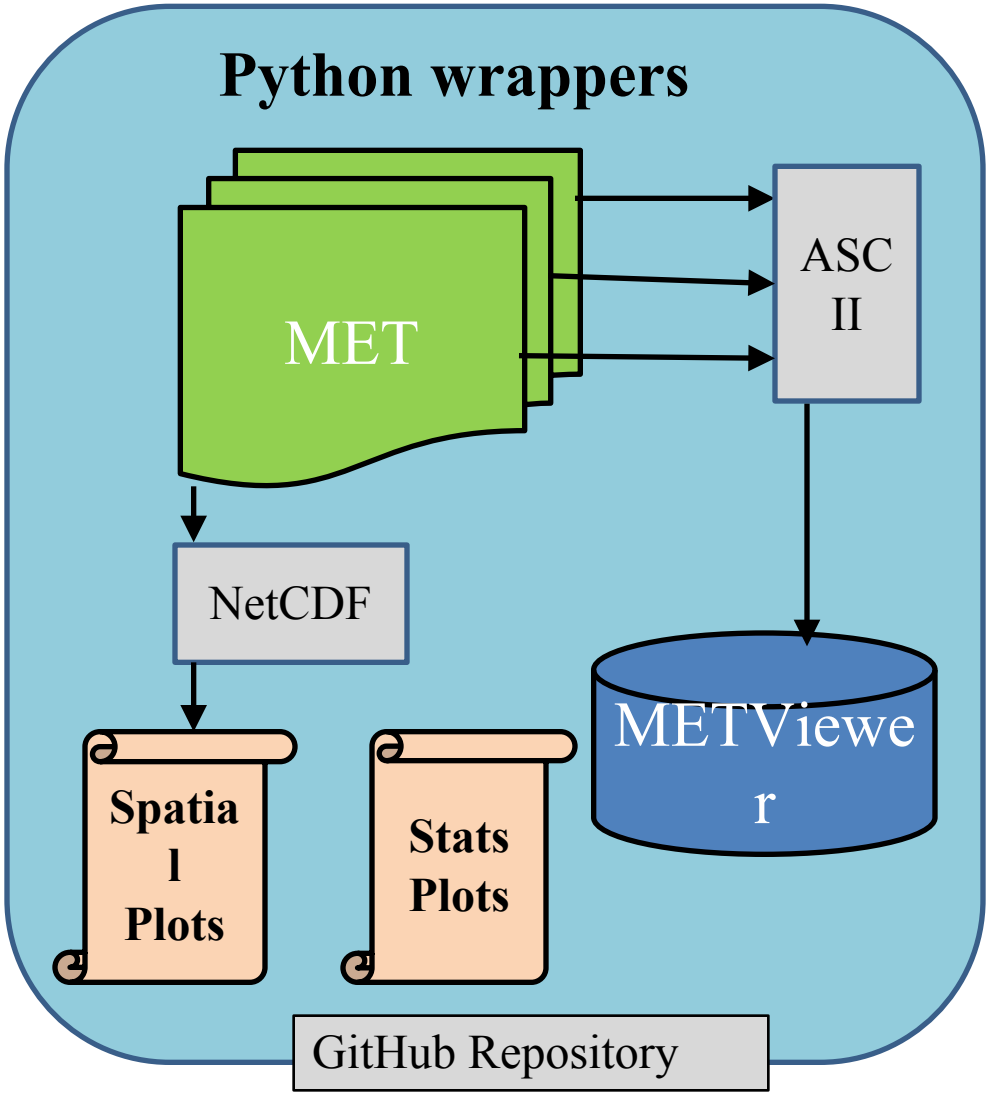
MET+ Overview

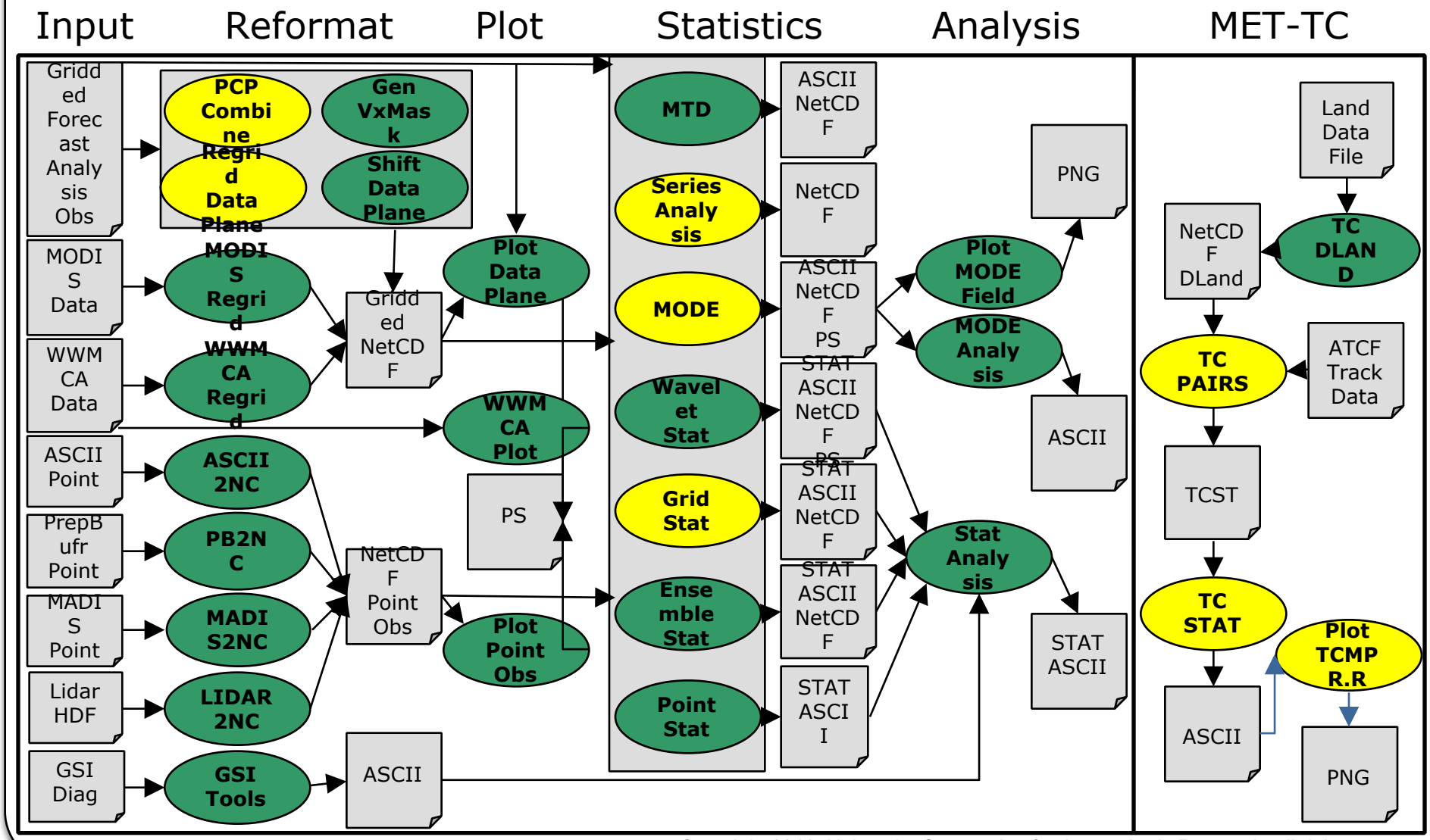
MET+ Unified Package

- Python wrappers around MET and METViewer:
- Simple to set-up and run
- Automated plotting of 2D fields and statistics

