

METplus Grid to Point Overview

George McCabe
(Originally created by Minna Win)

Background

- The Grid to Point (also referred to Grid to Obs) use case embodies the steps needed to obtain statistics via the MET point-stat tool for forecasts at observation points (vs a gridded analysis).
- Point observation data can be in prepBUFR format or netCDF. If the data is in prepBUFR format, it will first need to be converted to netCDF for the MET point_stat tool to read.

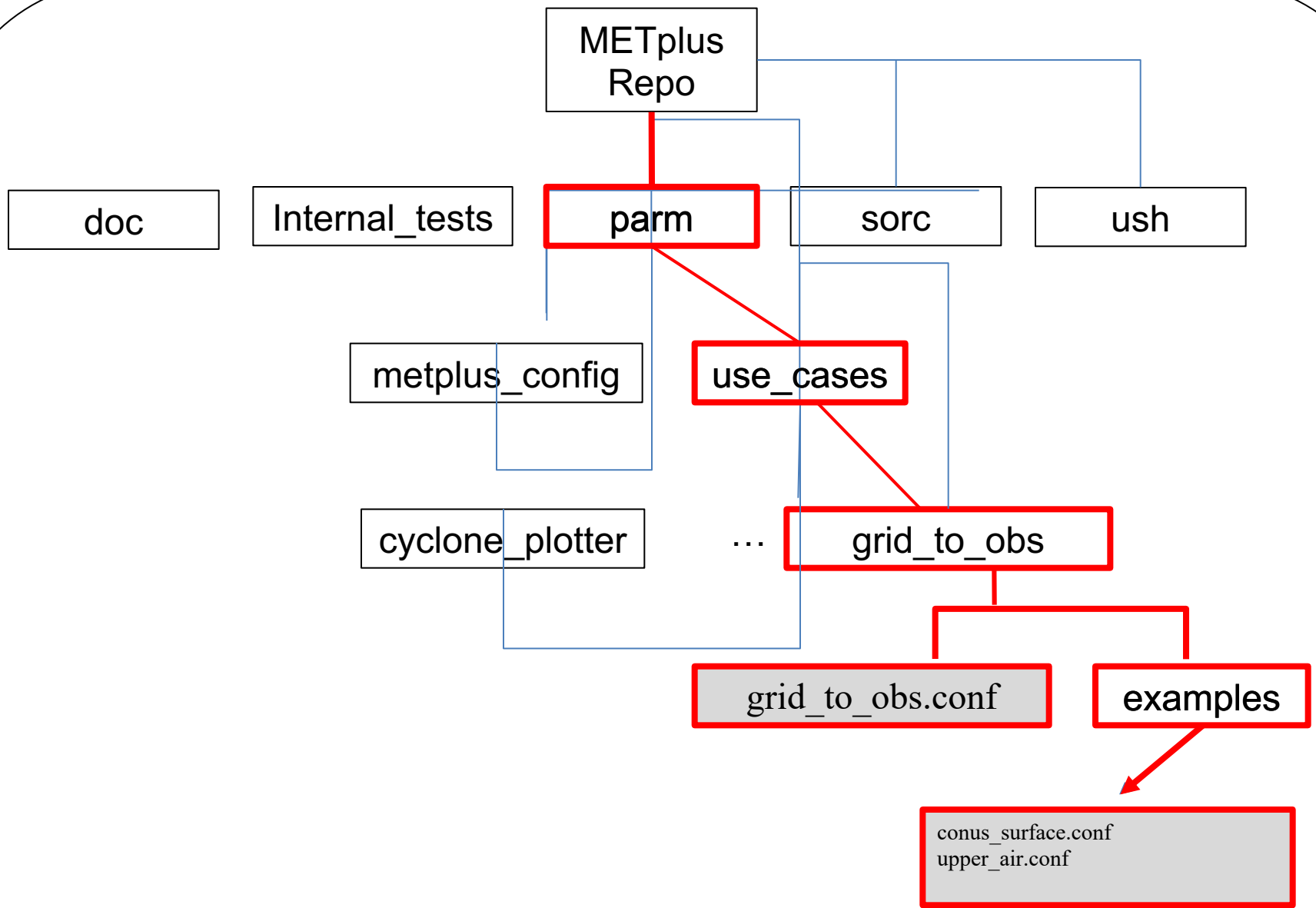
Tools

MET PB2NC

- Wrapped by PB2NC wrapper
- Converts prepBuf files into netCDF

MET Point-Stat

- Wrapped by PointStat wrapper
- Provides verification statistics for forecasts at observation points



