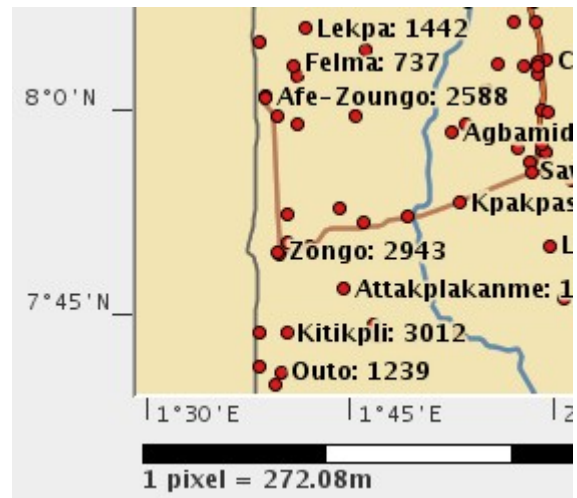


AtlasStyler 1.3 labeling tutorial

Submitted by [Stefan A. Krüger](#) on Mon, 10/19/2009 - 11:59

In the last months Geopublishing.org focussed its development efforts on the Geopublisher application. Besides internal changes like switching the underlying [Geotools library](#) to the latest version, upcoming [Geopublisher](#) 1.3 will come with an easy-to-use statistics module which allows to publish interactive geo-statistical charts integrated in your digital atlases. Wait for a dedicated blog entry about the feature when 1.3 will be released in early October.

But also did some nice new features slip into the [AtlasStyler SLD Editor](#) dialog, especially into the labeling pane. The following pages provide a step-by-step tutorial on how to use the new labeling functions like new *placement options*, a *beautiful halo effect*, *multiple attribute labeling* and better generalization using a *label priority attribute*.



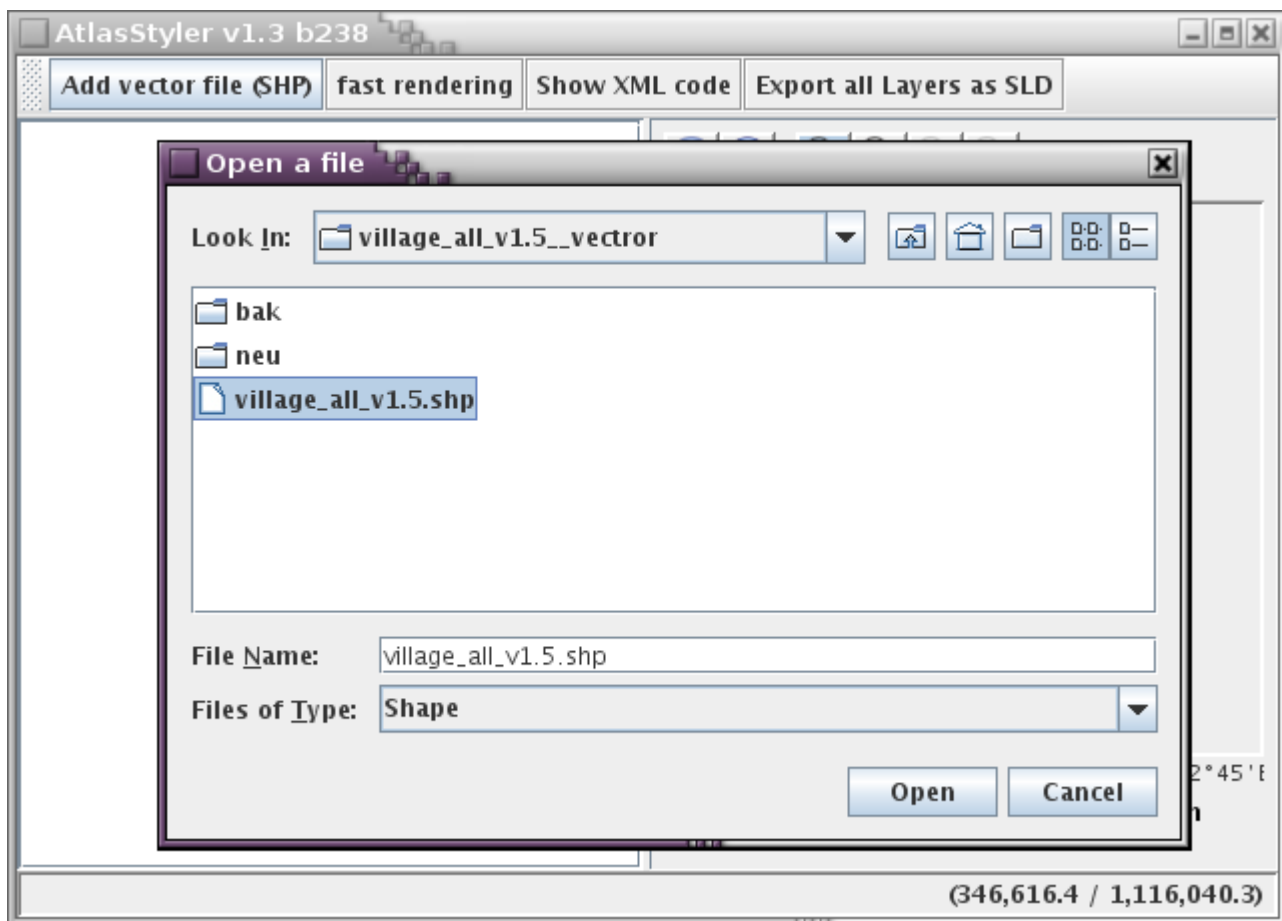
Start AtlasStyler:

Start [AtlasStyler](#) by clicking the green start button:

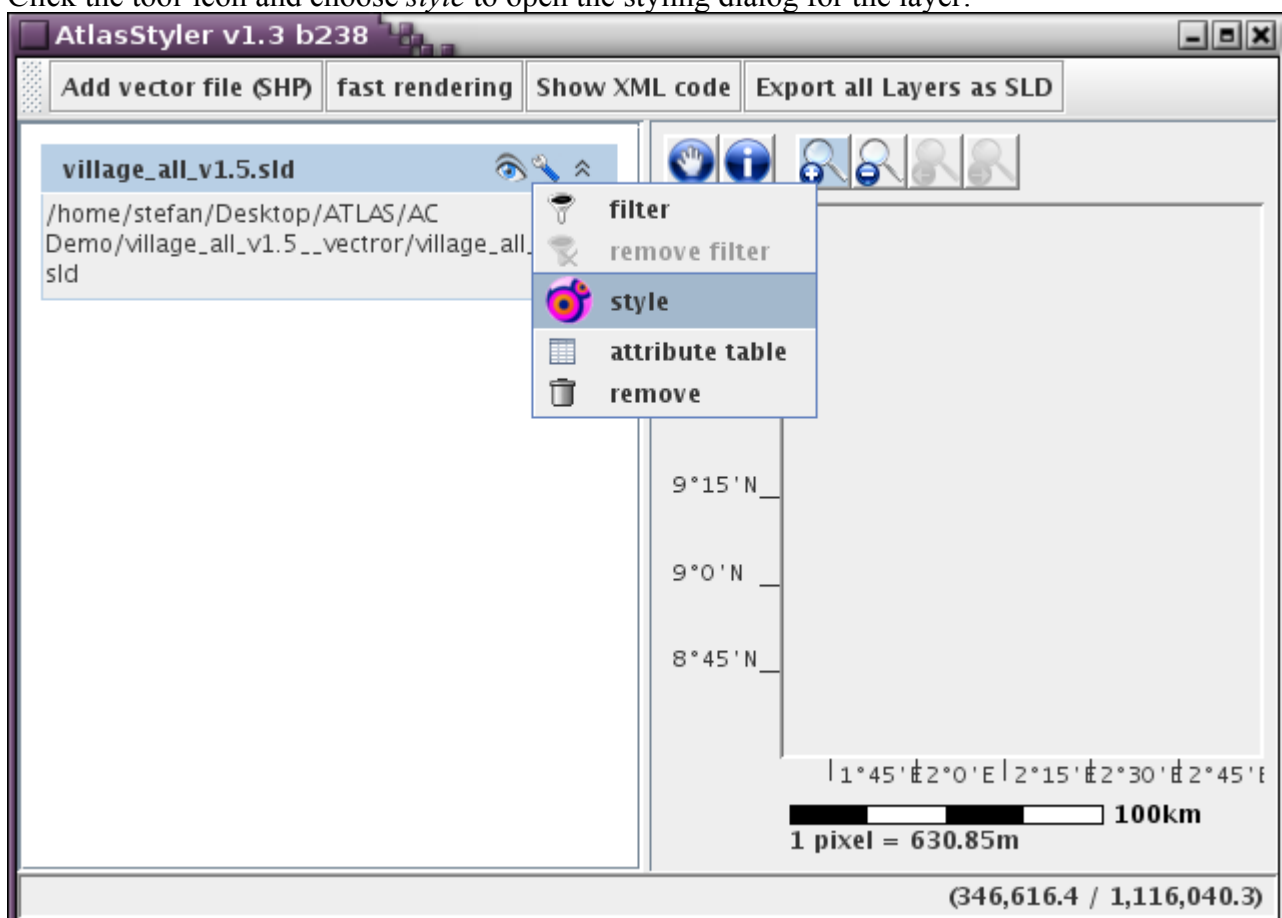


Some JavaWebStart windows will appear and you will be asked to trust our application enough to let it run on your computer. If starting AtlasStyler via JavaWebStart is not working, please download the archive from the [download page](#).