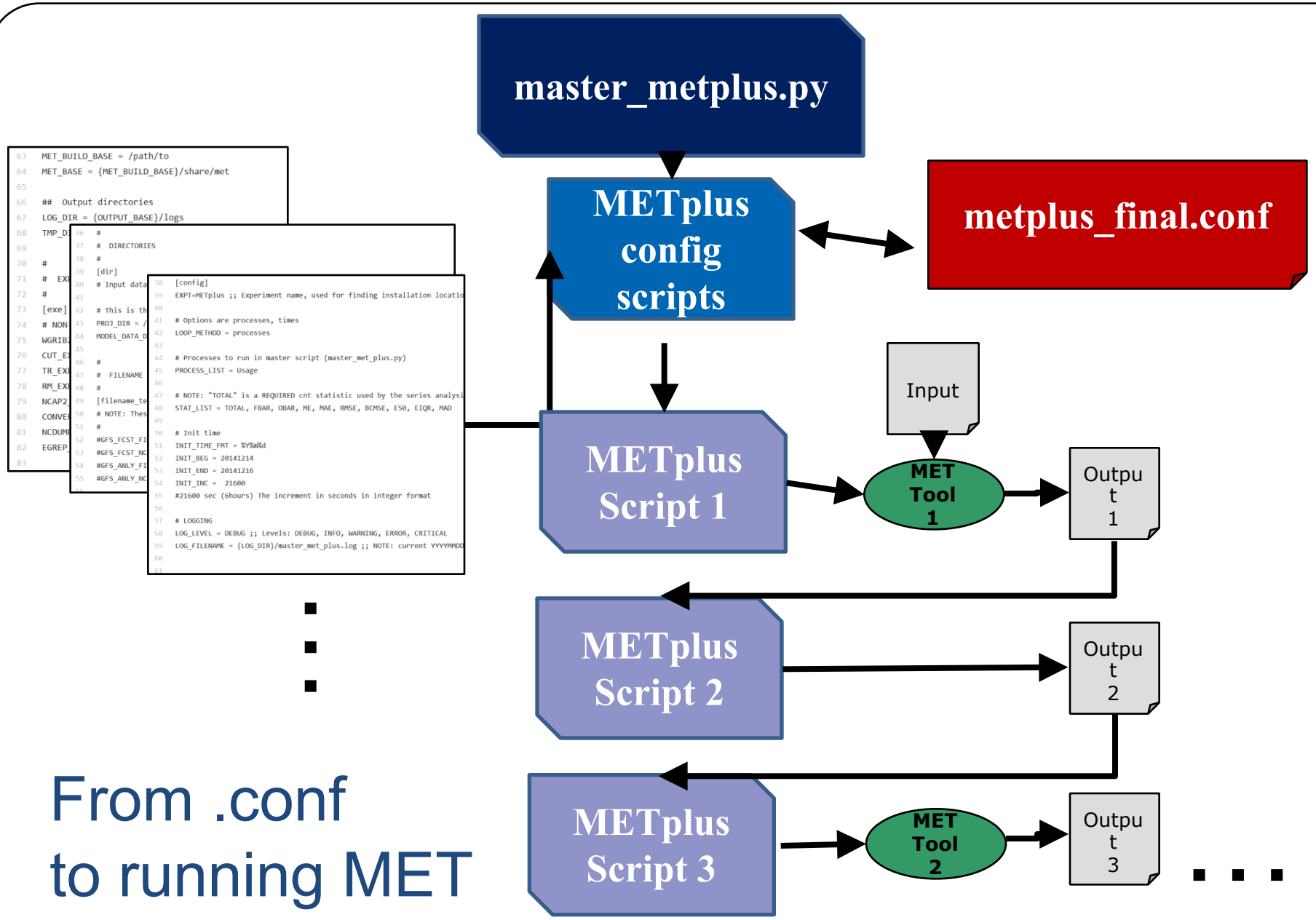
Initial system - Global deterministic with plans to generalize across scales when possible to quickly spin-up Ensembles, High Resolution & Global Components



What is currently wrapped with Python?



