads processed: [24, 30, 36, 42]

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_GROUP_LABELS**
Label strings to use for the forecast groups.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FHR_INC**
Stride to use for incrementing forecast lead times used in the analysis. Use in combination with FHR_BEG and FHR_END or FHR_GROUP_BEG and FHR_GROUP_END.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**FILTER**

Corresponds to the optional -filter argument to the plot_TCMPR.R script which is wrapped by tcmpr_plotter_wrapper.py. This is a list of filtering options for the tc_stat tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FILTERED_TCST_DATA_FILE**

Corresponds to the optional -tcst argument to the plot_TCMPR.R script which is wrapped by tcmpr_plotter_wrapper.py. This is a tcst data file to be used instead of running the tc_stat tool. Indicate a full path to the data file.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FOOTNOTE_FLAG**

This corresponds to the optional -footnote flag in the plot_TCMPR.R script which is wrapped by tcmpr_plotter_wrapper.py. According to the plot_TCMPR.R usage, this flag is used to disable footnote (date).

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**FORECAST_TMPL**

Filename template used to filter forecast files.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**FOURIER_HEIGHT_DECOMP**

Specify whether to perform a Fourier height decomposition or not.

Acceptable values: true/false

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.7 G

---

**GEMPAKTOCF_INPUT_DIR**
Specify the input directory for the tool used to convert GEMPAK files to netCDF.

***Used by:*** gempak_to_cf_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**GEMPAKTOCF_INPUT_TEMPLATE**
Filename template used for input files to the tool used to convert GEMPAK files to netCDF.

***Used by:*** gempak_to_cf_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**GEMPAKTOCF_OUTPUT_DIR**
Specify the output directory for files generated by the tool used to convert GEMPAK files to netCDF.

***Used by:*** gempak_to_cf_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**GEMPAKTOCF_OUTPUT_TEMPLATE**
Filename template used for output files from the tool used to convert GEMPAK files to netCDF.

***Used by:*** gempak_to_cf_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**GENERATE_TRACK_ASCII**
Specify whether or not to produce an ASCII file containing all of the tracks in the plot.

Acceptable values: true/false

***Used by:*** cyclone_plotter_wrapper.py
***Family:*** [conf]
***Default:*** Varies

---

**[deprecated] GEN_SEQ**

***Used by:***
***Family:***

***Default:***

---

**GFS_ANLY_FILE_TMPL**
Filename template used to identify the GFS analysis file.

***Used by:*** feature_util.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**GFS_FCST_FILE_TMPL**
Filename templated used to identify the GFS forecast files.

***Used by:*** feature_util.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**GRID_STAT_CONFIG**
Specify the absolute path to the configuration file used by the MET grid_stat tool.

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**GRID_STAT_ONCE_PER_FIELD**
STrue/False. If True, grid_stat will run once to process all name/level/threshold combinations specified. If False, it will run once for each name/level. Some cases require this to be set to False, for example processing probablistic forecasts or precipitation accumulations.

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** False

---

**GRID_STAT_OUT_DIR**
Specify the output directory where files from the MET grid_stat tool are written.

***Used by:*** grid_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

### 4.4.8 H

---

**HFIP_BASELINE**

Corresponds to the optional -hfip_bsln flag in the plot_TCMPR.R script which is wrapped by tcmpr_plotter_wrapper.py. This is a string that indicates whether to add the HFIP baseline, and indicates the version (no, 0, 5, 10 year goal).

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.9 I

---

**INIT_BEG**
Specify the beginning initialization time to be used in the analysis. Format can be controlled by INIT_TIME_FMT.

***Used by:*** command_builder.py, extract_tiles_wrapper.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_BEG_HOUR**
Specify the beginning initialization hour to be used in the analysis. Format is HH.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_END**
Specify the ending initialization time to be used in the analysis. Format can be controlled by INIT_TIME_FMT.

***Used by:*** command_builder.py, extract_tiles_wrapper.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_END_HOUR**
Specify the ending initialization hour to be used in the analysis. Format is HH.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_EXCLUDE**
Specify which, if any, forecast initializations to exclude from the analysis.

***Used by:*** tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_HOUR_END**
Specify the ending initialization hour to be used in the analysis. Format is HH.

***Used by:*** extract_tiles_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_INCLUDE**
Specify which forecast initializations to include in the analysis.

***Used by:*** tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_INCREMENT**
Control the increment or stride to use when stepping between forecast initializations. Units are seconds.

***Used by:*** command_builder.py, extract_tiles_wrapper.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**INIT_TIME_FMT**
Specify a formatting string to use for INIT_BEG and INIT_END.

***Used by:*** command_builder.py, master_metplus.py
***Family:***
***Default:***

---

**INTERVAL_TIME**
Define the interval time in hours (HH) to be used by the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.10 J

### 4.4.11 K

### 4.4.12 L

---

**LAT_ADJ**
Specify a latitude adjustment, in degrees to be used in the analysis.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**LEAD**
For cyclone_plotter_wrapper.py, this refers to the column of interest in the input ASCII cyclone file.

In the tcmpr_plotter_wrapper.py, this corresponds to the optional -lead argument in the plot_TCMPR.R script (which is wrapped by tcmpr_plotter.py). This argument is set to a comma-separted list of lead times (h) to be plotted.

In feature_util.py, this corresponds to the name of the column of interest in the input ASCII data file.

In tc_stat_wrapper.py, this corresponds to the name of the column of interest in the input ASCII data file.

***Used by:*** cyclone_plotter_wrapper.py, tcmpr_plotter_wrapper.py, feature_util.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**LEAD_LIST**
Specify a list of forecast leads to include in the analysis. Comma separated list format, e.g.:
0, 24, 48, 72, 96, 120

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**LEAD_SEQ**
Specify the sequence of forecast lead times to include in the analysis. Comma separated list format, e.g.:
0, 6, 12

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, gempak_to_cf_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py, reformat_gridded_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**LEGEND**
The text to be includede in the legend of your plot.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**LOG_DIR**
Specify the directory where log files from MET and METplus should be written.

***Used by:*** command_builder.py, met_util.py
***Family:*** [dir]
***Default:*** Varies

---

**LOG_LEVEL**
Specify the level of logging.

Everything above this level is sent to standard output. To quiet the output to a comfortable level, set this to "ERROR".

