

METplus Installation and Configuration

Getting the source

<https://github.com/NCAR/METplus>

*No GitHub account is necessary (repository is public)

Clone the repository (get the source) three ways:

1. Command line

- Best if planning on contributing source

2a. Your browser - downloading the tarball

- Source and docs only

2b. Your browser - release link

- Source, docs, and sample data

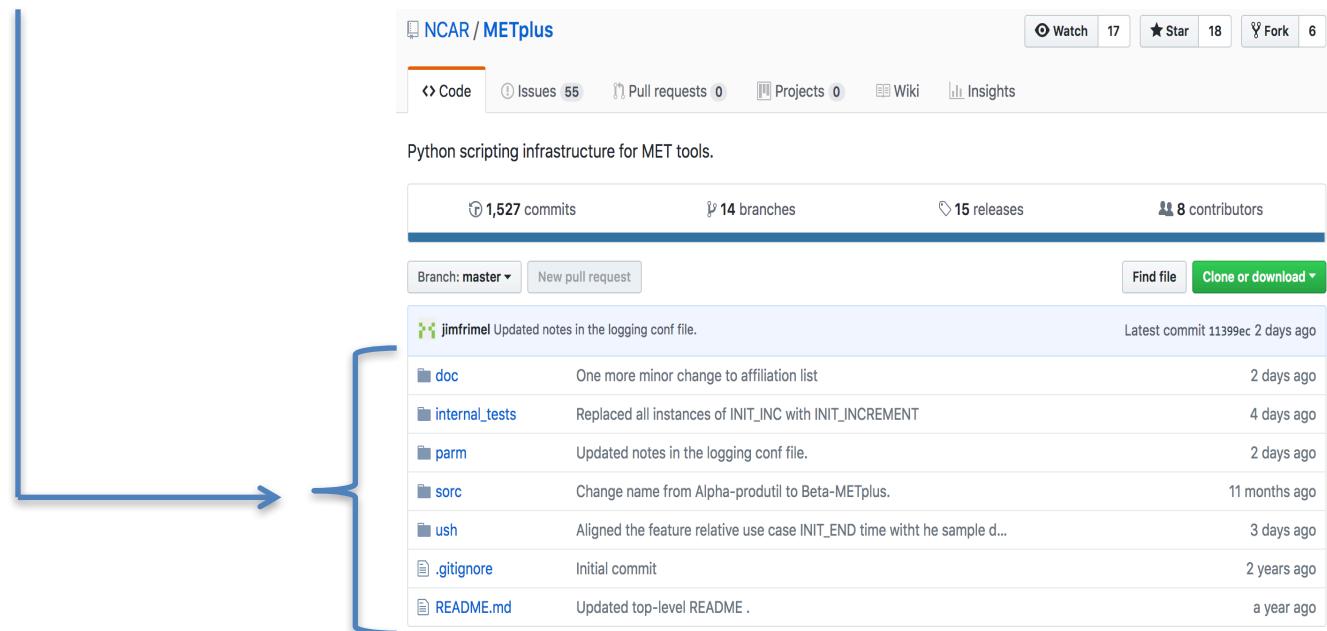
FIRST: Create the directory where you want to store the repository

Cloning from the command line

1. Change to the directory you created
2. Run the following from your terminal:

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

The entire repository resides under the *METplus* directory



Cloning from your web browser (2a)

Step 1

A screenshot of a web browser window. The address bar shows the URL <https://github.com/NCAR/METplus>. A yellow oval highlights the URL. Below the address bar, the GitHub navigation bar includes links for Features, Business, Explore, Marketplace, and Pricing. The main content area shows the repository name "NCAR / METplus" and a brief description: "Python scripting infrastructure for MET tools." Below this are statistics: 1,527 commits, 14 branches, 15 releases, and 8 contributors. At the bottom, there are buttons for "Find file" and "Clone or download".

Clone with HTTPS ⓘ

Use Git or checkout with SVN using the web URL.

<https://github.com/NCAR/METplus.git>



[Open in Desktop](#)

[Download ZIP](#)

Step 2

Step 3



Python scripting infrastructure for MET tools.

1,527 commits

14 branches

15 releases

8 contributors

Branch: master ▾

New pull request

Find file

Clone or download ▾

jimfrimel Updated notes in the logging conf file.