What does wrapped by Python mean?



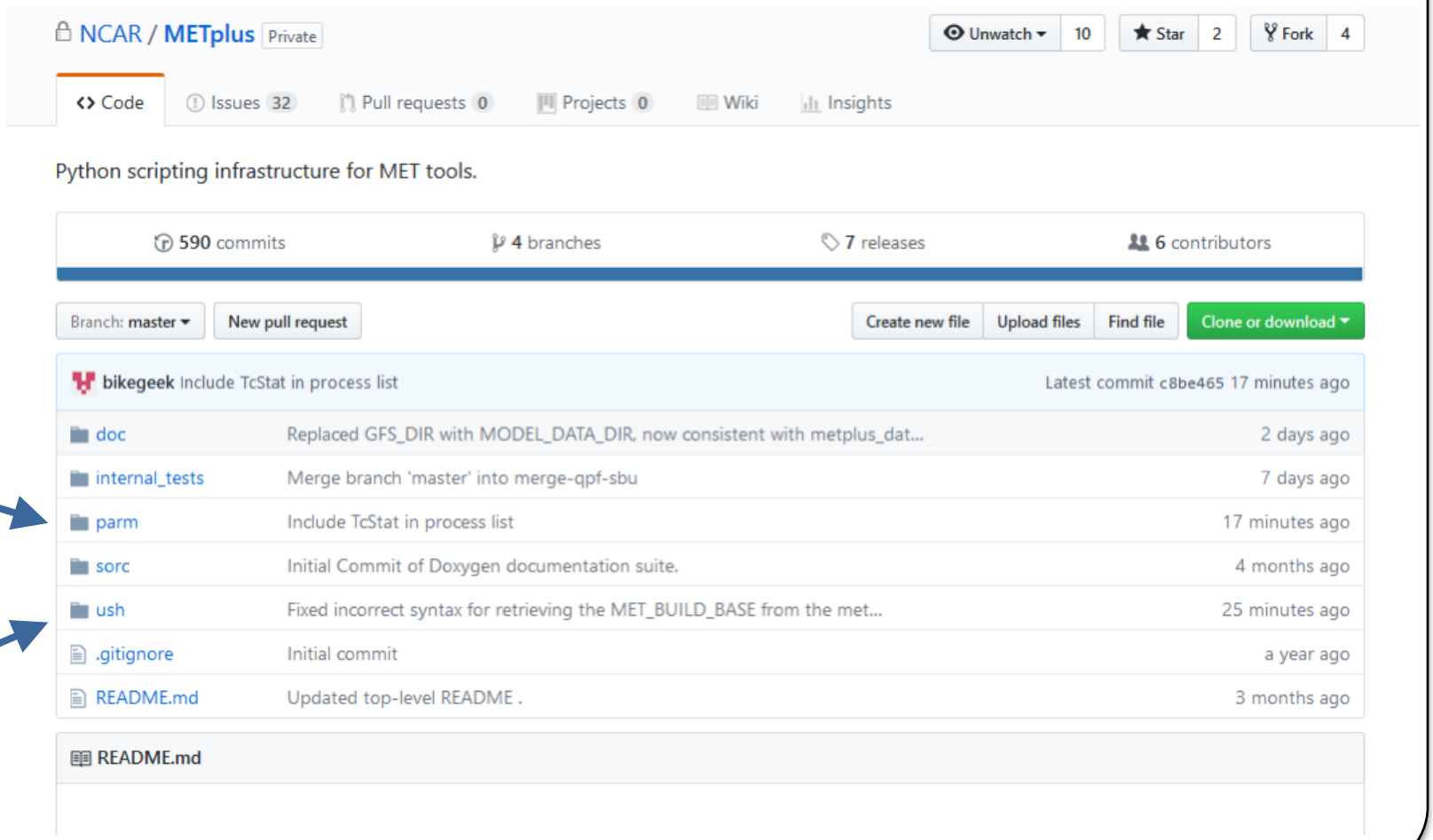
From .conf
to running MET

What does wrapped by Python mean?

At https://github.com/NCAR/METplus/

Control File and Config

Python Scripts



What does wrapped by Python mean?

METplus/parm/use_cases/feature_relative/examples/series_by_lead_all_fhrs.conf

series_by_lead_all_fhrs.conf

```
[config]
PROCESS_LIST = TcPairs, ExtractTiles, SeriesByLead

# Series analysis config file used by MET
SERIES_ANALYSIS_BY_LEAD_CONFIG_FILE =
{PARM_BASE}/met_config/SeriesAnalysisConfig_by_lead

# Variables and levels of interest
VAR_LIST = TMP/P850, HGT/P500

# Statistics of interest (Must always have include TOTAL)
STAT_LIST = TOTAL, FBAR, OBAR, ME

#TC-STAT filtering options used to extract tiles
EXTRACT_TILES_FILTER_OPTS = -basin ML

# The init time begin and end times, increment, and last init hour.
INIT_TIME_FMT = %Y%m%d
INIT_BEG = 20141214
INIT_END = 20141216
INIT_INC =  21600
```

What does wrapped by Python mean?