Options (ordered MOST verbose to LEAST verbose):
NOTSET
DEBUG
INFO
WARNING
ERROR
CRITICAL

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**LOG_METPLUS**
Control the filename of the METplus log file. Control the timestamp appended to the filename with LOG_TIMESTAMP_TEMPLATE. To turn OFF all logging, do not set this option.

***Used by:*** master_metplus.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**LOG_MET_OUTPUT_TO_METPLUS**
Control whether logging output from the MET tools is sent to the METplus log file, or individual log files for each MET tool.

***Used by:*** command_runner.py
***Family:*** [config]
***Default:*** yes/no

---

**LOG_MET_VERBOSITY**
Control the verbosity of the logging from the MET tools.
0 = Least amount of logging (lowest verbosity)
5 = Most amount of logging (highest verbosity)

***Used by:*** command_builder.py
***Family:*** [config]
***Default:*** 2

---

**LOG_TIMESTAMP_TEMPLATE**
Set the timestamp template for the METplus log file. Use Python strftime directives, e.g. %Y%m%d for YYYYMMDD.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** %Y%m%d

---

**LOG_TIMESTAMP_USE_DATATIME**
STrue/False. Determines which time to use for the log filenames. If True, use INIT_BEG if LOOP_BY_INIT is True or VALID_BEG if LOOP_BY_INIT is False. If False, use current time.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** *False*

---

**LON_ADJ**
Specify a longitude adjustment, in degrees to be used in the analysis.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**LOOP_BY_INIT**
Control whether the analysis is processed across initialization times or not. If set to false, loop by valid time

***Used by:*** command_builder.py, compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, make_plots_wrapper.py, master_metplus.py, mode_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** true

---

**LOOP_METHOD**

Control the looping method for METplus. Valid options are "times" or "processes". "times" runs all items in the PROCESS_LIST for a single run time, then repeat until all times have been evaluated. "processes" runs each item in the PROCESS_LIST for all times specified, then repeat for the next item in the PROCESS_LIST

***Used by:*** master_metplus.py, pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.13 M

---

**METPLUS_BASE**
This variable will automatically be set by METplus when it is started. It will be set to the location of METplus that is currently being run. Setting this variable in a config file will have no effect and will report a warning that it is being overridden.

***Used by:*** All
***Family:*** [dir]
***Default:*** Location METplus is being run from

---

**METPLUS_CONF**
Provide the absolute path to the METplus final configuration file. This file will contain every configuration option and value used when METplus was run.

***Used by:*** config_launcher.py
***Family:*** [config]
***Default:*** Varies

---

**MET_BASE**
The base directory where your MET installation resides.

***Used by:*** cyclone_plotter_wrapper.py, extract_tiles_wrapper.py, master_metplus.py, met_util.py, pb2nc_wrapper.py, point_stat_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py, tcmpr_plotter_wrapper.py, tc_pairs_wrapper.py, usage_wrapper.py
***Family:*** [dir]
***Default:***

---

**MET_BIN**
The location of MET binaries.

***Used by:***
***Family:***

***Default:***

---

**MET_BUILD_BASE**
The base directory of the MET install. Only needed if using MET version 6.0

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**MET_INSTALL_DIR**
The base directory of the MET install. To be defined when using MET version 6.1 and beyond

***Used by:*** compare_gridded_wrapper.py, cyclone_plotter_wrapper.py, ensemble_stat_wrapper.py, extract_tiles_wrapper.py, feature_util.py, grid_stat_wrapper.py, mode_wrapper.py, pb2nc_wrapper.py, pcp_combine_wrap
point_stat_wrapper.py, regrid_data_plane_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py,
stat_analysis_wrapper.py, tcmpr_plotter_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py, wavelet_stat_wrapper.
***Family:*** [dir]
***Default:*** Varies

---

**MISSING_VAL**
Specify the missing value code.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MISSING_VAL_TO_REPLACE**
Specify the missing value code to replace.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODEL**
Specify the model name.

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODEL1_NAME**
Define the model name for the first model to be used in the analysis. There can be N number of models defined in configuration files, simply increment the "MODEL1_" string to match the total number of models being used, e.g.:
MODEL1_NAME

MODEL2_NAME
.
.
.
MODELN_NAME

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODEL1_DIR**
Define the stat file directrory for the first model to be used in the analysis. There can be N number of model directories defined in configuration files, simply increment the "MODEL1_" string to match the total number of models being used, e.g.:
MODEL1_DIR
MODEL2_DIR
.
.
.
MODELN_DIR

***Used by:*** stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODEL_DATA_DIR**
Specify the directory where the model data are located.

***Used by:*** feature_util.py
***Family:*** [dir]
***Default:*** Varies

---

**MODEL_NAME**
Specify the model name.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODE_CONFIG**
Path to mode configuration file.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODE_CONV_RADIUS**
Comma separated list of convolution radius values used by mode for both forecast and observation fields. Has the same behavior as setting MODE_FCST_CONV_RADIUS and MODE_OBS_CONV_RADIUS to the same value.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** 5

---

**MODE_CONV_THRESH**
Comma separated list of convolution threshold values used by mode for both forecast and observation fields. Has the same behavior as setting MODE_FCST_CONV_THRESH and MODE_OBS_CONV_THRESH to the same value.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** >0.5

---

**MODE_FCST_CONV_RADIUS**
Comma separated list of convolution radius values used by mode for forecast fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:***5

---

**MODE_FCST_CONV_THRESH**
Comma separated list of convolution threshold values used by mode for forecast fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:***5

---

**MODE_FCST_MERGE_FLAG**
Sets the merge_flag value in the mode config file for forecast fields. Valid values are NONE, THRESH, ENGINE, and BOTH.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** THRESH

---

**MODE_FCST_MERGE_THRESH**
Comma separated list of merge threshold values used by mode for forecast fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]

***Default:*** >0.45

---

**MODE_MERGE_CONFIG_FILE**
Path to mode merge config file.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MODE_MERGE_FLAG**
Sets the merge_flag value in the mode config file for both forecast and observation fields. Has the same behavior as setting MODE_FCST_MERGE_FLAG and MODE_OBS_MERGE_FLAG to the same value. Valid values are NONE, THRESH, ENGINE, and BOTH.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** THRESH

