How to generate a vector map for use with jVectorMap:

**Here is a good StackOverflow guide:**

<http://stackoverflow.com/questions/11068645/how-to-generate-a-new-map-for-jvectormap-jquery-plugin>

**Text of above info in case that page goes away:**

Just in case anyone will need the same. To run converter, you need (this instruction is valid for Windows environment):

download and install OSGeo4W (use Express Desktop Install)

download desired shape file (this natural data file works fine).

put anyjson to converter directory

create makemap.bat in converter directory, like this:

python ^

converter.py ^

../../ne\_10m\_admin\_1\_states\_provinces\_shp.shp ^

test-map.js ^

--width 400 ^

--where "ISO\_3166\_2 = 'RU-' and code\_hasc!=''" ^

--country\_name\_index 12 ^

--country\_code\_index 18 ^

--minimal\_area 4000000 ^

--buffer\_distance -0.5 ^

--simplify\_tolerance 10000 ^

--longitude0 54.8270 ^

--name russia

run OSGeo4W shell (it will be added to start menu)

run makemap.bat

enjoy generated map

As a sample output attaching this fiddle with russian map generated using steps mentioned above http://jsfiddle.net/dyP4c/3/

About parameters (what I know)

where condition is used to filter out shapes from shapefile using shapefile attributes

simplify\_tolerance will affect map quality and size

country\_name\_index index of REGION NAME attribute in shapefile or column in tab separated file if codes\_file is used

country\_code\_index index of REGION CODE attribute in shapefile or column in tab separated file if codes\_file is used

codes\_file is tab separated file (if you want to use names/codes not from shapefile)

longitude0 is longitude of area (used this)

name is map name

P.S.: OSGeo4W package has nice Quantium GIS Browser to browse shapefile attributes. You can try to search other shapefiles on the web (not only natural data) to convert to jvectormap

**What I did:**

1. Installed the programs the above guide told me to
2. Downloaded the jVectorMap converter scripts (it is contained in the ‘Full Project’ download)
   1. <http://jvectormap.com/download/>
3. Created a .bat file with the above Python script
4. To create a country-level map, download this file
   1. <http://www.naturalearthdata.com/http//www.naturalearthdata.com/download/110m/cultural/ne_110m_admin_0_countries.zip>
   2. I used 1:110m to keep the generated file size down. You can download maps from Natural Earth that are very detailed. For example, there are a few island nations off the coast of Africa. The 1:110m map will not pick these up, but the 1:10m map will. Since I just needed the outlines of the larger countries, the 1:110m worked just fine.
5. To create a state-level map, download this file
   1. <http://www.naturalearthdata.com/http//www.naturalearthdata.com/download/10m/cultural/ne_10m_admin_1_states_provinces_shp.zip>
6. Create the script:
   1. There are a few parameters that must correct in order for the script to run successfully. The above StackOverflow post does a pretty good job explaining them. The jVectorMap documentation has an explanation of each as well:
      1. <http://jvectormap.com/documentation/gis-converter/>
   2. where clause – filter what you want on the map
      1. example: --where "(continent = 'Africa')" ^
   3. country name index – the index on the .dbf file of the name of the country. You can either open the .dbf file that comes in the above downloads (opens in excel) or open the .shp file in OSGeo4W Quantum GIS browser you downloaded and installed earlier. Opening the .shp file in the GIS browser will give you 4 tabs, one being attributes where you can see the indexes.
      1. Example: --country\_name\_index 3 ^
      2. Note: the .dbf file is zero-indexed
   4. country code index – very similar to the country name index. This one is most often the three letter abbreviation of the country. You may see this in multiple columns. I believe this is because there can be regions inside regions that are treated differently such as Hong Kong. It’s in China but has a different government that China.
      1. Example: --country\_code\_index 4 ^
      2. Note: the .dbf file is zero-indexed
      3. **Note: for my specific use, I made the country name and code index the same. This way the .js file for the map used the same verbiage for countries and states as an excel spreadsheet that was being used.**
   5. simplify-tolerance – this controls how detailed the map is. The higher the number, the lower the detail. 1000 is what I used.
      1. Example: --simplify\_tolerance 1000 ^
   6. longitude0 – this is the longitude of the region you are generating. You can find this out by typing something like “Africa Longitude” in google. The first number you see is your longitude.
      1. Example: --longitude0 9.1021 ^
   7. Projection – this is the type of projection the map will have. There are four options: aea, lcc, mill and merc. I generally chose lcc, but if a map looks funny, choose a different one. I picked the one that looked the best.
      1. --projection lcc ^
7. In the end, this is what the Africa.bat file looked like

