

# Regulatory Complexity Documentation

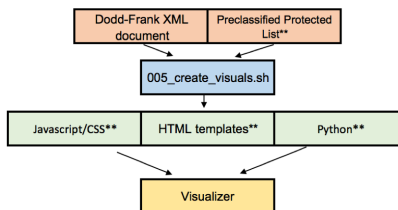
## 1 How to change the classifications?

The HTML application is based on predefined classifications that you can change. In this section you will learn how to do it.

We will use the last version of the visualizer:

```
RegulatoryComplexity\050_results\Dodd_Frank\Visuals\Visualizer_Versions\  
V8_visualizer
```

The following diagram shows the overview of the process to build the visualizer. In order to achieve the classification change, you must update some parts of the process (identified in the diagram with \*\*)



### 1.1 Protected List

The list of words already classified are independent .txt files located in:

```
RegulatoryComplexity\020_auxiliary_data\Sections\Protected_list\*.txt
```

Each file represents a classification category. Update the folder with the new files or just change the name of each file.

If you want to merge two different lists for one category, add “\_extra” at the final of the name.

## 1.2 Run Create Visuals Shell

Once the .txt files have the proper names, run the shell:

```
RegulatoryComplexity\100_code\shells\005_create_visuals.sh
```

The output are the html files representing each title of the dodd-frank document. These files contain span tags for the words found from the protected lists. Put all these files in the app directory:

```
V8_visualizer\app\templates\PreClass\
```

By default the html files are created in:

```
RegulatoryComplexity\050_results\Dodd_Frank\Visuals\DF_Templates_Protected\
```

If you want to switch the output directory, update the shell with the new one. (**argument -o**)

## 1.3 Python dictionary

In the application, go to the Python File:

```
V8_visualizer\app\helper_functions.py
```

At the top of the file, change the pre-defined dictionary with the new classifications and colors:

```
word_class = {'GrammaticalOperators': '#554600',
              'LegalOperators': '#806D15',
              'LegalReferences': '#D4C26A',
              'LogicalOperators': '#FF0AA',
              'RegulationOperators': '#AA0739',
              'Attributes': 'green',
              'EconomicOperand': 'yellow',
              'Other': 'red',
              'white': 'white'}
```

**Note: Do not change the item white**

## 1.4 CSS style file and images

There are two different CSS files that must be updated.

1) The file which contains the style background within the dodd-frank document for the classified words is located in:

```
V8_visualizer\app\static\styles\stylesheet.css
```

Change the names and colors for each span id.

```

}span.GrammaticalOperators {
    background-color:#554600;
} }

}span.LegalOperators {
    background-color:#806D15;
}

}span.LegalReferences {
    background-color:#D4C26A;
}

```

2) The file which contains the style for the context menu located in:

V8\_visualizer\app\static\styles\contextmenu.css

Change the #names for each #divID. Be sure to update the url(“img/name.png”) too

```

/* Put the proper classification names
# name
# .img/name.png
*/
}
#divID #GrammaticalOperators {
    background: url('img/GrammaticalOperators.png') no-repeat 6px 50%;
    background-size: 10px 10px;
    font-family: Arial, Sans-Serif;
}
}
#divID #LegalOperators {
    background: url('img/LegalOperators.png') no-repeat 6px 50%;
    background-size: 10px 10px;
}
}
#divID #LegalReferences {
    background: url('img/LegalReferences.png') no-repeat 6px 50%;
    background-size: 10px 10px;
}
}

```

Finally, put the proper .png color images for each category in:

V8\_visualizer\app\static\styles\img\

## 1.5 JavaScript and HTML view

Update the ”select case” statement inside of the main javascript file:

V8\_visualizer\app\static\js\main.js

Each case represents a word category. Change the case name (case ”name”) and the color htlr.Color (”color”). Be sure to have as many cases as categories without taking into account the additional buttons (Update and Remove).

```

/*
/* Put the proper classification names and colors
# case "Name":
# hltr.setColor(Color);
# Be sure to have the same number of cases and Names
*/

$(".custom-menu li").click(function(){
    // This is the triggered action name
    switch($(this).attr("data-action")) {
        // A case for each action. Your actions here
        case "GrammaticalOperators":
            flag = true;
            hltr.setColor("#554600");
            hltr.setClick(true);
            hltr.setReady(true);
            break;
        case "LegalOperators":
            flag = true;
            hltr.setColor("#806D15");
            hltr.setClick(true);
            hltr.setReady(true);
            break;
    }
});

```

**Note:** The select case statement is located at the bottom of the file.

Finally, go to the main html view:

V8\_visualizer\app\templates\index.html

Update the divID with the new ID, data-action and inner Text

```

<!-- Context button menu -->
<!-- Change context button menu -->
<!-- Change id, data Action and Text inside the tags -->

```

## 1.6 Update flask application in Pythonanywhere

Go to the following link: <https://www.pythonanywhere.com/>)

### Log in

username: alimon

password: complexity

Go to the site of your app

Files (Main Menu) – > mysite/ ( Folder located on the left)

Update the following files in the pythonanywhere site

1)

V8\_visualizer\app\helper\_functions.py

⇒

mysite\helper\_functions.py

2)

V8\_visualizer\app\templates\index.html

⇒

mysite\templates\index.html

3)

V8\_visualizer\app\templates\PreClass\\*.html

⇒

mysite\templates\PreClass\\*.html

4)

V8\_visualizer\app\static\styles\stylesheet.css

⇒

mysite\static\styles\stylesheet.css

5)

V8\_visualizer\app\static\styles\img\\*

⇒

mysite\static\styles\img\\*

6)

V8\_visualizer\app\static\js\main.js

⇒

mysite\static\styles\js\main.js

Finally, update the contextmenu.css with the new names and colors (Same as section 1.4):

mysite\static\styles\contextmenut.css

**Note: It must be changed manually because the url() has different functions in local-host compared to pythonanywhere. The contextmenu file of V8\_visualizer is slightly different to the pythonanywhere contextmenu file )**

To reload the website with the updates, go to Web (Main Menu) and click the green button: reload alimon.python ...