---

**MODE_MERGE_THRESH**
Comma separated list of merge threshold values used by mode for forecast and observation fields. Has the same behavior as setting MODE_FCST_MERGE_THRESH and MODE_OBS_MERGE_THRESH to the same value.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** >0.45

---

**MODE_OBS_CONV_RADIUS**
Comma separated list of convolution radius values used by mode for observation fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:***5

---

**MODE_OBS_CONV_THRESH**
Comma separated list of convolution threshold values used by mode for observation fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:***5

---

**MODE_OBS_MERGE_FLAG**
Sets the merge_flag value in the mode config file for observation fields. Valid values are NONE, THRESH, ENGINE, and BOTH.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** THRESH

---

**MODE_OBS_MERGE_THRESH**
Comma separated list of merge threshold values used by mode for observation fields.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** >0.45

---

**MODE_OUT_DIR**
Ouptut directory to write mode files.

***Used by:*** mode_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**MODE_QUILT**
True/False. If True, run all permutations of radius and threshold.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** False

---

**MTD_CONFIG**
Path to mode-TD configuration file.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**MTD_CONV_RADIUS**
Comma separated list of convolution radius values used by mode-TD for both forecast and observation files. Has the same behavior as setting MTD_FCST_CONV_RADIUS and MTD_OBS_CONV_RADIUS to the same value.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** 5

---

**MTD_CONV_THRESH**
Comma separated list of convolution threshold values used by mode-TD for both forecast and observation files. Has the same behavior as setting MTD_FCST_CONV_THRESH and MTD_OBS_CONV_THRESH to the same value.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** >0.5

---

**MTD_FCST_CONV_RADIUS**
Comma separated list of convolution radius values used by mode-TD for forecast files.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** 5

---

**MTD_FCST_CONV_THRESH**
Comma separated list of convolution threshold values used by mode-TD for forecast files.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** >0.5

---

**MTD_OBS_CONV_RADIUS**
Comma separated list of convolution radius values used by mode-TD for observation files.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** 5

---

**MTD_OBS_CONV_THRESH**
Comma separated list of convolution threshold values used by mode-TD for observation files.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** >0.5

---

**MTD_OUT_DIR**
Ouptut directory to write mode-TD files.

***Used by:*** mtd_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**MTD_SINGLE_DATA_SRC**
Only used ifMTD_SINGLE_RUN is True. Determines which data set to process. Valid options are FCST and OBS.

***Used by:*** mtd_wrapper.py
***Family:*** [config]

***Default:*** FCST

---

**MTD_SINGLE_RUN**
Run mode-TD with -single option. Must set MTD_SINGLE_DATA_SRC to specify which data set to process.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** False

---

### 4.4.14 N

---

**NCAP2_EXE**
Path to the "ncap2" executable.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py, series_by_lead_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**NCDUMP_EXE**
Path to the "ncdump" executable.

***Used by:*** met_util.py, pb2nc_wrapper.py, point_stat_wrapper.py, series_by_lead_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**NC_FILE_TMPL**
File template used to match netCDF files used for analysis.

***Used by:*** pb2nc_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**NLAT**
The number of latitude points.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**NLON**
The number of longitude points.

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**NO_EE**
Set the "NO_EE" flag for the TC Matched Pairs plotting utility.

Acceptable values: yes/no

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** no

---

**NO_LOG**
Set the "NO_LOG" flag for the TC Matched Pairs plotting utility.

Acceptable values: yes/no

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** no

---

### 4.4.15 O

---

**OBS_[N]_FIELD_NAME**
This variable is used to define a [N] hour accumulation NetCDF field in the observation dataset used in the MET tool pcp_combine. [N] must be an integer >= 1.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_BUFR_VAR_LIST**
Specify which BUFR codes to use from the observation dataset when using the MET pb2nc tool. Format is comma separated list, e.g.:

PMO, TOB, TDO

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**[deprecated] OBS_DATA_INTERVAL**
Specify the accumulation interval of the observation dataset used by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**[deprecated] OBS_GEMPAK_INPUT_DIR**
Specify the input directory for GEMPAK formatted observation files. Use GEMPAKTOCF_INPUT_DIR if running GempakToCF from the PROCESS_LIST.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**[deprecated] OBS_GEMPAK_TEMPLATE**
Filename template used to filter GEMPAK formatted observation files. Use GEMPAKTOCF_INPUT_TEMPLATE if running GempakToCF from the PROCESS_LIST.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_GRID_STAT_INPUT_DATATYPE**
Specify the data type of the input directory for observation files used with the MET grid_stat tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_GRID_STAT_INPUT_DIR**
Specify the directory where the input observation files are for the MET grid_stat tool.

***Used by:*** grid_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_GRID_STAT_INPUT_TEMPLATE**
Filename template used to filter input observation files used by the MET grid_stat tool.

***Used by:*** grid_stat_wrapper.py
***Family:*** [filename_templates]

***Default:*** Varies

---

**OBS_GRID_STAT_PROB_THRESH**
Threshold values to be used for probabilistic data in grid_stat. Used when setting OBS_* variables to probabilistic forecast data for comparison. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le).

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** ==0.1

---

**OBS_INPUT_DIR**
Specify the input directory for observation files.

***Used by:*** point_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_INPUT_DIR_REGEX**
Specify the regular expression to use when searching for observation file input directories.

***Used by:*** point_stat_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**OBS_INPUT_FILE_REGEX**
Regular expression used to filter observation input files used in the analysis.

***Used by:*** point_stat_wrapper.py,
***Family:*** [regex_pattern]
***Default:*** Varies

---

**OBS_INPUT_FILE_TEMPL**
Specify the filename template to use for observation input files.

***Used by:*** point_stat_wrapper.py,
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_IS_DAILY_FILE**
Specify whether the forecast file is a daily file or not.

Acceptable values: true/false

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**OBS_IS_PROB**
Used when setting OBS_* variables to process forecast data for comparisons with mtd. Specify whether the observation data are probabilistic or not. See FCST_IS_PROB.

