METplus Version 2.1

Automation for the Model Evaluation Tools

Developmental Testbed Center
Boulder, Colorado

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

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

May 2019

Contents

Foreword: A note to METplus users

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

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

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

New for METplus v2.1

METplus v2.1 Release Notes:

Configuration:

- Added gather_by_date.conf for grid-to-grid and grid-to-obs use cases
- Created specific directory for plotting use case config files
- Added error checking so it is more clear what config items need to be added or changed to run
- Renamed many config variables for clarity

  - Variables specific to a wrapper will contain the app name in the variable name
  - Error message at startup listing all deprecated config items and the suggested new item to use

- MET configuration files updated to contain new default values that were recently added
- Current time environment variables set to be used in MET or METplus config files if needed (See section describing Timing Control 4.3.1).
- Added INIT_SEQ for looping by valid time to calculate forecast lead list (instead of specifying with LEAD_SEQ)
- Adding window begin and end variables for individual wrappers instead of having the same window for every wrapper (See OBS_WINDOW_BEGIN and OBS_WINDOW_END in section describing A-Z Config Glossary4.5)
- Added option to skip processing of a file in some wrappers (i.e. pb2nc or regrid_data_plane) if the output file already exists (See PB2NC_SKIP_IF_OUTPUT_EXISTS in section describing A-Z Config Glossary 4.5)

Wrapper specific:

- make_plots_wrapper.py
  - Reworked following the changes to the work done on stat_analysis_wrapper.py
- stat_analysis_wrapper.py
  - Reworked to give users greater use of the MET stat_analysis tool
- pcp_combine_wrapper.py
  - Added support for -derive mode
  - Fixed bugs in -subtract mode and -sum mode
- pb2nc_wrapper.py
  - Refactored to allow looping by valid time and utilize filename templates instead of regex
- point_stat_wrapper.py
  - Refactored to allow looping by valid time and utilize filename templates instead of regex
  - Added verification mask and neighborhood width/shape configurations
- ensemble_stat_wrapper.py
  - Refactored to allow looping by valid time
- grid_stat_wrapper.py
  - Added verification mask and neighborhood width/shape configurations

- series_by_lead_wrapper.py
  - Changed configuration names for forecast hour to more closely match other wrappers (See LEAD_SEQ_[N], LEAD_SEQ_[N]_LABEL, SERIES_BY_LEAD_GROUP_FCSTS, and LEAD_SEQ in section describing A-Z Config Glossary 4.5)

General:

- Improved string template substitution and extraction functionality to be more flexible and to allow time shifting/truncating (See section describing Directory and Filename Template Info 4.3.3)
- Improve timed handling so filename templates can be more flexible
- Added 'now' time item to allow start and end time of run to be defined relative to the current time at execution (See section describing Timing Control 4.3.1)
- Logfiles now contains METplus version number and start/end logs to easily discern multiple runs within a single log file
- Various bug fixes

METplus v2.0.4 Release Notes:

Configuration:

- Updated config files to match sample data directory structure

General:

- Moved large mask files from repository to sample data tarballs
- Improved logging message clarity
- List METplus version number in final configuration file and logging output

METplus v2.0.3 Release Notes:

Configuration:

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

- Added automatically generated CLOCK_TIME config variable to keep track of time METplus was run

Wrapper specific:

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

General:

- Compressed input files with certain file extensions (gz, zip, bz2) will be automatically uncompressed and placed into a staging area for use in METplus (with option to scrub staging directory after run) - Gempak files now can automatically be converted to NetCDF for use in METplus (See [FCST/OBS]_[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 describing User Defined Config 4.6)

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

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

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

The citation for this User's Guide should be:

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

Acknowledgments

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

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

Chapter 1

Overview of METplus

1.1 Purpose and organization of the User's Guide

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

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

1.2 The Developmental Testbed Center (DTC)

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

METplus will also be available to DTC visitors and to the WRF modeling community for testing and evaluation of new model capabilities, applications in new environments, and so on. The METplus release schedule is coincident with the MET release schedule and the METplus major release number is six less than the MET major release number (e.g. MET 8.0 is released with METplus 2.0).

1.3 METplus goals and design philosophy

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

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

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

1.4 METplus components

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

1.5 Future development plans

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

1.6 Code support

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

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

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

Chapter 2

Software Installation/Getting Started

2.1 Introduction

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

2.2 Supported architectures

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

2.3 Programming/scripting languages

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

2.4 Pre-requisites

The following software is required to run METplus:

- Python 2.7

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

2.5 METplus directory structure

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

```
METplus

    doc
    internal_tests
    parm
    sorc
    ush
    README.md
```

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

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

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

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

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

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

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

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

2.6 Getting the METplus source code

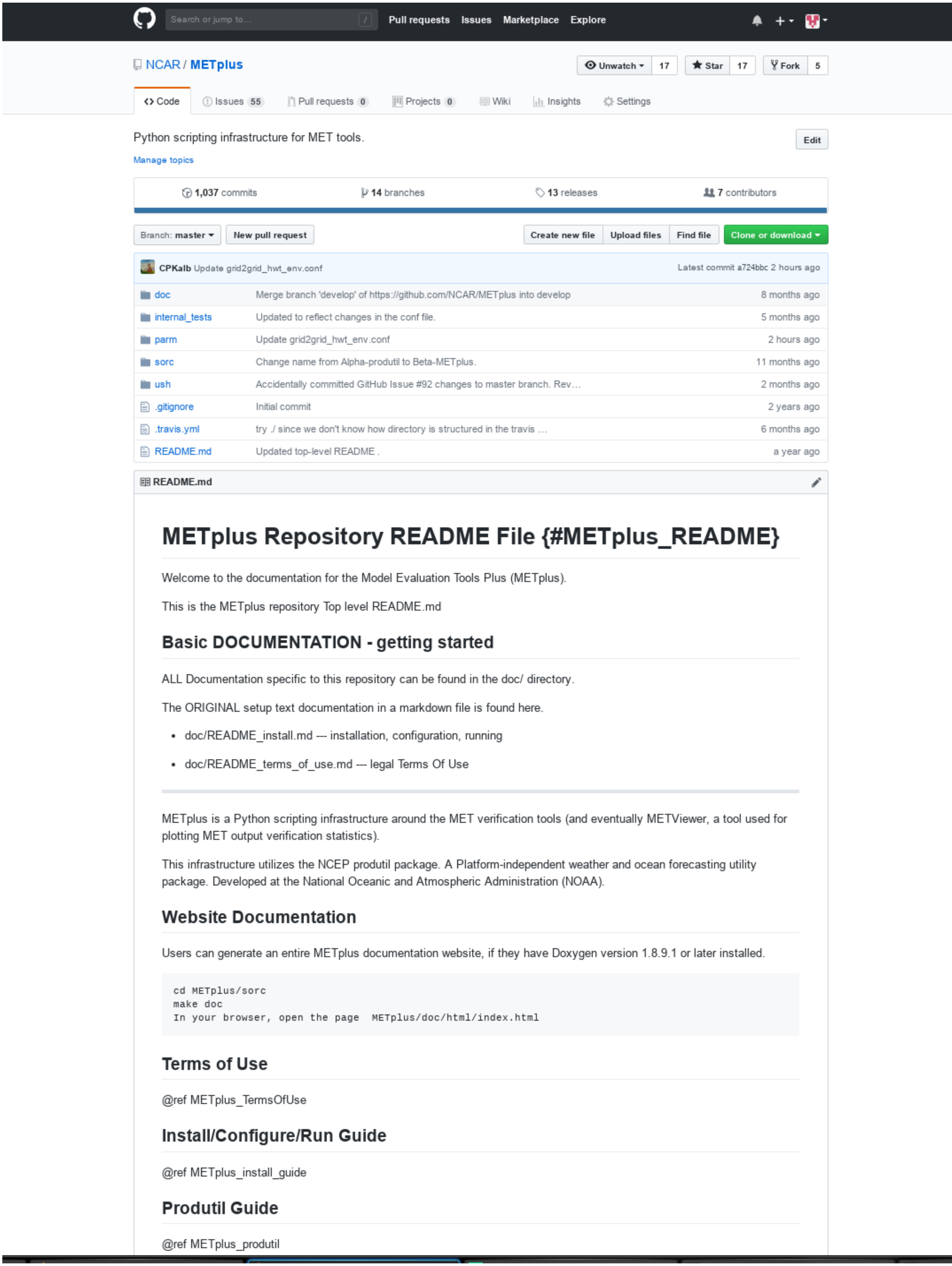
The METplus source code is available for download from a public GitHub repository. You can retrieve the source code through your web browser or the command line.

2.6.1 Get the source code via your Web Browser

2.6.1.1 Source code only:

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

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



Python scripting infrastructure for MET tools.

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

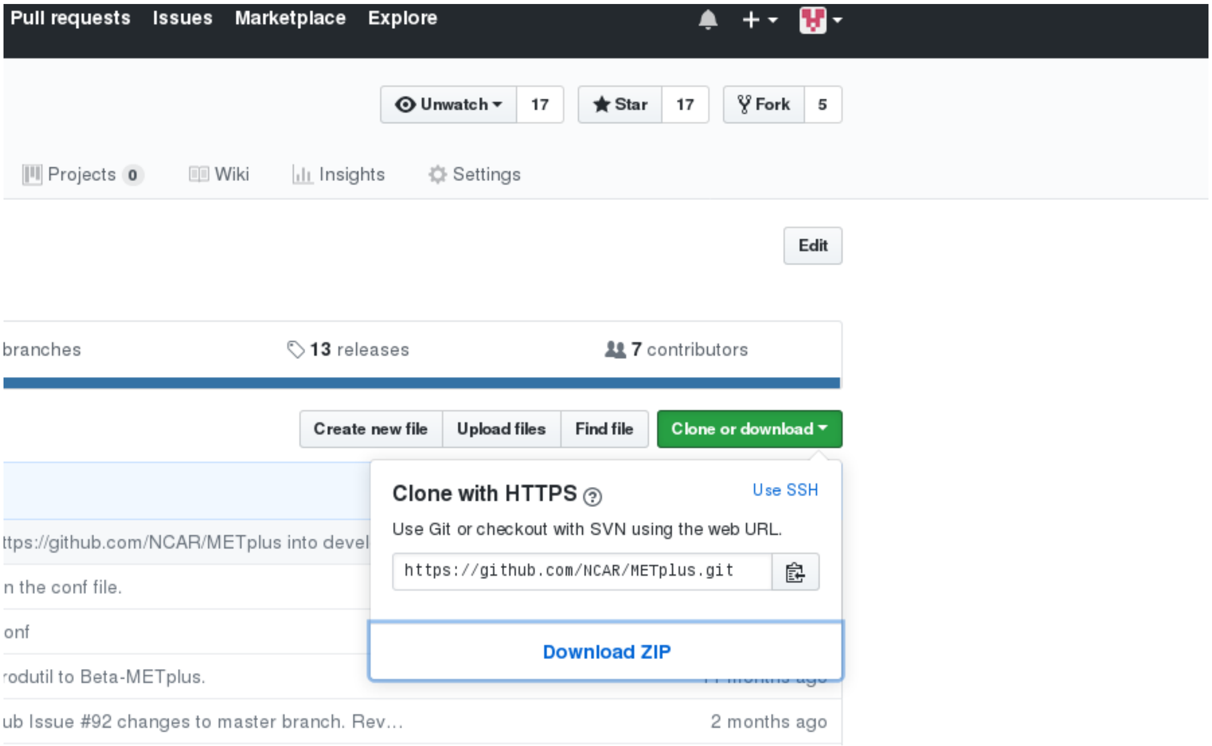
Install/Configure/Run Guide

@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" :



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/NCAR/METplus.git
```

ttps://github.com/NCAR/METplus into devel

n the conf file.

onf

rodutil to Beta-METplus.

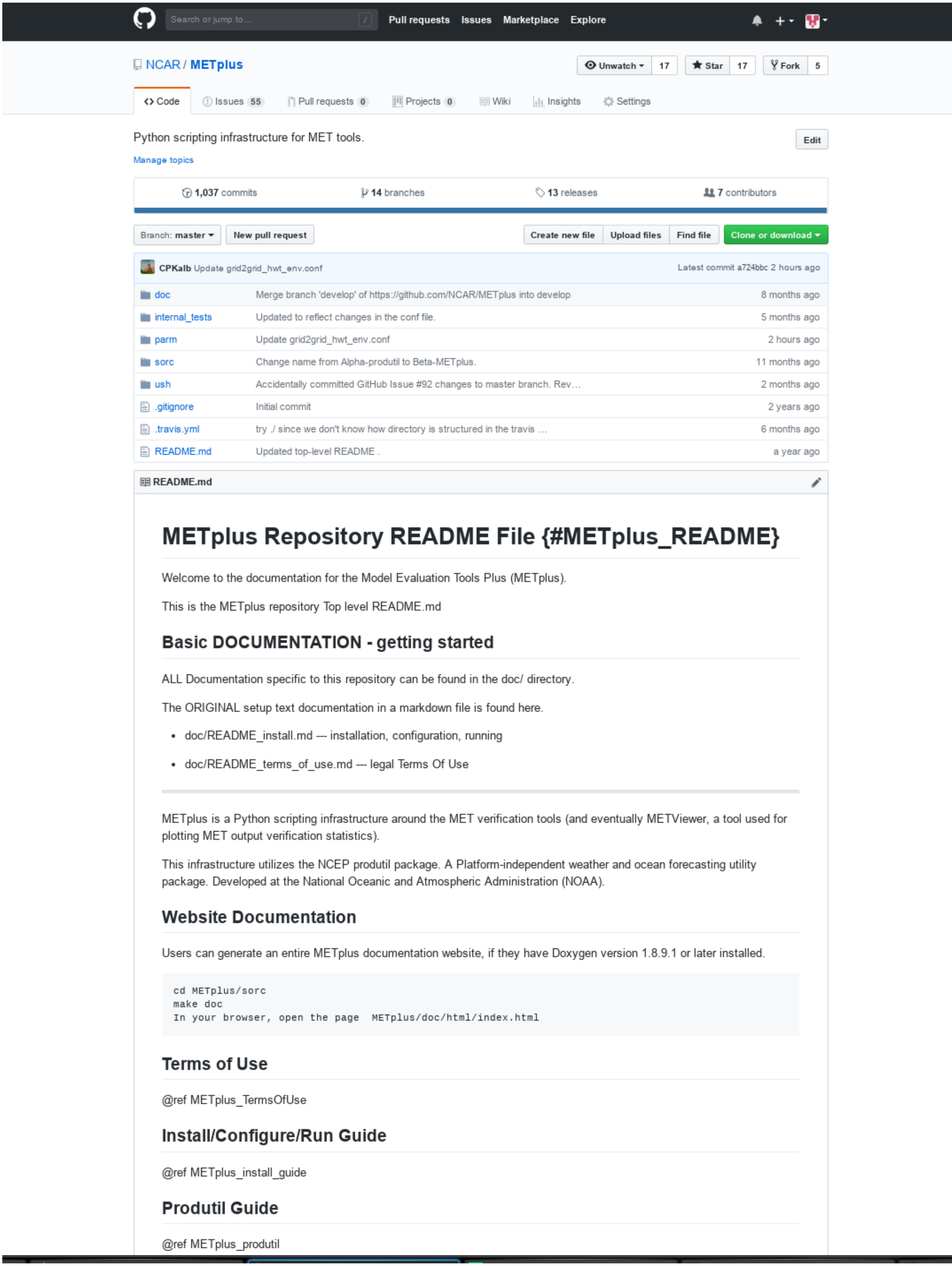
ub Issue #92 changes to master branch. Rev...

- 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:



NCAR / METplus

Python scripting infrastructure for MET tools.

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

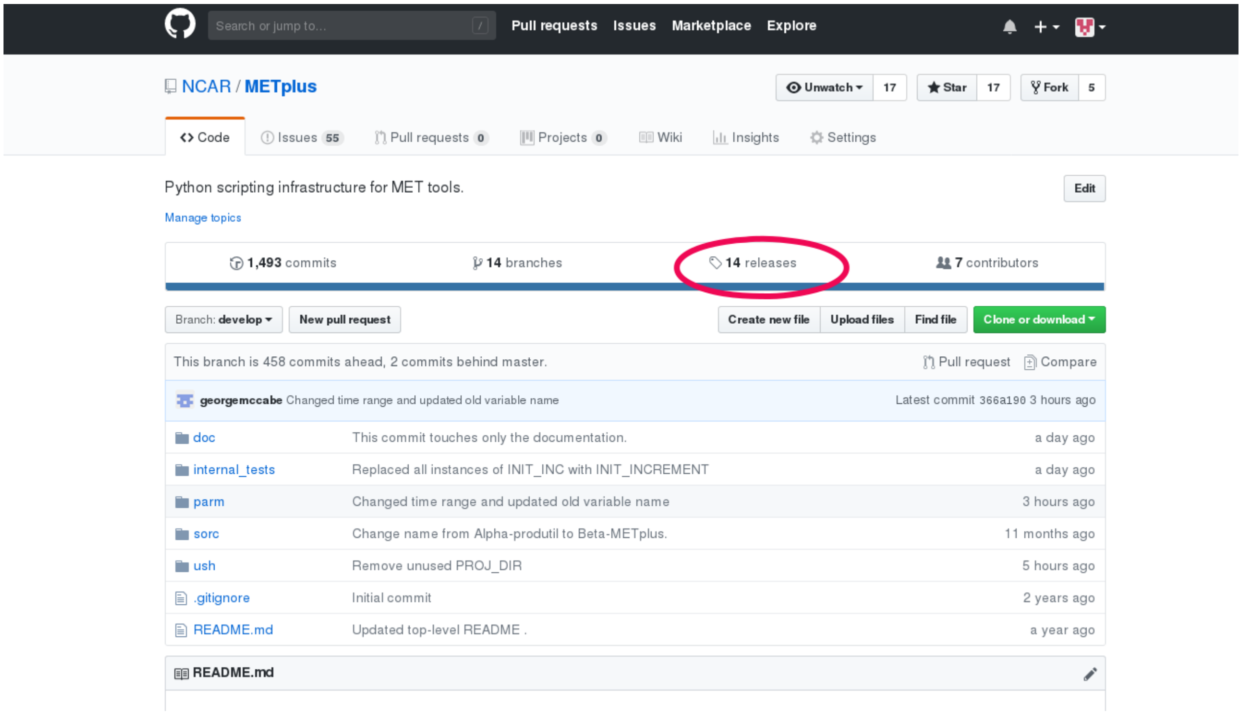
Install/Configure/Run Guide

@ref METplus_install_guide

Produtil Guide

@ref METplus_produtil

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



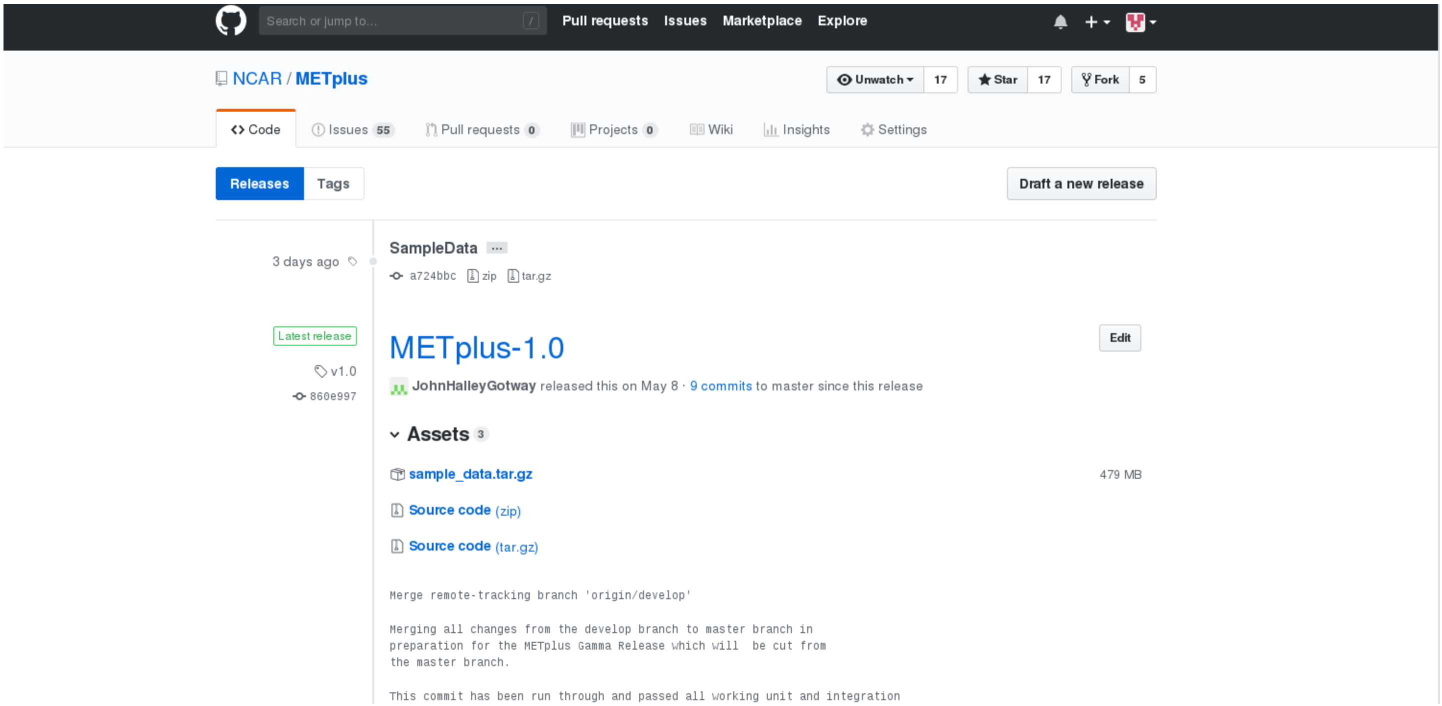
Python scripting infrastructure for MET tools.

This branch is 458 commits ahead, 2 commits behind master.

georgemccabe Changed time range and updated old variable name — Latest commit 366a190 3 hours ago

doc	This commit touches only the documentation.	a day ago
internal_tests	Replaced all instances of INIT_INC with INIT_INCREMENT	a day ago
parm	Changed time range and updated old variable name	3 hours ago
sorc	Change name from Alpha-produtil to Beta-METplus.	11 months ago
ush	Remove unused PROJ_DIR	5 hours ago
.gitignore	Initial commit	2 years ago
README.md	Updated top-level README .	a year ago

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



METplus-1.0

JohnHalleyGotway released this on May 8 · 9 commits to master since this release

sample_data.tar.gz

Source code (zip)

Source code (tar.gz)

