

# Update Rate Plan

## Load Pickle File

Import the pickled rate plan.

```
import pickle

with open("rate_plan.pkl", "rb") as fp:
    rate_plan = pickle.load(fp)
```

Rate the policy as before to confirm everything still works as expected.

```
json_input = {
    "Asset Size": 1_200_000,
    "Limit": 5_000_000,
    "Retention": 1_000_000,
    "Industry": "Hazard Group 2",
}

rate = rate_plan.calculate(**json_input)
print(rate)
```

4411.059983333334

## Update the Rate Plan

There is not an easy way to traverse the tree at the moment so I used this recursive function to collect the coordinates and parameters of each factor.

```

params = {}
def get_params(node, params={}, coord=""):
    if hasattr(node, "lhs") and hasattr(node, "rhs"):
        lhs_params = get_params(node.lhs, params, coord + "L")
        rhs_params = get_params(node.rhs, params, coord + "R")
        params.update(lhs_params)
        params.update(rhs_params)
        return params
    else:
        node_params = {coord+"__"+k: v for k, v in node.__dict__.items()}
        params.update(node_params)
        return params

params = get_params(rate_plan)
{k: v for k, v in params.items() if "name" in k if "Industry" in v}

```

```
{'LR__name': 'Industry'}
```

The coordinates for the Industry Factor are “LR”. We can update that lookup table following those coordinates.

```

rate_plan.lhs.rhs.lookup.update({"Hazard Group 2": 1.7913})
rate_plan.lhs.rhs.lookup

```

```
{'Hazard Group 1': 1.0, 'Hazard Group 2': 1.7913, 'Hazard Group 3': 1.5}
```

And now we can rerate the policy with the updated factor.

```

rate = rate_plan.calculate(**json_input)
print(rate)

```

```
6321.225398516
```