Acceptable values: true/false

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** False

---

**OBS_LEVEL**
Specify what accumulation level should be used from the observation data for the analysis. See FCST_LEVEL for more information

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_MODE_INPUT_DATATYPE**
Specify the data type of the input directory for observation files used with the MET mode tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** mode_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_MODE_INPUT_DIR**
Input directory for observation files to use with the MET tool mode.

***Used by:*** mode_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_MODE_INPUT_TEMPLATE**
Template used to specify observation input filenames for the MET tool mode.

***Used by:*** mode_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_MTD_INPUT_DATATYPE**
Specify the data type of the input directory for observation files used with the MET mode-TD tool. Currently

valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** mtd_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_MTD_INPUT_DIR**
Input directory for observation files to use with the MET tool mode-TD.

***Used by:*** mtd_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_MTD_INPUT_TEMPLATE**
Template used to specify observation input filenames for the MET tool mode-TD.

***Used by:*** mtd_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_NAME**
Provide a string to identify the observation dataset name.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**[deprecated] OBS_NATIVE_DATA_TYPE**
Specify the data format of the observation data. Use OBS_PCP_COMBINE_INPUT_DATATYPE instead.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_PCP_COMBINE_INPUT_DATATYPE**
Specify the data type of the input directory for observation files used with the MET pcp_combine tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF. Required by pcp_combine if OBS_PCP_COMBINE_RUN is True. Replaces deprecated variable OBS_NATIVE_DATA_TYPE.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_PCP_COMBINE_INPUT_DIR**
Specify the input directory for the observation data used by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_PCP_COMBINE_INPUT_TEMPLATE**
Filename template used to filter input observation files used by the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_PCP_COMBINE_OUTPUT_DIR**
Specify the output directory where files from the MET pcp_combine tool are written.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_PCP_COMBINE_OUTPUT_TEMPLATE**
Filename template used for writing output files from the MET pcp_combine tool.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_PCP_COMBINE_RUN**
Specify whether to run pcp_combine on the observation data or not.

Acceptable values: True/False

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_INPUT_DATATYPE**
Specify the data type of the input directory for observation files used with the MET regrid_data_plane tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_INPUT_DIR**
Specify the input directory for observation files used by the MET regrid_data_plane tool.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_OUTPUT_DIR**
Specify the output directory where files are written from the MET regrid_data_plane tool.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_RUN**
Specify whether to run regrid_data_plane on the observation data or not.

Acceptable values: True/False

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_INPUT_TEMPLATE**
Specify the input filename template to use for observation files (input and output) used by the MET regrid_data_plane tool. If not set, METplus will use OBS_REGRID_DATA_PLANE_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_OUTPUT_TEMPLATE**
Specify the output filename template to use for observation files (input and output) used by the MET regrid_data_plane tool. If not set, METplus will use OBS_REGRID_DATA_PLANE_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**OBS_REGRID_DATA_PLANE_TEMPLATE**
Specify the filename template to use for observation files (input and output) used by the MET regrid_data_plane tool. To specify different templates for input and output files , use FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE and FCST_REGRID_DATA_PLANE_OUTPUT_TEMPLATE.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**[deprecated] OBS_VAR**

Specify the string for the observation variable used in the analysis. See OBS_VARn_NAME, OBS_VARn_LEVELS, OBS_VARn_OPTIONS and OBS_VARn_THRESH where n = integer >= 1.

***Used by:*** grid_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_VAR[N]_LEVELS**

Define the levels for the [N]th observation variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items. You can define NetCDF levels, such as (0,*,*), but you will need to surround these values with quotation marks so that the commas in the item are not interpreted as an item delimeter. Some examples:

OBS_VAR1_LEVELS = A06, P500
OBS_VAR2_LEVELS = "(0,*,*)", "(1,*,*)"

If OBS_VAR[N]_LEVELS is not set but FCST_VAR[N]_LEVELS is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
OBS_VAR1_LEVELS
OBS_VAR2_LEVELS
...
OBS_VAR[N]_LEVELS

***Used by:*** make_plots_wrapper.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_VAR[N]_NAME**

Define the name for the [N]th observation variable to be used in the analysis where [N] is an integer >= 1. If OBS_VAR[N]_NAME is not set but FCST_VAR[N]_NAME is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
OBS_VAR1_NAME
OBS_VAR2_NAME
...
OBS_VAR[N]_NAME

***Used by:*** make_plots_wrapper.py, met_util.py

***Family:*** [config]
***Default:*** Varies

---

**OBS_VAR[N]_OPTIONS**
Define the options for the [N]th observation variable to be used in the analysis where [N] is an integer >= 1. These addition options will be applied to every name/level/threshold combination for VAR[N]. If OBS_VAR[N]_OPTIONS is not set but FCST_VAR[N]_OPTIONS is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
OBS_VAR1_OPTIONS
OBS_VAR2_OPTIONS
...
OBS_VAR[N]_OPTIONS

***Used by:*** make_plots_wrapper.py, met_util.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_VAR[N]_THRESH**
Define the threshold(s) for the [N]th observation variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le). If OBS_VAR[N]_THRESH is not set but FCST_VAR[N]_THRESH is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
OBS_VAR1_THRESH
OBS_VAR2_THRESH
...
OBS_VAR[N]_THRESH

***Used by:*** met_util.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_WINDOW_BEG**
Corresponds to the OBS_WINDOW_BEG in the MET config file for pb2nc. Please refer to Chapter 4 of the MET User's Guide.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OBS_WINDOW_END**
Corresponds to the OBS_WINDOW_END in the MET config file for pb2nc. Please refer to Chapter 4 of

the MET User's Guide.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OB_TYPE**
Provide a string to represent the type of observation data used in the analysis. Used in setting output filename

***Used by:*** compare_gridded_wrapper.py, ensemble_stat_wrapper.py, grid_stat_wrapper.py, mode_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**OUTPUT_BASE**
Provide a path to the top level output directory for METplus.

***Used by:*** config_launcher.py, pb2nc_wrapper.py, point_stat_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**OVERWRITE_NC_OUTPUT**
Specify whether to overwrite the netCDF output or not when using the MET pb2nc tool.

