

# METplus Version 2.0.3

## Automation for the Model Evaluation Tools

Developmental Testbed Center  
Boulder, Colorado

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

*<sup>1</sup> National Center for Atmospheric Research,  
Research Applications Laboratory*

*<sup>2</sup> Cooperative Institute for Research in the Atmosphere at  
National Oceanic and Atmospheric Administration (NOAA)  
Earth System Research Laboratory*

*<sup>3</sup> I.M. Systems Group at  
NOAA Environmental Modeling Center*

*<sup>4</sup> Developmental Testbed Center*

January 2019

# Contents

<b>1</b>	<b>Overview of METplus</b>	<b>12</b>
1.1	Purpose and organization of the User's Guide . . . . .	12
1.2	The Developmental Testbed Center (DTC) . . . . .	12
1.3	METplus goals and design philosophy . . . . .	13
1.4	METplus components . . . . .	13
1.5	Future development plans . . . . .	13
1.6	Code support . . . . .	14
<b>2</b>	<b>Software Installation/Getting Started</b>	<b>15</b>
2.1	Introduction . . . . .	15
2.2	Supported architectures . . . . .	15
2.3	Programming/scripting languages . . . . .	15
2.4	Pre-requisites . . . . .	15
2.5	METplus directory structure . . . . .	16
2.6	Getting the METplus source code . . . . .	17
2.6.1	Get the source code via your Web Browser . . . . .	17
2.6.1.1	Source code only: . . . . .	17
2.6.1.2	Source code, additional documentation, and sample data . . . . .	20
2.6.2	Get the source code via Command line . . . . .	23
2.7	Set up your environment . . . . .	23
2.8	Set up METplus Configuration files . . . . .	24
2.9	Running METplus . . . . .	25

<i>CONTENTS</i>	2
<b>3 METplus System Configuration</b>	<b>26</b>
3.1 Config Best Practices . . . . .	26
3.2 Config File Structure . . . . .	27
3.3 Config Quick Start Example . . . . .	27
3.4 A-Z Config Glossary . . . . .	29
3.4.1 A . . . . .	29
3.4.2 B . . . . .	30
3.4.3 C . . . . .	32
3.4.4 D . . . . .	34
3.4.5 E . . . . .	35
3.4.6 F . . . . .	36
3.4.7 G . . . . .	48
3.4.8 H . . . . .	49
3.4.9 I . . . . .	50
3.4.10 J . . . . .	52
3.4.11 K . . . . .	52
3.4.12 L . . . . .	52
3.4.13 M . . . . .	55
3.4.14 N . . . . .	62
3.4.15 O . . . . .	63
3.4.16 P . . . . .	72
3.4.17 Q . . . . .	77
3.4.18 R . . . . .	77
3.4.19 S . . . . .	78
3.4.20 T . . . . .	82

<i>CONTENTS</i>	3
3.4.21 U . . . . .	93
3.4.22 V . . . . .	93
3.4.23 W . . . . .	95
3.4.24 X . . . . .	96
3.4.25 Y . . . . .	96
3.4.26 Z . . . . .	97
3.5 User Defined Config . . . . .	97

# Foreword: A note to METplus users

This User's Guide is provided as an aid to users of the Model Evaluation Tools (MET) and its 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 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 Wingen-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 v2.0.3 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]\_[MET-APP]\_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 Prediction 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 its 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 3

### 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](http://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](mailto: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 <sup>1</sup>
- 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
├── src
├── 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 documentation. 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 src/ directory contains Doxygen executables to generate documentation for developers.

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

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

---

<sup>1</sup>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:



[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

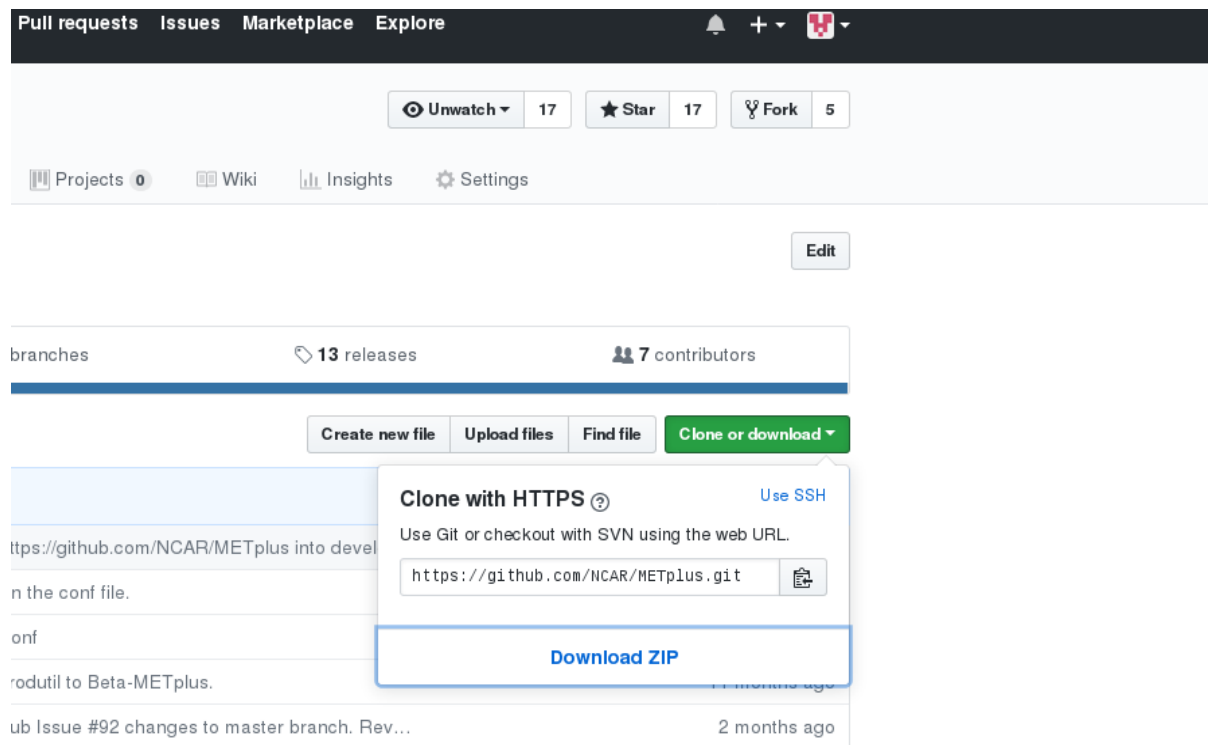
### Install/Configure/Run Guide

@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:



[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

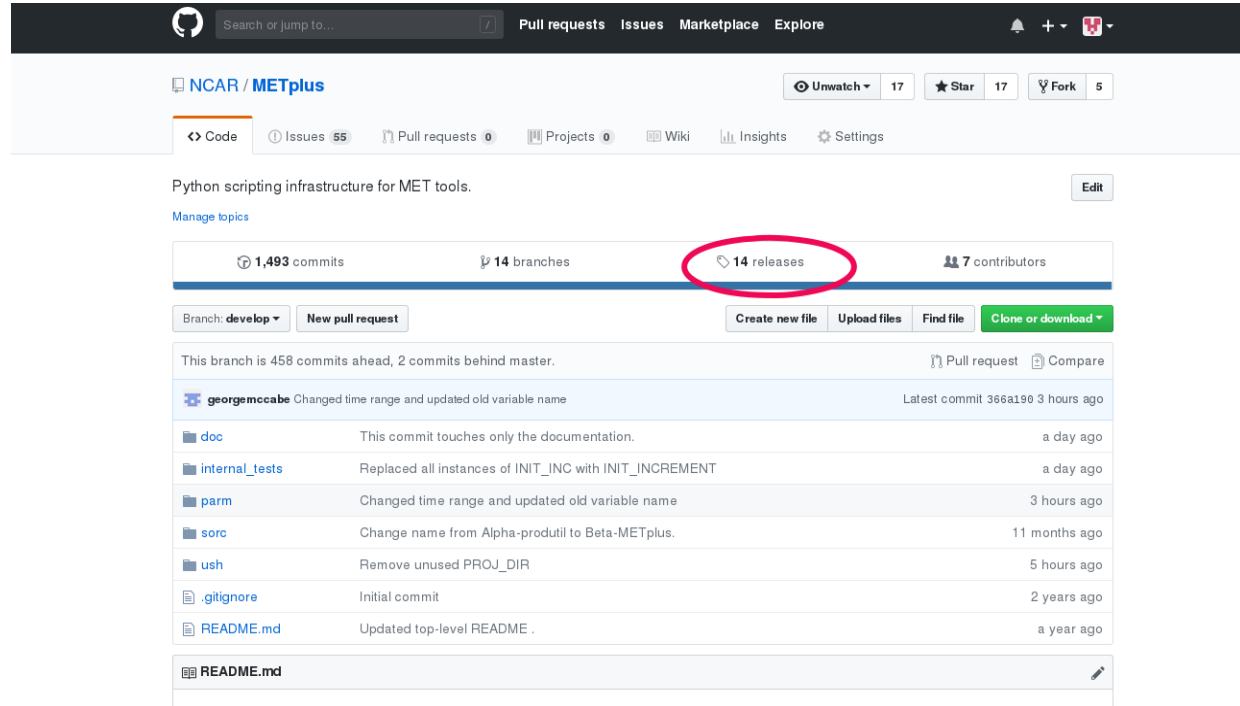
### Install/Configure/Run Guide

@ref METplus\_install\_guide

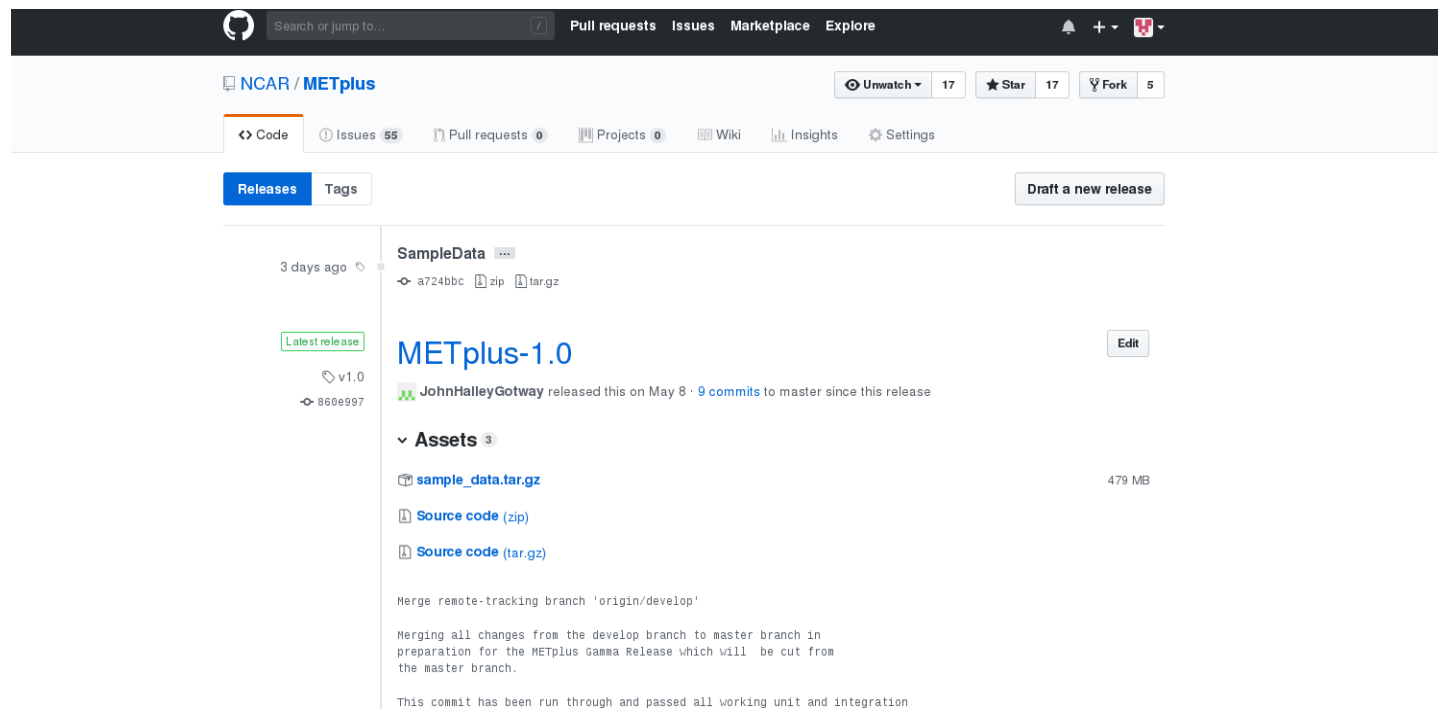
### Produtil Guide

@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*
- To your PYTHONPATH, add: *full-path-to-METplus/ush:full-path-to-METplus/parm*
- 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`  
`setenv PYTHONPATH ~/METplus/ush:~/METplus/parm:$PYTHONPATH`  
`# optional`  
`setenv JLOGFILE ~/jlog_out`

**.bashrc:**

- Open your .bashrc file and do the following:
- To your PATH, add : *full-path-to-METplus/ush*
- To your PYTHONPATH, add *full-path-to-METplus/parm*
- 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:$PATH`  
`export PYTHONPATH="~/METplus/ush:~/METplus/parm:$PYTHONPATH"`  
`#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 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

```
>master_metplus.py
```

Does nothing, a usage message appears, indicating that other config files are required to perform useful tasks.

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 config 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 System Configuration

This chapter is a guide on configuring METplus.

