**Open Integration – Getting Started**

The purpose of an Open Integration architecture is to allow you to interface your data with the analytic tools of the Mapping Broadband Health in America platform so you can examine the picture of connectivity and health with respect to datasets of interest to you. Simply put, it’s to put the power of this platform’s visual analytics into your hands and your data. This can be done quickly once your data is in the correct format. Steps I and II below help you get your data into the correct format. Step III shows you how to modify the code (html) in 3 easy steps. There are also completed examples (Disability Population Files and Veterans Population Files) that you can examine and modify.

**(I) DOWNLOAD EXTENSIBILITY FILES:** Download the extensibility files to your local environment. You will see one html file ('extensible.html') and four folders (css, data, img, js). You should save your data to the 'data' file and you will only modify the html file.

**(II) SAVE YOUR DATA IN THE CORRECT FORMAT - This is the only hard part (we promise)!**

1. CONVERT YOUR DATA INTO A JAVASCRIPT OBJECT: Save your dataset as a JavaScript object. An easy way to do this is to convert your CSV or other format to JSON format first, and then assign it to a variable. Because you are going to be 'merging' datasets by county geo-ids, make sure your dataset: (1) has data at the county level; (2) has a county FIPS code column; (3) the county FIPS code column has text values (not numeric) and named countyFIPS (for example, Autuga county data object would contain have the following key and value for the county FIPS code: "countyFIPS": "01001"

2. NAME YOUR DATA: Assign your data objects to a variable. We have included two example files (Populations with Disabilities, and Veterans Populations). In the 'Veterans Population Files' the data variable can be found in the data folder, in the vetspop.js file. There it's defined as:

***var*** vetsdata**=[**

**{**

"countyFIPS"**:** "01001"**,**

"mapdata\_pop\_2014"**:** 55395**,**

"STATECODE"**:** 1**,**

"pop\_pct\_65andabove"**:** 14**,**

"COUNTYNAME"**:** "Autauga County"**,**

"STATE"**:** "AL"**,**

"vets\_all"**:** 6063**,**

"vets\_all\_pct"**:** 11**,**

"national\_vet\_pop"**:** 7**,**

"vets\_all\_include"**:** 1

**},...**

There are many converters online that can help you convert your data into the JSON format.

3. SAVE YOUR DATA: Once you have the data in the correct format (see the diablity.js or the vetspop.js files as examples), save it as a JavaScript file to the 'data' folder. You can open these files with notepad or any text editor.

**(III) MODIFY THE HTML - the easy part...we have done all the coding for you!**

Open the 'exntesible.html' file and scroll down to the extensibility section...it comes with instructions in comments and looks like this:

<!--

(1) SET OPTION VALUES: The format for setting option values is: VARIABLENAME$LOWERBOUNDARY\_UPPERBOUNDARY

The code parses everything before the '$' sign as the variable name; and everything after the '$' sign as the numeric range.

-->

<div class=**"row row-space"** style=**"background-color:lightgray; border: solid 0.1em darkblue; padding-top:0.5em"**>

<div class=**"col-sm-5"**>

<div class=**"data-label"**>

<label for=**"ov-select-extend"**>**DATA\_NAME:**</label>

</div>

</div>

<div class=**"col-sm-7 no-left"**>

<select id=**"ov-select-extend"** class=**"selectpicker select-advanced advanced-extend"**>

<optgroup label=**"ENTER\_SUB\_GROUP\_NAME 1"**>

<option value=**""** selected>**None** </option>

<option value=**"VARNAME1$LB\_UB"**>**LB\_UB**</option>

<option value=**"VARNAME1$LB\_UB"**>**LB\_UB**</option>

</optgroup>

<optgroup label=**"ENTER\_SUB\_GROUP\_NAME 2"**>

<option value=**"VARNAME2$LB\_UB"**>**LB\_UB**</option>

<option value=**"VARNAME2$LB\_UB"**>**LB\_UB**</option>

</optgroup>

</select>

</div>

</div>

<!--

(2) ENTER PATH TO FILE WITH OBJECTS W/ GEOGRAPHIC FEATURES & DATA; MAKE SURE COUNTY FIPS CODES ARE TEXT VALUES IN THE OBJECT AND IS NAMED AS: countyFIPS...SO FOR AUTAUGA COUNTY IT SHOULD BE: "countyFIPS": "01001"

-->

<script src=**"data/PATH\_TO\_DATA\_OBJECT.js"**></script>

<!--

(3) SET extsdata VARIABLE BELOW TO REFER TO OBJECT CONTAINING GEOGRAPHIC FEATURES

-->

<script>

***var*** extsdata**=**OBJECT\_NAME\_W\_GEOGRAPHIC\_FEATURES**;**

</script>

<!--

To increase page load speed, the above scripts can be placed at the bottom of the <body> element

-->

(1) SET OPTION VALUES: This is where you define the variables in the dataset that you want displayed and the value levels that you want to use as selection options. The format for setting option values is:

VARIABLENAME$LOWERBOUNDARY\_UPPERBOUNDARY

The code parses everything before the '$' sign as the variable name; and everything after the '$' sign as the numeric range. For example, if your variable name is 'vets\_all\_pct' representing the percentage of the total population in a given county that is veterans, then you would place that name before the '$' sign: vets\_all\_pct$LOWERBOUNDARY\_UPPERBOUNDARY.