Add your Shapefile by pressing *Add vector file (SHP)*. For our example I have chosen a layer with 10.000+ settlements in Benin. You can download it [here](#). If any *.sld* file with the same basename would already exists, the layer would be rendered with it in the map area. Otherwise no points will be visible until we create a style.

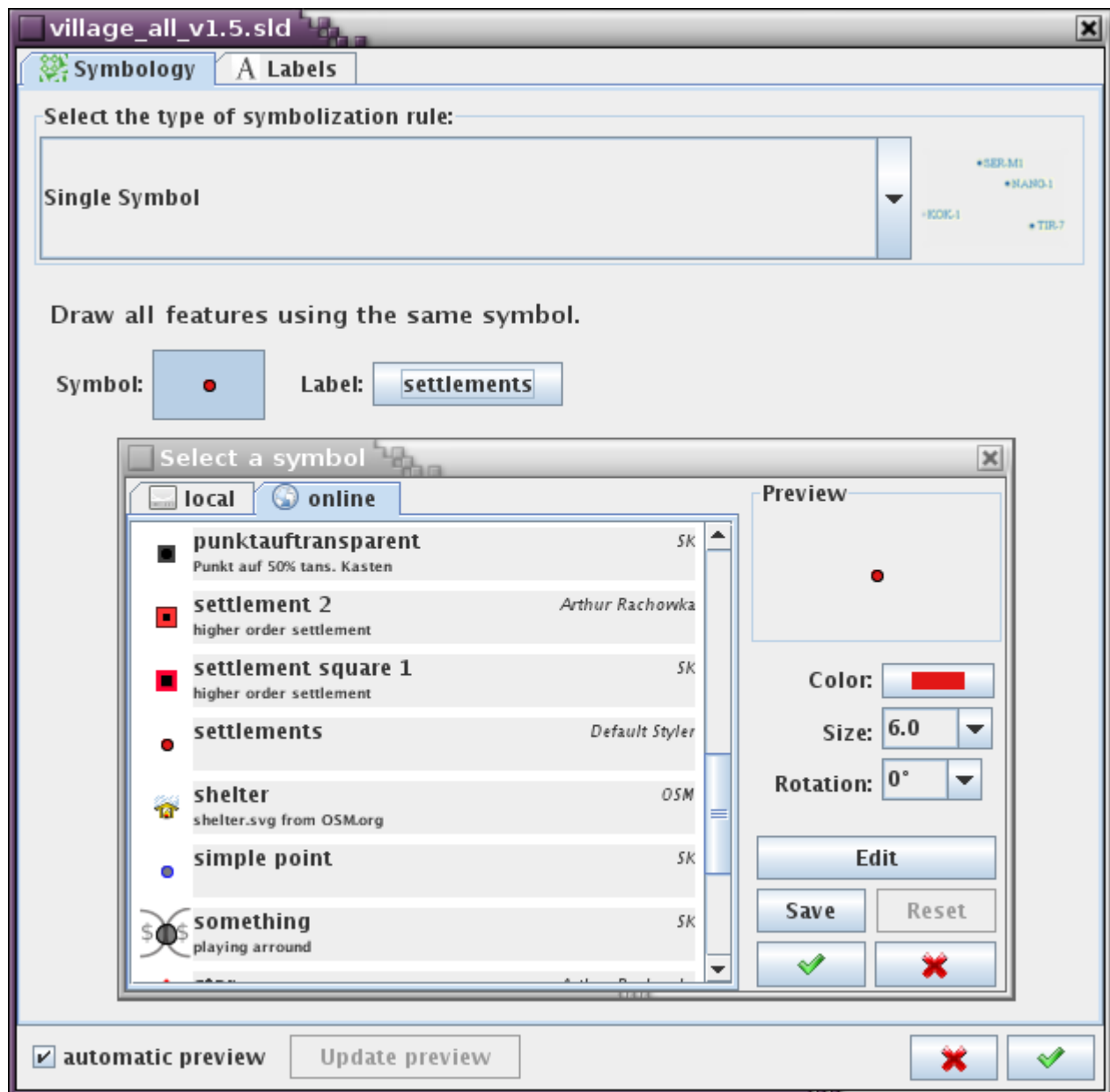


Click the tool-icon and choose *style* to open the styling dialog for the layer:



As we focus on text-labeling in this tutorial, let's skip any classification and just choose a single

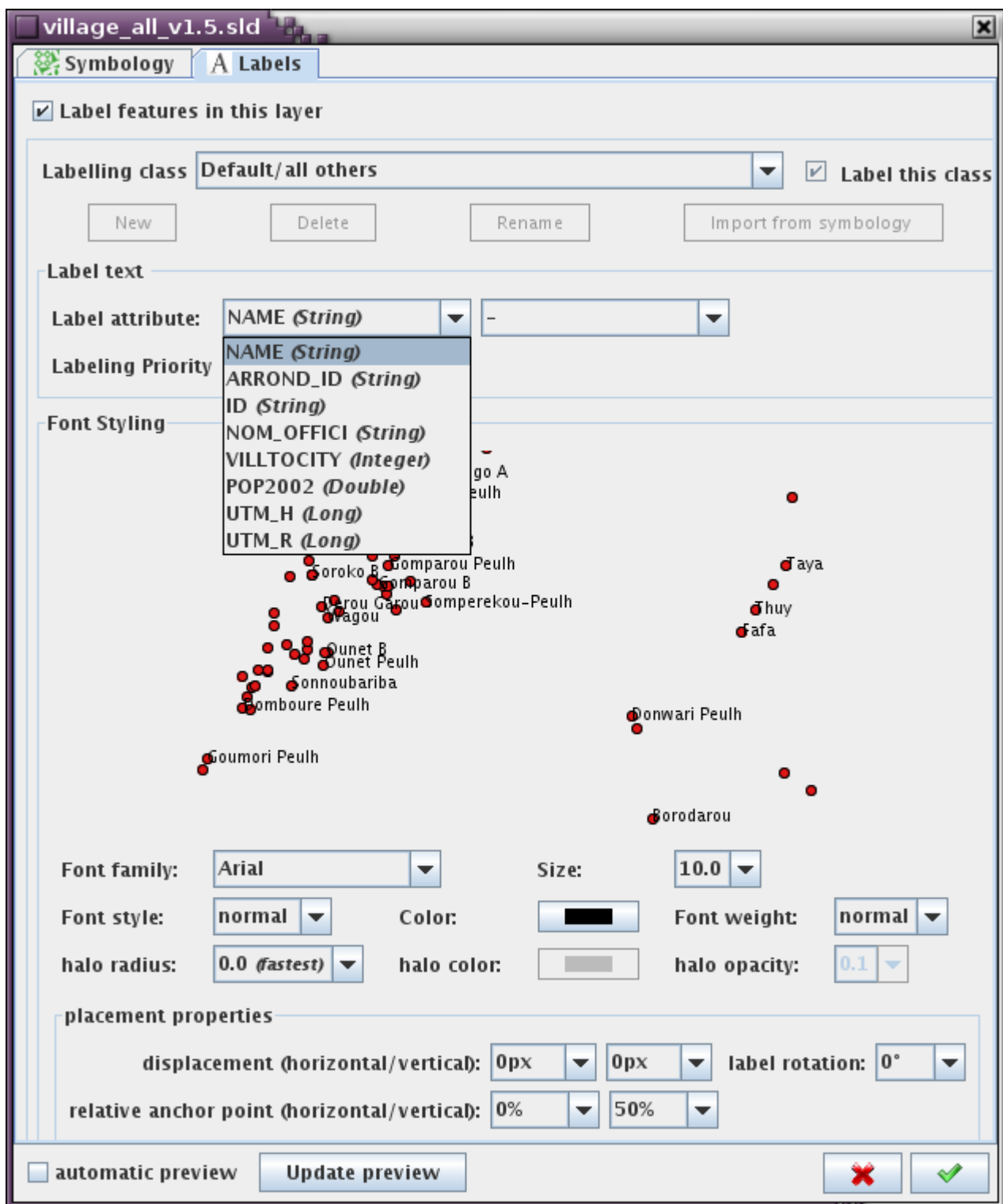
symbol for all points. Therefore we don't change the default *Single Symbol* selection and just click on the empty button. If you didn't save any symbols locally with AtlasStyler yet, you can select one of the symbols available in the *online*-tab. The online-tab allows you to browse all SLD and SVG symbols from the [openmapsymbols project](http://openmapsymbols.org). All symbols there are royalty-free and can be freely used for your maps:



Switch to the *Labels* tab:

Switch to the *Labels* tab and check the *Enable features in this layer* checkbox. The preview pane will automatically paint some example labels from the selected layer (To reflect all settings directly and to ensure reactivity while editing the SLD, the preview map only contains a subset of all features of the layer.).