At https://github.com/NCAR/METplus/

In Configs: Environment variables passed in from Constants File

Series_Analysis_Config

```
30
31  ////////////////////////////////////////////////////////////////////////////////
32
33  cat_thresh  = [ NA ];
34  cnt_thresh  = [ NA ];
35  cnt_logic   = UNION;
36
37  //
38  // Forecast and observation fields to be verified
39  //
40  fcst = {
41
42     field = [
43        {
44           name  = "${NAME}";
45           level = [ "${LEVEL}" ];
46        }
47     ];
48
49  }
50  obs = fcst;
51
52  ////////////////////////////////////////////////////////////////////////////////
53
54  //
55  // Climatology mean data
```

What does wrapped by Python mean?

METplus/parm/use_cases/feature_relative/examples/series_by_lead_all_fhrs.conf

series_by_lead_all_fhrs.conf

```
[config]
PROCESS_LIST = TcPairs, ExtractTiles, SeriesByLead

# Series analysis config file used by MET
SERIES_ANALYSIS_BY_LEAD_CONFIG_FILE =
{PARM_BASE}/met_config/SeriesAnalysisConfig_by_lead

# Variables and levels of interest
VAR_LIST = TMP/P850, HGT/P500

# Statistics of interest (Must always have include TOTAL)
STAT_LIST = TOTAL, FBAR, OBAR, ME

#TC-STAT filtering options used to extract tiles
EXTRACT_TILES_FILTER_OPTS = -basin ML

# The init time begin and end times, increment, and last init hour.
INIT_TIME_FMT = %Y%m%d
INIT_BEG = 20141214
INIT_END = 20141216
INIT_INC =  21600
```

MET+ Beta - Prerequisites

- Python 2.7 *** When we started this was specified by NCO*
- R version 3.25 *** Only if you are using plot_tcmpr.R tool*
- nco (netCDF operators)
- MET version 6.0 or later installed
  *** Tool is designed to sit on-top of MET and should be version insensitive after METv6.0*
- Basic familiarity with MET

MET+ Beta Installations

- Theia
- /scratch4/BMC/dtc/Tara.Jensen/METplus
- WCOSS
- Gyre: /global/noscrub/Julie.Prestopnik/METplus
- Surge: /gpfs/hps3/emc/global/noscrub/Julie.Prestopnik/METplus

Getting Started

Instructions for grabbing release:

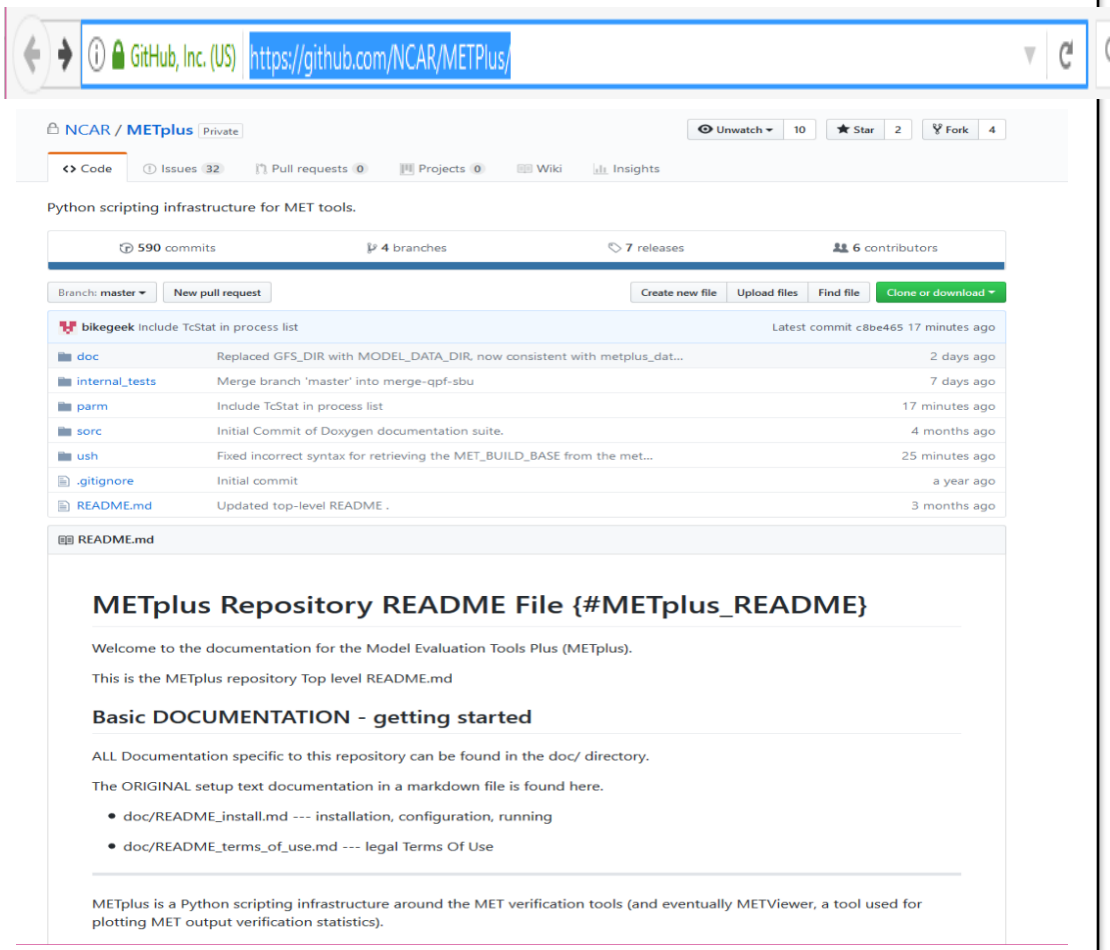
https://github.com/NCAR/METplus

Instructions for downloading:

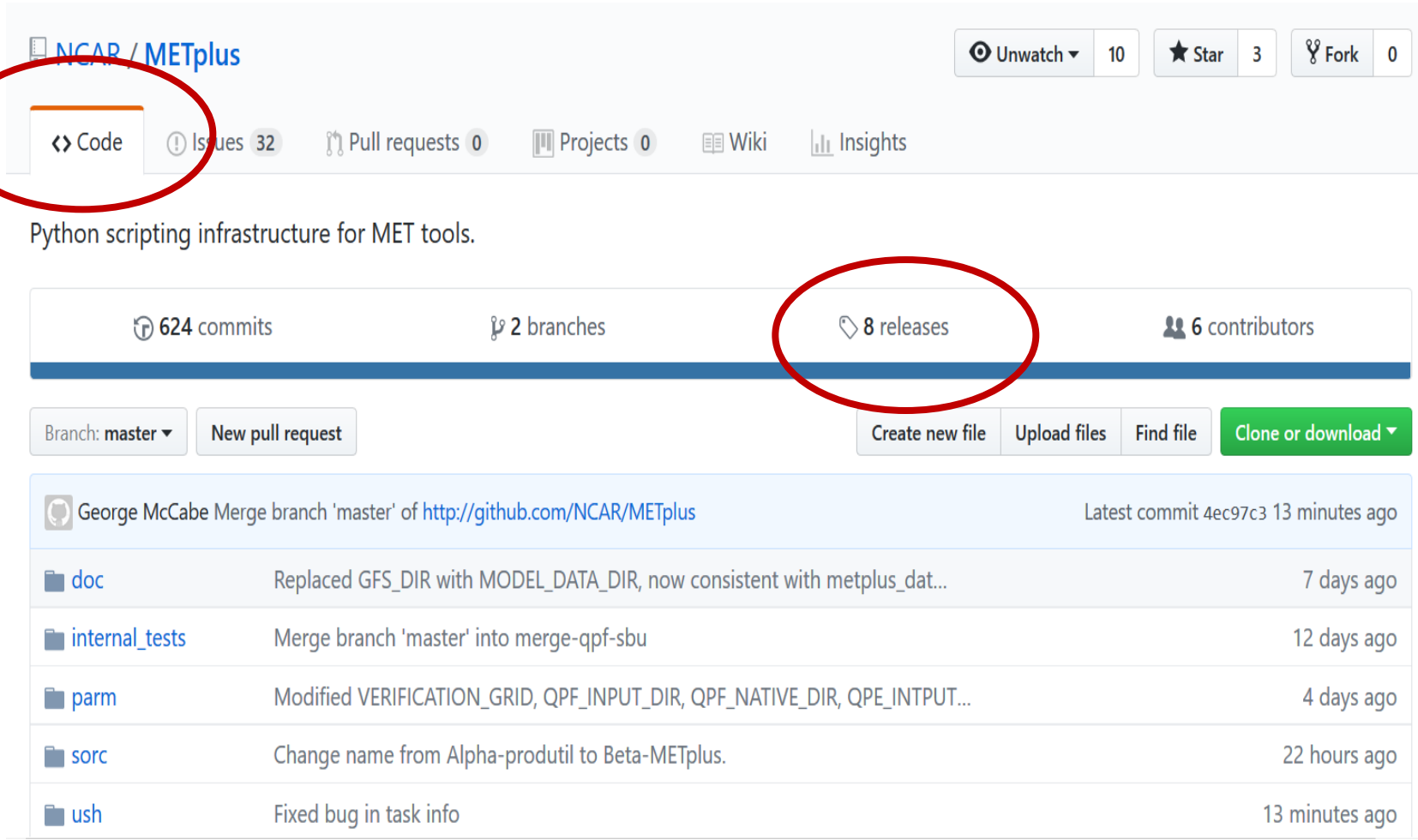
Click on the green download button on right side

Instructions for cloning:

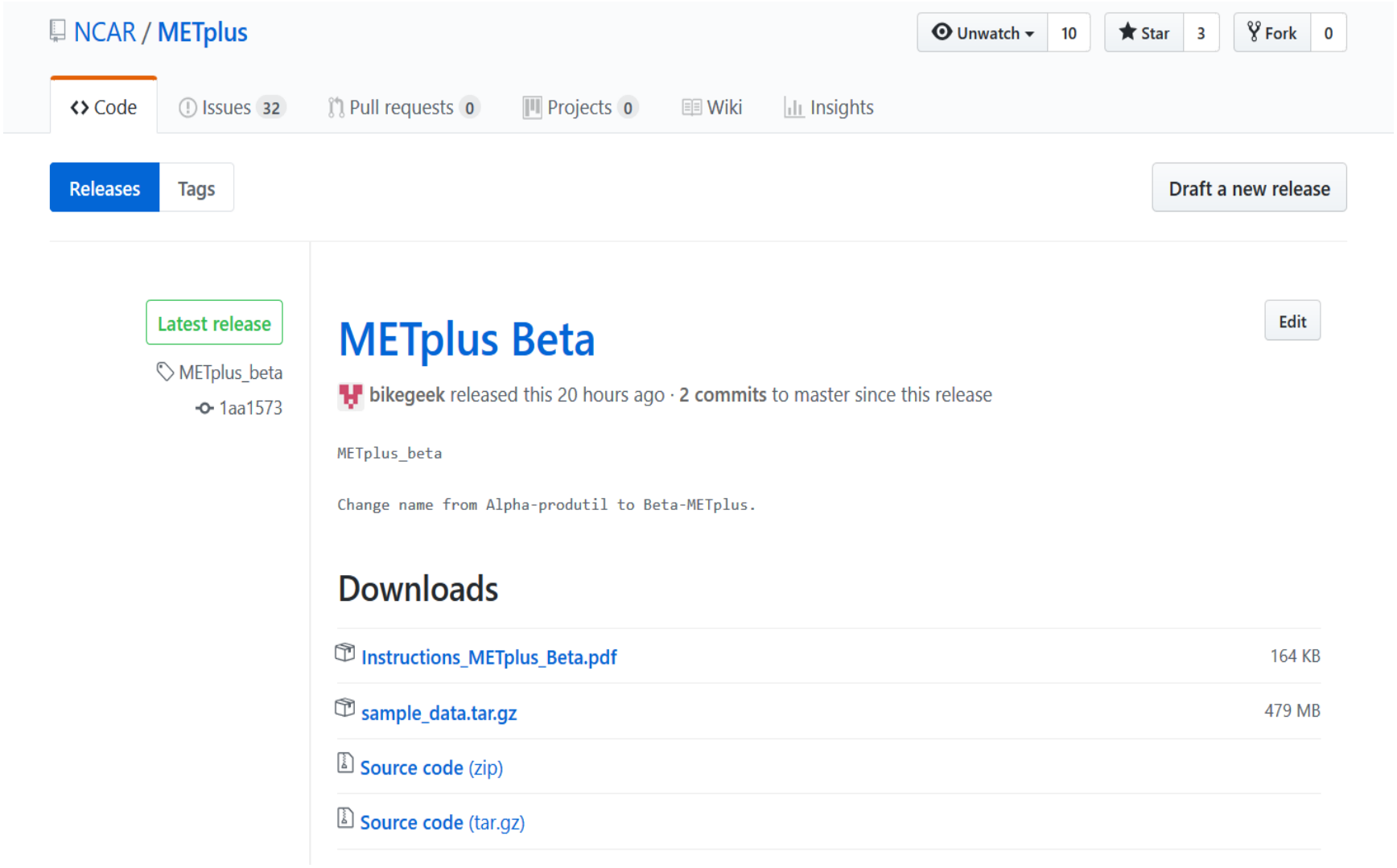
- git clone https://github.com/NCAR/METplus
- You should now have a METplus directory



Grabbing the Release



Grabbing the Release



Recommended Procedure - User

https://github.com/NCAR/METplus/wiki/GitHub-Repo-Information

GitHub Repository

For Users

Downloading a release from the GitHub web-page:

1. Click on the link 'Releases' link below the solid blue line near the top of the main NCAR/METplus web page and save the sample data, source code, and instructions.
2. Copy the source code (either .zip or .tar.gz) to your desired location.
3. Uncompress the source code:

   unzip <file.zip>

   or tar xvfz <file.tar.gz>
4. Copy the sample_data.tar.gz to the METplus/ directory
5. Uncompress the sample_data.tar.gz: tar xvfz sample_data.tar.gz