Acceptable values: yes/no

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** yes

---

**OVERWRITE_TRACK**
Specify whether to overwrite the track data or not.

Acceptable values: yes/no

***Used by:*** extract_tiles_wrapper.py, feature_util.py
***Family:*** [config]
***Default:*** no

---

### 4.4.16 P

---

**PARM_BASE**

Specify the top level METplus parameter file directory.

***Used by:*** config_launcher.py, pb2nc_wrapper.py, point_stat_wrapper.py, tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**PB2NC_CONFIG_FILE**
Specify the absolute path to the configuration file for the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PB2NC_GRID**
Specify a grid to use with the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PB2NC_MESSAGE_TYPE**
Specify which PREPBUFR (PB) message types to convert using the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PB2NC_OUTPUT_DIR**
Specify the directory where files will be written from the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**PB2NC_POLY**
Specify a polygon to be used with the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PB2NC_STATION_ID**
Specify the ID of the station to use with the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**PCP_COMBINE_METHOD**
Specify the method to be used with the MET pcp_combine tool. Valid options are ADD, SUM, and SUBTRACT.

***Used by:*** pcp_combine_wrapper.py
***Family:*** [config]
***Default:*** ADD

---

**PLOTTING_OUT_DIR**
Specify the output directory where plots will be saved.

***Used by:*** make_plots_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**PLOTTING_SCRIPTS_DIR**
Specify the directory where the plotting scripts are located.

***Used by:*** make_plots_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**PLOT_CONFIG_OPTS**
Specify plot configuration options for the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PLOT_STATS_LIST**
Specify which statistics should be plotted in a comma separated list, e.g.:
acc, bias, rmse

***Used by:*** make_plots_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PLOT_TYPES**
Specify what plot types are desired for the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**POINT_STAT_CONFIG_FILE**
Specify the absolute path to the configuration file to be used with the MET point_stat tool.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**POINT_STAT_GRID**
Specify the grid to use with the MET point_stat tool.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**POINT_STAT_MESSAGE_TYPE**
Specify which PREPBUFR message types to process with the MET point_stat tool.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**POINT_STAT_OUTPUT_DIR**
Specify the directory where output files from the MET point_stat tool are written.

***Used by:*** point_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**POINT_STAT_POLY**
Specify a polygon to use with the MET point_stat tool.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**POINT_STAT_STATION_ID**
Specify the ID of a specific station to use with the MET point_stat tool.

***Used by:*** point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PREFIX**

This corresponds to the optional -prefix flag of the plot_TCMPR.R script (which is wrapped by tcmpr_plotter_wrapper.py). This is the output file name prefix.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PREPBUFR_DATA_DIR**
Specify the directory where the PREPBUFR data are located for the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**PREPBUFR_DIR_REGEX**
Regular expression to use when searching for PREPBUFR data.

***Used by:*** pb2nc_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**PREPBUFR_FILE_REGEX**
Regular expression to use when searching for PREPBUFR files.

***Used by:*** pb2nc_wrapper.py
***Family:*** [regex_pattern]
***Default:*** Varies

---

**PREPBUFR_MODEL_DIR_NAME**
Specify the name of the model being used with the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**PROCESS_LIST**
Specify the list of processes for METplus to perform, in a comma separated list.

***Used by:*** master_metplus.py
***Family:*** [config]
***Default:*** Varies

---

**PROJ_DIR**
A directory for generic use. The user can store input files (if INPUT_BASE is not defined), intermediate files, and any other project-related files.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py, tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

### 4.4.17 Q

### 4.4.18 R

---

**REFERENCE_TMPL**
The filename template describing the observation/reference data.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [filename_templates]
***Default:*** Varies

---

**REGION_LIST**
A list of the regions of interest.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**REGRID_TO_GRID**
If supported, provide the output grid that is desired from the MET tool being used in the analysis.

***Used by:*** make_plots_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**REGRID_USING_MET_TOOL**
Specify whether to regrid using the MET regrid_data_plane tool or not.

Acceptable values: yes/no

***Used by:*** feature_util.py, met_util.py, series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** yes

---

**RM_EXE**
Specify the path to the Linux "rm" executable.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py, series_by_lead_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

**RP_DIFF**
This corresponds to the optional -rp_diff flag of the plot_TCMPR.R script (which is wrapped by tcmpr_plotter_wrapper.py). This a comma-separated list of thresholds to specify meaningful differences for the relative performance plot.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.19 S

---

**SAVE**
Corresponds to the optional -save flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is a yes/no value to indicate whether to save the image (yes).

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SAVE_DATA**
Corresponds to the optional -save_data flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). Indicates whether to save the filtered track data to a file instead of deleting it.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SCATTER_X**
Corresponds to the optional -scatter_x flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is a comma-separated list of x-axis variable columns to plot.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SCATTER_Y**
Corresponds to the optional -scatter_y flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is a comma-separated list of y-axis variable columns to plot.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SCRUB_STAGING_DIR**
Remove staging directory after METplus has completed running if set to True. Set to False to preserve data for subsequent runs.

***Used by:*** master_metplus.py
***Family:*** [config]
***Default:*** False

---

**SERIES**
Corresponds to the optional -series flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is the column whose unique values define the series on the plot, optionally followed by a comma-separated list of values, including: ALL, OTHER, and colon-separated groups.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SERIES_ANALYSIS_BY_INIT_CONFIG_FILE**
Specify the absolute path for the configuration file to use with the MET series_analysis tool by initialization time.

***Used by:*** series_by_init_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SERIES_ANALYSIS_BY_LEAD_CONFIG_FILE**
Specify the absolute path for the configuration file to use with the MET series_analysis tool by lead time.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SERIES_ANALYSIS_FILTER_OPTS**
Filtering options to be applied during series analysis. Filter options are performed by invoking the MET tc_stat tool within the METplus wrapper.

***Used by:*** series_by_lead_wrapper.py, series_by_init_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SERIES_CI**

Corresponds to the optional -series_ci flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is a list of true/false for confidence intervals. This list can be optionally followed by a comma-separated list of values, including ALL, OTHER, and colon-separated groups.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SERIES_INIT_FILTERED_OUT_DIR**

Specify the directory where filtered files will be written from the MET series_analysis tool when processing by initialization time.

***Used by:*** series_by_init_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**SERIES_INIT_OUT_DIR**

Specify the directory where files will be written from the MET series analysis tool when processing by initialization time.