python ^

converter.py ^

ne\_110m\_admin\_0\_countries/ne\_110m\_admin\_0\_countries.shp ^

../tests/assets/africa.js ^

--width 400 ^

--where "(continent = 'Africa')" ^

--country\_name\_index 3 ^

--country\_code\_index 4 ^

--minimal\_area 4000000 ^

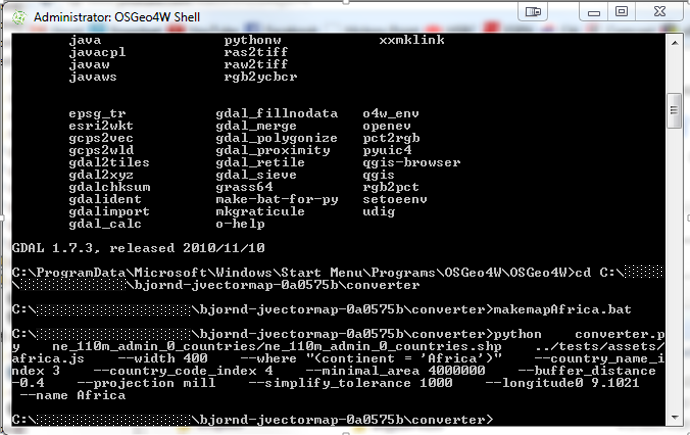
--buffer\_distance -0.4 ^

--projection mill ^

--simplify\_tolerance 1000 ^

--longitude0 9.1021 ^

--name Africa

1. Open up the OSGeo4W Shell, change directory to your converter, run the .bat file.
   1. A successful run will look something like this:
2. After it runs, you should see a JavaScript file in the directory you specified. Just include that on your web page and set it as the map to test it out.

**Problems**

1. A big problem I had was figuring out the correct parameters to use. If you are getting errors, it’s most likely a parameter issue. Make sure all of your paths are correct as well (e.g. the .shp file is in the right place)
2. Sometimes I would generate a map and only a few countries or states would show. This was because I had the country name and code indexes wrong in the script.

**Notes:**

1. For my specific use, I made the country name and code index the same. This way the .js file for the map used the same verbiage for countries and states as an excel spreadsheet that was being used.
2. Some regions are going to be too small to display on the map. One example is Singapore. This issue has been reported to the maker of jVectorMap and he has said that there is nothing that can be done because they are so small.
   1. <https://github.com/bjornd/jvectormap/issues/21>
   2. <http://stackoverflow.com/questions/14676195/jvectormap-missing-countries>
   3. What I did to get around this was to generate Singapore by itself. But instead of giving the map a full width of something like 400, I just gave it a width of 2.5. This way, the generated map of Singapore was very small. Then I just grabbed the coordinates from the .js file and dropped them on the Asia map. I had to manually adjust the placement of it though. Once I got it in the right place, I had a small Singapore on the map. It is quite a bit larger scale wise than the real Singapore, but I had to make it clickable. The happy medium I found was large enough to show but not too large to look “too” wrong.
3. For some maps like Europe, you may need to handle the little islands a little differently. For Europe, the islands for Portugal, Spain and Norway were throwing off the centering of the map. The map by default shows all clickable regions. If the islands are far out, they might leave giant blank spaces and the map won’t be centered and look nice. What I did was to just remove the coordinates of these islands from the .js file. The paths start with an ‘M’ and end with a ‘Z’, so you can use that to find smaller chunks of coordinates and it will most likely be one of the islands. I didn’t need the islands, so leaving them off was just fine.