```
Merge remote-tracking branch 'origin/develop'

Merging all changes from the develop branch to master branch in
preparation for the METplus Gamma Release which will  be cut from
the master branch.

This commit has been run through and passed all working unit and integration
```

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

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

2.6.2 Get the source code via Command line

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

2.7 Set up your environment

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

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

.cshrc:

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

.bashrc:

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

  ```
  export PATH=~/METplus/ush:$PATH
  #optional
  export JLOGFILE=~/
  ```

2.8 Set up METplus Configuration files

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

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

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

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

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

2.9 Running METplus

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

Example: Using a “default”configuration

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

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

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

or

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

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

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

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

Example: Using a use-case configuration

```
>master_metplus.py -c use_cases/feature_relative/feature_relative.conf
```

Runs METplus using the defaults set in the three config files found in parm/metplus_config. Any variables defined in these three config files can be over-ridden in the parm/use_cases/feature_relative/feature_relative.conf file. METplus will run using the values specified in the feature_relative.conf file.

Example: Using example configuration to perform specific evaluation (e.g. Model 1 vs. Obs1, Model 1 vs Obs 2, Model 2 vs. Obs 1, etc.)

```
>master_metplus.py -c use_cases/feature_relative/feature_relative.conf \
-c use_cases/feature_relative/example/series_by_lead_all_fhrs.conf
```

This runs METplus using the defaults set in the three config files found in parm/metplus_config, where variables can be over-ridden by parm/use_cases/feature_relative/feature_relative.conf or in parm/use_cases/feature_relative/example/series_by_lead_all_fhrs.conf. The order in which conf files are called is important. Variables that are defined in intermediate conf files will be over-ridden by the same variables set in the conf file following it, or the last conf file.

Chapter 3

METplus Python Wrappers

This chapter provides a description of each supported Python wrapper in METplus. A METplus wrapper is generally a Python script that encapsulates the behavior of a corresponding MET tool. Each of these sections can be added to the PROCESS_LIST configuration list variable. The Configuration section of each wrapper section below lists the METplus configuration variables that are specific to that wrapper organized by config file section. You can find more information about each item in the A-Z Config Glossary (4.5).

3.1 CyclonePlotter

3.1.1 Description

This wrapper does not have a corresponding MET tool but instead wraps the logic necessary to create plots of cyclone tracks. Currently only the output from the MET tc-pairs tool can be plotted.

3.1.2 Configuration

[dir]

CYCLONE_INPUT_DIR 4.5.3

CYCLONE_OUTPUT_DIR 4.5.3

[config]

CYCLONE_INIT_DATE 4.5.3

CYCLONE_INIT_HOUR 4.5.3

CYCLONE_MODEL 4.5.3

CYCLONE_PLOT_TITLE 4.5.3

CIRCLE_MARKDER_SIZE 4.5.3

CROSS_MARKER_SIZE 4.5.3

GENERATE_TRACK_ASCII 4.5.7

Deprecated:

CYCLONE_OUT_DIR

3.2 EnsembleStat

3.2.1 Description

Used to configure the MET tool ensemble_stat.

3.2.2 Configuration

[dir]

OBS_ENSEMBLE_STAT_POINT_INPUT_DIR 4.5.15

OBS_ENSEMBLE_STAT_GRID_INPUT_DIR 4.5.15

FCST_ENSEMBLE_STAT_INPUT_DIR 4.5.6

ENSEMBLE_STAT_OUTPUT_DIR 4.5.5

[filename_templates]

OBS_ENSEMBLE_STAT_POINT_INPUT_TEMPLATE 4.5.15

OBS_ENSEMBLE_STAT_GRID_INPUT_TEMPLATE 4.5.15

FCST_ENSEMBLE_STAT_INPUT_TEMPLATE 4.5.6

[config]

ENSEMBLE_STAT_ONCE_PER_FIELD 4.5.5

FCST_ENSEMBLE_STAT_INPUT_DATATYPE 4.5.6

OBS_ENSEMBLE_STAT_INPUT_POINT_DATATYPE 4.5.15

OBS_ENSEMBLE_STAT_INPUT_GRID_DATATYPE 4.5.15

ENSEMBLE_STAT_GRID_VX 4.5.5

ENSEMBLE_STAT_CONFIG_FILE 4.5.5

ENSEMBLE_STAT_MET_OBS_ERROR_TABLE 4.5.5

ENSEMBLE_STAT_N_MEMBERS 4.5.5

OBS_ENSEMBLE_STAT_WINDOW_BEGIN 4.5.15

OBS_ENSEMBLE_STAT_WINDOW_END 4.5.15

ENS_VAR[N]_NAME (optional) 4.5.5

ENS_VAR[N]_LEVELS (optional) 4.5.5

ENS_VAR[N]_THRESH (optional) 4.5.5

ENS_VAR[N]_OPTIONS (optional) 4.5.5

Deprecated:

ENSEMBLE_STAT_OUT_DIR

ENSEMBLE_STAT_CONFIG

3.3 ExtractTiles

3.5.1 Description

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

3.3.1 Configuration

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

[dir]

EXTRACT_TILES_OUTPUT_DIR 4.5.5

[config]

LON_ADJ4.5.12

LAT_ADJ 4.5.12

NLAT 4.5.14

NLON 4.5.14

DLON 4.5.4

DLAT 4.5.4

EXTRACT_TILES_FILTER_OPTS 4.5.5

EXTRACT_TILES_VAR_LIST 4.5.5

Deprecated:

EXTRACT_OUT_DIR

3.4 GempakToCF

3.4.1 Description

Used to configure the utility GempakToCF.

3.4.2 Configuration

[exe]

GEMPAKTOCF_CLASSPATH 4.5.7

[dir]

GEMPAKTOCF_INPUT_DIR 4.5.7

GEMPAKTOCF_OUTPUT_DIR 4.5.7

[filename_templates]

GEMPAKTOCF_INPUT_TEMPLATE 4.5.7

GEMPAKTOCF_OUTPUT_TEMPLATE 4.5.7

[config]

GEMPAKTOCF_SKIP_IF_OUTPUT_EXISTS 4.5.7

3.5 GridStat

3.5.1 Description

Used to configure the MET tool grid_stat.

3.5.2 Configuration

[dir]

FCST_GRID_STAT_INPUT_DIR 4.5.6

OBS_GRID_STAT_INPUT_DIR 4.5.15

GRID_STAT_OUTPUT_DIR 4.5.7

[filename_templates]

FCST_GRID_STAT_INPUT_TEMPLATE 4.5.6

OBS_GRID_STAT_INPUT_TEMPLATE 4.5.15

GRID_STAT_VERIFICATION_MASK_TEMPLATE (optional) 4.5.7

[config]

GRID_STAT_CONFIG_FILE 4.5.7

FCST_GRID_STAT_INPUT_DATATYPE 4.5.6

OBS_GRID_STAT_INPUT_DATATYPE 4.5.15

GRID_STAT_ONCE_PER_FIELD 4.5.7

FCST_GRID_STAT_PROB_THRESH (optional) 4.5.6

OBS_GRID_STAT_PROB_THRESH (optional) 4.5.15

GRID_STAT_NEIGHBORHOOD_WIDTH (optional) 4.5.7

GRID_STAT_NEIGHBORHOOD_SHAPE (optional) 4.5.7

FCST_GRID_STAT_WINDOW_BEGIN (optional) 4.5.6

FCST_GRID_STAT_WINDOW_END (optional) 4.5.6

OBS_GRID_STAT_WINDOW_BEGIN (optional) 4.5.15

OBS_GRID_STAT_WINDOW_END (optional) 4.5.15

Deprecated:

GRID_STAT_OUT_DIR

GRID_STAT_CONFIG

3.6 MakePlots

3.6.1 Description

The make_plots_wrapper creates various statistical plots using python scripts for the various METplus use cases. This can only be run following stat_analysis_wrapper when LOOP_ORDER = processes. To run make_plots_wrapper, include MakePlots in PROCESS_LIST.

3.6.2 Configuration

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

[dir]

PLOTTING_SCRIPTS_DIR 4.5.16

STAT_FILES_INPUT_DIR 4.5.19

PLOTTING_OUTPUT_DIR 4.5.16

[config]

VERIF_CASE 4.5.22

VERIF_TYPE 4.5.22

PLOT_TIME 4.5.16

VALID/INIT_BEG 4.5.22 4.5.22

VALID/INIT_END 4.5.22 4.5.9

VALID_HOUR_METHOD 4.5.22

VALID_HOUR_BEG 4.5.22

VALID_HOUR_END 4.5.22

VALID_HOUR_INCREMENT 4.5.22

INIT_HOUR_METHOD 4.5.9

INIT_HOUR_BEG 4.5.9

INIT_HOUR_END 4.5.9

INIT_HOUR_INCREMENT 4.5.9

MODELn_NAME 4.5.13

MODELn_OBS_NAME 4.5.13

MODELn_NAME_ON_PLOT 4.5.13

FCST_VARn_NAME 4.5.6

FCST_VARn_LEVELS 4.5.6

REGION_LIST 4.5.18

LEAD_LIST 4.5.12

INTERP 4.5.9

PLOT_STATS_LIST 4.5.16

CI_METHOD 4.5.3

VERIF_GRID 4.5.22

EVENT_EQUALIZATION 4.5.5

The following values are **optional** in the METplus configuration file:

FCST_VARn_THRESH 4.5.6

FCST_VARn_OPTIONS 4.5.6

VARn_FOURIER_DECOMP 4.5.22

VARn_WAVE_NUM_LIST 4.5.22

3.7 Mode

3.7.1 Description

Used to configure the MET tool mode.

3.7.2 Configuration

[dir]

FCST_MODE_INPUT_DIR 4.5.6

OBS_MODE_INPUT_DIR 4.5.15

MODE_OUTPUT_DIR 4.5.13

[filename_templates]

FCST_MODE_INPUT_TEMPLATE 4.5.6

OBS_MODE_INPUT_TEMPLATE 4.5.15

[config]

MODE_CONFIG_FILE 4.5.13

FCST_MODE_INPUT_DATATYPE 4.5.6

OBS_MODE_INPUT_DATATYPE 4.5.15

MODE_QUILT 4.5.13

MODE_CONV_RADIUS 4.5.13

FCST_MODE_CONV_RADIUS 4.5.6

OBS_MODE_CONV_RADIUS 4.5.15

MODE_CONV_THRESH 4.5.13

FCST_MODE_CONV_THRESH 4.5.6

OBS_MODE_CONV_THRESH 4.5.15

MODE_MERGE_THRESH 4.5.13

FCST_MODE_MERGE_THRESH 4.5.6

OBS_MODE_MERGE_THRESH 4.5.15

MODE_MERGE_FLAG 4.5.13

FCST_MODE_MERGE_FLAG 4.5.6

OBS_MODE_MERGE_FLAG 4.5.15

MODE_MERGE_CONFIG_FILE 4.5.13

FCST_MODE_WINDOW_BEGIN 4.5.6

FCST_MODE_WINDOW_END 4.5.6

OBS_MODE_WINDOW_BEGIN 4.5.15

OBS_MODE_WINDOW_END 4.5.15

Deprecated:

MODE_OUT_DIR

MODE_CONFIG

3.8 MTD

3.8.1 Description

Used to configure the MET tool mtd (mode time domain).

3.8.2 Configuration

[dir]

FCST_MTD_INPUT_DIR 4.5.6

OBS_MTD_INPUT_DIR 4.5.15

MTD_OUTPUT_DIR 4.5.13

[filename_templates]

FCST_MTD_INPUT_TEMPLATE 4.5.6

OBS_MTD_INPUT_TEMPLATE 4.5.15

[config]

MTD_CONFIG_FILE 4.5.13

MTD_MIN_VOLUME 4.5.13

MTD_SINGLE_RUN 4.5.13

MTD_SINGLE_DATA_SRC 4.5.13

FCST_MTD_INPUT_DATATYPE 4.5.6

OBS_MTD_INPUT_DATATYPE 4.5.15

FCST_MTD_CONV_RADIUS 4.5.6

FCST_MTD_CONV_THRESH 4.5.6

OBS_MTD_CONV_RADIUS 4.5.15

OBS_MTD_CONV_THRESH 4.5.15

Deprecated:

MTD_OUT_DIR

MTD_CONFIG

3.9 PB2NC

3.9.1 Description

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

3.9.2 Configuration

[dir]

PB2NC_INPUT_DIR 4.5.16

PB2NC_OUTPUT_DIR 4.5.16

[filename_templates]

PB2NC_INPUT_TEMPLATE 4.5.16

PB2NC_OUTPUT_TEMPLATE 4.5.16

[config]

PB2NC_SKIP_IF_OUTPUT_EXISTS 4.5.16

PB2NC_OFFSETS 4.5.16

PB2NC_INPUT_DATATYPE 4.5.16

PB2NC_CONFIG_FILE 4.5.16

PB2NC_MESSAGE_TYPE (optional) 4.5.16

PB2NC_STATION_ID (optional) 4.5.16

PB2NC_GRID (optional) 4.5.16

PB2NC_POLY 4.5.16

PB2NC_OBS_BUFR_VAR_LIST (optional) 4.5.16

PB2NC_TIME_SUMMARY_FLAG 4.5.16

PB2NC_TIME_SUMMARY_BEG 4.5.16

PB2NC_TIME_SUMMARY_END 4.5.16

PB2NC_TIME_SUMMARY_VAR_NAMES 4.5.16

PB2NC_TIME_SUMMARY_TYPES 4.5.16

PB2NC_WINDOW_BEGIN 4.5.16

PB2NC_WINDOW_END 4.5.16

Deprecated:

PREPBUFR_DATA_DIR

PREPBUFR_MODEL_DIR_NAME

PREPBUFR_DIR_REGEX

PREPBUFR_FILE_REGEX

NC_FILE_TMPL

PB2NC_VERTICAL_LEVEL

OBS_BUFR_VAR_LIST

TIME_SUMMARY_FLAG

TIME_SUMMARY_BEG

TIME_SUMMARY_END

TIME_SUMMARY_VAR_NAMES

TIME_SUMMARY_TYPE

OVERWRITE_NC_OUTPUT

VERTICAL_LOCATION

3.10 PcpCombine

3.10.1 Description

The pcp_combine_wrapper is a Python script that encapsulates the MET pcp_combine tool. It provides the infrastructure to combine or extract from files to build desired accumulations.

3.10.2 Configuration

[dir]

FCST_PCP_COMBINE_INPUT_DIR 4.5.6

FCST_PCP_COMBINE_OUTPUT_DIR 4.5.6

OBS_PCP_COMBINE_INPUT_DIR 4.5.15

OBS_PCP_COMBINE_OUTPUT_DIR 4.5.15

[filename_templates]

FCST_PCP_COMBINE_INPUT_TEMPLATE 4.5.6

FCST_PCP_COMBINE_OUTPUT_TEMPLATE 4.5.6

OBS_PCP_COMBINE_INPUT_TEMPLATE 4.5.15

OBS_PCP_COMBINE_OUTPUT_TEMPLATE 4.5.15

[config]

FCST_IS_PROB 4.5.6

OBS_IS_PROB 4.5.15

FCST_PCP_COMBINE_[N]_FIELD_NAME 4.5.6

OBS_PCP_COMBINE_[N]_FIELD_NAME 4.5.15

FCST_PCP_COMBINE_DATA_INTERVAL 4.5.6

OBS_PCP_COMBINE_DATA_INTERVAL 4.5.15

FCST_PCP_COMBINE_TIMES_PER_FILE 4.5.6

OBS_PCP_COMBINE_TIMES_PER_FILE 4.5.15

FCST_PCP_COMBINE_IS_DAILY_FILE 4.5.6

OBS_PCP_COMBINE_IS_DAILY_FILE 4.5.15

FCST_PCP_COMBINE_INPUT_DATATYPE 4.5.6

OBS_PCP_COMBINE_INPUT_DATATYPE 4.5.15

FCST_PCP_COMBINE_INPUT_LEVEL 4.5.6

OBS_PCP_COMBINE_INPUT_LEVEL 4.5.15

FCST_PCP_COMBINE_RUN 4.5.6

OBS_PCP_COMBINE_RUN 4.5.15

FCST_PCP_COMBINE_METHOD 4.5.6

OBS_PCP_COMBINE_METHOD 4.5.15

FCST_PCP_COMBINE_MIN_FORECAST 4.5.6

OBS_PCP_COMBINE_MIN_FORECAST 4.5.15

FCST_PCP_COMBINE_MAX_FORECAST 4.5.6

OBS_PCP_COMBINE_MAX_FORECAST 4.5.15

FCST_PCP_COMBINE_STAT_LIST 4.5.6

OBS_PCP_COMBINE_STAT_LIST 4.5.15

FCST_PCP_COMBINE_DERIVE_LOOKBACK 4.5.6

OBS_PCP_COMBINE_DERIVE_LOOKBACK 4.5.15

PCP_COMBINE_SKIP_IF_OUTPUT_EXISTS 4.5.16

Deprecated:

PCP_COMBINE_METHOD

FCST_MIN_FORECAST

FCST_MAX_FORECAST

OBS_MIN_FORECAST

OBS_MAX_FORECAST

FCST_DATA_INTERVAL

OBS_DATA_INTERVAL

FCST_IS_DAILY_FILE

OBS_IS_DAILY_FILE

FCST_TIMES_PER_FILE

OBS_TIMES_PER_FILE

FCST_LEVEL

OBS_LEVEL

3.11 PointStat

3.11.1 Description

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

3.11.2 Configuration

[dir]

FCST_POINT_STAT_INPUT_DIR 4.5.6

OBS_POINT_STAT_INPUT_DIR 4.5.15

POINT_STAT_OUTPUT_DIR 4.5.16

[filename_templates]

FCST_POINT_STAT_INPUT_TEMPLATE 4.5.6

OBS_POINT_STAT_INPUT_TEMPLATE 4.5.15

POINT_STAT_VERIFICATION_MASK_TEMPLATE (optional) 4.5.16

[config]

POINT_STAT_OFFSETS 4.5.16

FCST_POINT_STAT_INPUT_DATATYPE 4.5.6

OBS_POINT_STAT_INPUT_DATATYPE 4.5.15

POINT_STAT_CONFIG_FILE 4.5.16

MODEL 4.5.13

POINT_STAT_REGRID_TO_GRID 4.5.16

POINT_STAT_GRID 4.5.16

POINT_STAT_POLY 4.5.16

POINT_STAT_STATION_ID 4.5.16

POINT_STAT_MESSAGE_TYPE 4.5.16

FCST_POINT_STAT_WINDOW_BEGIN (optional) 4.5.6

FCST_POINT_STAT_WINDOW_END (optional) 4.5.6

OBS_POINT_STAT_WINDOW_BEGIN (optional) 4.5.15

OBS_POINT_STAT_WINDOW_END (optional) 4.5.15

POINT_STAT_NEIGHBORHOOD_WIDTH (optional) 4.5.16

POINT_STAT_NEIGHBORHOOD_SHAPE (optional) 4.5.16

Deprecated:

FCST_INPUT_DIR

OBS_INPUT_DIR

START_HOUR

END_HOUR

BEG_TIME

FCST_HR_START

FCST_HR_END

FCST_HR_INTERVAL

OBS_INPUT_DIR_REGEX

FCST_INPUT_DIR_REGEX

FCST_INPUT_FILE_REGEX

OBS_INPUT_FILE_REGEX

OBS_INPUT_FILE_TMPL

FCST_INPUT_FILE_TPOINTMPL

REGRID_TO_GRID

3.12 RegridDataPlane

3.12.1 Description

Used to configure the MET tool regrid_data_plane.

3.12.2 Configuration

[dir]

FCST_REGRID_DATA_PLANE_INPUT_DIR 4.5.6

OBS_REGRID_DATA_PLANE_INPUT_DIR 4.5.15

[filename_templates]

FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE 4.5.6

