1/29/21

# WT3 Developer Resource Page

## Resources:

* Resources located on OpenEI:
* Data references:
  + McGivney
  + TWB (IT3PR)
  + DEEP manual
  + BOR WaTER manual
* Models
  + Carlsbad Excel
  + Carlsbad Python
* Useful Links:
* OpenEI: <https://collab.openei.org/67/water-tap3>
* WT3 GitHub: https://github.com/NREL/NAWI-WaterTAP3
* NAWI GitHub: https://github.com/nawi-hub
* ProteusLib GitHub: <https://github.com/nawi-hub/proteuslib>
* GUI: <https://www.figma.com/proto/7JApf5g5vFCV4JIIccEYxf/Water-TAP-3-Prototype?node-id=1%3A27&viewport=1830%2C-1743%2C1.489366054534912&scaling=min-zoom>
* Meetings:
* WT3 NREL team meeting- Every Thursday at 3:00

## Adding to WT3:

* Adding a unit processes (UP) to WT3
* Copy the code from sulfuric\_acid\_addition.py (simple unit process that provides a good template)
* Create a new Jupyter script labeled as type\_of\_unit\_process.py
* *flow\_recovery\_factor* = .9999 because pyomo doesn’t like 1
* update the *base\_fixed\_cap\_cost*, *cap\_scaling\_exp*, *base\_year*, and *fixed\_op\_cost\_scaling\_exp* (default = .7) according to your particular UP
* in the build function, change *up\_name\_test* to *“type\_of\_unit\_process”*
* if there are equations for fixed capital beyond a simple cost curve accounted for in based\_fixed\_cap\_cost and cap\_scaling\_exp, add your fixed cost equations into *up\_costing* by creating a *fixed\_cap(flow\_in)* function
  + - *change self.fixed\_cap\_inv\_unadjusted to self.fixed\_cap\_inv\_unadjusted = fixed\_cap(flow\_in)*
    - change *self.base\_employee\_salary\_cost* to *self.base\_employee\_salary\_cost = fixed\_cap(flow\_in) \* salaries\_percent\_FCI*
* check what unit of flow rate the UP’s source uses
  + change *flow\_in =* to the appropriate pyunits.convert values
  + within cost\_method = “wt”, the flow\_in within the self.salaries equations requires MGD
* read comments and change/update where appropriate (ex. sources)
* open watertap.py and add your UP to the *unit\_process\_library\_list*
* open module\_import.py and add your UP to *get\_module*
* open unit\_process\_equations.py and add your UP to *build\_up*

## GitHub Commands:

Current model is always on GitHub

Here is a workflow for pushing model changes to GitHub:

* Make your changes
* Navigate to git folder on local computer
* *Git pull origin master* (to make sure the code you use is up to date)
* *Git status* (to see what’s going on)
* *Git branch new\_branch\_name* (to create a branch to later commit)
* *Git checkout new\_branch\_name* (to switch to this branch)
* *Git add file\_name.py* or *git add \**
* *Git status*
* *Git commit -m “Explain what changes were made”*
* *Git push -u origin master new\_branch\_name*
* *Git branch -d new\_branch\_name* (to delete local branch)
* On GitHub, someone else will review the pull request and ask for changes or accept your commit

## WT3 Team Contacts:

* NREL WT3 lead: Ariel Miara ([ariel.miara@nrel.gov](mailto:ariel.miara@nrel.gov))
* NREL/Mines developer: Anna Evans ([anna.evans@nrel.gov](mailto:anna.evans@nrel.gov))
* NREL developer: Kurby Sitterley ([kurban.sitterley@nrel.gov](mailto:kurban.sitterley@nrel.gov))
* NREL developer: Ann Wang ([ann.wang@nrel.gov](mailto:ann.wang@nrel.gov))