Configuring METplus Best Practices

1) Copy parm directory to user location

- Users may work with a shared installation of METplus wrappers
 - Scripts are shared by all users
 - Changes to the parm (configuration) directory would affect all users – on installation, system settings should be set (location of MET version to use)
 - May not have permission to edit parm directory in shared location
- May want some settings to be static over all runs

2) Create user specific configuration files

- Used to override configuration settings for a given run
- Contains only the settings overridden from default in one place
- Preserves user settings when moving to new version of scripts

Configuring METplus Best Practices (continued)

- Review the configuration files and look for variables with a value containing the phrase “/path/to”
- These variables will need to be set. You can either:
 - Change them directly in the base configuration files
 - Copy them into a new file and modify the values (remember to use correct headers)

```
[dir]
# This is a comment.
# Use comments to help you remember important things.

TMP_DIR = /path/to
OUTPUT_BASE = /path/to
INPUT_BASE = /path/to
MET_INSTALL_DIR = /path/to
```

Configuring METplus

What to Override?

```
[dir]  
INPUT_BASE = /home/mccabe/data/input  
OUTPUT_BASE = /home/mccabe/data/output-g2g  
MET_INSTALL_DIR = /usr/local/met-8.0  
TMP_DIR = /tmp
```

```
[config]
```

```
[exe]  
WGTRIB2 = /usr/local/bin/wgrib2
```

```
CUT = cut  
TR = tr  
RM = rm  
NCAP2 = /usr/local/nco/bin/ncap2  
CONVERT = convert  
NCUMP = ncdump  
EGREP = egrep
```

Configuring METplus for Use Case

In your last "custom" configuration file, under the [config] section, set the PROCESS_LIST:

- **Scenario 1:** Point observation data is in prepBUFR format, require conversion to netCDF

PROCESS_LIST = PB2NC, PointStat

- **Scenario 2:** Point observation data is in netCDF format:

PROCESS_LIST = PointStat

Configuring METplus for Use Case

1. In your custom config file

```
PROCESS_LIST = Usage
```

2. Run METplus*, on command line:

```
master_metplus.parm -c path_to_your/custom_conf_file.conf
```

*Don't forget to replace all */path/to's*

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
- RegridDataPlane
- CyclonePlotter
- TCMPRPlotter
- PB2NC
- PointStat

Configuring METplus for Use Case

Question: What do I need to do if I want to use my own data?

Specify the appropriate values for the input data

PrepBUFR point data

```
# prepBUFR Point Data
[dir]
PB2NC_INPUT_DIR = /path/to/your/prepbuf_data
FCST_POINT_STAT_INPUT_DIR = /Model_dat/gfs
OBS_POINT_STAT_INPUT_DIR = {PB2NC_OUTPUT_DIR}
```

NetCDF point data

```
# netCDF Point Data
[dir]
FCST_POINT_STAT_INPUT_DIR = /Model_dat/gfs
OBS_POINT_STAT_INPUT_DIR = /path/to/your/nc_datas
```

Configuring METplus for Use Case

Question: What do I need to do if I want to use my own data?

Specify the appropriate filename structure under [filename templates] for the filename or directory:

```
[filename_templates]
# Input file template

# For NAM (conus surface)
# prepbufr.nam20170811.t06z.tm06
#PB2NC_INPUT_TEMPLATE =
prepbufr.nam.{init?fmt=%Y%m%d}.t{cycle?fmt=%2H}z.tm{offset?fmt=%2H}

# For GDAS (upper air)
# prepbufr.gdas.2018011312
PB2NC_INPUT_TEMPLATE = prepbufr.gdas.{valid?fmt=%Y%m%d%H}
```

Configuring METplus for Use Case

Defining Filename Templates:

Example 1: How to set up a variable in [file_templates]:

prepbufr.gdas.2017061618.nc is described by:

```
PB2NC_OUTPUT_TEMPLATE =  
prepbufr.gdas.{valid?fmt=%Y%m%d%H}.nc
```

- The parts of the file that will vary from file to file is contained within {}'s

Example 2: A little more than one piece of time information:

pgbf021.gfs.201706110 is described by:

```
FCST_POINT_STAT_INPUT_TEMPLATE =  
pgbf{lead?fmt=%3H}.gfs.{valid?fmt=%Y%m%d%H}
```

Configuring METplus for Use Case

NOTE

Not all MET dictionary values have a corresponding value in the METplus Config file

```
message_type = `${POINT_STAT_MESSAGE_TYPE}`;  
sid_exc      = [];  
obs_quality  = [ "1", "2", "3" ];  
duplicate_flag = NONE;
```