OBS_REGRID_DATA_PLANE_INPUT_TEMPLATE 4.5.15

[config]

FCST_REGRID_DATA_PLANE_RUN 4.5.6

OBS_REGRID_DATA_PLANE_RUN 4.5.15

REGRID_DATA_PLANE_SKIP_IF_OUTPUT_EXISTS 4.5.18

REGRID_DATA_PLANE_VERIF_GRID 4.5.18

FCST_REGRID_DATA_PLANE_INPUT_DATATYPE 4.5.6

OBS_REGRID_DATA_PLANE_INPUT_DATATYPE 4.5.15

Deprecated:

VERIFICATION_GRID

3.13 SeriesByInit

3.13.1 Description

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

3.13.2 Configuration

[dir]

SERIES_BY_INIT_FILTERED_OUTPUT_DIR 4.5.19

SERIES_BY_INIT_OUTPUT_DIR 4.5.19

[regex_patterns]

FCST_TILE_PREFIX 4.5.6

ANLY_TILE_PREFIX 4.5.1

FCST_TILE_REGEX 4.5.6

ANLY_TILE_REGEX 4.5.1

FCST_NC_TILE_REGEX 4.5.6

ANLY_NC_TILE_REGEX 4.5.1

FCST_ASCII_REGEX_LEAD 4.5.6

ANLY_ASCII_REGEX_LEAD 4.5.1

[config]

INIT_BEG 4.5.9

INIT_END 4.5.9

INIT_INCREMENT 4.5.9

INIT_HOUR_END 4.5.9

INIT_INCLUDE 4.5.9

INIT_EXCLUDE 4.5.9

SERIES_ANALYSIS_FILTER_OPTS 4.5.19

Deprecated:

SERIES_INIT_FILTERED_OUT_DIR

3.14 SeriesByLead

3.14.1 Description

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

3.14.2 Configuration

The input track and model data files are defined in any one of the user's METplus configuration files. If creating a final configuration file that overrides all other config files, it is customary to define the MODEL_DATA_DIR, pointing to the directory where all model data resides. The full file path to the INIT_INCLUDE and INIT_EXCLUDE are used to list the times in YYYYMMDD_HH format to include or exclude from your time window. If these values are undefined (i.e. no value is set for the variable), then all available times in your time window will be considered. For example, if your data is available every 6 hours and you are interested in creating a series analysis from init time 20180601 to 20180615 for all available times, from 00z to 23z, you would set the following:

[dir]

SERIES_BY_LEAD_FILTERED_OUTPUT 4.5.19

SERIES_BY_LEAD_OUTPUT_DIR 4.5.19

[config]

INIT_BEG 4.5.9

INIT_TIME_FMT 4.5.9

INIT_END 4.5.9

INIT_INCREMENT 4.5.9

SERIES_BY_LEAD_GROUP_FCSTS 4.5.19

LEAD_SEQ_*n 4.5.12*

LEAD_SEQ_*n*_LABEL 4.5.12

SERIES_ANALYSIS_FILTER_OPT 4.5.19

VAR_LIST

STAT_LIST

Deprecated:

SERIES_LEAD_FILTERED_OUT_DIR

3.15 StatAnalysis

3.15.1 Description

The stat_analysis_wrapper encapsulates the behavior of the MET stat_analysis tool. It provides the infrastructure to summarize and filter the MET .stat files. stat_analysis_wrapper can be run in two different methods. First is to look at the STAT lines for a single date, to use this method set LOOP_ORDER = times. Second is to look at the STAT lines over a span of dates, to use this method set LOOP_ORDER = processes. To run stat_analysis_wrapper, include StatAnalysis in PROCESS_LIST.

3.15.2 Configuration

The following values must be defined in the METplus configuration file for running with LOOP_ORDER = times:

[dir]

STAT_ANALYSIS_LOOKIN_DIR4.5.19

STAT_ANALYSIS_OUTPUT_DIR 4.5.19

LOOP_BY 4.5.12

VALID/INIT_TIME_FMT 4.5.22

VALID/INIT_BEG 4.5.22 4.5.9

VALID/INIT_END 4.5.22 4.5.9

VALID_HOUR_METHOD 4.5.22

VALID_HOUR_BEG 4.5.22

VALID_HOUR_END 4.5.22

VALID_HOUR_INCREMENT 4.5.22

INIT_HOUR_METHOD 4.5.9

INIT_HOUR_BEG 4.5.9

INIT_HOUR_END 4.5.9

INIT_HOUR_INCREMENT 4.5.9

STAT_ANALYSIS_CONFIG 4.5.19

MODEL 4.5.13

OBTYPE 4.5.15

JOB_NAME 4.5.10

JOB_ARGS 4.5.10

The following values are **optional** in the METplus configuration file for running with LOOP_ORDER = times:

DESC 4.5.4

FCST_LEAD 4.5.6

FCST_VAR_NAME 4.5.6

FCST_VAR_LEVEL 4.5.6

OBS_VAR_NAME 4.5.15

OBS_VAR_LEVEL4.5.15

REGION 4.5.18

INTERP 4.5.9

INTERP_PTS 4.5.9

FCST_THRESH 4.5.6

COV_THRESH 4.5.3

LINE_TYPE 4.5.12

STAT_ANALYSIS_DUMP_ROW_TMPL 4.5.19

STAT_ANALYSIS_OUT_STAT_TMPL 4.5.19

The following values **must** be defined in the METplus configuration file for running with LOOP_ORDER = processes:

STAT_ANALYSIS_OUTPUT_DIR 4.5.19

VERIF_CASE 4.5.22

VERIF_TYPE 4.5.22

PLOT_TIME 4.5.16

VALID/INIT_BEG 4.5.22 4.5.9

VALID/INIT_END 4.5.22 4.5.9

VALID_HOUR_METHOD 4.5.22

VALID_HOUR_BEG 4.5.22

VALID_HOUR_END 4.5.22

VALID_HOUR_INCREMENT 4.5.22

INIT_HOUR_METHOD 4.5.9

INIT_HOUR_BEG 4.5.9

INIT_HOUR_END 4.5.9

INIT_HOUR_INCREMENT 4.5.9

STAT_ANALYSIS_CONFIG 4.5.19

MODELn_NAME 4.5.13

MODELn_OBS_NAME 4.5.13

MODELn_NAME_ON_PLOT 4.5.13

FCST_VARn_NAME 4.5.6

FCST_VARn_LEVELS 4.5.6

REGION_LIST 4.5.18

LEAD_LIST 4.5.12

INTERP 4.5.9

LINE_TYPE 4.5.12

The following values are optional in the METplus configuration file for running with LOOP_ORDER = processes:

FCST_VARn_THRESH 4.5.6

FCST_VARn_THRESH 4.5.6

FCST_VARn_OPTIONS 4.5.6

VARn_FOURIER_DECOMP 4.5.22

VARn_WAVE_NUM_LIST 4.5.22

Deprecated:

STAT_ANALYSIS_OUT_DIR

3.16 TcPairs

3.16.1 Description

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

3.16.2 Configuration

[dir]

ADECK_TRACK_DATA_DIR 4.5.1

BDECK_TRACK_DATA_DIR 4.5.2

TRACK_DATA_SUBDIR_MOD 4.5.20

TC_PAIRS_DIR 4.5.20

[config]

TC_PAIRS_CONFIG_FILE 4.5.20

INIT_BEG 4.5.9

INIT_END 4.5.9

INIT_INCREMENT 4.5.9

INIT_HOUR_END 4.5.9

INIT_INCLUDE 4.5.9

INIT_EXCLUDE 4.5.9

TOP_LEVEL_DIRS 4.5.20

MODEL 4.5.13

STORM_ID 4.5.19

BASIN 4.5.2

CYCLONE 4.5.3

STORM_NAME 4.5.19

DLAND_FILE 4.5.4

TRACK_TYPE 4.5.20

ADECK_FILE_PREFIX 4.5.1

BDECK_FILE_PREFIX 4.5.2

MISSING_VAL_TO_REPLACE 4.5.13

MISSING_VAL 4.5.13

3.17 TcStat

3.17.1 Description

Used to configure the MET tool tc_stat. This wrapper can be run by listing it in the PROCESS_LIST, or can be called from the ExtractTiles wrapper (via the MET tc-stat command line commands).

3.17.2 Configuration

[dir]

TC_STAT_INPUT_DIR 4.5.20

TC_STAT_OUTPUT_DIR 4.5.20

[config]

TC_STAT_RUN_VIA 4.5.20

TC_STAT_CONFIG_FILE 4.5.20

TC_STAT_CMD_LINE_JOB 4.5.20

TC_STAT_JOBS_LIST 4.5.20

TC_STAT_AMODEL 4.5.20

TC_STAT_BMODEL 4.5.20

TC_STAT_DESC 4.5.20

TC_STAT_STORM_ID 4.5.20

TC_STAT_BASIN 4.5.20

TC_STAT_CYCLONE 4.5.20

TC_STAT_STORM_NAME 4.5.20

TC_STAT_INIT_BEG 4.5.20

TC_STAT_INIT_INCLUDE 4.5.20

TC_STAT_INIT_EXCLUDE 4.5.20

TC_STAT_INIT_HOUR 4.5.20

TC_STAT_VALID_BEG 4.5.20

TC_STAT_VALID_END 4.5.20

TC_STAT_VALID_INCLUDE 4.5.20

TC_STAT_VALID_EXCLUDE 4.5.20

TC_STAT_VALID_HOUR 4.5.20

TC_STAT_LEAD_REQ 4.5.20

TC_STAT_INIT_MASK 4.5.20

TC_STAT_VALID_MASK 4.5.20

TC_STAT_VALID_HOUR 4.5.20

TC_STAT_LEAD 4.5.20

TC_STAT_TRACK_WATCH_WARN 4.5.20

TC_STAT_COLUMN_THRESH_NAME 4.5.20

TC_STAT_COLUNN_THRESH_VAL 4.5.20

TC_STAT_COLUMN_STR_NAME 4.5.20

TC_STAT_COLUMN_STR_VAL 4.5.20

TC_STAT_INIT_THRESH_NAME 4.5.20

TC_STAT_INIT_THRESH_VAL 4.5.20

TC_STAT_INIT_STR_NAME 4.5.20

TC_STAT_INIT_STR_VAL 4.5.20

TC_STAT_WATER_ONLY 4.5.20

TC_STAT_LANDFALL 4.5.20

TC_STAT_LANDFALL_BEG 4.5.20

TC_STAT_LANDFALL_END 4.5.20

TC_STAT_MATCH_POINTS 4.5.20

3.18 TCMPRPlotter

3.18.1 Description

The tcmpr_plotter_wrapper is a Python script that wraps the R script plot_tcmpr.R. This script is usefule for plotting the calculated statistics for the output from the MET-TC tools. This script, and other R scripts are included in the MET installation. Please refer to section 21.2.3 of the MET User's Guide for usage information.

3.18.2 Configuration

The following are configuration settings that correspond to the tcmpr_plotter wrapper. A description of what these values represent are found in the appendix in chapter 4, METplus System Configuration

LOOP ORDER 4.5.12

CONFIG_FILE 4.5.3

PREFIX 4.5.16

TITLE 4.5.20

SUBTITLE 4.5.19

XLAB 4.5.24

YLAB 4.5.25

XLIM4.5.24

YLIM 4.5.25

FILTER 4.5.6

FILTERED_TCST_DATA_FILE 4.5.6

DEP_VARS 4.5.4

SCATTER_X 4.5.4

SCATTER_Y 4.5.4

SKILL_REF 4.5.4

SERIES 4.5.4

SERIES_CI 4.5.4

LEGEND 4.5.12

LEAD 4.5.12

PLOT_TYPES 4.5.16

RP_DIFF 4.5.18

DEMO_YR 4.5.4

HFIP_BASELINE 4.5.8

FOOTNOTE_FLAG 4.5.6

PLOT_CONFIG_OPTS 4.5.16

SAVE_DATA 4.5.4

The following are TCMPR flags, if set to ’no’, then don’t set flag, if set to ’yes’, then set the flag

NO_EE 4.5.14

NO_LOG 4.5.14

SAVE 4.5.19

TCMPR_DATA_DIR 4.5.20

TCMPR_PLOT_OUTPUT_DIR 4.5.20

Deprecated:

TCMPR_PLOT_OUT_DIR

3.19 WaveletStat

3.19.1 Description

NOTE: This wrapper has not been tested for use in METplus.

Chapter 4

METplus System Configuration

This chapter is a guide on configuring METplus.

4.1 Config Best Practices

Below is a list of Best Practices:

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

4.2 Config File Structure

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

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

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

4.3 Common Config Variables

4.3.1 Timing Control

This section describes the METplus configuration variables that are used to control which times are processed. It also covers functionality that is useful for processing data in realtime by setting run times based on the clock time when METplus is started.

4.3.1.1 LOOP_BY

METplus can be configured to loop over a set of valid times or a set of initialization times. This is controlled by the METplus configuration variable called LOOP_BY. If the value of this variable is set to INIT or RETRO, looping will be relative to initialization time. If the value is set to VALID or REALTIME, looping will be relative to valid time. Older versions of METplus used a True/False variable named LOOP_BY_INIT. METplus still supports using this variable, although it is recommended that users update their config files to use LOOP_BY instead. LOOP_BY_INIT will eventually be removed.

4.3.1.2 Looping by Valid Time

When looping over valid time (LOOP_BY = VALID or LOOP_BY = REALTIME), the following variables must be set:

VALID_TIME_FMT
This is the format of the valid times the user can configure in METplus. The value of VALID_BEG and VALID_END must correspond to this format. Example: VALID_TIME_FMT = %Y%m%d%H. Using this format, the valid time range values specified must be defined as YYYYMMDDHH, i.e. 2019020112.

VALID_BEG
This is the first valid time that will be processed. The format of this variable is controlled by VALID_TIME_FMT. For example, if VALID_TIME_FMT = %Y%m%d, then VALID_BEG must be set to a valid time matching YYYYMMDD, such as 20190201.

VALID_END
This is the last valid time that can be processed. The format of this variable is controlled by VALID_TIME_FMT. For example, if VALID_TIME_FMT = %Y%m%d, then VALID_END must be set to a valid time matching YYYYMMDD, such as 20190202. Note that the time specified for this variable will not necessary be processed. It is used to determine the cutoff of run times that can be processed. For example, if METplus is configured to start at 20190201 and end at 20190202 processing data in 48 hour increments, it will process valid time 20190201 then increment the run time to 20190203. This is later than the VALID_END value, so execution will stop. However, if the increment is set to 24 hours (see VALID_INCREMENT), then METplus will process valid times 20190201 and 20190202 before ending execution.

VALID_INCREMENT
This is the number of seconds to add to each run time to determine the next run time to process. This value must be greater than or equal to 60 because METplus currently does not support processing intervals less than one minute.

The following is a configuration that will process valid time 20190201 at 0Z until 20190202 at 0Z at 6 hour (21600 second) increments:

```
[config]
LOOP_BY = VALID
VALID_TIME_FMT = %Y%m%d%H
VALID_BEG = 2019020100
VALID_END = 2019020200
VALID_INCREMENT = 21600
```

This will process data valid on 20190201 at 0Z, 6Z, 12Z, and 18Z, as well as 20190202 at 0Z. For each of these valid times, METplus can also loop over a set of forecast leads that are all valid at the current run time. See 'Looping Over Forecast Leads' 4.3.1.4 for more information.

4.3.1.3 Looping by Initialization Time

When looping over inialization time (LOOP_BY = INIT or LOOP_BY = RETRO), the following variables must be set:

INIT_TIME_FMT
This is the format of the initialization times the user can configure in METplus. The value of INIT_BEG and INIT_END must correspond to this format. Example: INIT_TIME_FMT = %Y%m%d%H. Using this format, the initialization time range values specified must be defined as YYYYMMDDHH, i.e. 2019020112.

INIT_BEG
This is the first initialization time that will be processed. The format of this variable is controlled by INIT_TIME_FMT. For example, if INIT_TIME_FMT = %Y%m%d, then INIT_BEG must be set to an initialization time matching YYYYMMDD, such as 20190201.

INIT_END
This is the last initialization time that can be processed. The format of this variable is controlled by INIT_TIME_FMT. For example, if INIT_TIME_FMT = %Y%m%d, then INIT_END must be set to an initialization time matching YYYYMMDD, such as 20190202. Note that the time specified for this variable will not necessary be processed. It is used to determine the cutoff of run times that can be processed. For example, if METplus is configured to start at 20190201 and end at 20190202 processing data in 48 hour increments, it will process 20190201 then increment the run time to 20190203. This is later than the INIT_END value, so execution will stop. However, if the increment is set to 24 hours (see INIT_INCREMENT), then METplus will process initialization times 20190201 and 20190202 before ending execution.

INIT_INCREMENT
This is the number of seconds to add to each run time to determine the next run time to process. This value must be greater than or equal to 60 because METplus currently does not support processing intervals less than one minute.

The following is a configuration that will process initialization time 20190201 at 0Z until 20190202 at 0Z at 6 hour (21600 second) increments:

```
[config]
LOOP_BY = INIT
INIT_TIME_FMT = %Y%m%d%H
INIT_BEG = 2019020100
INIT_END = 2019020200
INIT_INCREMENT = 21600
```

This will process data initialized on 20190201 at 0Z, 6Z, 12Z, and 18Z, as well as 20190202 at 0Z. For each of these initialization times, METplus can also loop over a set of forecast leads that are all initialized at the current run time. See 'Looping Over Forecast Leads' 4.3.1.4 for more information.

4.3.1.4 Looping over Forecast Leads

Many of the METplus wrappers will also loop over a list of forecast leads relative to the current valid/initialization time that is being processed.

LEAD_SEQ

This variable can be set to a comma-separated list of integers to define the forecast leads (hours) that will be processed relative to the initialization/valid time. Currently only hours are supported for these values. Future versions of METplus will allow sub-hourly forecast leads. For example:

```
[config]
LEAD_SEQ = 3, 6, 9
```

If LOOP_BY = VALID and the current run time is 20190201 at 0Z, then three times will be processed:

1) Initialized on 20190131 at 21Z / valid on 20190201 at 0Z

2) Initialized on 20190131 at 18Z / valid on 20190201 at 0Z

3) Initialized on 20190131 at 15Z / valid on 20190201 at 0Z

If LOOP_BY = INIT and the current run time is 20190201 at 0Z, then three times will be processed:

1) Initialized on 20190201 at 0Z / valid on 20190201 at 3Z

2) Initialized on 20190201 at 0Z / valid on 20190201 at 6Z

3) Initialized on 20190201 at 0Z / valid on 20190201 at 9Z

Defining lists with begin_end_incr

You can also define LEAD_SEQ using begin_end_incr(b, e, i) where b = the first value, e = the last value (inclusive), and i = the increment. For example:

```
LEAD_SEQ = begin_end_incr(0, 12, 3)
```

Is equivalent to setting:

```
LEAD_SEQ = 0, 3, 6, 9, 12
```

INIT_SEQ

If METplus is configured to loop by valid time (LOOP_BY = VALID), you can use INIT_SEQ instead of LEAD_SEQ. This is a list of initialization hours that are available in the data. This is useful if you know when the data is initialized and you need to use a different list of forecast leads depending on the valid time being evaluated. For example:

```
[config]
LOOP_BY = VALID
INIT_SEQ = 0, 6, 12, 18
```

At valid time 20190201_0Z, this initialization sequence will build a forecast lead list of 0, 6, 12, 18, 24, 30, etc.
At valid time 20190201_1Z, this initialization sequence will build a forecast least list of 1, 7, 13, 19, 25, 31, etc.

You can also restrict the forecast leads that will be used by setting LEAD_SEQ_MIN and LEAD_SEQ_MAX. For example, if you want to only process forecast leads between 12 and 24 you can set:

```
[config]
LEAD_SEQ_MIN = 12
LEAD_SEQ_MAX = 24
```

Using the initialization sequence specified above, you will get the following results:

At valid time 20190201_0Z, this initialization sequence will build a forecast lead list of 12, 18, 24.
At valid time 20190201_1Z, this initialization sequence will build a forecast least list of 13, 19.

Setting minimum and maximum values will also affect the list of forecast leads if you use LEAD_SEQ. LEAD_SEQ takes precedence over INIT_SEQ, so if you have both variables set in your configuration, INIT_SEQ will be ignored in favor of LEAD_SEQ.

Grouping Forecast Leads (LEAD_SEQ_[N])

This is used by SeriesByLead wrapper only. If SERIES_BY_LEAD_GROUP_FCSTS is set to True, then you can define groups of forecast leads that will be evaluated together. You can define any number of these groups by setting configuration variables LEAD_SEQ_1, LEAD_SEQ_2, LEAD_SEQ_3, etc. You can define the value with a comma-separated list of integers (hours) or using the begin_end_incr notation (See ’Defining lists with begin_end_incr’ earlier in this section). Each list must have a corresponding label to describe it using LEAD_SEQ_[N]_LABEL, i.e. LEAD_SEQ_1 must have the corresponding variable LEAD_SEQ_1_LABEL set. For example:

```
[config]
SERIES_BY_LEAD_GROUP_FCSTS = True
LEAD_SEQ_1 = 0, 6, 12, 18
LEAD_SEQ_1_LABEL = Day1
LEAD_SEQ_2 = begin_end_incr(24, 42, 6)
LEAD_SEQ_2_LABEL = Day2
```

4.3.1.5 Realtime Looping

To make running in realtime easier, METplus supports defining the begin and end times relative to the current clock time. For example, if the current time is 20190426_08:17 and you start METplus with:

```
VALID_END = {now?fmt=%Y%m%d%H}
```

then the value of VALID_END will be set to 2019042608. You can also use {today} to substitute the current YYYYMMDD, i.e. 20190426. You cannot change the formatting for the 'today' keyword.

You can use the 'shift' keyword to shift the current time by any number of seconds. For example, if you start METplus at the same clock time with:

```
VALID_BEG = {now?fmt=%Y%m%d%H?shift=-86400}
```

then the value of VALID_BEG will be set to the current clock time shifted by -86400 (24 hours backwards), or 2019042508.

If VALID_INCREMENT is set to 21600 (seconds or 6 hours), then METplus will process valid times:

```
20190425_08
20190425_14
20190425_20
20190426_02
20190426_08
```

However, you may want to configure METplus to process at 0Z, 6Z, 12Z, and 18Z of a given day instead of 2Z, 8Z, 14Z, and 20Z. Having to adjust the shift amount differently if you are running at 8Z or 9Z to get the times to line up would be tedious. Instead, use the 'truncate' keyword. The value set here is the number of seconds that is used to determine the interval of time to round down. If you want to process every 6 hours, set 'truncate' to 21600 (seconds).

```
VALID_BEG = {now?fmt=%Y%m%d%H?shift=-86400?truncate=21600}
```

This will round down the value to the nearest 6 hour interval of time. Starting METplus on or after 6Z but before 12Z on 20190426 will result in VALID_BEG = 2019042506 (clock time shifted backwards by 24 hours then truncated to 6 hours).

Starting METplus on 20190426 at 8:16 with the following configuration:

```
[config]
LOOP_BY = VALID
VALID_TIME_FMT = %Y%m%d%H
VALID_BEG = {now?fmt=%Y%m%d%H?shift=-86400?truncate=21600}
VALID_END = {now?fmt=%Y%m%d%H}
VALID_INCREMENT = 21600
```

will process valid times starting on 20190425 at 6Z every 6 hours until the current run time is later than 20190426 at 8Z, which will result in processing:

20190425 at 6Z
20190425 at 12Z
20190425 at 18Z
20190426 at 0Z
20190426 at 6Z

IMPORTANT NOTE: When using the 'now' keyword, the value of VALID_TIME_FMT must be set to the same value as the 'fmt' used in the 'now' item in VALID_BEG and VALID_END. The relevant values are highlighted in bold in the above example.

4.3.2 Field Info

This section describes how METplus configuration variables can be used to define field information that is sent to the MET applications to read forecast and observation fields.

FCST_VAR[N]_NAME
Set this to the name of a forecast variable that you want to evaluate. N is any integer greater or equal to 1, i.e.

```
[config]
FCST_VAR1_NAME = TMP
FCST_VAR2_NAME = RH
```

FCST_VAR[N]_LEVELS
Set this to a comma-separated list of levels or a single value. FCST_VAR1_LEVELS corresponds to FCST_VAR1_NAME, FCST_VAR2_LEVELS corresponds to FCST_VAR2_NAME, etc.

For example:

```
[config]
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500, P750
```

will process TMP at P500 and TMP at P750.

OBS_VAR[N]_NAME

Set this to the corresponding observation variable that you want to evaluate with FCST_VAR[N]_NAME. If this value is not set for a given [N] value, then it will be assumed that the same name as the forecast name will be used.

OBS_VAR[N]_LEVELS

Set this to a comma-separated list of levels or a single value. If OBS_VAR[N]_LEVELS and OBS_VAR[N]_NAME are not set, it will be assumed that the same name/level as the forecast data will be used.

For example, setting

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500
FCST_VAR2_NAME = RH
FCST_VAR2_LEVELS = P750, P250
```

without setting OBS_VAR1_NAME is the equivalent of setting:

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500
FCST_VAR2_NAME = RH
FCST_VAR2_LEVELS = P750, P250
OBS_VAR1_NAME = TMP
OBS_VAR1_LEVELS = P500
OBS_VAR2_NAME = RH
OBS_VAR2_LEVELS = P750, P250
```

This will compare:
TMP/P500 in the forecast data to TMP/P500 in the observation data
RH/P750 in the forecast data to RH/P750 in the observation data
RH/P250 in the forecast data to RH/P250 in the observation data

If you set:

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500, P750
OBS_VAR1_NAME = TEMP
OBS_VAR1_LEVELS = “(0,*,*)”, “(1,*,*)”
```

This will compare:
Forecast TMP/P500 to observation TEMP at (0,*,*)
Forecast TMP/P750 to observation TEMP at (1,*,*)

IMPORTANT NOTE: NetCDF level values that contain (*,*) notation must be surrounded by quotation marks so it will not be misinterpreted as a list of items.

The number of level items must be equal in each list for a given comparison. If you define separate names for a forecast and observation, you will need to define separate levels for both even if they are equivalent.

For example, setting FCST_VAR1_NAME, FCST_VAR1_LEVELS, and OBS_VAR1_NAME, but not setting OBS_VAR1_LEVELS will result in an error.

The field information will be formatted to match the field info dictionary in the MET config files and passed to the appropriate config file to evaluate the data. The previous configuration comparing TMP (P500 and P750) and TEMP will generate the following in the MET config file:

```
fcst = {
field = [ { name="TMP"; level="P500"; } ];
}

obs = {
field = [ {name=”TEMP”; level=“(0,*,*)”; } ];
}
```

for the first run and:

```
fcst = {
field = [ { name="TMP"; level="P750"; } ];
}

obs = {
field = [ {name=”TEMP”; level=“(1,*,*)”; } ];
}
```

for the second run. Note that some MET applications allow multiple fields to be specified for a single run. If the MET tool allows it and METplus is configured accordingly, these two comparisons would be configured in a single run.

[FCST/OBS]_VAR[N]_THRESH: Set this to a comma-separated list of threshold values to use in the comparison. Each of these values must begin with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt, or le). It is recommended that you use the alphabetic versions (gt, ge, eq, ne, lt, le) because the > and < symbols can cause errors when inserted into MET commands as command line arguments, but most uses of thresholds via METplus are passed into the MET config files and will not cause any issues. For example, setting:

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500
FCST_VAR1_THRESH = le0.5, gt0.4, gt0.5, gt0.8
```

will add the following information to the MET config file:

```
fcst = {
field = [ { name=”TMP”; level=“P500”; cat_thresh=[ le0.5, gt0.4, gt0.5, gt0.8 ]; } ];
}
```

Note that the value for FCST_VAR[N]_THRESH will not be copied to OBS_VAR[N]_THRESH if it is not set. These items are independent of each other.

[FCST/OBS]_VAR[N]_OPTIONS: Set this to add additional information to field dictionary in the MET config file. The item must end with a semi-colon. For example:

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500
FCST_VAR1_OPTIONS = GRIB_lvl_typ = 105;
```

will add the following to the MET config file:

```
fcst = {
field = [ { name=”TMP”; level=“P500”; GRIB_lvl_typ = 105; } ];
}
```

You can specify multiple items as long as they are separated by and end with a semi-colon.

```
FCST_VAR1_NAME = TMP
FCST_VAR1_LEVELS = P500
FCST_VAR1_OPTIONS = GRIB_lvl_typ = 105; ens_phist_bin_size = 0.05;
```

will add the following to the MET config file:

```
fcst = {
field = [ { name=”TMP”; level=“P500”; GRIB_lvl_typ = 105; ens_phist_bin_size = 0.05; } ];
}
```

Note that the value for FCST_VAR[N]_OPTIONS will not be copied to OBS_VAR[N]_OPTIONS if it is not set. These items are independent of each other.

ENS_VAR[N]_[NAME/LEVELS/THRESH/OPTIONS]: Used with EnsembleStat only. Users may want to define the ens dictionary item in the MET EnsembleStat config file differently than the fcst dictionary item. If this is the case, you can use this variable. If it is not set, the value set in the corresponding FCST_VAR[N]_NAME will be used in the ens dictionary.

4.3.3 Directory and Filename Template Info

The METplus wrappers use directory and filename template configuration variables to find the desired files for a given run.

Using Templates to Find Observation Data:
The following configuration variables describe input observation data:

```
[dir]
OBS_GRID_STAT_INPUT_DIR = /my/path/to/grid_stat/input/obs
[filename_templates]
OBS_GRID_STAT_INPUT_TEMPLATE = {valid?fmt=%Y%m%d}/prefix.{valid?fmt=%Y%m%d%H}.ext
```

The input directory is the top level directory containing all of the observation data. The template contains items with keywords that will be substituted with time values for each run. After the values are substituted, METplus will check to see if the desired file exists relative to the input directory. At valid time 20190201_12Z, the full desired path of the observation input data to grid_stat is:

/my/path/to/grid_stat/input/obs/20190201/prefix.2019020112.ext

Note that the template contains a dated subdirectory. This cannot go in the directory variable because it changes based on the run time.

METplus does not need be configured to loop by valid time to find files using a template containing valid time information. For example, at initialization time 20190201_12Z and forecast lead 3, the valid time is calculated to be 20190201_15Z and the full desired path of the observation input data to grid_stat is:

/my/path/to/grid_stat/input/obs/20190201/prefix.2019020115.ext

'init' and 'valid' are keywords to notate initalization and valid times respectively. 'lead', 'offset', 'da_init', and 'cycle' are also keywords that can be used to find forecast data and data assimilation data.

Using Templates to Find Forecast Data

Most forecast files contain the initialization time and the forecast lead in the filename. The keywords 'init' and 'lead' can be used to describe the template of these files:

```
[dir]
FCST_GRID_STAT_INPUT_DIR = /my/path/to/grid_stat/input/fcst
[filename_templates]
FCST_GRID_STAT_INPUT_TEMPLATE = prefix.{init?fmt=%Y%m%d%H}_f{lead?fmt=%3H}.ext
```

For a valid time of 20190201_0Z and a forecast lead of 3, METplus will look for the following forecast file with initialization time of 20190131_21Z:

/my/path/to/grid_stat/input/fcst/prefix.2019013121_f003.ext

Using Templates to Find Data Assimilation Data

Some data assimilation files contain offset and da_init (data assimilation initialization) values in the filename. These values are used to determine the value time of the data. Consider the following configuration:

```
[config]
PB2NC_OFFSETS = 6, 3
[dir]
PB2NC_INPUT_DIR = /my/path/to/prepbufr
[filename_templates]
PB2NC_INPUT_TEMPLATE = prefix.{da_init?fmt=%Y%m%d}_{cycle?fmt=%H}_off{offset?fmt=%2H}.ext
```

The PB2NC_OFFSETS list tells METplus the order to prioritize files with offsets in the name. At valid time 20190201_12Z, METplus will check if the following file exists:

/my/path/to/prepbufr/prefix.20190201_18_off06.ext

The offset is added to the valid time to get the data assimilation initialization time. Note that 'cycle' can be used interchangeably with 'da_init.' It is generally used to specify the hour of the data that was generated. If that file doesn't exists, it will check:

```
/my/path/to/prepbufr/prefix.20190201_15_off03.ext
```

Shifting Times in Filename Templates

Users can use the 'shift' keyword to adjust the time referenced in the filename template relative to the run time. For example, the input files contain data from 1Z on the date specified in the filename until 1Z the following day. This means that for a 0Z run you want to use the file from the previous day and for the 1Z to 23Z runs you want to use the file that corresponds to the current day.

```
[filename_templates]
OBS_POINT_STAT_INPUT_TEMPLATE = {valid?fmt=%Y%m%d?shift=-3600}.ext
```

Running at valid time 20190201_12Z, shifting the valid time backwards by 1 hour will result in 20190201_11Z, so it will substitute the current day into the template, giving you 20190201.ext. Running at valid time 20190201_0Z, the shift will result in a file time of 20190131_23Z, so 20190131.ext will be generated by the template.

Using Windows to Find Valid Files

The [FCST/OBS]_FILE_WINDOW_[BEGIN/END] configuration variables can be used if the time information in the input data does not exactly line up with the run time but you still want to process the data. The default value is the file window begin and end variables is 0 (seconds). If both values are set to 0, METplus will require that a file matching a template with the exact time requested exists. If either value is non-zero, METplus will examine all of the files under the input directory that match the template, pull out the time information from the files, and use the file with time closest to the run time. For example, consider the configuration:

```
[config]
OBS_FILE_WINDOW_BEGIN = -7200
OBS_FILE_WINDOW_END = 7200
[dir]
OBS_GRID_STAT_INPUT_DIR = /my/grid_stat/input/obs
[filename_templates]
OBS_GRID_STAT_INPUT_TEMPLATE = {valid?fmt=%Y%m%d}/pre.{valid?fmt=%Y%m%d}_{valid?fmt=%H}.ext
```

The run time is 20190201_0Z and the following files exist in the input directory:

```
/my/grid_stat/input/obs/20190131/pre.20190131_22.ext
/my/grid_stat/input/obs/20190131/pre.20190131_23.ext
/my/grid_stat/input/obs/20190201/othertype.20190201_00.ext
/my/grid_stat/input/obs/20190201/pre.20190201_01.ext
/my/grid_stat/input/obs/20190201/pre.20190201_02.ext
```

The first file matches the template and the file time is within the window, so the filename and time difference relative to the valid time (7200 seconds or 2 hours) is saved.

The second file matches the template, the file time is within the window, and the time difference is less than the closest file, so the filename and time difference from the valid time (3600 seconds or 1 hour) is saved.
The third file does not match the template, so it is ignored.
The fourth file matches the template and is within the time range, but it is the same distance away from the valid time as the closest file. GridStat only allows one file to be processed, so it is ignored (PB2NC is currently the only wrapper to allow multiples files to be processed).
The first file matches the template but it is valid outside of the -2 to +2 hour window range, so it is ignored.