First we select an appropriate attribute for labeling: In the case of our example layer of Benin settlements, this is the *NAME* column.



Let's have a look at the labels we see now: The text begins directly at the feature's center, which is ugly and hard to read - especially with bigger symbols. Hence we need to change the placement properties. *Placement* can be adjusted by setting the *absolute displacement* and the relative *anchor point* position.

The *anchor point*:

The *anchor point* is measured in percentage of the label's width and height - which directly depends on every feature's label text. Setting the horizontal and vertical anchor point to 50% will center the labels directly above the feature's coordinate. Such a setting is often used to center a label inside a

polygon.

Label attribute: NAME (String) -

Labeling Priority -

Font Styling

Font family: Arial Size: 14.0

Font style: normal Color: [black] Font weight: normal

halo radius: 0.0 (fastest) halo color: [grey] halo opacity: 0.1

placement properties

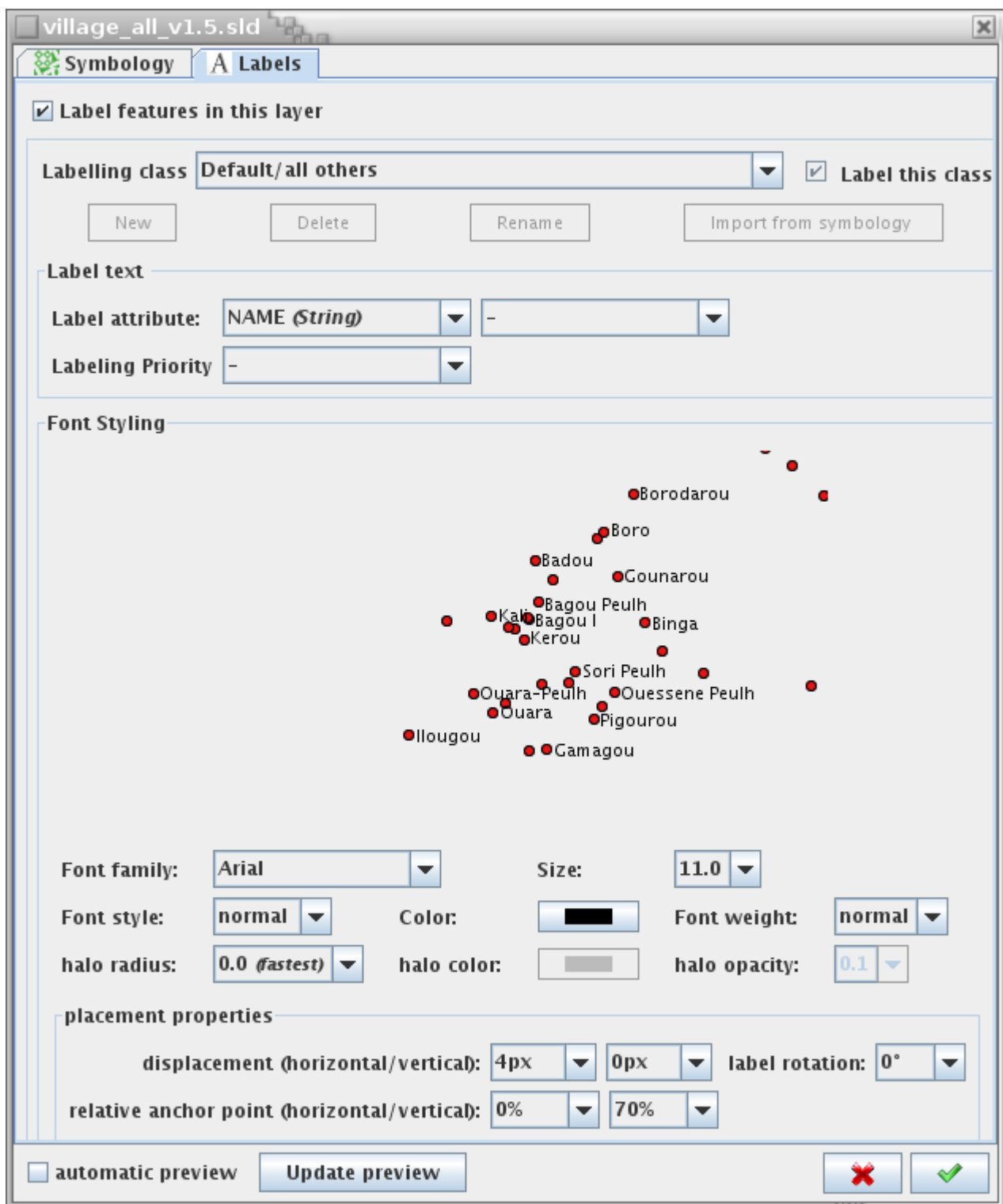
displacement (horizontal/vertical): 0px 0px label rotation: 0°