### 3.1 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.

## 3.2 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.

## 3.3 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/metplus\_system.conf files
- Two configuration files are used in this use case, track\_and\_intensity.conf file and tcpr\_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 tcpr\_plotter\_wrapper.py is then used (wraps the MET tool plot\_tcpr.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 '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/tcpr\_mean\_median.conf file
    - \* 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 RSCRIPTS\_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\_intensity/tcpr\_mean\_median.conf
    - \* When complete, you will have a log file in the output directory you specified, and under the tc\_pairs directory you will see .tct 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 tcpr\_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

## 3.4 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

---

### 3.4.1 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:**

---

#### **ONLY\_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:**

## **3.4.2 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:**

---



### 3.4.3 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

---

**3.4.4 D****DEMO\_YR**

The demo year. This is an optional value used by the plot\_TCMR.R script, (which is wrapped by tcmr\_plotter\_wrapper.py). Please refer to Chapter 21 in the MET User's Guide for more details.

*Used by:* tcmr\_plotter\_wrapper.py

*Family:* [config]

*Default:* Varies

---

**DEP\_VARS**

Corresponds to the optional flag -dep in the plot\_TCMR.R, which is wrapped by tcmr\_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:* tcmr\_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

---

### 3.4.5 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

---

### 3.4.6 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  $\geq 1$ .

*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. If 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. If 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 delimiter. 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  $\geq 1$ . 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  $\geq 1$ . The value can be a single item or a comma separated list of items that must start with a comparison operator ( $>$ ,  $\geq$ ,  $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $gt$ ,  $ge$ ,  $eq$ ,  $ne$ ,  $lt$ ,  $le$ ). 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 tcpr\_plotter\_wrapper.py. This is a list of filtering options for the tc\_stat tool.

**Used by:** tcpr\_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 tcpr\_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:** tcpr\_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 tcpr\_plotter\_wrapper.py. According to the plot\_TCMPR.R usage, this flag is used to disable footnote (date).

**Used by:** tcpr\_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

---



### 3.4.7 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 probabilistic 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

---

### **3.4.8 H**

---

#### **HFIP\_BASELINE**

Corresponds to the optional `-hfp_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

---

### 3.4.9 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

---

**3.4.10 J****3.4.11 K****3.4.12 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 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-separated 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 included 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\_DATETIME

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

---

### 3.4.13 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, tcpr\_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\_wrapper.py, 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.py

**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 directory 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

Ouput 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

Output directory to write mode-TD files.

**Used by:** mtd\_wrapper.py

**Family:** [dir]

**Default:** Varies

---

### MTD\_SINGLE\_DATA\_SRC

Only used if MTD\_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

---

### 3.4.14 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

---

## 3.4.15 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  $\geq 1$ .

**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 delimiter. 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  $\geq 1$ . 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  $\geq 1$ . 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

---

### 3.4.16 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

---

### 3.4.17 Q

### 3.4.18 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\_TCMR.R script (which is wrapped by tcmr\_plotter\_wrapper.py).

This a comma-separated list of thresholds to specify meaningful differences for the relative performance plot.

**Used by:** tcmr\_plotter\_wrapper.py

**Family:** [config]

**Default:** Varies

---

### 3.4.19 S

---

#### SAVE

Corresponds to the optional -save flag in plot\_TCMR.R (which is wrapped by tcmr\_plotter\_wrapper.py).

This is a yes/no value to indicate whether to save the image (yes).

**Used by:** tcmr\_plotter\_wrapper.py

**Family:** [config]

**Default:** Varies

---

#### SAVE\_DATA

Corresponds to the optional -save\_data flag in plot\_TCMR.R (which is wrapped by tcmr\_plotter\_wrapper.py).

Indicates whether to save the filtered track data to a file instead of deleting it.

**Used by:** tcmr\_plotter\_wrapper.py

**Family:** [config]

**Default:** Varies

---

#### SCATTER\_X

Corresponds to the optional -scatter\_x flag in plot\_TCMR.R (which is wrapped by tcmr\_plotter\_wrapper.py).

This is a comma-separated list of x-axis variable columns to plot.

**Used by:** tcmr\_plotter\_wrapper.py

**Family:** [config]

**Default:** Varies

---

#### SCATTER\_Y

Corresponds to the optional -scatter\_y flag in plot\_TCMR.R (which is wrapped by tcmr\_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 YYYYMMDDHH.

**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

---

### 3.4.20 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\_HH:mm:ss

*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\_HH:mm:ss

*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\_HH:mm:ss

*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\_HH:mm:ss

*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\_HHmms

*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  $\leq 0$ . 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

---

**3.4.21 U****3.4.22 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

---

### 3.4.23 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

---

### 3.4.24 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

---

### 3.4.25 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

---

### 3.4.26 Z

## 3.5 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