Wrapper Specific Windows
A user may need to specify a different window on a wrapper-by-wrapper basis. If this is the case, you can override the file window values for each wrapper. Consider the following configuration:

```
[config]
PROCESS_LIST = PcpCombine, GridStat, EnsembleStat
OBS_FILE_WINDOW_BEGIN = 0
OBS_FILE_WINDOW_END = 0
OBS_GRID_STAT_FILE_WINDOW_BEGIN = -1800
OBS_GRID_STAT_FILE_WINDOW_END = 1800
OBS_ENSEMBLE_STAT_FILE_WINDOW_END = 3600
```

Using this configuration, PcpCombine will use (0,0) and require exact file times. GridStat will use (-1800, 1800) for observation data. EnsembleStat will use (0, 3600) for observation data. OBS_ENSEMBLE_STAT_FILE_WINDOV was not set, so it will use OBS_FILE_WINDOW_BEGIN.

4.4 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/me and METplus_INSTALL/parm/metplus_conf/metplus_system.conf files

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

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

4.5 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_temUplates], [exe], [regex_pattern], etc...
Default: If it makes sense to include a default value (or value shipped in a release), do it here

---

4.5.1 A

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

Used by: TcPairs
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: TcPairs
Family: [dir]
Default: Varies

AMODEL
The model name of the ADeck model data

Used by: CyclonePlotter, TcStat
Family: [config]
Default:

ANLY_ASCII_REGEX_LEAD

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

Used by: SeriesByLead
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: SeriesByLead, SeriesByInit
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: SeriesByLead, SeriesByInit
Family: [regex_pattern]
Default:

4.5.2 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: SeriesByLead, SeriesByInit
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: CyclonePlotter, TcPairs, TcStat
Family: [config]
Default: Varies

BDECK_FILE_PREFIX

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

Used by: TcPairs
Family: [config]
Default: Varies

BDECK_TRACK_DATA_DIR
The input directory where the BDeck track data resides.

Used by: TcPairs
Family: [dir]
Default: Varies

[deprecated] BEG_TIME
Please use INIT_BEG or VALID_BEG instead. Beginning time for analysis in YYYYMMDD format.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

BMODEL
The model name of the BDeck model data.

Used by: TcStat
Family: [config]
Default:

4.5.3 C

CI_METHOD
The method for creating confidence intervals. Valid options are EMC, or NONE.

Used by: MakePlots
Family: [config]
Default:

CIRCLE_MA

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

Used by: CyclonePlotter
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, EnsembleStat, GridStat, Mode, StatAnalysis
Family: [dir]
Default: Varies

CONFIG_FILE
Specific configuration file name to use for MET tools.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

CONVERT_EXE
Path to the ImageMagick “convert” executable.

Used by: PB2NC, PointStat, SeriesByInit, SeriesByLead
Family: [exe]
Default: /path/to

COV_THRESH
Specify the values of the COV_THRESH column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: StatAnalysis
Family: [config]
Default:

CROSS_MAR

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

Used by: CyclonePlotter
Family: [config]
Default: 51

CUT_EXE
Path to the Linux "cut" executable.

Used by: PB2NC, PointStat
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: TcPairs, TcStat
Family: [config]
Default: Varies

CYCLONE_INIT_DATE
Initialization date for the cyclone forecasts in YYYYMMDD format.

Used by: CyclonePlotter
Family: [config]
Default: Varies

CYCLONE_INIT_HR
Initialization hour for the cyclone forecasts in HH format.

Used by: CyclonePlotter
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: CyclonePlotter
Family: [dir]
Default: Varies

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

Used by: CyclonePlotter
Family: [config]
Default: Varies

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

Used by: CyclonePlotter
Family: [dir]
Default: Varies

CYCLONE_PLOT_TITLE
Title string for the cyclone plotter.

Used by: CyclonePlotter
Family: [config]
Default: Varies

4.5.4 D

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

DEP_VARS
Corresponds to the optional flag -dep in the plot_TCMPR.R script, which is wrapped by TCMPRPlotter. The value to this flag is a comma-separated list (no whitespace) of dependent variable columns to plot (e.g.

AMSLP-BMSLP, AMAX_WIND-BMAX_WIND, TK_ERR). If this is undefined, then the default plot for TK_ERR (track error) is generated. Note, if you want the track error plot generated, in addition to other plots, then you need to explicitly list this with the other variables. Please refer to Chapter 21 in the MET User's Guide for more details.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

DESC
A single value or list of values used in the stat_analysis data stratification. Specifies the values of the DESC column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times

Used by: StatAnalysis
Family: [config]
Default: Varies

DLAND_FIL

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: TcPairs
Family: [config]
Default: Varies

DLAT
The value that defines the resolution of the data (in decimal degrees).

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

DLON
The longitude value, in degrees. Set to the value that defines the resolution of the data (in decimal degrees).

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

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

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

4.5.5 E

EGREP_EXE
Path to the Linux "egrep" executable.

Used by: feature_util.py, PB2NC, PointStat
Family: [exe]
Default: /path/to

[deprecated] END_DATE
Please use INIT_END or VALID_END instead.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

[deprecated] END_HOUR
Ending hour for analysis with format HH.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

[deprecated] END_TIME
Ending date string for analysis with format YYYYMMDD.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

ENSEMBLE_STAT_CONFIG_FILE
Specify the absolute path to the configuration file for the MET ensemble_stat tool.

Used by: EnsembleStat
Family: [config]
Default:

ENSEMBLE_STAT_GRID_VX

Used to set the regrid dictionary item 'to_grid' in the MET ensemble_stat config file. See the MET User's Guide for more information.

Used by: EnsembleStat
Family: [config]
Default: FCST

ENSEMBLE_STAT_MET_OBS_ERROR_TABLE

Used by: EnsembleStat
Family: [config]
Default:

ENSEMBLE_STAT_N_MEMBERS

Expected number of ensemble members found. This should correspond to the number of items in FCST_ENSEMBLE_STAT_
If this number differs from the number of files are found for a given run, then ensemble_stat will not run for that time.

Used by: EnsembleStat
Family: [config]
Default:

ENSEMBLE_STAT_ONCE_PER_FIELD

If True, run ensemble_stat separately for each field name/level combination specified in the configuration file. See 4.3.2 for more information on how fields are specified. If False, run ensemble_stat once with all of the fields specified.

Used by: EnsembleStat
Family: [config]
Default: False

ENSEMBLE_STAT_OUTPUT_DIR

Specify the output directory where files from the MET ensemble_stat tool are written.

Used by: EnsembleStat
Family: [dir]
Default: Varies

ENSEMBLE_STAT_OUTPUT_TEMPLATE

Sets the subdirectories below ENSEMBLE_STAT_OUTPUT_DIR using a template to allow run time information. If LOOP_BY = VALID, default value is valid time YYYYMMDDHHMM/ensemble_stat. If LOOP_BY = INIT, default value is init time YYYYMMDDHHMM/ensemble_stat.

Used by: EnsembleStat
Family: [filename_templates]

Default: Varies

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

ENS_VAR1_LEVELS = A06, P500
ENS_VAR2_LEVELS = "(0,*,*)", "(1,*,*)"

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.:
ENS_VAR1_LEVELS
ENS_VAR2_LEVELS
...
ENS_VAR[N]_LEVELS

Used by: EnsembleStat
Family: [config]
Default: Varies

ENS_VAR[N]_NAME
Define the name for the [N]th ensemble variable to be used in the analysis where [N] is an integer >= 1. 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.:
ENS_VAR1_NAME
ENS_VAR2_NAME
...
ENS_VAR[N]_NAME

Used by: EnsembleStat
Family: [config]
Default: Varies

ENS_VAR[N]_OPTIONS
Define the options for the [N]th ensemble variable to be used in the analysis where [N] is an integer >= 1. These addition options will be applied to every name/level/threshold combination for VAR[N]. 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.:
ENS_VAR1_OPTIONS
ENS_VAR2_OPTIONS
...

ENS_VAR[N]_OPTIONS

Used by: EnsembleStat
Family: [config]
Default: Varies

ENS_VAR[N]_THRESH
Define the threshold(s) for the [N]th ensemble variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le). 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.:
ENS_VAR1_THRESH
ENS_VAR2_THRESH
...
ENS_VAR[N]_THRESH

Used by: EnsembleStat
Family: [config]
Default: Varies

EVENT_EQUALIZATION
If event equalization is to be used (True) or not (False). If set to True, if any of the listed models are missing data for a particular time, data for all models will be masked out for this time. If set to False, there are no changes to the data.

Used by: MakePlots
Family: [config]
Default: True

[deprecated]

EXTRACT_OUT_DIR
Please use EXTRACT_TILES_OUTPUT_DIR. Set the output directory for the METplus extract_tiles utility.

Used by: ExtractTiles, SeriesByInit, SeriesByLead
Family: [dir]
Default: Varies

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

Used by: ExtractTiles
Family: [config]
Default: Varies

EXTRACT_T

Set the output directory for the METplus extract_tiles utility.

Used by: ExtractTiles, SeriesByInit, SeriesByLead
Family: [dir]
Default: Varies

EXTRACT_TILES_VAR_LIST
Control what variables the METplus extract_tiles utility runs on. Additional filtering by summary (via the MET tc_stat tool). Please refer to Chapter 20 in the MET Users Guide (TC-STAT Tools) for all the available options for filtering by summary method in tc-stat. If no additional filtering is required, simply leave the value to EXTRACT_TILES_FILTER_OPTS blank/empty in the METplus configuration file.

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

4.5.6 F

[deprecated] FCST_EXACT_VALID_TIME
No longer used. Please use FCST_WINDOW_BEGIN and FCST_WINDOW_END instead. If both of those variables are set to 0, the functionality is the same as FCST_EXACT_VALID_TIME = True.

Used by: GridStat Mode, mtd_wrapper.py
Family: [config]
Default: False

[deprecated] FCST_[N]_FIELD_NAME
Please use FCST_PCP_COMBINE_[N]_FIELD_NAME where N >=1 instead.

Used by: PcpCombine
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: SeriesByLead
Family: [regex_pattern]
Default: Varies

FCST_ENSEMBLE_STAT_FILE_WINDOW_BEGIN
See OBS_ENSEMBLE_STAT_FILE_WINDOW_BEGIN 4.5.15.

Used by: EnsembleStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

FCST_ENSEMBLE_STAT_FILE_WINDOW_END
See OBS_ENSEMBLE_STAT_FILE_WINDOW_END 4.5.15.

Used by: EnsembleStat
Family: [config]
Default: OBS_FILE_WINDOW_END

FCST_ENSEMBLE_STAT_INPUT_DIR
Input directory for forecast files to use with the MET tool ensemble_stat. A corresponding variable exists for observation data called OBS_ENSEMBLE_STAT_INPUT_DIR.

Used by: EnsembleStat
Family: [dir]
Default: Varies

FCST_ENSEMBLE_STAT_INPUT_TEMPLATE
Template used to specify forecast input filenames for the MET tool ensemble_stat. A corresponding variable exists for observation data called OBS_ENSEMBLE_STAT_INPUT_TEMPLATE.

Used by: EnsembleStat
Family: [filename_templates]
Default: Varies

FCST_FILE_WINDOW_BEGIN
See OBS_FILE_WINDOW_BEGIN4.5.15.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
Family: [config]
Default: Varies

FCST_FILE_WINDOW_END
See OBS_FILE_WINDOW_END4.5.15.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
Family: [config]
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:
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:
Family: [filename_templates]
Default: Varies

FCST_GRID_STAT_FILE_WINDOW_BEGIN

See OBS_GRID_STAT_FILE_WINDOW_BEGIN 4.5.15.

Used by: GridStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

FCST_GRID_STAT_FILE_WINDOW_END

See OBS_GRID_STAT_FILE_WINDOW_END 4.5.15.

Used by: GridStat
Family: [config]
Default: OBS_FILE_WINDOW_END

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. A corresponding variable exists for observation data called OBS_GRID_STAT_INPUT_DATATYPE.

Used by: GridStat
Family: [config]
Default: Varies

FCST_GRID_STAT_INPUT_DIR

Input directory for forecast files to use with the MET tool grid_stat. A corresponding variable exists for observation data called OBS_GRID_STAT_INPUT_DIR.

Used by: GridStat
Family: [dir]

Default: Varies

FCST_GRID_STAT_INPUT_TEMPLATE
Template used to specify forecast input filenames for the MET tool grid_stat. A corresponding variable exists for observation data called OBS_GRID_STAT_INPUT_TEMPLATE.

Used by: GridStat
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). A corresponding variable exists for observation data called OBS_GRID_STAT_PROB_THRESH.

Used by: GridStat
Family: [config]
Default: ==0.1

[deprecated] FCST_HR_END
Please use LEAD_SEQ instead.

Used by:
Family: [config]
Default:

[deprecated] FCST_HR_INTERVAL
Please use LEAD_SEQ instead.

Used by:
Family: [config]
Default:

[deprecated] FCST_HR_START
Please use LEAD_SEQ instead.

Used by:
Family: [config]
Default:

[deprecated] FCST_INIT_INTERVAL
Specify the stride for forecast initializations.

Used by: compare_gridded_wrapper.py, EnsembleStat, GridStat, Mode
Family: [config]

Default: Varies

[deprecated] FCST_INPUT_DIR_REGEX
Please use FCST_POINT_STAT_INPUT_DIR instead.

Used by: PointStat
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, GridStat, Mode, PointStat, PcpCombine
Family: [dir]
Default: Varies

[deprecated] FCST_INPUT_FILE_REGEX
Regular expression to use when identifying which forecast file to use.

Used by: PointStat
Family: [regex_pattern]
Default: Varies

[deprecated] FCST_INPUT_FILE_TMPL
Please use FCST_POINT_STAT_INPUT_TEMPLATE instead.

Used by: PointStat
Family: [filename_templates]
Default: Varies

[deprecated] FCST_IS_DAILY_FILE
Please use FCST_PCP_COMBINE_IS_DAILY_FILE instead.

Acceptable values: true/false

Used by: PcpCombine
Family: [config]
Default: False

FCST_IS_PROB
Specify whether the forecast data are probabilistic or not.

Acceptable values: true/false

Used by: EnsembleStat, GridStat, Mode, MTD, PointStat
Family: [config]

Default: False

FCST_LEAD
Specify the values of the FCST_LEAD column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times

Used by: StatAnalysis
Family: [config]
Default: Varies

[deprecated] FCST_LEVEL
Please use FCST_PCP_COMBINE_INPUT_LEVEL instead.

Used by: PcpCombine
Family: [config]
Default: Varies

[deprecated] FCST_MAX_FORECAST
Please use LEAD_SEQ_MAX instead. Specify the maximum forecast lead time to use for the analysis.

Used by: compare_gridded_wrapper.py, EnsembleStat, GridStat, Mode
Family: [config]
Default: Varies

FCST_MODE_CONV_RADIUS
Comma separated list of convolution radius values used by mode for forecast fields. A corresponding variable exists for observation data called OBS_MODE_CONV_RADIUS.

Used by: Mode
Family: [config]
Default:

FCST_MODE_CONV_THRESH
Comma separated list of convolution threshold values used by mode for forecast fields. A corresponding variable exists for observation data called OBS_MODE_CONV_THRESH.

Used by: Mode
Family: [config]
Default:

FCST_MODE_FILE_WINDOW_BEGIN
See OBS_MODE_FILE_WINDOW_BEGIN 4.5.15.

Used by: Mode
Family: [config]

Default: OBS_FILE_WINDOW_BEGIN

FCST_MODE_FILE_WINDOW_END
See OBS_MODE_FILE_WINDOW_END 4.5.15.

Used by: Mode
Family: [config]
Default: OBS_FILE_WINDOW_END

FCST_MODE_MERGE_FLAG
Sets the merge_flag value in the mode config file for forecast fields. Valid values are NONE, THRESH, ENGINE, and BOTH. A corresponding variable exists for observation data called OBS_MODE_MERGE_FLAG.

Used by: Mode
Family: [config]
Default:

FCST_MODE_MERGE_THRESH
Comma separated list of merge threshold values used by mode for forecast fields. A corresponding variable exists for observation data called OBS_MODE_MERGE_THRESH.

Used by: Mode
Family: [config]
Default:

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. A corresponding variable exists for observation data called OBS_MODE_INPUT_DATATYPE.

Used by: Mode
Family: [config]
Default: Varies

FCST_MODE_INPUT_DIR
Input directory for forecast files to use with the MET tool mode. A corresponding variable exists for observation data called OBS_MODE_INPUT_DIR.

Used by: Mode
Family: [dir]
Default: Varies

FCST_MODE_INPUT_TEMPLATE
Template used to specify forecast input filenames for the MET tool mode. A corresponding variable exists for observation data called OBS_MODE_INPUT_TEMPLATE.

Used by: Mode
Family: [filename_templates]
Default: Varies

FCST_MTD_CONV_RADIUS

Comma separated list of convolution radius values used by mode-TD for forecast files. A corresponding variable exists for observation data called OBS_MTD_CONV_RADIUS.

Used by: MTD
Family: [config]
Default:

FCST_MTD_CONV_THRESH

Comma separated list of convolution threshold values used by mode-TD for forecast files. A corresponding variable exists for observation data called OBS_MTD_CONV_THRESH.

Used by: MTD
Family: [config]
Default:

FCST_MTD_FILE_WINDOW_BEGIN

See OBS_MTD_FILE_WINDOW_BEGIN 4.5.15.

Used by: MTD
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

FCST_MTD_FILE_WINDOW_END

See OBS_MTD_FILE_WINDOW_END 4.5.15.

Used by: MTD
Family: [config]
Default: OBS_FILE_WINDOW_END

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. A corresponding variable exists for observation data called OBS_MTD_INPUT_DATATYPE.

Used by: MTD
Family: [config]
Default: Varies

FCST_MTD_INPUT_DIR

Input directory for forecast files to use with the MET tool mode-TD. A corresponding variable exists for observation data called OBS_MTD_INPUT_DIR.

Used by: MTD
Family: [dir]
Default: Varies

FCST_MTD_INPUT_TEMPLATE
Template used to specify forecast input filenames for the MET tool mode-TD. A corresponding variable exists for observation data called OBS_MTD_INPUT_TEMPLATE.

Used by: MTD
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: PcpCombine
Family: [config]
Default: Varies

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

Used by: SeriesByLead, SeriesByInit
Family: [regex_pattern]
Default: Varies

FCST_PCP_COMBINE_[N]_FIELD_NAME
This variable is used to define a [N] hour accumulation NetCDF field in the forecast dataset used in the MET tool pcp_combine. [N] must be an integer >= 1. A corresponding variable exists for observation data called OBS_PCP_COMBINE_[N]_FIELD_NAME.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_DATA_INTERVAL
Specify the accumulation interval of the forecast dataset used by the MET pcp_combine tool when processing daily input files. A corresponding variable exists for observation data called OBS_PCP_COMBINE_DATA_INTERVAL.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_DERIVE_LOOKBACK
Specify how far to look back in time in hours to find files for running the MET pcp_combine tool in derive mode. A corresponding variable exists for observation data called OBS_PCP_COMBINE_DERIVE_LOOKBACK.

Used by: PcpCombine
Family: [config]
Default: ADD

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. A corresponding variable exists for observation data called OBS_PCP_COMBINE_INPUT_DATA_TYPE.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_INPUT_DIR
Specify the input directory for forecast files used with the MET pcp_combine tool. A corresponding variable exists for observation data called OBS_PCP_COMBINE_INPUT_DIR.

Used by: PcpCombine
Family: [dir]
Default: Varies

FCST_PCP_COMBINE_INPUT_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. A corresponding variable exists for observation data called OBS_PCP_COMBINE_INPUT_LEVEL.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_INPUT_TEMPLATE
Template used to specify input filenames for forecast files used by the MET pcp_combine tool. A corresponding variable exists for observation data called OBS_PCP_COMBINE_INPUT_TEMPLATE.