relative anchor point (horizontal/vertical): 50% 50%

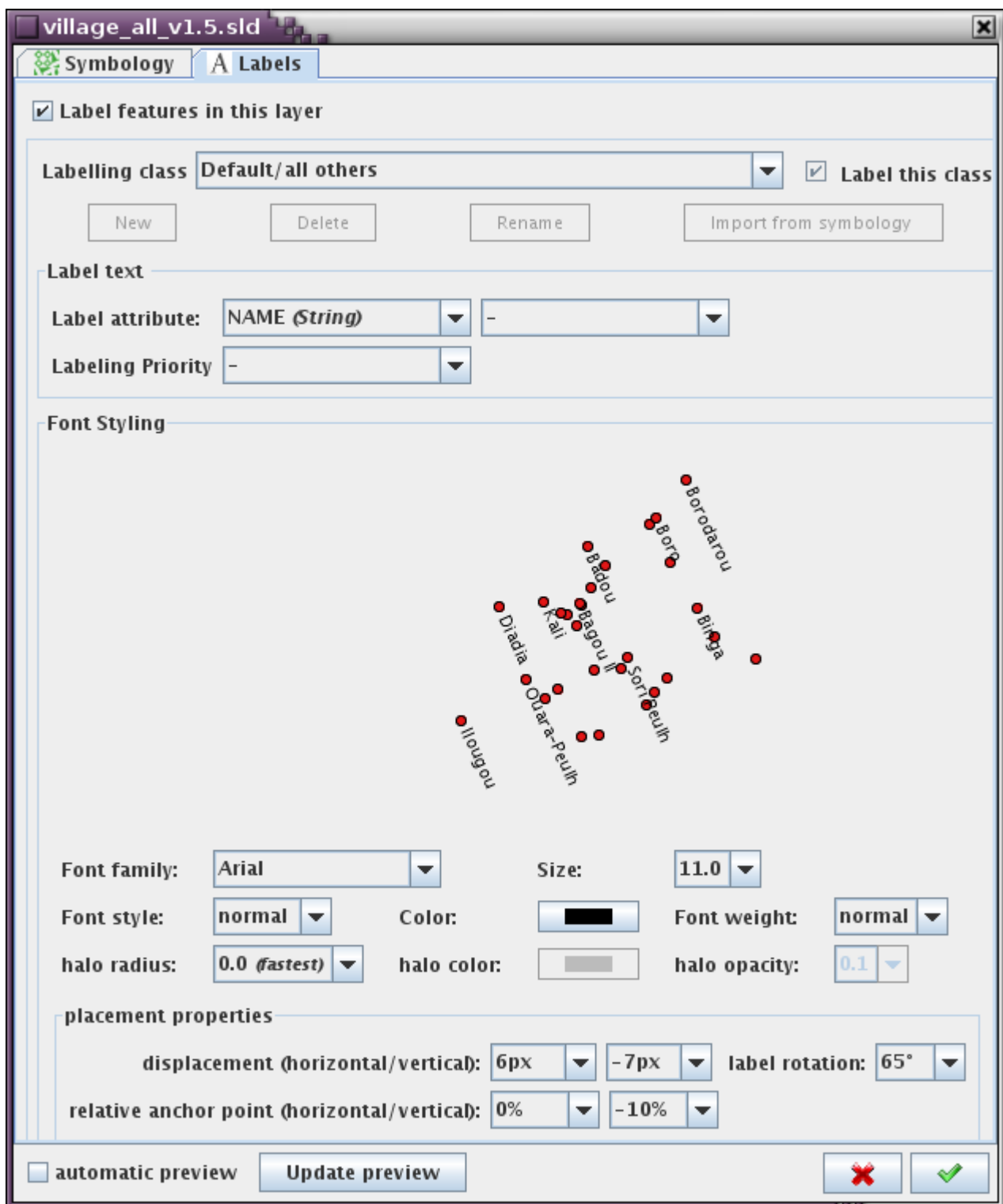
☐ automatic preview Update preview [X] [checkmark]

Absolute displacement:

Horizontal and vertical *displacements* are measured in absolute pixels and define a fixed gap between the feature's coordinates, and the anchor point of the label. If the point's symbol has a radius of 3 pixels, setting the horizontal displacement to 4 pixels will ensure that the labeling is not covering the symbol.