***Used by:*** series_by_init_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**SERIES_LEAD_FILTERED_OUT_DIR**

Specify the directory where filtered files will be written from the MET series_analysis tool when processing by lead time.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**SERIES_LEAD_OUT_DIR**

Specify the directory where files will be written from the MET series analysis tool when processing by lead time.

***Used by:*** series_by_lead_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**SKILL_REF**

This corresponds to the optional -skill_ref flag in plot_TCMPR.R (which is wrapped by tcmpr_plotter_wrapper.py). This is the identifier for the skill score reference.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**START_DATE**
Specify the start data for the analysis time period. Format is YYYYYMMDDHH.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**STAGING_DIR**
Directory to uncompress or convert data into for use in METplus.

***Used by:*** All
***Family:*** [dir]
***Default:*** OUTPUT_BASE/stage

---

**START_HOUR**
Specify the start hour for the analysis time period. Format is HH.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**STAT_ANALYSIS_CONFIG**
Specify the absolute path for the configuration file used with the MET stat_analysis tool.

***Used by:*** stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**STAT_ANALYSIS_LOOKIN_DIR**
Specify the input directory where the MET stat_analysis tool will find input files.

***Used by:*** stat_analysis_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**STAT_ANALYSIS_OUT_DIR**
Specify the output directory where files will be written from the MET stat_analysis tool.

***Used by:*** stat_analysis_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**STAT_FILES_INPUT_DIR**
Specify the directory where stat files exist that plots can be generated from.

***Used by:*** make_plots_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**STAT_LIST**
Specify a list of statistics to be computed by the MET series_analysis tool.

***Used by:*** series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**STORM_ID**
The identifier of the storm(s) of interest.

***Used by:*** cyclone_plotter_wrapper.py, met_util.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**STORM_NAME**
The name(s) of the storm of interest.

***Used by:*** tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**SUBTITLE**
The subtitle of the plot.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.20 T

---

**TCMPR_DATA**
Provide the input directory for the track data for the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [dir]

***Default:*** Varies

---

**TCMPR_PLOT_OUT_DIR**
Provide the output directory where the TC Matched Pairs plotting tool will create files.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TC_PAIRS_CONFIG_FILE**
Provide the absolute path to the configuration file for the MET tc_pairs tool.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_PAIRS_DIR**
Specify the directory where the MET tc_pairs tool will write files.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TC_PAIRS_FORCE_OVERWRITE**
Specify whether to overwrite the output from the MET tc_pairs tool or not.

Acceptable values: yes/no

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** no

---

**TC_STAT_AMODEL**
Specify the AMODEL for the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_BASIN**
Specify the BASIN for the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_BMODEL**
Specify the BMODEL for the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_CMD_LINE_JOB**
Specify expression(s) that will be passed to the MET tc_stat tool via the command line.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_COLUMN_STR_NAME**
Specify the string names of the columns for stratification with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_COLUMN_STR_VAL**
Specify the values for the columns set via the TC_STAT_COLUMN_STR_NAME option for use with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_COLUMN_THRESH_NAME**
Specify the string names of the columns for stratification by threshold with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_COLUMN_THRESH_VAL**
Specify the values used for thresholding the columns specified in the TC_STAT_COLUMN_THRESH_NAME option for use with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_CYCLONE**
Specify the CYCLONE of interest for use with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_DESC**
Specify the DESC option for use with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_BEG**
Specify the beginning initialization time for stratification when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_END**
Specify the ending initialization time for stratification when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_EXCLUDE**
Specify the initialization times to exclude when using the MET tc_stat tool, via a comma separated list e.g.:

20141220_18, 20141221_00

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_HOUR**
The beginning hour (HH) of the initialization time of interest.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_INCLUDE**
Specify the initialization times to include when using the MET tc_stat tool, via a comma separated list e.g.:

20141220_00, 20141220_06, 20141220_12

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_MASK**
This corresponds to the INIT_MASK keyword in the MET tc_stat config file. For more information, please refer to Chapter 20 in the MET User's Guide.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_STR_NAME**
This corresponds to the INIT_STR_NAME keyword in the MET tc_stat config file. Please refer to Chapter 20 in the MET User's Guide for more details.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INIT_STR_VAL**
This corresponds to the INIT_STR_VAL keyword in the MET tc_stat config file. Please refer to Chapter 20 in the MET User's Guide for more information.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_INPUT_DIR**
Specify the input directory where the MET tc_stat tool will look for files.

***Used by:*** tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TC_STAT_JOBS_LIST**
Specify expressions for the MET tc_stat tool to execute.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_LANDFALL**
Specify whether only those points occurring near landfall should be retained when using the MET tc_stat tool.

Acceptable values: True/False

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** False

---

**TC_STAT_LANDFALL_BEG**
Specify the beginning of the landfall window for use with the MET tc_stat tool.

Acceptable formats: HH, HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** -24

---

**TC_STAT_LANDFALL_END**
Specify the end of the landfall window for use with the MET tc_stat tool.

Acceptable formats: HH, HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_LEAD**
Specify the lead times to stratify by when using the MET tc_stat tool.

Acceptable formats: HH, HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]

***Default:*** Varies

---

**TC_STAT_LEAD_REQ**
Specify the LEAD_REQ when using the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_MATCH_POINTS**
Specify whether only those points common to both the ADECK and BDECK tracks should be written out or not when using the MET tc_stat tool.

Acceptable values: True/False

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** false

---

**TC_STAT_OUTPUT_DIR**
Specify the output directory where the MET tc_stat tool will write files.

***Used by:*** tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TC_STAT_RUN_VIA**
Specify the method for running the MET tc_stat tool.

Acceptable values: CONFIG

If left blank (unset), tc_stat will run via the command line.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** CONFIG

---

**TC_STAT_STORM_ID**
Set the STORM_ID(s) of interest with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_STORM_NAME**
Set the STORM_NAME for use with the MET tc_stat tool.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_TRACK_WATCH_WARN**
Specify which watches and warnings to stratify over when using the MET tc_stat tool.