Used by: PcpCombine
Family: [filename_templates]
Default: Varies

FCST_PCP_COMBINE_IS_DAILY_FILE

Specify whether the forecast file is a daily file or not. A corresponding variable exists for observation data called OBS_PCP_COMBINE_IS_DAILY_FILE.

Acceptable values: true/false

Used by: PcpCombine
Family: [config]
Default: False

FCST_PCP_COMBINE_METHOD
Specify the method to be used with the MET pcp_combine tool processing forecast data.
Valid options are ADD, SUM, SUBTRACT, and DERIVE. A corresponding variable exists for observation data called OBS_PCP_COMBINE_METHOD.

Used by: PcpCombine
Family: [config]
Default:_None_

FCST_PCP_COMBINE_MIN_FORECAST
Specify the minimum forecast lead time to use when finding the lowest forecast lead to use in pcp_combine. A corresponding variable exists for observation data called OBS_PCP_COMBINE_MIN_FORECAST.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_MAX_FORECAST
Specify the maximum forecast lead time to use when finding the lowest forecast lead to use in pcp_combine. A corresponding variable exists for observation data called OBS_PCP_COMBINE_MAX_FORECAST.

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_OUTPUT_DIR
Specify the output directory for forecast files generated by the MET pcp_combine tool. A corresponding variable exists for observation data called OBS_PCP_COMBINE_OUTPUT_DIR.

Used by: PcpCombine
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. A corresponding variable exists for observation data called OBS_PCP_COMBINE_OUTPUT_TEMPLATE.

Used by: PcpCombine
Family: [filename_templates]
Default: Varies

FCST_PCP_COMBINE_RUN
Specify whether to run the MET pcp_combine tool on forecast data or not. A corresponding variable exists for observation data called OBS_PCP_COMBINE_RUN.

Acceptable values: true/false

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_STAT_LIST
List of statistics to process when using the MET pcp_combine tool on forecast data in derive mode. A corresponding variable exists for observation data called OBS_PCP_COMBINE_STAT_LIST.

Acceptable values: sum, min, max, range, mean, stdev, vld_count

Used by: PcpCombine
Family: [config]
Default: Varies

FCST_PCP_COMBINE_TIMES_PER_FILE
Specify the number of accumulation intervals of the forecast dataset used by the MET pcp_combine tool when processing daily input files. A corresponding variable exists for observation data called OBS_PCP_COMBINE_TIMES

Used by: PcpCombine
Family: [config]
Default:

FCST_POINT_STAT_FILE_WINDOW_BEGIN
See OBS_POINT_STAT_FILE_WINDOW_BEGIN 4.5.15.

Used by: PointStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

FCST_POINT_STAT_FILE_WINDOW_END
See OBS_POINT_STAT_FILE_WINDOW_END 4.5.15.

Used by: PointStat
Family: [config]
Default: OBS_FILE_WINDOW_END

FCST_POINT_STAT_INPUT_DATATYPE
Specify the data type of the input directory for forecast files used with the MET point_stat tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF. A corresponding variable exists for observation data called OBS_POINT_STAT_INPUT_DATATYPE.

Used by: PointStat
Family: [config]
Default: Varies

FCST_POINT_STAT_INPUT_DIR
Input directory for forecast files to use with the MET tool point_stat. A corresponding variable exists for observation data called OBS_POINT_STAT_INPUT_DIR.

Used by: PointStat
Family: [dir]
Default: Varies

FCST_POINT_STAT_INPUT_TEMPLATE
Template used to specify forecast input filenames for the MET tool point_stat. A corresponding variable exists for observation data called OBS_POINT_STAT_INPUT_TEMPLATE.

Used by: GriPointStat
Family: [filename_templates]
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. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_INPUT_DATATYPE.

Used by: RegridDataPlane
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. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_INPUT_DIR.

Used by: RegridDataPlane
Family: [dir]
Default: Varies

FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE
Template used to specify input filenames for forecast data used by the MET regrid_data_plane tool. It not

set, METplus will use FCST_REGRID_DATA_PLANE_TEMPLATE. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_INPUT_TEMPLATE.

Used by: RegridDataPlane
Family: [filename_templates]
Default: Varies

FCST_REGRID_DATA_PLANE_OUTPUT_TEMPLATE
Template used to specify output filenames for forecast data used by the MET regrid_data_plane tool. It not set, METplus will use FCST_REGRID_DATA_PLANE_TEMPLATE. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_OUTPUT_TEMPLATE.

Used by: RegridDataPlane
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. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_TEMPLATE.

Used by: RegridDataPlane
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. A corresponding variable exists for observation data called OBS_REGRID_DATA_PLANE_OUTPUT_DIR.

Used by: RegridDataPlane
Family: [dir]
Default: Varies

FCST_THRESH
Specify the values of the FCST_THRESH column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: stat_analysis_lead_wrapper.py
Family: [config]
Default:

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: SeriesByInit, SeriesByLead
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, EnsembleStat, MakePlots, met_util.py
Family: [config]
Default: Varies

FCST_VAR_LEVEL
Specify the values of the FCST_VAR_LEVEL column in the MET .stat file to use.

Used by: StatAnalysis
Family: [config]
Default: Varies

FCST_VAR_NAME
Specify the values of the FCST_VAR_NAME column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times

Used by: StatAnalysis
Family: [config]
Default: Varies

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

FCST_VAR1_LEVELS = A06, P500
FCST_VAR2_LEVELS = “(0,*,*)”, “(1,*,*)”

If FCST_VAR[N]_LEVELS is not set but OBS_VAR[N]_LEVELS is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment

the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_LEVELS
FCST_VAR2_LEVELS
...
FCST_VAR[N]_LEVELS

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

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

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

FCST_VAR[N]_THRESH
Define the threshold(s) for the [N]th forecast variable to be used in the analysis where [N] is an integer >= 1. The value can be a single item or a comma separated list of items that must start with a comparison operator (>,>=,==,!=,<,<=,gt,ge,eq,ne,lt,le). If FCST_VAR[N]_THRESH is not set but OBS_VAR[N]_THRESH

is, the same information will be used for both variables. There can be [N] number of these variables defined in configuration files, simply increment the "_VAR1_" string to match the total number of variables being used, e.g.:
FCST_VAR1_THRESH
FCST_VAR2_THRESH
...
FCST_VAR[N]_THRESH

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

FCST_WINDOW_BEGIN
See OBS_WINDOW_BEGIN4.5.15.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
Family: [config]
Default: Varies

FCST_WINDOW_END
See OBS_WINDOW_END4.5.15.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
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: SeriesByLead
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: SeriesByLead
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: SeriesByLead
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: SeriesByLead
Family: [config]
Default: Varies

FHR_GROUP_LABELS
Label strings to use for the forecast groups.

Used by: SeriesByLead
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: SeriesByLead
Family: [config]
Default: Varies

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

Used by: TCMPRPlotter
Family: [config]

Default: Varies

FILTERED_TCST_DATA_FILE
Corresponds to the optional -tcst argument to the plot_TCMPR.R script which is wrapped by TCMPRPlotter. 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: TCMPRPlotter
Family: [config]
Default: Varies

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

FORECAST_TMPL
Filename template used to filter forecast files.

Used by: TcPairs
Family: [filename_templates]
Default: Varies

4.5.7 G

GEMPAKTOCF_CLASSPATH
Path to the GempakToCF binary file and the NetCDF jar file required to run GempakToCF.

Used by: GempakToCF
Family: [exe]
Default: Varies

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

Used by: GempakToCF
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: GempakToCF
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: GempakToCF
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: GempakToCF
Family: [filename_templates]
Default: Varies

GEMPAKTOCF_SKIP_IF_OUTPUT_EXISTS
If True, do not run GempakToCF if output file already exists. Set to False to overwrite files.

Used by: GempakToCF
Family: [config]
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: CyclonePlotter
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

[deprecated] GRID_STAT_CONFIG
Please use GRID_STAT_CONFIG_FILE instead. Specify the absolute path to the configuration file used by the MET grid_stat tool.

Used by: GridStat
Family: [config]
Default: Varies

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

Used by: GridStat
Family: [config]
Default: Varies

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

Used by: GridStat
Family: [config]
Default: False

[deprecated] GRID_STAT_OUT_DIR
Specify the output directory where files from the MET grid_stat tool are written. Please use GRID_STAT_OUTPUT_DIR instead.

Used by: GridStat
Family: [dir]
Default: Varies

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

Used by: GridStat
Family: [dir]
Default: Varies

GRID_STAT_OUTPUT_TEMPLATE
Sets the subdirectories below GRID_STAT_OUTPUT_DIR using a template to allow run time information. If LOOP_BY = VALID, default value is valid time YYYYMMDDHHMM/grid_stat. If LOOP_BY = INIT, default value is init time YYYYMMDDHHMM/grid_stat.

Used by: GridStat
Family: [filename_templates]
Default: Varies

GRID_STAT_VERIFICATION_MASK_TEMPLATE
Template used to specify the verification mask filename for the MET tool grid_stat.

Used by: GridStat
Family: [filename_templates]
Default: Varies

4.5.8 H

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

4.5.9 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, ExtractTiles, MakePlots, master_metplus.py, StatAnalysis, TcPairs, TcStat
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, ExtractTiles, MakePlots, master_metplus.py, StatAnalysis, TcPairs, TcStat
Family: [config]
Default: Varies

INIT_EXCLUDE

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

Used by: TcPairs, TcStat
Family: [config]
Default: Varies

INIT_HOUR_BEG

Specify the beginning initialization hour to be used in the analysis. Format is HHMM.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

INIT_HOUR_

Specify the ending initialization hour to be used in the analysis. Format is HH or HHMM.

Used by: ExtractTiles, MakePlots, StatAnalysis, TcPairs, TcStat
Family: [config]
Default: Varies

INIT_HOUR_INCREMENT

Specify a time increment for valid times for use in the analysis. This is an integer defined in seconds.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

INIT_HOUR_

Specify the method for the treatment of valid hours. Valid options are LOOP or GROUP. LOOP will consider the initialization hours individually, and GROUP will consider them as a whole.

Used by: MakePlots, StatAnalysis
Family: [config]

Default: Varies

INIT_INCLUDE

Specify which forecast initializations to include in the analysis.

Used by: TcPairs, TcStat
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, ExtractTiles, MakePlots, master_metplus.py, StatAnalysis, TcPairs, TcStat
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:

INTERP

Specify the interpolation used to create the MET .stat files. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: MakePlots, StatAnalysis
Family: [config]
Default:

INTERP_PTS

Corresponds to the interpolation in the MET .stat files. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: MakePlots, StatAnalysis
Family: [config]
Default:

INTERVAL_T

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

Used by: PB2NC
Family: [config]
Default: Varies

4.5.10 J

JOB_ARGS
Specify stat_analysis job arguments to run. The job agruments that are to be run with the cooresponding JOB_NAME. If using -dump_row, use -dump_row [dump_row_filename]. If using -out_stat, -out_stat [out_stat_filename]. For more information on these job agruments, please see the MET Users Guide.

Used by: StatAnalysis
Family: [config]
Default:

JOB_NAME
Specify stat_analysis job name to run. Valid options are filter, summary, aggregate, aggregate_stat, go_index, and ramp. For more information on these job names and what they do, please see the MET Users Guide.

Used by: StatAnalysis
Family: [config]
Default:

4.5.11 K

4.5.12 L

LAT_ADJ
Specify a latitude adjustment, in degrees to be used in the analysis. In the ExtractTiles wrapper, this corresponds to the 2m portion of the 2n x 2m subregion tile.

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

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

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

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

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

Used by: CyclonePlotter, TCMPRPlotter, feature_util.py, TcStat
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: MakePlots, StatAnalysis
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: EnsembleStat, GridStat, Mode, MTD, PB2NC, PcpCombine, PointStat, RegridDataPlane, SeriesByLead
Family: [config]
Default: Varies

LEAD_SEQ_*n*
Required when SERIES_BY_LEAD_GROUP_FCSTS=True. Not necessary otherwise. Specify the sequence of forecast lead times to include in the analysis. Comma separated list format, e.g.:
0, 6, 12. The n corresponds to the bin in which the user wishes to aggregate series by lead results.

Used by: SeriesByLead
Family: [config]
Default: Varies

LEAD_SEQ_*n*_LABEL
Required when SERIES_BY_LEAD_GROUP_FCSTS=True. Specify the label of the corresponding bin of series by lead results.

Used by: SeriesByLead
Family: [config]
Default:

LEGEND

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

LINE_TYPE
Specify the MET STAT line types to be considered. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: TCMPRPlotter
Family: [config]
Default:

LOG_DIR

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

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

LOG_LEVEL
Specify the level of logging.

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

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

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

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

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

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

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

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

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

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

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

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

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

LON_ADJ
Specify a longitude adjustment, in degrees to be used in the analysis. In the ExtractTiles wrapper, this corresponds to the 2n portion of the 2n x 2m subregion tile.

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

LOOP_BY
Control whether the analysis is processed across valid or initialization times.

Used by: command_builder.py, compare_gridded_wrapper.py, EnsembleStat, GridStat, MakePlots, master_metplus.py, Mode, StatAnalysis
Family: [config]
Default: true

LOOP_ORDER
Control the looping order 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: MakePlots, master_metplus.py, PB2NC, PointStat, StatAnalysis
Family: [config]
Default: Varies

4.5.13 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: CyclonePlotter, ExtractTiles, master_metplus.py, met_util.py, PB2NC, PointStat, SeriesByInit, SeriesByLead, TCMPRPlotter, TcPairs, 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: TCMPRPlotter
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, CyclonePlotter, EnsembleStat, ExtractTiles, feature_util.py, GridStat, Mode, PB2NC, PcpCombine, PointStat, regrid_data_plane_wrapper.py, SeriesByInit, SeriesByLead, StatAnalysis, TCMPRPlotter, TcPairs, TcStat, wavelet_stat_wrapper.py
Family: [dir]
Default: Varies

MISSING_VAL

Specify the missing value code.

Used by: TcPairs
Family: [config]
Default: Varies

MISSING_VAL_TO_REPLACE

Specify the missing value code to replace.

Used by: TcPairs
Family: [config]
Default: Varies

MODEL

Specify the model name. This is the model name listed in the MET .stat files.

Used by: compare_gridded_wrapper.py, EnsembleStat, StatAnalysis, TcPairs
Family: [config]
Default: Varies

MODEL1_NAME

Define the model name for the first model to be used in the analysis. This is the model name listed in the MET .stat files.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: MakePlots, StatAnalysis
Family: [config]
Default: Varies

MODEL1_NAME_ON_PLOT
Define the name he first model will be listed as on the plots. 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_ON_PLOT
MODEL2_NAME_ON_PLOT
.
.
.
MODELN_NAME_ON_PLOT

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

MODEL1_OBS_NAME
Define the observation name that was used to compare the first model to be. This is the observation name listed in the MET .stat files. There can be N number of observation names defined in configuration files, simply increment the "MODEL1_" string to match the total number of models being used, e.g.:
MODEL1_OBS_NAME
MODEL2_OBS_NAME
.
.
.
MODELN_OBS_NAME

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

MODEL1_STAT_DIR
Define the stat file directrory for the first model to be used in the analysis. There can be N number of model directories defined in configuration files, simply increment the "MODEL1_" string to match the total number of models being used, e.g.:
MODEL1_DIR
MODEL2_DIR

.

.

.

MODELN_DIR

Used by: StatAnalysis
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

[deprecated] MODEL_NAME
Please use MODEL instead.

Used by: PointStat
Family: [config]
Default: Varies

[deprecated] MODE_CONFIG
Please use MODE_CONFIG_FILE instead. Path to mode configuration file.

Used by: Mode
Family: [config]
Default: Varies

MODE_CONFIG_FILE
Path to mode configuration file.

Used by: Mode
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 FCST_MODE_CONV_RADIUS and OBS_MODE_CONV_RADIUS to the same value.

Used by: Mode
Family: [config]
Default:

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 FCST_MODE_CONV_THRESH and OBS_MODE_CONV_THRESH to the same value.

Used by: Mode
Family: [config]
Default:

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

Used by: Mode
Family: [config]
Default:5

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

Used by: Mode
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
Family: [config]
Default: THRESH

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

Used by: Mode
Family: [config]
Default: >0.45

MODE_MERGE_CONFIG_FILE
Path to mode merge config file.

Used by: Mode
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
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
Family: [config]
Default: >0.45

[deprecated]MODE_OBS_CONV_RADIUS

Please use OBS_CONV_MODE_RADIUS instead. Comma separated list of convolution radius values used by mode for observation fields.

Used by: Mode
Family: [config]
Default:5

[deprecated]MODE_OBS_CONV_THRESH

Please use OBS_MODE_CONV_THRESH instead. Comma separated list of convolution threshold values used by mode for observation fields.

Used by: Mode
Family: [config]
Default:5

[deprecated] MODE_OBS_MERGE_FLAG

Please use OBS_MODE_MERGE_FLAG instead. Sets the merge_flag value in the mode config file for observation fields. Valid values are NONE, THRESH, ENGINE, and BOTH.

Used by: Mode
Family: [config]
Default: THRESH

[deprecated]MODE_OBS_MERGE_THRESH

Please use OBS_MODE_MERGE_THRESH_INSTEAD. Comma separated list of merge threshold values used by mode for observation fields.

Used by: Mode
Family: [config]
Default: >0.45

[deprecated]MODE_OUT_DIR
Please use MODE_OUTPUT_DIR instead. Ouptut directory to write mode files.

Used by: Mode
Family: [dir]
Default: Varies

MODE_OUTPUT_DIR
Ouptut directory to write mode files.

Used by: Mode
Family: [dir]
Default: Varies

MODE_OUTPUT_TEMPLATE
Sets the subdirectories below MODE_OUTPUT_DIR using a template to allow run time information. If LOOP_BY = VALID, default value is valid time YYYYMMDDHHMM/mode. If LOOP_BY = INIT, default value is init time YYYYMMDDHHMM/mode.

Used by: Mode
Family: [filename_templates]
Default: Varies

MODE_VERIFICATION_MASK_TEMPLATE
Template used to specify the verification mask filename for the MET tool mode.

Used by: Mode
Family: [filename_templates]
Default: Varies

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

Used by: Mode
Family: [config]
Default: False

[deprecated]MTD_CONFIG
Please use MTD_CONFIG_FILE instead. Path to mode-TD configuration file.

Used by: MTD
Family: [config]
Default: Varies

MTD_CONFIG_FILE
Path to mode-TD configuration file.

Used by: MTD
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 FCST_MTD_CONV_RADIUS and OBS_MTD_CONV_RADIUS to the same value.

Used by: MTD
Family: [config]
Default:

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 FCST_MTD_CONV_THRESH and OBS_MTD_CONV_THRESH to the same value.

Used by: MTD
Family: [config]
Default:

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_MIN_VOLUME
Sets min_volume in the MET Mode-TD config file. Refer to the MET User's Guide for more information.

Used by: MTD
Family: [config]
Default:

MTD_SINGLE_RUN
Set to True to only process one data set (forecast or observation) in Mode-TD. If True, must set MTD_SINGLE_RUN_SRC to either 'FCST' or 'OBS'.

Used by: MTD
Family: [config]
Default:

MTD_SINGLE_RUN_SRC
Used only if MTD_SINGLE_RUN is set to True. Valid options are 'FCST' or 'OBS'.

Used by: MTD
Family: [config]
Default:

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

[deprecated] MTD_OUT_DIR
Please use MTD_OUTPUT_DIR.

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

MTD_OUTPUT_DIR
Ouptut directory to write mode-TD files.