Latest commit 11399ec 2 days ago

doc One more minor change to affiliation list

2 days ago

internal_tests Replaced all instances of INIT_INC with INIT_INCREMENT

4 days ago

parm Updated notes in the logging conf file.

2 days ago

sorc Change name from Alpha-produtil to Beta-METplus.

11 months ago

ush Aligned the feature relative use case INIT_END time with the sample d...

3 days ago

.gitignore Initial commit

2 years ago

README.md Updated top-level README .

a year ago

Cloning from your web browser (2a)

NCAR / METplus

Code Issues 55 Pull requests 0 Projects 0 Wiki Insights

Watch 17 Star 18 Fork 6

Python scripting infrastructure for MET tools.

1,527 commits 14 branches 15 releases 8 contributors

Branch: master New pull request Find file Clone or download

File	Description	Time
.gitignore	Initial commit	2 years ago
doc	One more minor change to affiliation list	2 days ago
internal_tests	Replaced all instances of INIT_INC with INIT_INCREMENT	4 days ago
parm	Updated notes in the logging conf file.	2 days ago
sorc	Change name from Alpha-produtil to Beta-METplus.	11 months ago
ush	Aligned the feature relative use case INIT_END time with the sample d...	3 days ago
README.md	Updated top-level README .	a year ago

```
[Daniels-MacBook-Pro:METplus daniel$ unzip -qq METplus-master.zip  
[Daniels-MacBook-Pro:METplus daniel$ ls -1
```

METplus-master

METplus-master.zip

```
[Daniels-MacBook-Pro:METplus daniel$ ls -1 METplus-master  
README.md
```

doc

internal_tests

parm

sorc

ush

```
Daniels-MacBook-Pro:METplus daniel$
```

Cloning from browser (2b)

Step 1



The screenshot shows a GitHub repository page for 'NCAR / METplus'. A yellow circle highlights the URL in the browser's address bar: <https://github.com/NCAR/METplus>. Another yellow circle highlights the '15 releases' link in the top navigation bar. A black box labeled 'Step 2' is positioned over the '15 releases' link, with a blue arrow pointing down to it from the 'Step 1' label.

Python scripting infrastructure for MET tools.

1,527 commits 14 branches 15 releases 8 contributors

Branch: master ▾ New pull request Find file Clone or download ▾

File / Commit	Description	Date
jimfrimel	Updated notes in the logging conf file.	Latest commit 11399ec 2 days ago
doc	One more minor change to affiliation list	2 days ago
internal_tests	Replaced all instances of INIT_INC with INIT_INCREMENT	4 days ago
parm	Updated notes in the logging conf file.	2 days ago
sorc	Change name from Alpha-produtil to Beta-METplus.	11 months ago
ush	Aligned the feature relative use case INIT_END time with the sample d...	3 days ago
.gitignore	Initial commit	2 years ago
README.md	Updated top-level README .	a year ago

Cloning from browser (2b)

NCAR / METplus

Code Issues 55 Pull requests 0 Projects 0 Wiki Insights

Releases Tags

Latest release v2.0 · 11399ec

METplus-2.0

jimfrimel released this 2 days ago

Assets 6

- [METplus_Users_Guide.pdf](#)
- [sample_data-cyclone_track_feature.tgz](#)
- [sample_data-grid_to_grid.tgz](#)
- [sample_data-qpf.tar.gz](#)
- [Source code \(zip\)](#)
- [Source code \(tar.gz\)](#)

Note other options-
User guide and
sample data available
at the release page.

Step 3

METplus Common Installations

Theia:

/contrib/met/METplus/METplus-2.0

Tide:

/global/noscrub/Julie.Prestopnik/METplus/METplus-2.0

METplus Common Installations

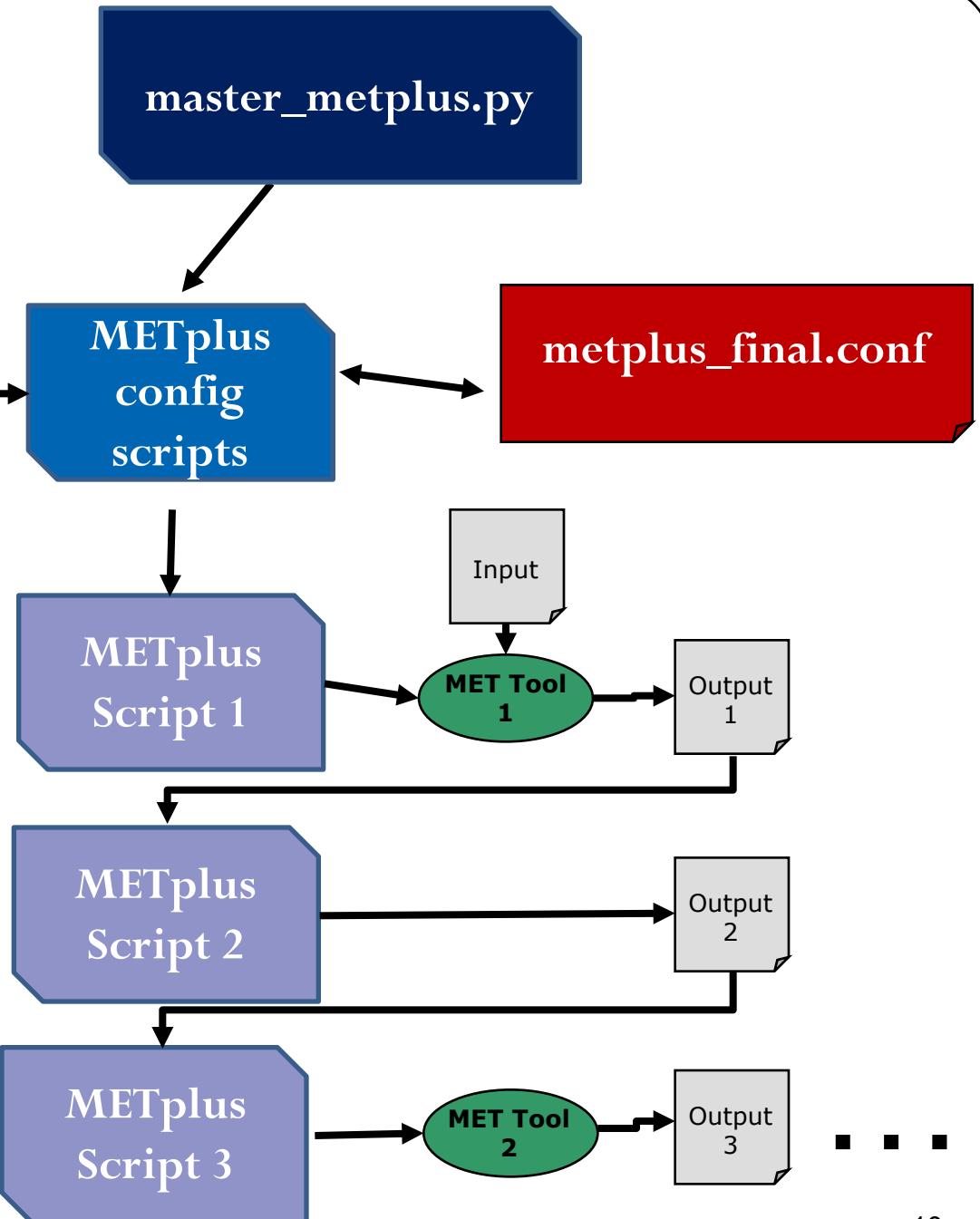
WCOSS - /u/<user>/.bashrc

```
loadmetplus='yes'
if [ ${loadmetplus} == 'yes' ];
then
    echo "Loading METplus environment"
    module use /global/noscrub/Julie.Prestopnik/modulefiles
    module load met/8.0
    module load nco
    module load grib_util
    module use /usrx/local/dev/modulefiles
    module load python/2.7.14
    export METPLUS_PATH=<path to top level METplus-2.0 directory for user>
    export MET_PATH="/global/noscrub/Julie.Prestopnik/met/8.0"
    export JLOGFILE="${METPLUS_PATH}/logs/metplus_jlogfile"
    export PYTHONPATH="${METPLUS_PATH}/ush:${METPLUS_PATH}/parm"
    export PATH="${PATH}:${METPLUS_PATH}/ush:."
fi
```

