Setup as of 04.09.14

Workstation environment

**Bold** = Dependency file found below

1. Sublime Text 3 + Dreamweaver

Sublime addons needed:

* SideBarEnhancements
* Emmet
* SFTP- sftp-config.json in root of working dir - **Preview URLs**

1. WampServer 2.4 (versions as of this document)

Apache 2.4.4

PHP 5.4.12

MySQL 5.6.12

1. Install Ruby – to run everything. (During install select “Add Ruby executables to your PATH”) <http://www.ruby-lang.org>
2. Install DevKit (**install to c:\DevKit**) – <http://goo.gl/k2HIKb>
3. Make sure RubyGems (package manager) is installed. $>gem –v

If not – <http://rubygems.org> and install.

Make sure RubyGems is current. $>gem update --system

1. Install Sass as RubyGem – $>gem install sass

Confirm successful install –$>sass -?

1. Install Compass as RubyGem - $>gem install compass
2. Reboot computer
3. Install Nodejs – go to nondejs.org/download (make sure to get the 64 bit installer)
4. Install Grunt

$>npm install –g grunt-cli

Do this in working directory for each new project.

**Bold** = Dependency file found below

1. Setup New Project Directory
2. Create new folder with Project Name in c:\wamp\www
3. Copy everything from “c:\wamp\www\Robert Means\new\_client” into new folder
4. Create the package.json file in working directory – **package.json (open and name project before step #3 below)**

[This allows you to specify plugins and dependencies that your project needs to run. ]

1. [see red instructions above] From working directory - $>npm install

It reads the package.json file just created and installs dependencies in a node\_modules folder w/in your project.

1. Confirm Nodejs was bundled in npm install - $>node –v

If, for some weird reason it is not there - install it – <http://nodejs.org>

[See: <https://npmjs.org/doc/json.html> - This specifies all the different values you can add to your package.json file.]

1. Create gruntfile.js – in root of working directory - **gruntfile.js**

The grunt-cli installed in workstation setup Step 8 will use this to read and handle the tasks. It runs the plugins we added when we created the project.json file.

1. Create config.rb – in root of working directory. This is so you can use Grunt instead of just using Compass. It tells Sass & Compass how it needs to be structured. – **config.rb**

Fire it all up

Launch Wamp

Command line from working directory -$>grunt

Go to work!

**package.json**

You can Google the names of the different plugins/dependencies in order to see current versions that are available.

{

"name" : "name-of-project",

"version" : "0.0.1",

"dependencies" : {

"grunt": "~0.4.4",

"grunt-contrib-watch": "~0.6.1",

"grunt-contrib-compass": "~0.7.2",

"grunt-contrib-uglify": "~0.4.0",

"matchdep": "~0.3.0"

}

}

**config.rb**

css\_dir = '\_css'

sass\_dir = 'components/sass'

javascripts\_dir = '\_scripts'

#output style for compressed CSS - see Compass documentation at http://compass-style.org/help/tutorials/configuration-reference/

#-- uncomment line below when ready for production to put all css on one line

#output\_style = :compressed

#-- :expanded allows debugging wile in development

output\_style = :expanded

relative\_assets = true;

**Preview URLs**

Make sure Sublime Text will open hypertext to browser on F12 – rt click on hypertext file in root > Project > Edit Preview URLs

{

"C:\\wamp\\www\\Robert Means":{

"url\_testing":"http://localhost/Robert Means"

},

"C:\\wamp\\www\\Wildflower":{

"url\_testing":"http://localhost/Wildflower"

},

}

**gruntfile.js**

module.exports = function(grunt) {

grunt.loadNpmTasks('grunt-contrib-uglify'); //uglify minifies js upon save

grunt.loadNpmTasks('grunt-contrib-watch'); //watch keeps an eye out for saves

grunt.loadNpmTasks('grunt-contrib-compass'); //compass preprocesses scss files

grunt.initConfig({

uglify: {

my\_target: {

files: {

'\_scripts/scripts.js': ['components/js/\*.js']

} //files

} //my\_target

}, //uglify

compass: {

dev: {

options: {

config: 'config.rb'

} //options

} //dev

}, //compass

watch: {

options: { livereload: true }, //keeping an eye on files and updating Chrome upon save of any file

scripts: {

files: ['components/js/\*js'],

tasks: ['uglify']

}, //scripts

sass: {

files: ['components/sass/\*.scss'],

tasks: ['compass:dev']

}, //sass

hypertext: {

files: ['\*.php','\*.htm']

} //php

} //watch

}) //initConfig

grunt.registerTask('default', 'watch'); //allows you to do everything above when you run grunt from command line

} //exports