Recommended Procedure - Developer

https://github.com/NCAR/METplus/wiki/GitHub-Repo-Information

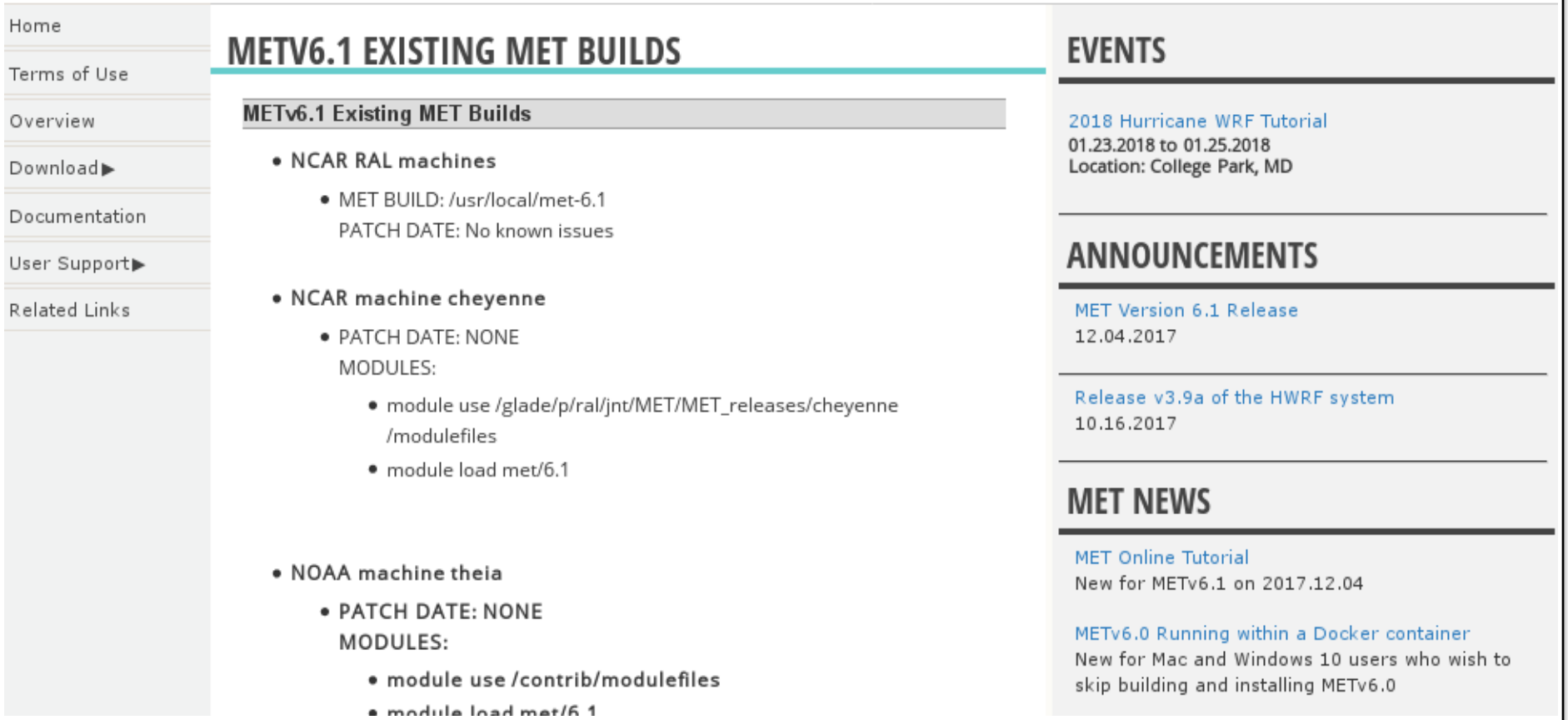
For Developers

1. Fork a copy of the NCAR/METplus code into your own GitHub repo, then clone from your GitHub repo to your local machine and make any changes.
2. When you want to save your work in progress, you can check into your GitHub repo.
3. When everything is working satisfactorily, you can do a pull request to the NCAR/METplus repo.

Diagram outlining GitHub process for developers:
https://github.com/NCAR/METplus/blob/master/doc/wiki_contents/diagrams/GitHub_process.png

Existing MET Builds

https://dtcenter.org/met/users/downloads/existing_met_builds/METv6.1_existing_met_builds.php



Home
Terms of Use
Overview
Download▶
Documentation
User Support▶
Related Links

METV6.1 EXISTING MET BUILDS

METv6.1 Existing MET Builds

- **NCAR RAL machines**
  - MET BUILD: /usr/local/met-6.1
    PATCH DATE: No known issues
- **NCAR machine cheyenne**
  - PATCH DATE: NONE
    MODULES:
    - module use /glade/p/ral/jnt/MET/MET_releases/cheyenne/modulefiles
    - module load met/6.1
- **NOAA machine theia**
  - **PATCH DATE: NONE**
    MODULES:
    - **module use /contrib/modulefiles**
    - **module load met/6.1**

EVENTS

2018 Hurricane WRF Tutorial
01.23.2018 to 01.25.2018
Location: College Park, MD

ANNOUNCEMENTS

MET Version 6.1 Release
12.04.2017

Release v3.9a of the HWRF system
10.16.2017

MET NEWS

MET Online Tutorial
New for METv6.1 on 2017.12.04

METv6.0 Running within a Docker container
New for Mac and Windows 10 users who wish to skip building and installing METv6.0

Setting up profile - Theia

Theia - .cshrc

```
set loadmemetplus='yes'
if ( $loadmemetplus == 'yes' ) then
    module use /contrib/modulefiles
    module load met
    module load nco
    module load wgrib2
    module load R
    set METPLUS_PATH=/scratch4/BMC/dtc/Tara.Jensen/METplus
    set MET_PATH=/contrib/met/6.1
    setenv JLOGFILE ${METPLUS_PATH}/logs/metplus_jlogfile
    setenv  PYTHONPATH
${METPLUS_PATH}/ush:${METPLUS_PATH}/parm
    setenv  PATH ${PATH}:${METPLUS_PATH}/ush:.
endif
```

Setting up profile - Gyre

WCOSS - /u/user/.bashrc

```
set loadmetplus='yes'
if [ $loadmetplus=='yes' ]; then
    echo "Loading METplus environment"
    module use /global/noscrub/Julie.Prestopnik/modulefiles
    module load met/6.1
    module load nco
    module load grib_util/v1.0.3
    module use /usrx/local/dev/modulefiles
    module load python
    export METPLUS_DEMO="/global/noscrub/Julie.Prestopnik/"
    export MET_DEMO="/global/noscrub/Julie.Prestopnik/met/6.1"
    export JLOGFILE="${METPLUS_DEMO}/METplus/logs/metplus_jlogfile"
    export
PYTHONPATH="${METPLUS_DEMO}/METplus/ush:${METPLUS_DEMO
}/METplus/parm"
    export PATH="${PATH}:${METPLUS_DEMO}/METplus/ush:."
fi
```

Setting up profile - Surge

```
set loadmetplus='yes'
if [ $loadmetplus=='yes' ]; then
    echo "Loading METplus environment"
    module use /gpfs/hps3/emc/global/noscrub/Julie.Prestopnik/modulefiles
    module load met/6.1
    module load grib_util/1.0.3
    module use /usrx/local/dev/modulefiles
    module load python
    module load nco-gnu-sandybridge/4.4.4
    export METPLUS_DEMO="/gpfs/hps3/emc/global/noscrub/Julie.Prestopnik/"
    export MET_DEMO="/gpfs/hps3/emc/global/noscrub/Julie.Prestopnik/met/6.1"
    export JLOGFILE="${METPLUS_DEMO}/METplus/logs/metplus_jlogfile"
    export
PYTHONPATH="${METPLUS_DEMO}/METplus/ush:${METPLUS_DEMO}/MET
plus/parm"
    export PATH="${PATH}:${METPLUS_DEMO}/METplus/ush:."
fi
```

Directory Structure

- doc/ - Doxygen documentation
- internal_tests/ - developer tests
- parm/ - where configs live
- README.md - general README
- sorc/ - executables
- ush/ - python scripts

METplus/doc

```
/scratch4/BMC/dtc/Tara.Jensen/METplus/doc$ ls
confguide.dox      NCEP_Coding_Standards.pdf  run.dox
install.dox        overview.dox               wiki_contents
install-main.dox   README_install.md          wiki_contents.dox
mainpage.dox       README_terms_of_use.md     wiki_contents_imagefiles.dox
metplus-conf.dox   rocoto.dox
```