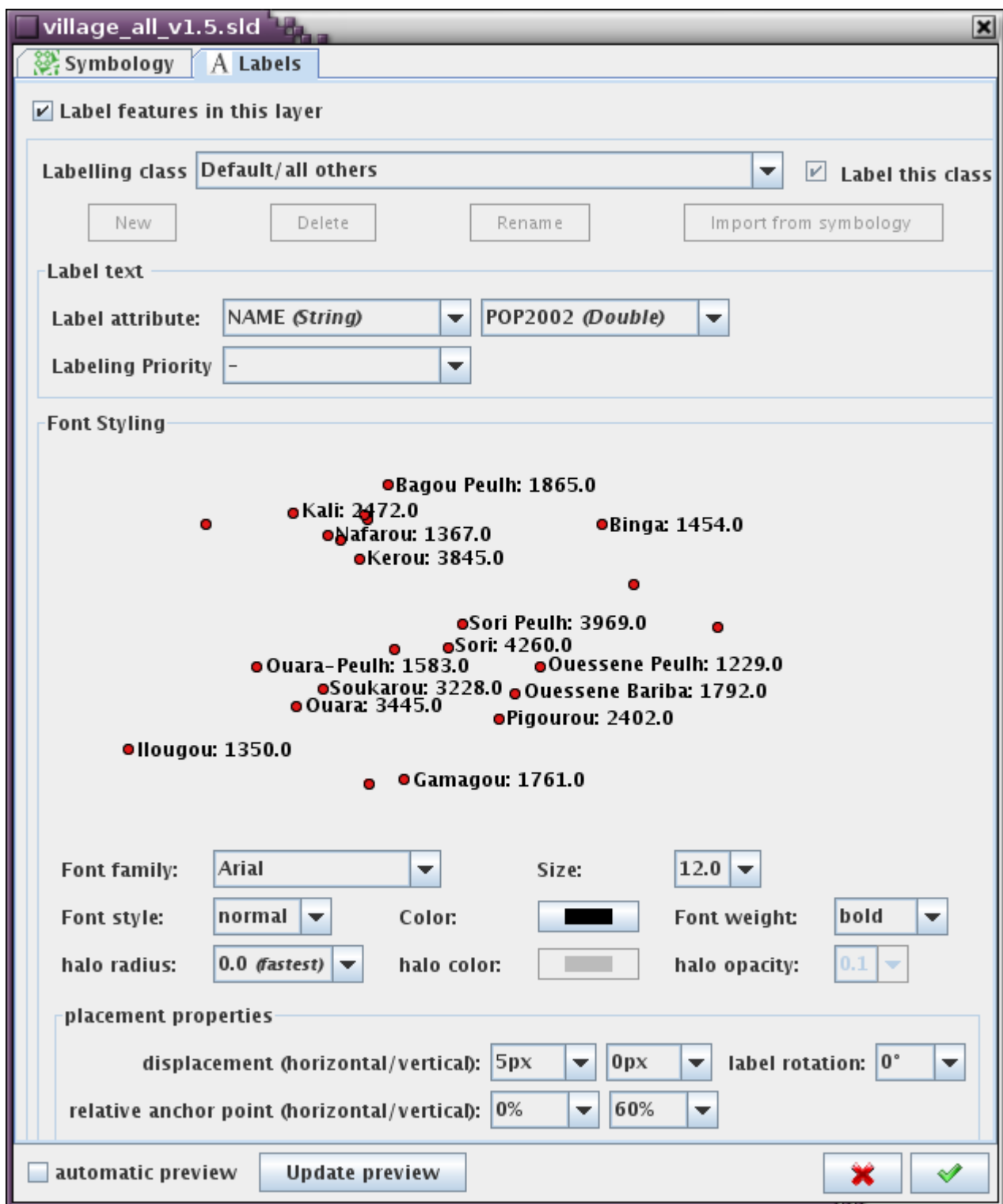


In combination with the *rotation* option, horizontal and vertical displacement allow us also to define pretty skewed labels:



Second labeling attribute:

Besides the placement options, the labeling panel also offers to select a second labeling attribute, which will print a second value behind the first label, separated by a colon. This setting is optional and only affects the features that have a not-null value in that field. The screenshot shows the population of each settlement next to its name:



The number of digits after the decimal point (.0) is dependent on the attribute's type. In the Benin settlements Shapefile, the *.dbf* file defines the column as *N,19,11*, which is a floating point number and is therefore presented with a digit after the decimal point.

