



MeqTrees Batch Mode: A Short Tutorial

An Example

```
$ meqtrees-pipeliner.py -help
```

```
$ meqtrees-pipeliner.py -c .tdl.conf
```

```
example-sim.py
```

```
=_tdl_job_1_simulate_MS
```

use this
config file

compile this
script

run this job
(look in meqtrees.log)

Config Files

- **.tdl.conf** is where the browser keeps the current option set associated with each script
 - Section is script name
- **tdlconf.profiles** is where you save/load options using the buttons at the bottom of the options window
 - Section is the profile name you supply
- **meqtrees.log** is where all the jobs you run are logged, along with a complete configuration

Typical Use Case

- Set up a simulation or calibration interactively
- “Save” options to a profile named “foo”

```
$ cp tdlconf.profiles my.conf
```

```
$ meqtrees-pipelinier.py -c my.conf
```

```
@foo
```



use section [foo]
from my.conf

```
example-sim.py
```

```
=_tdl_job_1_simulate_MS
```

```
\  
\  
\
```

Typical Config File

```
...  
ms_sel.ms_antenna_sel = None  
ms_sel.ms_corr_sel = 1  
ms_sel.ms_taql_str = None  
ms_sel.msname = MS/WSRT_hba.MS  
ms_sel.output_column = CORRECTED_DATA  
ms_sel.select_channels = 0  
ms_sel.tile_size = 32  
noise_stddev = None  
oms_gain_models.err-gain.error_model = SineError  
oms_gain_models.err-gain.max_period = 2  
oms_gain_models.err-gain.maxval = 1.5  
oms_gain_models.err-gain.min_period = 1  
oms_gain_models.err-gain.minval = 0.5  
oms_gain_models.err-phase.error_model = NoError  
run_purr = 0  
sim_mode = sim only  
use_wsrt_cos3 = 1  
uvw_refant = 0  
uvw_source = from MS  
...
```

Changing Things On-The-Fly

```
$ meqtrees-pipeliner.py -c my.conf \
@foo \
ms_sel.msname=vlad.ms \
example-sim.py \
=_tdl_job_1_simulate_MS
```

Load section [foo] from my.conf,
then set MS name to “vlad.ms”,
then compile script example-sim.py.

(MS name is a compile-time
option, and thus must be set
before we compile the script.)

Same For Runtime Options

```
$ meqtrees-pipeliner.py -c my.conf \
@foo \
ms_sel.msname=vlad.ms \
example-sim.py \
ms_sel.tile_size=128 ← \
=_tdl_job_1_simulate_MS \
img_sel.imaging_weight = natural \
=make_dirty_image ←
```

This is a run-time option, and may be set before or after compiling the script (but before running the job!)

More jobs may be run in the same invocation, with their own option changes

Going Deeper

- meqtrees-pipeliner.py is just a simple wrapper around the Python interface (~170 lines of code)
- If you want to run MeqTrees non-interactively directly from Python, look inside

And That's All, Folks