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

MTD_OUTPUT_TEMPLATE
Sets the subdirectories below MTD_OUTPUT_DIR using a template to allow run time information. If LOOP_BY = VALID, default value is valid time YYYYMMDDHHMM/mtd. If LOOP_BY = INIT, default value is init time YYYYMMDDHHMM/mtd.

Used by: MTD
Family: [filename_templates]
Default: Varies

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

Used by: mtd_wrapper.py
Family: [config]
Default: FCST

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

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

4.5.14 N

NCAP2_EXE
Path to the “ncap2” executable.

Used by: PB2NC, PointStat, SeriesByLead
Family: [exe]
Default: /path/to

NCDUMP_EXE
Path to the “ncdump” executable.

Used by: met_util.py, PB2NC, PointStat, SeriesByLead
Family: [exe]
Default: /path/to

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

Used by: PB2NC
Family: [filename_templates]
Default: Varies

NLAT
The number of latitude points, set to a whole number. This defines the number of latitude points to incorporate into the subregion (density).

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

NLON
The number of longitude points, set to a whole number. This defines the number of longitude points to incorporate into the subregion (density).

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: TCMPRPlotter
Family: [config]
Default: no

NO_LOG
Set the “NO_LOG” flag for the TC Matched Pairs plotting utility.

Acceptable values: yes/no

Used by: TCMPRPlotter
Family: [config]

Default: no

4.5.15 O

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

Used by: PcpCombine
Family: [config]
Default: Varies

[deprecated] OBS_BUFR_VAR_LIST
Please use PB2NC_OBS_BUFR_VAR_LIST instead. 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
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: PcpCombine
Family: [config]
Default: Varies

OBS_ENSEMBLE_STAT_GRID_INPUT_DATATYPE
Specify the data type of the input directory for grid observation files used with the MET ensemble_stat tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_DATATYPE.

Used by: EnsembleStat
Family: [config]
Default: Varies

OBS_ENSEMBLE_STAT_GRID_INPUT_DIR

Input directory for grid observation files to use with the MET tool ensemble_stat. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_DIR.

Used by: EnsembleStat
Family: [dir]
Default: Varies

OBS_ENSEMBLE_STAT_GRID_INPUT_TEMPLATE
Template used to specify grid observation input filenames for the MET tool ensemble_stat. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_TEMPLATE.

Used by: EnsembleStat
Family: [filename_templates]
Default: Varies

OBS_ENSEMBLE_STAT_POINT_INPUT_DATATYPE
Specify the data type of the input directory for point observation files used with the MET ensemble_stat tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_DATATYPE.

Used by: EnsembleStat
Family: [config]
Default: Varies

OBS_ENSEMBLE_STAT_POINT_INPUT_DIR
Input directory for point observation files to use with the MET tool ensemble_stat. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_DIR.

Used by: EnsembleStat
Family: [dir]
Default: Varies

OBS_ENSEMBLE_STAT_POINT_INPUT_TEMPLATE
Template used to specify point observation input filenames for the MET tool ensemble_stat. A similar variable exists for forecast data called FCST_ENSEMBLE_STAT_INPUT_TEMPLATE.

Used by: EnsembleStat
Family: [filename_templates]
Default: Varies

OBS_ENSEMBLE_STAT_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by EnsembleStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_ENSEMBLE_STAT_FILE_WINDOW_BEGIN is not set in the config file, the value of

OBS_FILE_WINDOW_BEGIN will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: EnsembleStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

OBS_ENSEMBLE_STAT_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by EnsembleStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_ENSEMBLE_STAT_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: EnsembleStat
Family: [config]
Default: OBS_FILE_WINDOW_END

OBS_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds.
This value will be used for all wrappers that look for an observation file unless it is overridden by a wrapper specific configuration variable. For example, if OBS_GRID_STAT_FILE_WINDOW_BEGIN is set, the GridStat wrapper will use that value. If PB2NC_FILE_WINDOW_BEGIN is not set, then the PB2NC wrapper will use OBS_FILE_WINDOW_BEGIN.
A corresponding variable exists for forecast data called FCST_FILE_WINDOW_BEGIN.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
Family: [config]
Default: Varies

OBS_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds.
This value will be used for all wrappers that look for an observation file unless it is overridden by a wrapper specific configuration variable. For example, if OBS_GRID_STAT_WINDOW_END is set, the GridStat wrapper will use that value. If PB2NC_WINDOW_END is not set, then the PB2NC wrapper will use OBS_WINDOW_END.
A corresponding variable exists for forecast data called FCST_FILE_WINDOW_END.

Used by: EnsembleStat, GridStat, Mode, MTD, PB2NC, PointStat
Family: [config]
Default: Varies

OBS_GRID_STAT_FILE_WINDOW_BEGIN

Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by GridStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_GRID_STAT_FILE_WINDOW_BEGIN is not set in the config file, the value of OBS_FILE_WINDOW_BEGIN will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: GridStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

OBS_GRID_STAT_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by GridStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_GRID_STAT_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: GridStat
Family: [config]
Default: OBS_FILE_WINDOW_END

OBS_GRID_STAT_INPUT_DATATYPE
See FCST_GRID_STAT_INPUT_DATATYPE 4.5.6.

Used by: GridStat
Family: [config]
Default: Varies

OBS_GRID_STAT_INPUT_DIR
See FCST_GRID_STAT_INPUT_DIR 4.5.6.

Used by: GridStat
Family: [dir]
Default: Varies

OBS_GRID_STAT_INPUT_TEMPLATE
See FCST_GRID_STAT_INPUT_TEMPLATE 4.5.6.

Used by: GridStat
Family: [filename_templates]
Default: Varies

OBS_GRID_STAT_PROB_THRESH
See FCST_GRID_STAT_PROB_THRESH 4.5.6.

Used by: GridStat
Family: [config]
Default: ==0.1

[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: PcpCombine
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: PcpCombine
Family: [filename_templates]
Default: Varies

[deprecated] OBS_INPUT_DIR
Please use OBS_POINT_STAT_INPUT_DIR instead. Specify the input directory for observation files.

Used by: PointStat
Family: [dir]
Default: Varies

[deprecated] OBS_INPUT_DIR_REGEX
Please use OBS_POINT_STAT_INPUT_DIR instead. Specify the regular expression to use when searching for observation file input directories.

Used by: PointStat
Family: [regex_pattern]
Default: Varies

[deprecated] OBS_INPUT_FILE_REGEX
Please use OBS_POINT_STAT_INPUT_TEMPLATE instead. Regular expression used to filter observation input files used in the analysis.

Used by: PointStat,
Family: [regex_pattern]
Default: Varies

[deprecated] OBS_INPUT_FILE_TEMPL

Please use OBS_POINT_STAT_INPUT_TEMPLATE instead. Specify the filename template to use for observation input files.

Used by: PointStat,
Family: [filename_templates]
Default: Varies

[deprecated] OBS_IS_DAILY_FILE

Please use OBS_PCP_COMBINE_IS_DAILY_FILE instead. Specify whether the forecast file is a daily file or not.

Acceptable values: true/false

Used by: PcpCombine
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 4.5.6.

Acceptable values: true/false

Used by: EnsembleStat, GridStat, Mode, MTD, PointStat
Family: [config]
Default: False

[deprecated] OBS_LEVEL

Please use OBS_PCP_COMBINE_INPUT_LEVEL instead. Specify what accumulation level should be used from the observation data for the analysis. See FCST_LEVEL for more information

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_MODE_CONV_RADIUS

See FCST_MODE_CONV_RADIUS 4.5.6.

Used by: Mode
Family: [config]
Default:

OBS_MODE_CONV_THRESH

See FCST_MODE_CONV_THRESH 4.5.6.

Used by: Mode
Family: [config]
Default:

OBS_MODE_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by Mode. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_MODE_FILE_WINDOW_BEGIN is not set in the config file, the value of OBS_FILE_WINDOW_BEGIN will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: Mode
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

OBS_MODE_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by Mode. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_MODE_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: Mode
Family: [config]
Default: OBS_FILE_WINDOW_END

OBS_MODE_MERGE_FLAG
See FCST_MODE_MERGE_FLAG 4.5.6.

Used by: Mode
Family: [config]
Default:

OBS_MODE_MERGE_THRESH
See FCST_MODE_MERGE_THRESH 4.5.6.

Used by: Mode
Family: [config]
Default:

OBS_MODE_INPUT_DATATYPE
See FCST_MODE_INPUT_DATATYPE 4.5.6.

Used by: Mode
Family: [config]

Default: Varies

OBS_MODE_INPUT_DIR
See FCST_MODE_INPUT_DIR 4.5.6.

Used by: Mode
Family: [dir]
Default: Varies

OBS_MODE_INPUT_TEMPLATE
See FCST_MODE_INPUT_TEMPLATE 4.5.6.

Used by: Mode
Family: [filename_templates]
Default: Varies

OBS_MTD_CONV_RADIUS
See FCST_MTD_CONV_RADIUS 4.5.6.

Used by: MTD
Family: [config]
Default:

OBS_MTD_CONV_THRESH
See FCST_MTD_CONV_THRESH 4.5.6.

Used by: MTD
Family: [config]
Default:

OBS_MTD_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by MTD. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_MTD_FILE_WINDOW_BEGIN is not set in the config file, the value of OBS_FILE_WINDOW_BEGIN will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: MTD
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

OBS_MTD_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by MTD. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_MTD_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END

will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: MTD
Family: [config]
Default: OBS_FILE_WINDOW_END

OBS_MTD_INPUT_DATATYPE
See FCST_MTD_INPUT_DATATYPE 4.5.6.

Used by: MTD
Family: [config]
Default: Varies

OBS_MTD_INPUT_DIR
See FCST_MTD_INPUT_DIR 4.5.6.

Used by: MTD
Family: [dir]
Default: Varies

OBS_MTD_INPUT_TEMPLATE
See FCST_MTD_INPUT_TEMPLATE 4.5.6.

Used by: MTD
Family: [filename_templates]
Default: Varies

[deprecated] OBS_NAME
No longer used. Provide a string to identify the observation dataset name.

Used by: PointStat
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: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_[N]_FIELD_NAME
See FCST_PCP_COMBINE_[N]_FIELD_NAME 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_DATA_INTERVAL
See FCST_PCP_COMBINE_DATA_INTERVAL 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_DERIVE_LOOKBACK
See FCST_PCP_COMBINE_DERIVE_LOOKBACK 4.5.6.

Used by: PcpCombine
Family: [config]
Default: ADD

OBS_PCP_COMBINE_INPUT_DATATYPE
See FCST_PCP_COMBINE_INPUT_DATA_TYPE 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_INPUT_DIR
See FCST_PCP_COMBINE_INPUT_DIR 4.5.6.

Used by: PcpCombine
Family: [dir]
Default: Varies

OBS_PCP_COMBINE_INPUT_LEVEL
See FCST_PCP_COMBINE_INPUT_LEVEL 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_INPUT_TEMPLATE
See FCST_PCP_COMBINE_INPUT_TEMPLATE 4.5.6.

Used by: PcpCombine
Family: [filename_templates]
Default: Varies

OBS_PCP_COMBINE_IS_DAILY_FILE
See FCST_PCP_COMBINE_IS_DAILY_FILE 4.5.6.

Acceptable values: true/false

Used by: PcpCombine
Family: [config]
Default: False

OBS_PCP_COMBINE_METHOD
See FCST_PCP_COMBINE_METHOD 4.5.6.

Used by: PcpCombine
Family: [config]
Default:*None*

OBS_PCP_COMBINE_MIN_FORECAST
See FCST_PCP_COMBINE_MIN_FORECAST 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_MAX_FORECAST
See FCST_PCP_COMBINE_MAX_FORECAST 4.5.6.

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_OUTPUT_DIR
See FCST_PCP_COMBINE_OUTPUT_DIR 4.5.6.

Used by: PcpCombine
Family: [dir]
Default: Varies

OBS_PCP_COMBINE_OUTPUT_TEMPLATE
See FCST_PCP_COMBINE_OUTPUT_TEMPLATE 4.5.6.

Used by: PcpCombine
Family: [filename_templates]
Default: Varies

OBS_PCP_COMBINE_RUN

See FCST_PCP_COMBINE_RUN 4.5.6.

Acceptable values: true/false

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_STAT_LIST
See FCST_PCP_COMBINE_STAT_LIST 4.5.6.

Acceptable values: sum, min, max, range, mean, stdev, vld_count

Used by: PcpCombine
Family: [config]
Default: Varies

OBS_PCP_COMBINE_TIMES_PER_FILE
See FCST_PCP_COMBINE_TIMES_PER_FILE 4.5.6.

Used by: PcpCombine
Family: [config]
Default:

OBS_POINT_STAT_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by PointStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_POINT_STAT_FILE_WINDOW_BEGIN is not set in the config file, the value of OBS_FILE_WINDOW_BEGIN will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: PointStat
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

OBS_POINT_STAT_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by PointStat. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If OBS_POINT_STAT_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: PointStat
Family: [config]
Default: OBS_FILE_WINDOW_END

OBS_POINT_STAT_INPUT_DATATYPE
See FCST_POINT_STAT_INPUT_DATATYPE 4.5.6.

Used by: PointStat
Family: [config]
Default: Varies

OBS_POINT_STAT_INPUT_DIR
See FCST_POINT_STAT_INPUT_DIR 4.5.6.

Used by: PointStat
Family: [dir]
Default: Varies

OBS_POINT_STAT_INPUT_TEMPLATE
See FCST_POINT_STAT_INPUT_TEMPLATE 4.5.6.

Used by: GriPointStat
Family: [filename_templates]
Default: Varies

OBS_POINT_STAT_WINDOW_BEGIN
Passed to the point_stat MET config file to determine the range of data within a file that should be used for processing.
Units are seconds. If the variable is not set, point_stat will use OBS_WINDOW_BEGIN.

Used by: PointStat
Family: [config]
Default: Varies

OBS_POINT_STAT_WINDOW_END
Passed to the point_stat MET config file to determine the range of data within a file that should be used for processing. Units are seconds. If the variable is not set, point_stat will use OBS_WINDOW_END.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

OBS_REGRID_DATA_PLANE_INPUT_DATATYPE
See FCST_REGRID_DATA_PLANE_INPUT_DATATYPE 4.5.6.

Used by: RegridDataPlane
Family: [config]
Default: Varies

OBS_REGRID_DATA_PLANE_INPUT_DIR
See FCST_REGRID_DATA_PLANE_INPUT_DIR 4.5.6.

Used by: RegridDataPlane
Family: [dir]
Default: Varies

OBS_REGRID_DATA_PLANE_INPUT_TEMPLATE
See FCST_REGRID_DATA_PLANE_INPUT_TEMPLATE 4.5.6.

Used by: RegridDataPlane
Family: [filename_templates]
Default: Varies

OBS_REGRID_DATA_PLANE_OUTPUT_TEMPLATE
See FCST_REGRID_DATA_PLANE_OUTPUT_TEMPLATE 4.5.6.

Used by: RegridDataPlane
Family: [filename_templates]
Default: Varies

OBS_REGRID_DATA_PLANE_TEMPLATE
See FCST_REGRID_DATA_PLANE_TEMPLATE 4.5.6.

Used by: RegridDataPlane
Family: [filename_templates]
Default: Varies

OBS_REGRID_DATA_PLANE_OUTPUT_DIR
See FCST_REGRID_DATA_PLANE_OUTPUT_DIR 4.5.6.

Used by: RegridDataPlane
Family: [dir]
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: GridStat
Family: [config]
Default: Varies

OBS_VAR_LEVEL
Specify the values of the OBS_VAR_LEVEL column in the MET .stat file to use.

Used by: StatAnalysis
Family: [config]
Default: Varies

OBS_VAR_NAME
Specify the values of the OBS_VAR_NAME column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times

Used by: StatAnalysis
Family: [config]
Default: Varies

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

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

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

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

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

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

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

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

[deprecated] OBS_WINDOW_BEG
Please use OBS_WINDOW_BEGIN.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

OBS_WINDOW_BEGIN

Passed to the MET config file to determine the range of data within a file that should be used for processing. Units are seconds. This value will be used for all wrappers that look for an observation file unless it is overridden by a wrapper specific configuration variable. For example, if OBS_POINT_STAT_WINDOW_BEGIN is set, the PointStat wrapper will use that value. If PB2NC_WINDOW_BEGIN is not set, then the PB2NC wrapper will use OBS_WINDOW_BEGIN.
A corresponding variable exists for forecast data called FCST_WINDOW_BEGIN.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

OBS_WINDOW_END

Passed to the MET config file to determine the range of data within a file that should be used for processing. Units are seconds. This value will be used for all wrappers that look for an observation file unless it is overridden by a wrapper specific configuration variable. For example, if OBS_POINT_STAT_WINDOW_END is set, the PointStat wrapper will use that value. If PB2NC_WINDOW_END is not set, then the PB2NC wrapper will use OBS_WINDOW_END.
A corresponding variable exists for forecast data called FCST_WINDOW_END.

Used by: PB2NC, PointStat
Family: [config]
Default: Varies

OBTYPE

Provide a string to represent the type of observation data used in the analysis. This is the observation time listed in the MET .stat files and is used in setting output filename.

Used by: compare_gridded_wrapper.py, EnsembleStat, GridStat, Mode, StatAnalysis
Family: [config]
Default: Varies

[deprecated] OB_TYPE

Please use OBTYPE instead.

Used by: compare_gridded_wrapper.py, EnsembleStat, GridStat, Mode, StatAnalysis
Family: [config]
Default: Varies

OUTPUT_BASE

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

Used by: config_launcher.py, PB2NC, PointStat, TcPairs, TcStat
Family: [dir]

Default: Varies

[deprecated] OVERWRITE_NC_OUTPUT
Please use PB2NC_SKIP_IF_OUTPUT_EXISTS instead. Specify whether to overwrite the netCDF output or not when using the MET pb2nc tool.

Acceptable values: yes/no

Used by: PB2NC
Family: [config]
Default: yes

OVERWRITE_TRACK
Specify whether to overwrite the track data or not.

Acceptable values: yes/no

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

4.5.16 P

PARM_BASE
Specify the top level METplus parameter file directory.

Used by: config_launcher.py, PB2NC, PointStat, TcStat
Family: [dir]
Default: Varies

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

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_FILE_WINDOW_BEGIN
Used to control the lower bound of the window around the valid time to determine if a file should be used for processing by PB2NC. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If PB2NC_FILE_WINDOW_BEGIN is not set in the config file, the value of OBS_FILE_WINDOW_BEGIN

will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: PB2NC
Family: [config]
Default: OBS_FILE_WINDOW_BEGIN

PB2NC_FILE_WINDOW_END
Used to control the upper bound of the window around the valid time to determine if a file should be used for processing by PB2NC. See 4.3.3 subsection called 'Use Windows to Find Valid Files.' Units are seconds. If PB2NC_FILE_WINDOW_END is not set in the config file, the value of OBS_FILE_WINDOW_END will be used instead. If both file window begin and window end values are set to 0, then METplus will require an input file with an exact time match to process.

Used by: PB2NC
Family: [config]
Default: OBS_FILE_WINDOW_END

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

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_INPUT_DATATYPE
Specify the data type of the input directory for prepbufr files used with the MET pb2nc tool. Currently valid options are NETCDF, GRIB, and GEMPAK. If set to GEMPAK, data will automatically be converted to NetCDF via GempakToCF.

Used by: PB2NC
Family: [config]
Default: Varies

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

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_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
Family: [config]
Default: Varies

PB2NC_OFFSETS
A list of potential offsets (in hours) that can be found in the prepbufr input template. METplus will check if a file with a given offset exists in the order specified in this list, to be sure to put favored offset values first.