Acceptable values: HUWARN, HUWATCH, TSWARN, TSWATCH, ALL

If left blank (unset), no stratification will be done.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_BEG**
Specify a comma separated list of beginning valid times to stratify with when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_END**
Specify a comma separated list of ending valid times to stratify with when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_EXCLUDE**
Specify a comma separated list of valid times to exclude from the stratification with when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_HOUR**
This corresponds to the VALID_HOUR keyword in the MET tc_stat config file. For more information, please refer to Chapter 20 of the MET User's Guide.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_INCLUDE**
Specify a comma separated list of valid times to include in the stratification with when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_VALID_MASK**
This corresponds to the VALID_MASK in the MET tc_stat config file. Please refer to Chapter 20 of the MET User's Guide for more information.

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TC_STAT_WATER_ONLY**
Specify whether to exclude points where the distance to land is <= 0. If set to TRUE, once land is encountered the remainder of the forecast track is not used for the verification, even if the track moves back over water.

Acceptable values: true/false

***Used by:*** tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TIME_METHOD**
Specify which time method to use with the MET pb2nc and point_stat tools.

Acceptable values: BY_VALID, BY_INIT

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:***

***Default:***

---

**TIME_SUMMARY_BEG**
Specify the starting time of the summary when using the MET pb2nc tool.

Acceptable formats: HHMMSS

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** 000000

---

**TIME_SUMMARY_END**
Specify the ending time of the summary when using the MET pb2nc tool.

Acceptable formats: HHMMSS

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** 235959

---

**TIME_SUMMARY_FLAG**
Specify whether to receive a time summary from the MET pb2nc tool or not.

Acceptable values: True/False

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** False

---

**TIME_SUMMARY_TYPES**
Specify a comma separated list of time summary types to receive from the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TIME_SUMMARY_VAR_NAMES**
Specify a comma separated list of time summary variable names to receive from the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TITLE**
Specify a title string for the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TMP_DIR**
Specify the path to a temporary directory where the user has write permissions.

***Used by:*** extract_tiles_wrapper.py, pb2nc_wrapper.py, point_stat_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py, tc_stat_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TOP_LEVEL_DIRS**
Specify whether to use top-level directories when using the MET tc_pairs utility or not.

Acceptable values: yes/no

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** no

---

**TRACK_DATA_DIR**
Specify the directory where track data are located for use with the MET tc_pairs tool.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TRACK_DATA_MOD_FORCE_OVERWRITE**
Specify whether to force an overwrite of the track data or not.

Acceptable values: yes/no

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** no

---

**TRACK_DATA_SUBDIR_MOD**
Specify the sub-directory where modified track data files are stored for use with the MET tc_pairs tool.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [dir]
***Default:*** Varies

---

**TRACK_TYPE**
Specify the track type to filter by when using the MET tc_pairs tool.

***Used by:*** tc_pairs_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**TR_EXE**
Specify the path to the Linux "tr" executable.

***Used by:*** pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [exe]
***Default:*** /path/to

---

### 4.4.21 U

### 4.4.22 V

---

**VALID_BEG**
Specify a begin time for valid times for use in the analysis.

Acceptable formats: YYYYMM[DD[_HH]]

***Used by:*** command_builder.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VALID_BEG_HOUR**
Specify a beginning hour for valid times for use in the analysis.

Acceptable formats: HH

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VALID_END**
Specify an end time for valid times for use in the analysis.

Acceptable formats: controlled via VALID_TIME_FMT

***Used by:*** command_builder.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_pairs_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VALID_END_HOUR**
Specify an end hour for valid times for use in the analysis.

Acceptable formats: controlled via VALID_TIME_FMT

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VALID_INCREMENT**
Specify the time increment for valid times for use in the analysis.

Acceptable formats: seconds

***Used by:*** command_builder.py, make_plots_wrapper.py, master_metplus.py, stat_analysis_wrapper.py, tc_stat_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VALID_TIME_FMT**
Specify a strftime formatting string for use with VALID_BEG and VALID_END.

***Used by:*** command_builder.py, master_metplus.py
***Family:*** [config]
***Default:*** Varies

---

**VAR_LIST**
Specify a comma separated list of variables to be used in the analysis.

***Used by:*** feature_util.py, pb2nc_wrapper.py, series_by_init_wrapper.py, series_by_lead_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VERIFICATION_GRID**
Specify the absolute path to a file containing information about the desired output grid from the MET regrid_data_plane tool.

***Used by:*** regrid_data_plane_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VERIF_CASE**
Specify a string identifying the verification case being performed.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VERIF_TYPE**
Specify a string describing the type of verification being performed.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**VERTICAL_LOCATION**
Specify the vertical location desired when using the MET pb2nc tool.

***Used by:*** pb2nc_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.23 W

---

**WAVE_NUM_BEG_LIST**
Specify a comma separated list of desired beginning wave numbers.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**WAVE_NUM_END_LIST**
Specify a comma separated list of desired ending wave numbers.

***Used by:*** make_plots_wrapper.py, stat_analysis_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**WGRIB2**
Specify the path to the "wgrib2" executable.

***Used by:*** feature_util.py, pb2nc_wrapper.py, point_stat_wrapper.py
***Family:*** [exe]

***Default:*** /path/to

---

### 4.4.24 X

---

**XLAB**
Specify the x-axis label when using the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**XLIM**
Specify the x-axis limit when using the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.25 Y

---

**YLAB**
Specify the y-axis label when using the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

**YLIM**
Specify the y-axis limit when using the TC Matched Pairs plotting tool.

***Used by:*** tcmpr_plotter_wrapper.py
***Family:*** [config]
***Default:*** Varies

---

### 4.4.26 Z

## 4.5 User Defined Config

You can define your own custom config variables that will be set as environment variables when METplus is run. MET config files can read environment variables, so this is a good way to customize information that is read by those files. To create add a custom config variable, add a section to one of your METplus config files called [user_env_vars]. Under this header, add as many variables as you'd like. For example, if you added the following to your METplus config:

**[user_env_vars]**
**VAR_NAME = some_text_for_feb_1_1987_run**

and you added the following to a MET config file that is used:

**output_prefix = ${VAR_NAME}**

then at run time, the MET application will be run with the configuration:

**output_prefix = some_text_for_feb_1_1987_run**

You can also reference other variables in the METplus config file. For example:

**[config]**
**INIT_BEG = 1987020104**
**...**
**[user_env_vars]**
**USE_CASE_TIME_ID = {INIT_BEG}**

This is the equivalent of calling

**export USE_CASE_TIME_ID=1987020104**

at the beginning of your METplus run. You can access the variable in the MET config file with ${USE_CASE_TIME_ID}.

