

Optional Exercise B: Experimentation with SLD

Goal

The goal of this exercise is to give you time to consolidate what you have learned so far, to experiment with SLD and to increase your confidence in reading and writing SLD. You may choose one of the options below

Part 1: Experiment with Manually Configuring Layers and Data Sources

Modify the rules and symbology from the “admin.sld” file according to your wishes by first manually adding one new WMS layer of type point (e.g. the nuclear power plants dataset or download the airports dataset from <http://www.naturalearthdata.com/downloads/10m-cultural-vectors/>) or polygon (e.g. eu_countries). Then experiment with new filters (e.g. PropertyIsBetween) or new Styles according to your interests. Please consult the Appendix B of the manual (<http://karlinapp.ethz.ch/qgismapservmanual.pdf>) for a list of supported SLD options.

Part 2: Experiment with Point Symbolization

In this step, you may continue editing the custom symbolization for the city of Bern or, in a similar manner, create a custom symbolization for the nuclear power plants.

You may use the symbol defined in the “greensphere.svg” file (symbol available on “<http://karlinapp.ethz.ch/giscience2010/>” and containing a SVG gradient definition) or you may use any of the svg symbols available in the QGIS symbol library, or you may create your own SVG symbols using Inkscape (available locally on your computer).

Part 3: Experiment with Area Symbolization

In this step, you can modify the pattern for Switzerland or for any country of your choice. You may use the patterns available at <http://www.carto.net/svg/samples/patterns.shtml> or you may create more complex SVG patterns using Inkscape (available locally on your computer).

Part 4: Experiment with Line Symbolization

In this step, you can first download the transportation dataset from <http://www.naturalearthdata.com/downloads/10m-cultural-vectors/>. Set it as a WMS layer and experiment with symbolizing the different classes of lines with different styles by using double lines and dash arrays (like explained in the lecture).