Sample Profile -
Tide

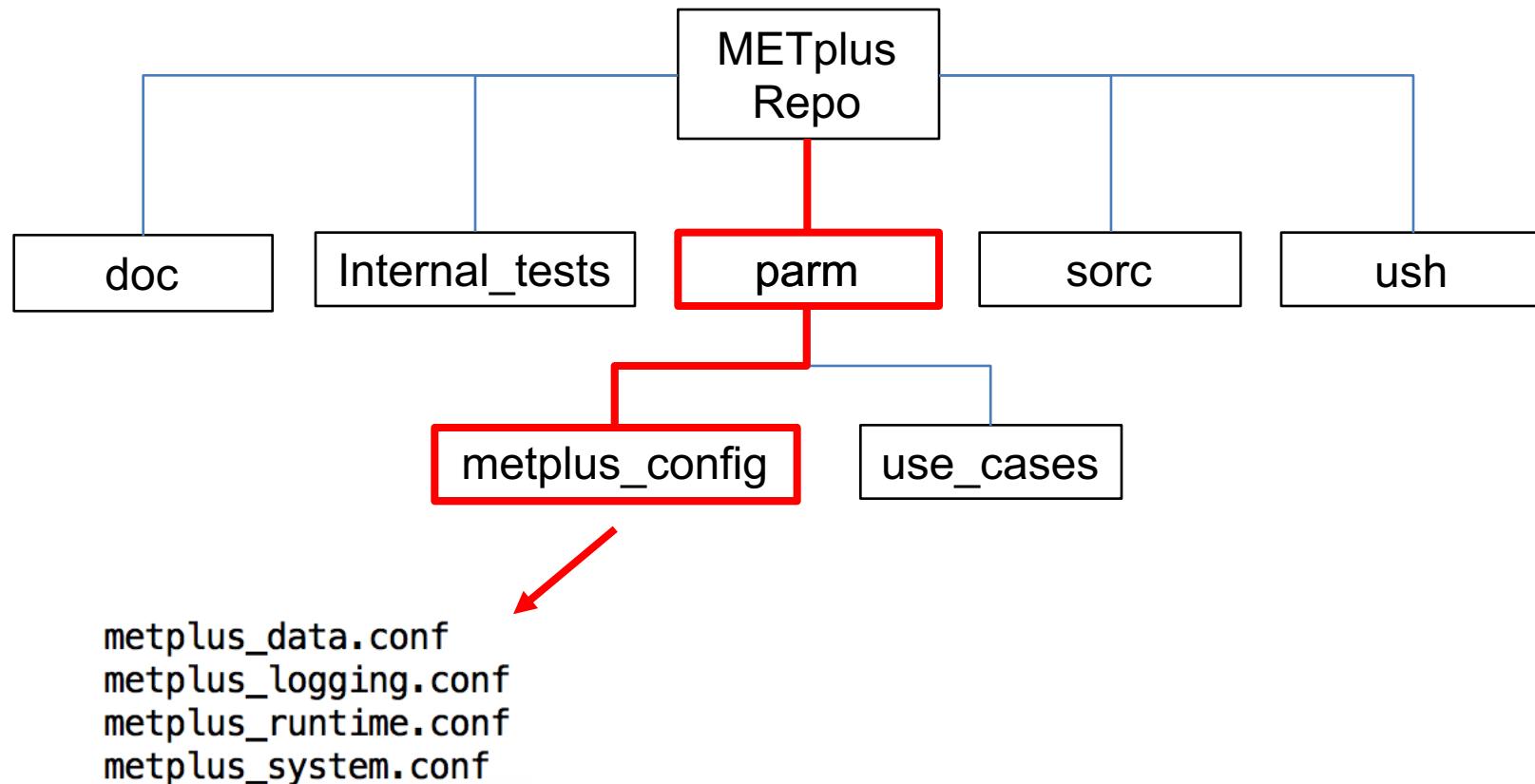
Configuration files

```
63 MET_BUILD_BASE = /path/to
64 MET_BASE = {MET_BUILD_BASE}/share/met
65
66 ## Output directories
67 LOG_DIR = {OUTPUT_BASE}/logs
68 TMP_DIR #
69 # [dir]
70 # [dir]
71 # EX # Input data
72 # A1
73 # [exe] # This is the
74 # PROJ_DIR = /A1
75 WGRIB #
76 CUT_E #
77 TR_EX # FILENAME
78 RM_EX #
79 NCAP2 # filename test
80 CONVE #
81 NCDDUM #
82 EGRER # GFS_FCST_FI
83 # GFS_FCST_NC
84 # GFS_ANLY_FI
85 # GFS_ANLY_NC
86
87 [config]
88 EXP=METplus ;; Experiment name, used for finding installation location
89 # Options are processes, times
90 LOOP_METHOD = processes
91
92 # Processes to run in master script (master_met_plus.py)
93 PROCESS_LIST = Usage
94
95 # NOTE: "TOTAL" is a REQUIRED cnt statistic used by the series analysis
96 STAT_LIST = TOTAL, FBAR, OBAR, ME, MAE, RMSE, BCMSE, E50, EIQR, MAD
97
98 # Init time
99 INIT_TIME_FMT = %Y%m%d
100 INIT_BEG = 20141214
101 INIT_END = 20141216
102 INIT_INC = 21600
103 #21600 sec (6hours) The increment in seconds in integer format
104
105 # LOGGING
106 LOG_LEVEL = DEBUG ;; Levels: DEBUG, INFO, WARNING, ERROR, CRITICAL
107 LOG_FILENAME = {LOG_DIR}/master_met_plus.log ;; NOTE: current YYYYMMDD
108
109
```