Used by: PB2NC
Family: [config]
Default: Varies

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

Used by: PB2NC
Family: [dir]
Default: Varies

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

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_SKIP_IF_OUTPUT_EXISTS
If True, do not run PB2NC if output file already exists. Set to False to overwrite files.

Used by: PB2NC
Family: [config]
Default: Varies

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

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_TIME_SUMMARY_FLAG

Specify the time summary flag item in the MET pb2nc config file. Refer to the MET User's Guide for more information.

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_TIME_SUMMARY_BEG

Specify the time summary beg item in the MET pb2nc config file. Refer to the MET User's Guide for more information.

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_TIME_SUMMARY_END

Specify the time summary end item in the MET pb2nc config file. Refer to the MET User's Guide for more information.

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_TIME_SUMMARY_VAR_NAMES

Specify the time summary obs_var list item in the MET pb2nc config file. Refer to the MET User's Guide for more information.

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_TIME_SUMMARY_TYPES

Specify the time summary type list item in the MET pb2nc config file. Refer to the MET User's Guide for more information.

Used by: PB2NC
Family: [config]
Default: Varies

PB2NC_WINDOW_BEGIN

Passed to the pb2nc MET config file to determine the range of data within a file that should be used for processing.
Units are seconds. If the variable is not set, pb2nc will use OBS_WINDOW_BEGIN.

Used by: PB2NC

Family: [config]
Default: Varies

PB2NC_WINDOW_END

Passed to the pb2nc MET config file to determine the range of data within a file that should be used for processing. Units are seconds. If the variable is not set, pb2nc will use OBS_WINDOW_END.

Used by: PB2NC
Family: [config]
Default: Varies

[deprecated] PCP_COMBINE_METHOD

SPlease use [FCST/OBS]_PCP_COMBINE_METHOD instead.

Used by: PcpCombine
Family: [config]
Default: ADD

PCP_COMBINE_SKIP_IF_OUTPUT_EXISTS

If True, do not run pcp_combine if output file already exists. Set to False to overwrite files.

Used by: PcpCombine
Family: [config]
Default: False

PLOTTING_OUTPUT_DIR

Specify the output directory where plots will be saved. This is the base directory where the output from running make_plots_wrapper will be put.

Used by: MakePlots
Family: [dir]
Default: Varies

PLOTTING_SCRIPTS_DIR

Specify the directory where the plotting scripts are located. It is recommended to set this to {METPLUS_BASE}/ush/plotting_scripts.

Used by: MakePlots
Family: [dir]
Default: Varies

PLOT_CONFIG_OPTS

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

Used by: TCMPRPlotter
Family: [config]

Default: Varies

PLOT_STATS_LIST
This is a list of the statistics to calculate and create plots for. Specify the list in a comma-separated list, e.g.:
acc, bias, rmse

The list of valid options varies depending on line type that was used during the filtering of stat_analysis_wrapper. For SL1L2, VL1L2 valid options are bias, rms, msess, rsd, rmse_md, rmse_pv, pcor, fbar, and fbar_obar. For SAL1L2, VAL1L2, the valid options is acc. For VCNT, bias, fbar, fbar_obar, speed_err, dir_err, rmsve, vdiff_speed, vdiff_dir, rsd, fbar_speed, fbar_dir, fbar_obar_speed, and fbar_obar_dir.

Used by: MakePlots
Family: [config]
Default: Varies

PLOT_TIME
In StatAnalysis, this specifies the way to treat the date information, where valid options are valid and init.

Used by: StatAnalysis
Family: [config]
Default: Varies

PLOT_TYPES
Specify what plot types are desired for the TC Matched Pairs plotting tool. By default, a boxplot is generated if this is undefined in the configuration file. If other plots are requested and a boxplot is also desired, you must explicitly list *boxplot* in your list of plot types. Supported plot types: BOXPLOT, POINT, MEAN, MEDIAN, RELPERF (relative performance), RANK (time series of ranks for the first model), SCATTER, SKILL_MN (mean skill scores) and SKILL_MD (median skill scores).

Used by: TCMPRPlotter
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: PointStat
Family: [config]
Default: Varies

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

Used by: PointStat
Family: [config]

Default: Varies

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

Used by: PointStat
Family: [config]
Default: Varies

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

Used by: PointStat
Family: [dir]
Default: Varies

POINT_STAT_OUTPUT_TEMPLATE
Sets the subdirectories below POINT_STAT_OUTPUT_DIR using a template to allow run time information. If LOOP_BY = VALID, default value is valid time YYYYMMDDHHMM/point_stat. If LOOP_BY = INIT, default value is init time YYYYMMDDHHMM/point_stat.

Used by: PointStat
Family: [filename_templates]
Default: Varies

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

Used by: PointStat
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: PointStat
Family: [config]
Default: Varies

POINT_STAT_VERIFICATION_MASK_TEMPLATE
Template used to specify the verification mask filename for the MET tool point_stat.

Used by: PointStat
Family: [filename_templates]
Default: Varies

PREFIX

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

[deprecated] PREPBUFR_DATA_DIR

Please use PB2NC_INPUT_DIR instead. Specify the directory where the PREPBUFR data are located for the MET pb2nc tool.

Used by: PB2NC
Family: [dir]
Default: Varies

[deprecated] PREPBUFR_DIR_REGEX

Regular expression to use when searching for PREPBUFR data.

Used by: PB2NC
Family: [regex_pattern]
Default: Varies

[deprecated] PREPBUFR_FILE_REGEX

Regular expression to use when searching for PREPBUFR files.

Used by: PB2NC
Family: [regex_pattern]
Default: Varies

[deprecated] PREPBUFR_MODEL_DIR_NAME

Please put the value previously used here in the PB2NC_INPUT_DIR path. Specify the name of the model being used with the MET pb2nc tool.

Used by: PB2NC
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

[deprecated] 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, PointStat, TcStat
Family: [dir]
Default: Varies

4.5.17 Q

4.5.18 R

REFERENCE_TMPL
The filename template describing the observation/reference data.

Used by: TcPairs
Family: [filename_templates]
Default: Varies

REGION
Specify the values of the VX_MASK column in the MET .stat file to use. This is optional in the METplus configuration file for running with LOOP_ORDER = times

Used by: StatAnalysis
Family: [config]
Default:

REGION_LIS
A list of the regions of interest. This is the list of regions for plotting verification.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

REGRID_DATA_PLANE_METHOD
Sets the method used by regrid_data_plane. See MET User's Guide for more information.

Used by: RegridDataPlane
Family: [config]
Default:

REGRID_DATA_PLANE_SKIP_IF_OUTPUT_EXISTS
If True, do not run regrid_data_plane if output file already exists. Set to False to overwrite files.

Used by: RegridDataPlane
Family: [config]
Default: False

REGRID_DATA_PLANE_WIDTH
Sets the width used by regrid_data_plane. See MET User's Guide for more information.

Used by: RegridDataPlane
Family: [config]
Default: 1

REGRID_DATA_PLANE_VERIF_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

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

Used by: MakePlots, PointStat
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, SeriesByInit, SeriesByLead
Family: [config]
Default: yes

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

Used by: PB2NC, PointStat, SeriesByLead
Family: [exe]
Default: /path/to

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

4.5.19 S

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

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

Used by: TCMPRPlotter
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 TCMPRPlotter). 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: TCMPRPlotter
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: SeriesByInit
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: SeriesByLead
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. Refer to Chapter 20 of the MET User's Guide for the syntax to use for performing filtering via the MET tc_stat tool.

Used by: SeriesByLead, SeriesByInit
Family: [config]
Default: Varies

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

Used by: SeriesByLead
Family: [config]
Default: Varies

SERIES_BY_LEAD_GROUP_FCSTS
Set to *True* to aggregate the series by lead results into bins of time.

Used by: SeriesByLead
Family: [config]
Default: Varies

SERIES_CI
Corresponds to the optional -series_ci flag in plot_TCMPR.R (which is wrapped by TCMPRPlotter). 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: TCMPRPlotter
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: SeriesByInit
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: SeriesByInit
Family: [dir]
Default: Varies

[deprecated] SERIES_LEAD_FILTERED_OUT_DIR
Please use SERIES_BY_LEAD_FILTERED_OUTPUT_DIR. Specify the directory where filtered files will be written from the MET series_analysis tool when processing by lead time.

Used by: SeriesByLead
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: SeriesByLead
Family: [dir]
Default: Varies

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

Used by: TCMPRPlotter
Family: [config]
Default: Varies

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

Used by: PB2NC, PointStat
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, PointStat
Family: [config]
Default: Varies

STAT_ANALYSIS_CONFIG
Specify the absolute path for the configuration file used with the MET stat_analysis tool. It is recommended to set this to {PARM_BASE}/use_cases/plotting/met_config/STATAnalysisConfig.

Used by: StatAnalysis
Family: [config]

Default: Varies

STAT_ANALYSIS_DUMP_ROW_TMPL
Specify the template to use for the stat_analysis dump_row file. A user customized template to use for the dump_row file. If left blank and a dump_row file is requested, a default version will be used. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: StatAnalysis
Family: [filename_templates]
Default:

STAT_ANALYSIS_LOOKIN_DIR
Specify the input directory where the MET stat_analysis tool will find input files. This is the directory that the stat_analysis wrapper will use to build the argument to -lookin for the MET stat_analysis tool. It can contain wildcards, i.e. *.

Used by: StatAnalysis
Family: [dir]
Default: Varies

STAT_ANALYSIS_OUT_STAT_TMPL
Specify the template to use for the stat_analysis out_stat file. A user customized template to use for the out_stat file. If left blank and a out_stat file is requested, a default version will be used. This is optional in the METplus configuration file for running with LOOP_ORDER = times.

Used by: StatAnalysis
Family: [filename_templates]
Default:

STAT_ANAL

This is the base directory where the output from running stat_analysis_wrapper will be put.

Used by: StatAnalysis
Family: [dir]
Default: Varies

STAT_FILES_INPUT_DIR
Specify the directory where stat files exist that plots can be generated from. This is the directory where the files from running previously running stat_analysis_wrapper are located. These are the files used as the data to create the plots. It is recommended to set this to {STAT_ANALYSIS_OUTPUT_DIR}.

Used by: MakePlots
Family: [dir]
Default: Varies

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

Used by: SeriesByInit, SeriesByLead
Family: [config]
Default: Varies

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

Used by: CyclonePlotter, met_util.py, TcPairs, TcStat
Family: [config]
Default: Varies

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

Used by: TcPairs, TcStat
Family: [config]
Default: Varies

SUBTITLE
The subtitle of the plot.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

4.5.20 T

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

Used by: TCMPRPlotter
Family: [dir]
Default: Varies

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

Used by: TCMPRPlotter
Family: [dir]

Default: Varies

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

Used by: TcPairs
Family: [config]
Default: Varies

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

Used by: TcPairs
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: TcPairs
Family: [config]
Default: no

TC_STAT_AMODEL
Specify the AMODEL for the MET tc_stat tool.

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_BASIN
Specify the BASIN for the MET tc_stat tool.

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_BMODEL
Specify the BMODEL for the MET tc_stat tool.

Used by: TcStat
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. Only specify if TC_STAT_RUN_VIA=CLI. Please refer to the MET User's Guide chapter for tc-stat for the details on performing job summaries and job filters.

Used by: TcStat
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: TcStat
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: TcStat
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: TcStat
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: TcStat
Family: [config]
Default: Varies

TC_STAT_CYCLONE

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

Used by: TcStat
Family: [config]

Default: Varies

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

Used by: TcStat
Family: [config]
Default: Varies

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

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

Used by: TcStat
Family: [config]
Default: Varies

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

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_INIT_EXCLUDE
Specify the initialization times to exclude when using the MET tc_stat tool, via a comma separated list e.g.:

20141220_18, 20141221_00

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_INIT_HOUR
The beginning hour (HH) of the initialization time of interest.

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_INIT_INCLUDE
Specify the initialization times to include when using the MET tc_stat tool, via a comma separated list e.g.:

20141220_00, 20141220_06, 20141220_12

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

Used by: TcStat
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: TcStat
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: TcStat
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: TcStat
Family: [config]
Default: Varies

TC_STAT_INPUT_DIR
Specify the input directory where the MET tc_stat tool will look for files.

Used by: TcStat
Family: [dir]
Default: Varies

TC_STAT_JOBS_LIST
Specify expressions for the MET tc_stat tool to execute.

Used by: TcStat
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: TcStat
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: TcStat
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: TcStat
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: TcStat
Family: [config]
Default: Varies

TC_STAT_LEAD_REQ
Specify the LEAD_REQ when using the MET tc_stat tool.

Used by: TcStat
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: TcStat
Family: [config]
Default: false

TC_STAT_OUTPUT_DIR
Specify the output directory where the MET tc_stat tool will write files.

Used by: TcStat
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: TcStat
Family: [config]
Default: CONFIG

TC_STAT_STORM_ID
Set the STORM_ID(s) of interest with the MET tc_stat tool.

Used by: TcStat
Family: [config]
Default: Varies

TC_STAT_STORM_NAME
Set the STORM_NAME for use with the MET tc_stat tool.

Used by: TcStat
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: TcStat
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: TcStat
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: TcStat
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: TcStat
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: TcStat
Family: [config]
Default: Varies

TC_STAT_VALID_INCLUDE

Specify a comma separated list of valid times to include in the stratification with when using the MET tc_stat tool.

Acceptable formats: YYYYMMDD_HH, YYYYMMDD_HHmmss

Used by: TcStat
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: TcStat
Family: [config]
Default: Varies

TC_STAT_WATER_ONLY

Specify whether to exclude points where the distance to land is <= 0. If set to TRUE, once land is encountered the remainder of the forecast track is not used for the verification, even if the track moves back over water.

Acceptable values: true/false

Used by: TcStat
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, PointStat
Family:
Default:

[deprecated] TIME_SUMMARY_BEG

Please use PB2NC_TIME_SUMMARY_BEG instead. Specify the starting time of the summary when using the MET pb2nc tool.

Acceptable formats: HHMMSS

Used by: PB2NC
Family: [config]
Default: 000000

[deprecated] TIME_SUMMARY_END

Please use PB2NC_TIME_SUMMARY_END instead. Specify the ending time of the summary when using the MET pb2nc tool.

Acceptable formats: HHMMSS

Used by: PB2NC
Family: [config]
Default: 235959

[deprecated] TIME_SUMMARY_FLAG

Please use PB2NC_TIME_SUMMARY_FLAG instead. Specify whether to receive a time summary from the MET pb2nc tool or not.

Acceptable values: True/False

Used by: PB2NC
Family: [config]
Default: False

[deprecated] TIME_SUMMARY_TYPES

Please use PB2NC_TIME_SUMMARY_TYPES instead. Specify a comma separated list of time summary types to receive from the MET pb2nc tool.

Used by: PB2NC
Family: [config]
Default: Varies

[deprecated]TIME_SUMMARY_VAR_NAMES

Please use PB2NC_TIME_SUMMARY_VAR_NAMES instead. Specify a comma separated list of time summary variable names to receive from the MET pb2nc tool.

Used by: PB2NC
Family: [config]
Default: Varies

TITLE

Specify a title string for the TC Matched Pairs plotting tool.

Used by: TCMPRPlotter
Family: [config]

Default: Varies

TMP_DIR

Specify the path to a temporary directory where the user has write permissions.

Used by: ExtractTiles, PB2NC, PointStat, SeriesByInit, SeriesByLead, TcStat
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: TcPairs
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: TcPairs
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: TcPairs
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: TcPairs
Family: [dir]
Default: Varies

TRACK_TYPE

Specify the track type to filter by when using the MET tc_pairs tool.

Used by: TcPairs
Family: [config]

Default: Varies

TR_EXE
Specify the path to the Linux "tr" executable.

Used by: PB2NC, PointStat
Family: [exe]
Default: /path/to

4.5.21 U

4.5.22 V

VALID_BEG
Specify a begin time for valid times for use in the analysis. This is the starting date in the format set in the TIME_FMT. It is named accordingly to the value set for LOOP_BY. However, in StatAnalysis, it is named accordingly to the value set for PLOT_TIME.

Acceptable formats: YYYYMM[DD[_HH]]

Used by: command_builder.py, MakePlots, master_metplus.py, StatAnalysis, TcPairs, TcStat
Family: [config]
Default: Varies

VALID_END
Specify an end time for valid times for use in the analysis. This is the ending date in the format set in the TIME_FMT. It is named accordingly to the value set for LOOP_BY.

Acceptable formats: controlled via VALID_TIME_FMT

Used by: command_builder.py, MakePlots, master_metplus.py, StatAnalysis, TcPairs, TcStat
Family: [config]
Default: Varies

VALID_HOUR_BEG
Specify a beginning hour for valid times for use in the analysis.

Acceptable formats: HHMM

Used by: MakePlots, StatAnalysis
Family: [config]

Default: Varies

VALID_HOUR_END

Specify an end hour for valid times for use in the analysis.

Acceptable formats: HHMM

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

VALID_HOUR_INCREMENT

Specify a time increment for valid times for use in the analysis.

Acceptable formats: seconds

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

VALID_HOUR_METHOD

Specify the method for the treatment of valid hours. Valid options are LOOP or GROUP. LOOP will consider the valid hours individually, and GROUP will consider them valid hours as a whole.

Acceptable formats: LOOP or GROUP

Used by: MakePlots, StatAnalysis
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, MakePlots, master_metplus.py, StatAnalysis, TcStat
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, StatAnalysis
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, SeriesByInit, SeriesByLead
Family: [config]
Default: Varies

VARn_FOURIER_DECOMP

Specify if Fourier decomposition is to be considered (True) or not (False). If this is set to True, data stratification will be done for the Fourier decomposition of FCS_VARn_NAME. This should have been previously run in grid_stat_wrapper. The default value is set to False.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: False

VARn_WAVE_NUM_LIST

Specify a comma separated list of wave numbers pairings of the Fourier decomposition.

Used by: MakePlots, StatAnalysis
Family: [config]
Default:

[deprecated] VERIFICATION_GRID

Please use REGRID_DATA_PLANE_VERIF_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. Valid options are grid2grid, grid2obs, and precip.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

VERIF_GRID

Specify a string describing the grid the verification was performed on. This is the name of the grid upon which the verification was done on, ex. G002.

Used by: MakePlots
Family: [config]

Default:

VERIF_TYPI

Specify a string describing the type of verification being performed. For VERIF_CASE = grid2grid, valid options are anom, pres, and sfc. For VERIF_CASE = grid2obs, valid options are conus_sfc and upper_air. For VERIF_CASE = precip, any accumulation amount is valid, ex. A24.

Used by: MakePlots, StatAnalysis
Family: [config]
Default: Varies

[deprecated] VERTICAL_LOCATION
Specify the vertical location desired when using the MET pb2nc tool.

Used by: PB2NC
Family: [config]
Default: Varies

4.5.23 W

WGRIB2
Specify the path to the “wgrib2” executable.

Used by: feature_util.py, PB2NC, PointStat
Family: [exe]
Default: /path/to

4.5.24 X

XLAB
Specify the x-axis label when using the TC Matched Pairs plotting tool.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

XLIM
Specify the x-axis limit when using the TC Matched Pairs plotting tool.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

4.5.25 Y

YLAB
Specify the y-axis label when using the TC Matched Pairs plotting tool.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

YLIM
Specify the y-axis limit when using the TC Matched Pairs plotting tool.

Used by: TCMPRPlotter
Family: [config]
Default: Varies

4.5.26 Z

4.6 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