METplus/internal_tests

```
/scratch4/BMC/dtc/Tara.Jensen/METplus/internal_tests$ ls
test_extract_tiles_unittest.py  test_series_by_init.py
test_original_run_tc_pairs.py   test_tc_pairs_wrapper.py
test_run_tc_pairs.py            test_util_unittest.py
```

METplus/parm

```
/scratch4/BMC/dtc/Tara.Jensen/METplus/parm$ ls */*
met_config/GridStatConfig_MEAN            met_config/TCPairsETCConfig
met_config/GridStatConfig_PROB            metplus_config/metplus_data.conf
met_config/MODEConfig_PROB                metplus_config/metplus_runtime.conf
met_config/MODEConfig_test                metplus_config/metplus_system.conf
met_config/SeriesAnalysisConfig_by_init   metplus_config/README
met_config/SeriesAnalysisConfig_by_lead

met_config/mask:
CONUS_HRRRTLE.nc  EAST_HRRRTLE.nc  HRRRTLE_GRID.grb2  WEST_HRRRTLE.nc

use_cases/feature_relative:
examples  feature_relative.conf  README

use_cases/qpf:
examples  qpf.conf  README

use_cases/track_and_intensity:
examples  track_and_intensity.conf
```

METplus/sorc

```
/scratch4/BMC/dtc/Tara.Jensen/METplus/sorc$ ls
doc  Makefile
```

METplus/ush

```
/scratch4/BMC/dtc/Tara.Jensen/METplus/ush$ ls
command_builder.py                  master_met_plus.py            string_template_substitution.py
command_builder.pyc                 met_util.py                   string_template_substitution.pyc
confdoc.py                          met_util.pyc                  task_info.py
config_launcher.py                  mode_wrapper.py               task_info.pyc
config_launcher.pyc                 pcp_combine_wrapper.py        tcmpr_plotter_wrapper.py
config_metplus.py                   pcp_combine_wrapper.pyc       tcmpr_plotter_wrapper.pyc
config_metplus.pyc                  produtil                      tc_pairs_wrapper.py
externals                           README_produtil.md            tc_pairs_wrapper.pyc
extract_tiles_wrapper.py            regrid_data_plane_wrapper.py  tc_stat_wrapper.py
extract_tiles_wrapper.pyc           regrid_data_plane_wrapper.pyc tc_stat_wrapper.pyc
extra_tropical_cyclone_plotter.py   run_example_uswrp.py          usage_wrapper.py
gempak_to_cf_wrapper.py             series_by_init_wrapper.py     usage_wrapper.pyc
gempak_to_cf_wrapper.pyc            series_by_init_wrapper.pyc    ush.dox
grid_stat_wrapper.py                series_by_lead_wrapper.py
grid_stat_wrapper.pyc               series_by_lead_wrapper.pyc
```

Key to running METplus – parm dir

- Met_config
  - All MET configuration files with Environment Variables should reside here
- Metplus_config
  - Three basic files that can be set by a system administrator for all to use
- Use_cases
  - feature_relative
    - Three ways of running feature_relative software
  - qpf
    - One example of grib to grib comparison and lots of GEMPAK examples
  - track_and_intensity
    - One example of computing track and intensity scores using plot_tcmpr.R script
- {user}_system.conf.{system_name}
  - Allows user to over-ride base system setting and write data into a given directory

Suggestions on how to set up parm dir

- Met_config
  - All MET configuration files with Environment Variables should reside here
- Metplus_config
  - Common install for BRANCH – includes paths to commonly used data
- Use_cases
  - Common install for FUNCTIONAL GROUP – includes paths for tests your conducting
- {user}_system.conf.{system_name}
  - Place your variances from use-cases in here, including pointing to your output directory, or pointing to a different config you are trying, etc…

Key to running METplus- parm dir

- Met_config
  - All MET configuration files with Environment Variables should reside here
- Metplus_config
  - Three basic files that can be set by a system administrator for all to use
- Use_cases
  - feature_relative
    - Three ways of running feature_relative software
  - qpf
    - One example of grib to grib comparison and lots of GEMPAK examples
  - track_and_intensity
    - One example of computing track and intensity scores using plot_tcmpr.R script
- {user}_system.conf.{system_name}
  - Allows user to over-ride base system setting and write data into a given directory

Three Use Cases

- Track and Intensity
  - Using MET-TC to pair up ATCF track files
  - plot_tcmpr.R to compute track and intensity errors and plot

- Feature Relative
  - Use MET-TC to pair up ATCF track files
  - Extract 30deg by 30deg tiles from GFS Forecast and Analysis files for comparison
  - Use Series-Analysis to compute statistics for the stack of tiles over CONUS
  - Use Plot-Data-Plane to generate quick look plots

- QPF
  - Use Pcp-Combine to accumulate 1-hr QPE into 3-hr accumulation
  - Use Grid-Stat to compute Categorical statistics

Phase I – Left to be done

- Moving MET code-base from SVN to GitHub
- Additional scripting to emulate base Global verification
- Scripting to push data to METViewer server, load data and make basic batch plots
- Install and test on Theia and WCOSS ✔
- Python scripting to emulate base Global Vx plots

Where to get help

- GitHub Instructions at Release link
  - https://github.com/NCAR/METplus/releases
  - Click on Instructions_METplus_Beta.pdf
- Contact met_help@ucar.edu

Supplementary Slides

MET+ Coding Standards

- NCEP Coding Standards
- NCO WCOSS Implementation Standards for directory structure and script naming conventions (http://www.nco.ncep.noaa.gov/idsb/implementation_standards/)
- pep8 for code style
- Doxygen and Python docstrings for documentation