From .conf
to running MET

Configuration files



Required config files – ‘set and forget’. These are ***automatically*** loaded when running METplus.

Configuration files

Anatomy of a configuration file (the “families”):

[config] → General options

[dir] → Data directories/paths

[exe] → Executable paths

[filename_templates] → Format templates for filenames

[regex_pattern] → Regular expression patterns

The family name (item in []) MUST precede configuration options of that type in the configuration file.

Configuration files: example

```
[config]
INIT_BEG=2018100100

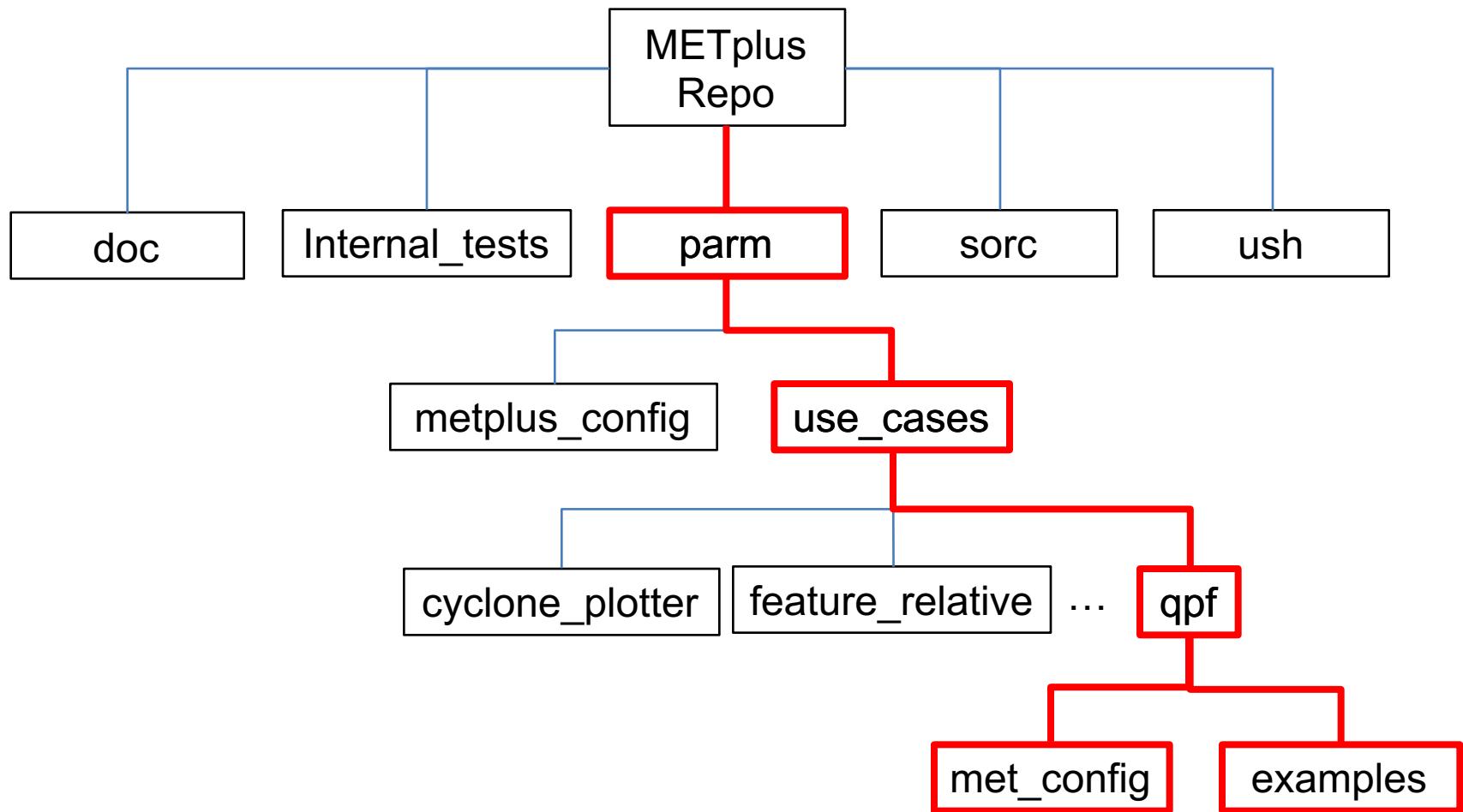
[dir]
CONFIG_DIR={PARM_BASE}/use_cases/qpf/met_config

[filename_templates]
OBS_PCP_COMBINE_INPUT_TEMPLATE = {valid?fmt=%Y%m%d}/ST4.{valid?fmt=%Y%m%d%H}.{level?fmt=%HH}h
```