Corresponds to
POINT_STAT_MESSAGE_TYPE
In METplus config file

No corresponding environment variable in
METplus config

- look at the following files in <https://github.com/NCAR/METplus>:
 - parm/use_cases/grid_to_obs/met_config/PointStatConfig_conus_sfc
 - parm/use_cases/grid_to_obs/met_config/PointStatConfig_upper_air
- MET environment variables look like the following:
 - model_name = “`\${MODEL}`”;
 - field = [`\${FCST_FIELD}`];
 - message_type = `\${POINT_STAT_MESSAGE_TYPE}`;
 - **General format: `\${env var name}`**

Running METplus Setup

Add METplus/ush to PATH to run
master_metplus.py from any directory

csh:

```
setenv PATH </path/to>/METplus/ush:$PATH
```

bash:

```
export PATH=</path/to>/METplus/ush:$PATH
```

Post-run overview: output

Run `master_metplus.py`, from command line:

Conus surface

```
master_metplus.py -c parm/use_cases/grid_to_obs/grid_to_obs.conf \  
-c /path_to_your/custom_conf_file.conf
```

```
master_metplus.py -c parm/use_cases/grid_to_obs/examples/conus_surface.conf \  
-c /path_to_your/custom_conf_file.conf
```

Upper air:

```
master_metplus.py -c parm/use_cases/grid_to_obs/examples/upper_air.conf \  
-c /path_to_your/custom_conf_file.conf
```

*The `metplus_final.conf` file, log files, intermediate files, and final output files are created in the output directory that were specified in your last custom config file

See **README** in each use case directory for instructions on how to run use cases

Post-run overview: logging

- Located in your output directory under the 'logs' directory
- Format is based on the timestamp template you indicated in your `metplus_logging.conf` file:

```
Building MET pb2nc command... 2018-03-21 23:13:13,943 :  
INFO|:extract_prepbufr_file_info|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py| Creating  
prepbufr file information 2018-03-21 23:13:13,943 :  
DEBUG|:generate_output_nc_filename|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py  
Generating output NetCDF file name... 2018-03-21 23:13:13,943 :  
DEBUG|:build_pb2nc_command|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py|pb2nc  
called with: /usr/local/met-6.1/bin/pb2nc /d1/METplus_Mallory/data/prepbufr/gdas/prepbufr  
.gdas.2017060200 /d1/minnawin/pb2nc_crow_test/gdas/upper_air/prepbufr.gdas.2017060200.nc -v  
5 -log /tmp/pb2nc_test.log /home/minnawin/wip/METplus/parm/met_config/PB2NCConfig_upper_air  
2018-03-21 23:13:13,943 : RUNNING: /usr/local/met-6.1/bin/pb2nc  
/d1/METplus_Mallory/data/prepbufr/gdas/prepbufr.gdas.2017060200  
/d1/minnawin/pb2nc_crow_test/gdas/upper_air/prepbufr.gdas.2017060200.n c -v 5 -log  
/tmp/pb2nc_test.log /home/minnawin/wip/METplus/parm/met_config/PB2NCConfig_upper_air
```


Post-run overview: output

Directories Created

Scenario 1: point obs data is in prepBUFR

PB2NC wrapper is run, followed by PointStat wrapper

- pb2nc directory with netCDF data created:
in directory specified by `{PB2NC_OUTPUT_DIR}`
- point_stat text output (.stat files):
In directory `{POINT_STAT_OUTPUT_DIR}`

Recall:

You set your input data in your last custom config file

The MODEL and PB2NC_VERTICAL_LOCATION are defined in the point stat section of the use case conf file

Post-run overview: output

Directories Created

Scenario 2: point obs data is netCDF

- PointStat wrapper runs point_stat
- point_stat (MET) text output (.stat files) are created:

In directory `<output_base>/<model_name>`:

```
point_stat_000000L_20170601_000000V.stat  
point_stat_000000L_20170602_000000V.stat  
point_stat_000000L_20170603_000000V.stat
```

point_stat_000000L_20170601_000000V.stat:

```
VERSION MODEL DESC FCST_LEAD FCST_VALID_BEG FCST_VALID_END OBS_LEAD OBS_VALID_BEG OBS_VALID_END  
V8.0 gfs NA 000000 20170601_000000 20170601_000000 000000 20170531_231500 20170601_004500  
V8.0 gfs NA 000000 20170601_000000 20170601_000000 000000 20170531_231500 20170601_004500  
V8.0 gfs NA 000000 20170601_000000 20170601_000000 000000 20170531_231500 20170601_004500  
...etc.
```

Post-run overview: output

Questions?