Then if you want the first option to be to show all counties where veteran population is 0-5%, then '0' is the lower boundary and '5' is the upper boundary (separated by an underscore character: \_ )...the option value would be: vets\_all\_pct$0\_5. A completed example from the Veterans Population Files is below ('vets\_all\_pct' and 'vets\_all' are variables in the dataset in the vetspops.js data file):

<div class=**"row row-space"**>

<div class=**"col-sm-5"**>

<div class=**"data-label"**>

<label for=**"ov-select-extend"**>**Vet Pop Filter:**</label>

</div>

</div>

<div class=**"col-sm-7 no-left"**>

<select id=**"ov-select-extend"** class=**"selectpicker select-advanced advanced-extend"**>

<optgroup label=**"Veteran Pop Filter 1"**>

<option value=**""** selected>**None** </option>

<option value=**"vets\_all\_pct$0\_5"**>**0-5% vet pop.** </option>

<option value=**"vets\_all\_pct$5\_10"**>**5-10% vet pop.** </option>

<option value=**"vets\_all\_pct$10\_15"**>**10-15% vet pop.** </option>

<option value=**"vets\_all\_pct$15\_35"**>**15%+ vet pop.** </option>

</optgroup>

<optgroup label=**"Veteran Pop Filter 2"**>

<option value=**"vets\_all\_pct$0\_10"**>**0-10% vet pop.** </option>

<option value=**"vets\_all\_pct$10\_35"**>**10%+ vet pop.** </option>

</optgroup>

<optgroup label=**"Veteran Pop Filter 3"**>

<option value=**"vets\_all$0\_5000"**>**0-5000 vet pop.** </option>

<option value=**"vets\_all$5000\_1000000"**>**5000+ vet pop.** </option>

</optgroup>

</select>

</div>

</div>

(2) ENTER PATH TO FILE WITH OBJECTS W/ GEOGRAPHIC FEATURES & DATA: In the portion of the code in the <script> tags, modify the path to your data file by simply entering the name of your data file. To do this, change "PATH\_TO\_DATA\_OBJECT" to the name of the data file that you saved into the data folder:

<script src=**"data/PATH\_TO\_DATA\_OBJECT.js"**></script>

For the example in the Veterans Population Files you will see that the data file name is vetspops.js and is saved under the data folder. Therefore, the path in the <script> tags is changed to the following:

<script src=**"data/vetspops.js"**></script>

(3) LET THE PLATFORM KNOW THE NAME OF YOUR DATA OBJECTS: To do this, set the extsdata variable to equal to the name of the name you gave your data object:

<script>

***var*** extsdata**=**OBJECT\_NAME\_W\_GEOGRAPHIC\_FEATURES**;**

</script>

For the example in the Veterans Population Files you will see that the data name is vetspops.js and is saved under the data folder. Therefore, the path in the <script> tags is changed to the following:

<script>

***var*** extsdata**=**vetsdata**;**

</script>

Once your code is completed, you can rename you can open the extensibility.html file in your browser and you should see that a feature to manipulate your dataset has been added to the user interface. You should now be able to manipulate the map and see the change in connectivity and health based on the varying levels of your variable.

The complete code for the example we just examined is given below (i.e. this the only part of the html file you need to modify once you have saved your dataset to the data folder):

<!--

(1) SET OPTION VALUES: The format for setting option values is: VARIABLENAME$LOWERBOUNDARY\_UPPERBOUNDARY

The code parses everything before the '$' sign as the variable name; and everything after the '$' sign as the numeric range.

-->

<div class=**"row row-space"**>

<div class=**"col-sm-5"**>

<div class=**"data-label"**>

<label for=**"ov-select-extend"**>**Vet Pop Filter:**</label>

</div>

</div>

<div class=**"col-sm-7 no-left"**>

<select id=**"ov-select-extend"** class=**"selectpicker select-advanced advanced-extend"**>

<optgroup label=**"Veteran Pop Filter 1"**>

<option value=**""** selected>**None** </option>

<option value=**"vets\_all\_pct$0\_5"**>**0-5% vet pop.** </option>

<option value=**"vets\_all\_pct$5\_10"**>**5-10% vet pop.** </option>

<option value=**"vets\_all\_pct$10\_15"**>**10-15% vet pop.** </option>

<option value=**"vets\_all\_pct$15\_35"**>**15%+ vet pop.** </option>

</optgroup>

<optgroup label=**"Veteran Pop Filter 2"**>

<option value=**"vets\_all\_pct$0\_10"**>**0-10% vet pop.** </option>

<option value=**"vets\_all\_pct$10\_35"**>**10%+ vet pop.** </option>

</optgroup>

<optgroup label=**"Veteran Pop Filter 3"**>

<option value=**"vets\_all$0\_5000"**>**0-5000 vet pop.** </option>

<option value=**"vets\_all$5000\_1000000"**>**5000+ vet pop.** </option>

</optgroup>

</select>

</div>

</div>

<!--

(2) ENTER PATH TO FILE WITH OBJECTS W/ GEOGRAPHIC FEATURES & DATA; MAKE SURE COUNTY FIPS CODES ARE TEXT VALUES IN THE OBJECT AND IS NAMED AS: countyFIPS...SO FOR AUTAUGA COUNTY IT SHOULD BE: "countyFIPS": "01001"

-->

<script src=**"data/vetspops.js"**></script>

<!--

(3) SET extsdata VARIABLE BELOW TO REFER TO OBJECT CONTAINING GEOGRAPHIC FEATURES

-->

<script>

***var*** extsdata**=**vetsdata**;**

</script>

<!--

To increase page load speed, the above scripts can be placed at the bottom of the <body> element

-->