Q: How do I know what “family” ([dir], [config], etc...) a config option belongs to?

A: The METplus User Guide contains a “Config Glossary” (Section 3.4) that contains every METplus configuration option with various information including which family the config option belongs to.

Use case config files



Configuration files

Major Configuration Note #1:

METplus configuration files are hierarchical; if multiple files contain the same config option, the value of that option in the last loaded configuration file will be used by METplus.

Major Configuration Note #2:

Custom user configuration files are encouraged! These DO NOT have to exist under the METplus Repository, and can be located anywhere that METplus has read access (local machine, shared disk, etc...).

Configuration – sample idea #1

Custom user conf files for test and system

1. mkdir /home/user/myconf
2. cp METplus/parm/metplus_config/metplus_system.conf
/home/user/myconf/metplus_system.conf.SYSTEM_NAME
3. Edit custom system file from step 2 for my system
4. cp METplus/parm/use_cases/qpf/examples/EXAMPLE.conf
/home/user/myconf/EXAMPLE.conf.SUFFIX
5. Edit custom use case file from step 4. Note: SUFFIX is some unique identifier to describe the test being performed, if it's not an exact replica of the canned use case delivered with METplus.
6. Run METplus using the two custom config files:

```
master_metplus.py -c  
/home/user/myconf/metplus_system.conf.SYSTEM_NAME -c  
/home/user/myconf/EXAMPLE.conf.SUFFIX
```

Configuration – sample idea #2

Separate CONF items by MET tool

USER.system.conf.gyre - fields specific to gyre/tide
and specific to my directories for output, etc.

pb2nc.conf - fields specific to pb2nc

point_stat.conf - fields specific to point_stat

shared.conf - fields shared between pb2nc and
point_stat

Configuration – your own idea!

Do what works for you!

Some considerations:

1. Your environment
2. System constraints
3. Shared resources
4. Ease of configuration

Running METplus

```
python master_metplus.py \
-c <use_case>.conf \
-c <custom1>.conf \
-c <custom2>.conf
```

REMEMBER: Order matters when providing config files-
Later configuration files over-ride earlier config files.

Config file list loaded with above command:

- metplus_data.conf (auto-load)
- metplus_system.conf (auto-load)
- metplus_runtime.conf (auto-load)
- metplus_logging.conf (auto-load)
- <use_case>.conf
- <custom1>.conf
- <custom2>.conf