# References


Alberson, S.D., 1998: Five-day Tropical cyclone track forecasts in the North Atlantic Basin. *Weather & Forecasting*, 13, 1005-1015.

Bradley, A.A., S.S. Schwartz, and T. Hashino, 2008: Sampling Uncertainty and Confidence Intervals for the Brier Score and Brier Skill Score. *Weather and Forecasting*, 23, 992-1006.

Brill, K. F., and F. Mesinger, 2009: Applying a general analytic method for assessing bias sensitivity to bias-adjusted threat and equitable threat scores. *Weather and Forecasting*, 24, 1748–1754.

Brown, B.G., R. Bullock, J. Halley Gotway, D. Ahijevych, C. Davis, E. Gilleland, and L. Holland, 2007: Application of the MODE object-based verification tool for the evaluation of model precipitation fields. *AMS 22nd Conference on Weather Analysis and Forecasting and 18th Conference on Numerical Weather Prediction*, 25-29 June, Park City, Utah, American Meteorological Society (Boston), Available at `http://ams.confex.com/ams/pdfpapers/124856.pdf`.

Bullock, R., T. Fowler, and B. Brown, 2016: Method for Object-Based Diagnostic Evaluation. NCAR Tech. Note NCAR/TN-532+STR, 66 pp.

Candille, G., and O. Talagrand, 2008: Impact of observational error on the validation of ensemble prediction systems. *Q. J. R. Meteorol. Soc.* 134: 959–971.

Casati, B., G. Ross, and D. Stephenson, 2004: A new intensity-scale approach for the verification of spatial precipitation forecasts. *Meteorol. Appl.* 11, 141-154.

Davis, C.A., B.G. Brown, and R.G. Bullock, 2006a: Object-based verification of precipitation forecasts, Part I: Methodology and application to mesoscale rain areas. *Monthly Weather Review*, 134, 1772-1784.

Davis, C.A., B.G. Brown, and R.G. Bullock, 2006b: Object-based verification of precipitation forecasts, Part II: Application to convective rain systems. *Monthly Weather Review*, 134, 1785-1795.

Dawid, A.P., 1984: Statistical theory: The prequential approach. *J. Roy. Stat. Soc*, A147, 278-292.

Ebert, E.E., 2008: Fuzzy verification of high-resolution gridded forecasts: a review and proposed framework. *Meteorological Applications*, 15, 51-64.

Eckel, F. A., M.S. Allen, M. C. Sittel, 2012: Estimation of Ambiguity in Ensemble Forecasts. *Wea. Forecasting*, 27, 50-69. doi: `http://dx.doi.org/10.1175/WAF-D-11-00015.1`

Efron, B. 2007: Correlation and large-scale significance testing. *Journal of the American Statistical Association*, 102(477), 93-103.

Gilleland, E., 2010: Confidence intervals for forecast verification. *NCAR Technical Note* NCAR/TN-479+STR, 71pp.

Gneiting, T., A. Westveld, A. Raferty, and T. Goldman, 2004: *Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation.* Technical Report no. 449, Department of Statistics, University of Washington. [Available online at `http://www.stat.washington.edu/www/research/reports/` ]

Hamill, T. M., 2001: Interpretation of rank histograms for verifying ensemble forecasts. *Mon. Wea. Rev.*, 129, 550-560.

Hogan, R., E. O'Connor, and A. Illingworth, 2009: Verification of cloud-fraction forecasts. *Quart. Jour. Roy. Meteorol. Soc.*, 135, 1494-1511.

Jolliffe, I.T., and D.B. Stephenson, 2012: *Forecast verification. A practitioner's guide in atmospheric science.* Wiley and Sons Ltd, 240 pp.

Knaff, J.A., M. DeMaria, C.R. Sampson, and J.M. Gross, 2003: Statistical, Five-Day Tropical Cyclone Intensity Forecasts Derived from Climatology and Persistence." *Weather & Forecasting,*" Vol. 18 Issue 2, p. 80-92.

Mason**,** S. J., 2004: On Using "Climatology" as a Reference Strategy in the Brier and Ranked Probability Skill Scores. *Mon. Wea. Rev.*, 132, 1891–1895.

Mittermaier, M., 2013: A strategy for verifying near-convection-resolving model forecasts at observing sites. Wea. Forecasting, 29, 185-204.

Mood**,** A. M., F. A. Graybill and D. C. Boes, 1974: *Introduction to the Theory of Statistics*, McGraw-Hill, 299-338.

Murphy, A.H., and R.L. Winkler, 1987: A general framework for forecast verification. *Monthly Weather Review*, 115, 1330-1338.

Roberts, N.M., and H.W. Lean, 2008: Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. *Monthly Weather Review*, 136, 78-97.

Saetra O., H. Hersbach, J-R Bidlot, D. Richardson, 2004: Effects of observation errors on the statistics for ensemble spread and reliability. *Mon. Weather Rev.* 132: 1487–1501.

Santos C. and A. Ghelli, 2012: Observational probability method to assess ensemble precipitation forecasts. *Q. J. R. Meteorol. Soc.* 138: 209–221.

Stephenson, D.B., 2000: Use of the "Odds Ratio" for diagnosing forecast skill. *Weather and Forecasting*, 15, 221-232.

Stephenson, D.B., B. Casati, C.A.T. Ferro, and C.A. Wilson, 2008: The extreme dependency score: A non-vanishing measure for forecasts of rare events. *Meteor. Appl.* 15, 41-50.

Weniger, M., F. Kapp, and P. Friederichs, 2016: Spatial Verification Using Wavelet Transforms: A Review. *Quarterly Journal of the Royal Meteorological Society*, 143, 120-136.

Wilks, D.S. 2010: Sampling distributions of the Brier score and Brier skill score under serial dependence. Q.J.R. Meteorol. Soc., 136, 2109–2118. doi:10.1002/qj.709

Wilks, D., 2011: *Statistical methods in the atmospheric sciences.* Elsevier, San Diego.

# List of Tables

# List of Figures