

# Versioning params\*.py

[params.yaml](#)

## Introduction : Importing params.py

Originally, cshell scripts for our GM sim workflow kept the parameters and code in the same file. When we translated to Python, we separated parameters and code, which was the right decision.

The parameter files, params.py, params\_base.py were imported using Python's "import" facility and all the parameters defined in these files became local variables.

This approach worked acceptably (with a lot of caveats) for a small scale individual event, but it showed significant limitations with Cybershake.

Note:

1. params\*.py includes: params.py, params\_base.py, params\_base\_bb.py, params\_vel.py, params\_uncertain.py, params\_bb\_uncertain.py
2. 'params.py' imports everything from 'params\_base.py'

## Limitations and risk/opportunity

1. Importing requires the params\*.py to be located in the same directory, or meddling with PYTHONPATH
2. One can not import different versions of params\*.py during the execution of the code
3. Two different ways of importing resulted in inconsistency.
4. from params import \* : Parameter x is accessed as a local variable "x"
5. import params : Parameter x is accessed as params.x
6. Growing technical debt : Roughly 100 places in our code base using this "import" method.
7. Right opportunity to fix this before introducing GM sim versions.

## Proposed Solution

Covert .py to .yaml: .yaml is an advanced, flexible file format that is ideal to keep configurations. More human readable than .json

Replace all "import" with execution of a standard yaml load function that returns a dictionary.

## Steps

1. Identify usage of params\*.py in all current working repositories and remove redundant params. (sprint 10)
2. Write proper code to import params (sprint 11)
3. Version params (sprint 11-12)

### Step 1. Cleaning up params\*.py

"from params\*.py import " is found in qcore, visualization and most frequently, slurm\_gm\_workflow.

The used/unused params and their specific usage are logged into 5 csv/txt files, which are attached to this document.

245 used, 398 unused params across all the params\*.py

### Step 2. Importing params

Proposed plan:

1. All useful params should be in a .yaml file; params should be imported via loading the .yaml file
2. Constructing a Param object. In that way, codes that use 'import params', 'params.xx' do not need to be changed

Implementation:

1. Removed unused params in params.py; Params that unsure if using reamins.
2. params\_base.py remains untouched as all its parameters are used.
3. A basic params.yaml that contains used parameters from params.py and all the parameters from params\_base.py. See attached params.yaml.

### Step 3. Versioning params

Please see:

1. Fix params imports in git repos
2. Setting up params.yaml with new Cybershake install structure
3. Remove params\*.py from the new Cybershake install structure
4. Create dummy templates for each level of yaml and Clean up