Running METplus

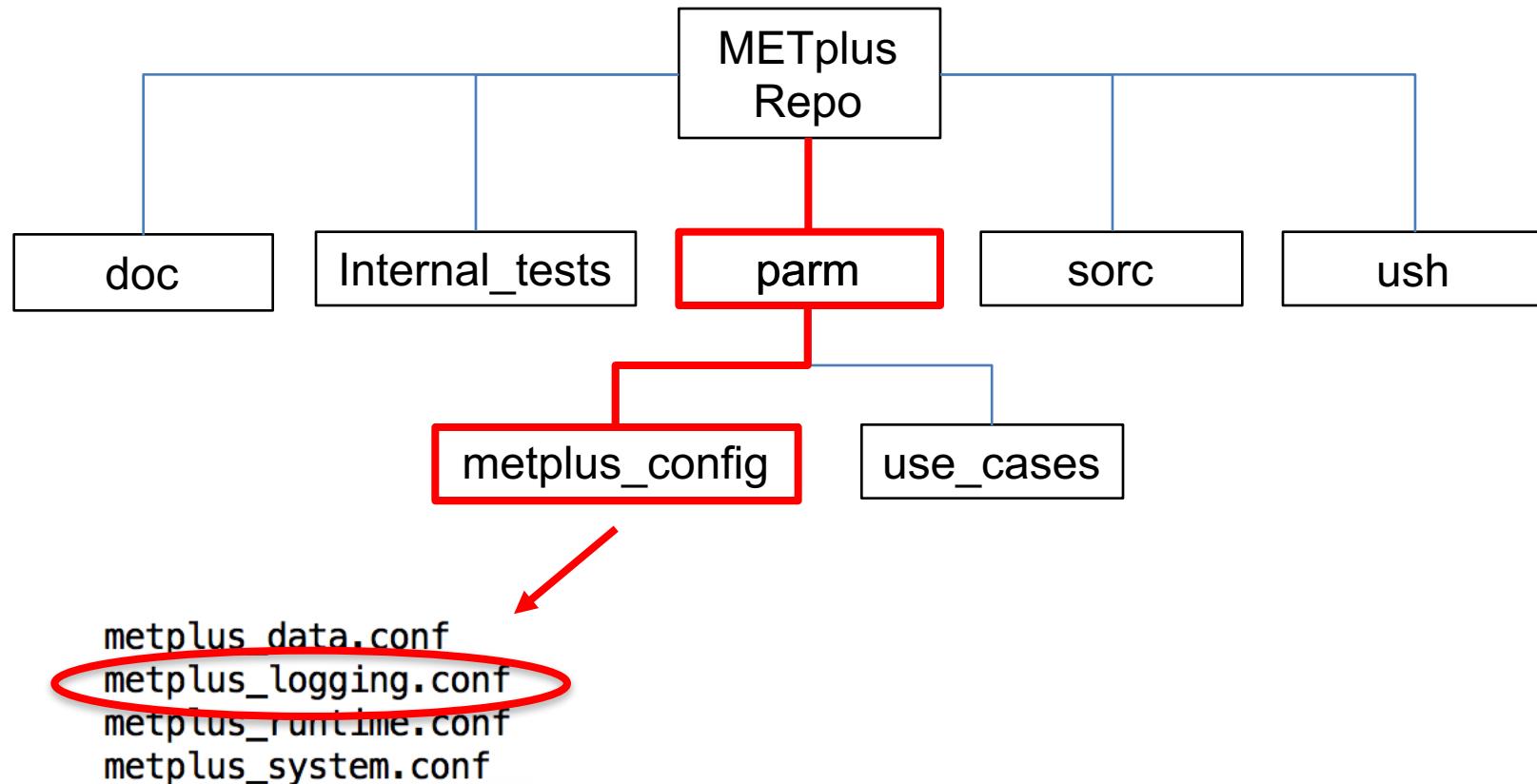
The final key-values from all the METplus config files are consolidated into `master_metplus.conf` file

- + Located in your output directory
- + Useful for troubleshooting/debugging

In Linux-speak, essentially:

```
cat metplus_data.conf metplus_system.conf metplus_runtime.conf  
metplus_logging.conf <use_case>.conf <custom1>.conf  
<custom2>.conf > /path/to/output/directory/metplus_final.conf
```

Set up logging – WORTH IT



Set up logging – WORTH IT

LOG_DIR: Control where the logs are written

LOG_METPLUS: Output from master_metplus.py is written to this file

LOG_TIMESTAMP_TEMPLATE: Time format string to append to LOG_METPLUS file

LOG_MET_OUTPUT_TO_METPLUS: If yes, MET tools send their output to the LOG_METPLUS file. If no, then each MET tool receives its own log file.

LOG_MET_VERBOSITY: MET verbosity level

LOG_LEVEL: METplus verbosity level

Personal
Favorite!