To remove the pending ".0" we have to change the type of the attribute. This can be done by editing the *.dbf* file in [OpenCalc](#) (or use the [qgis Table Manager Plugin](#)). The idea is to copy the column and change its type from floating point number (*N,15,11*) to text (*C,10*).

village_all_v1.5.dbf - OpenOffice.org Calc

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Hilfe

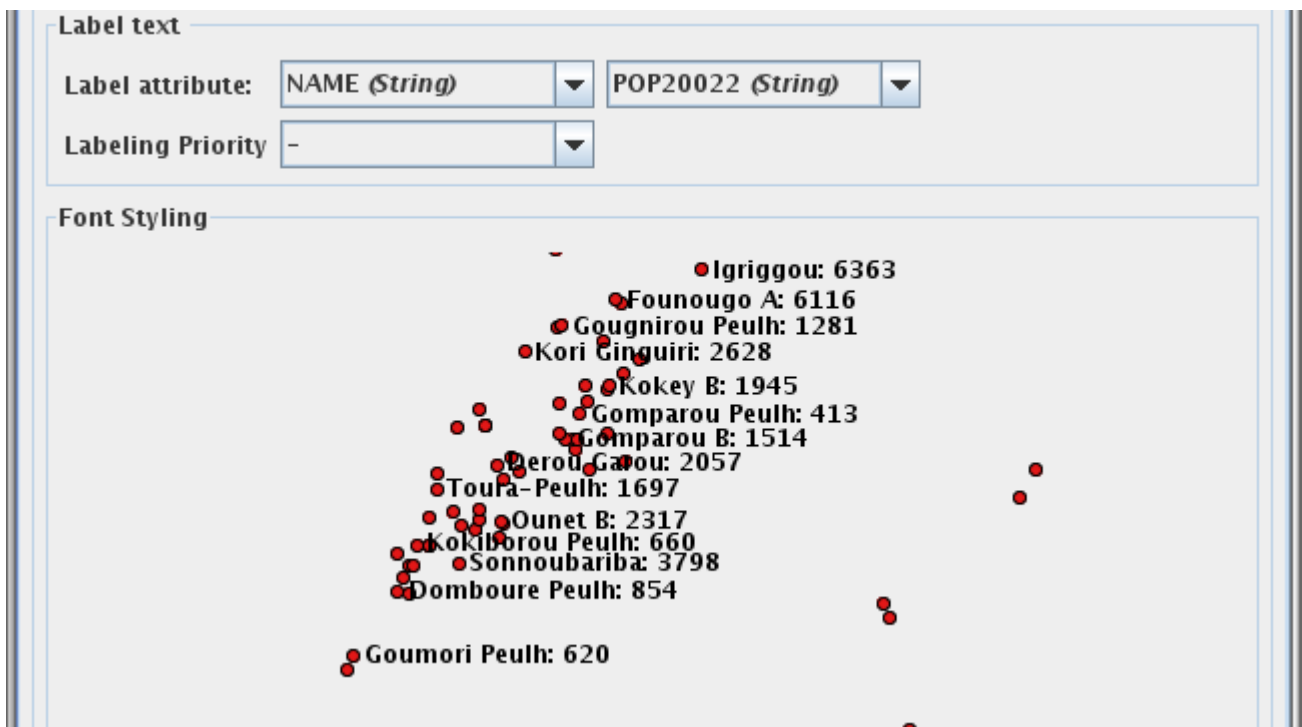
11:1048576 $f(x)$ Σ = POP20022,C,10

	E	F	G	H	I
1	UTM_H,N	ID,C,12	ARRON	POP2002,N,15,11	POP20022,C,10
2	1264324	01 1 01 01	01 1 01	236	236
3	1269577	01 1 01 02	01 1 01	6116	6116
4	1269017	01 1 01 03	01 1 01	4872	4872
5	1278331	01 1 01 04	01 1 01	2015	2015
6	1275878	01 1 01 05	01 1 01	1678	1678
7	1266048	01 1 01 06	01 1 01	1270	1270

Tabelle1

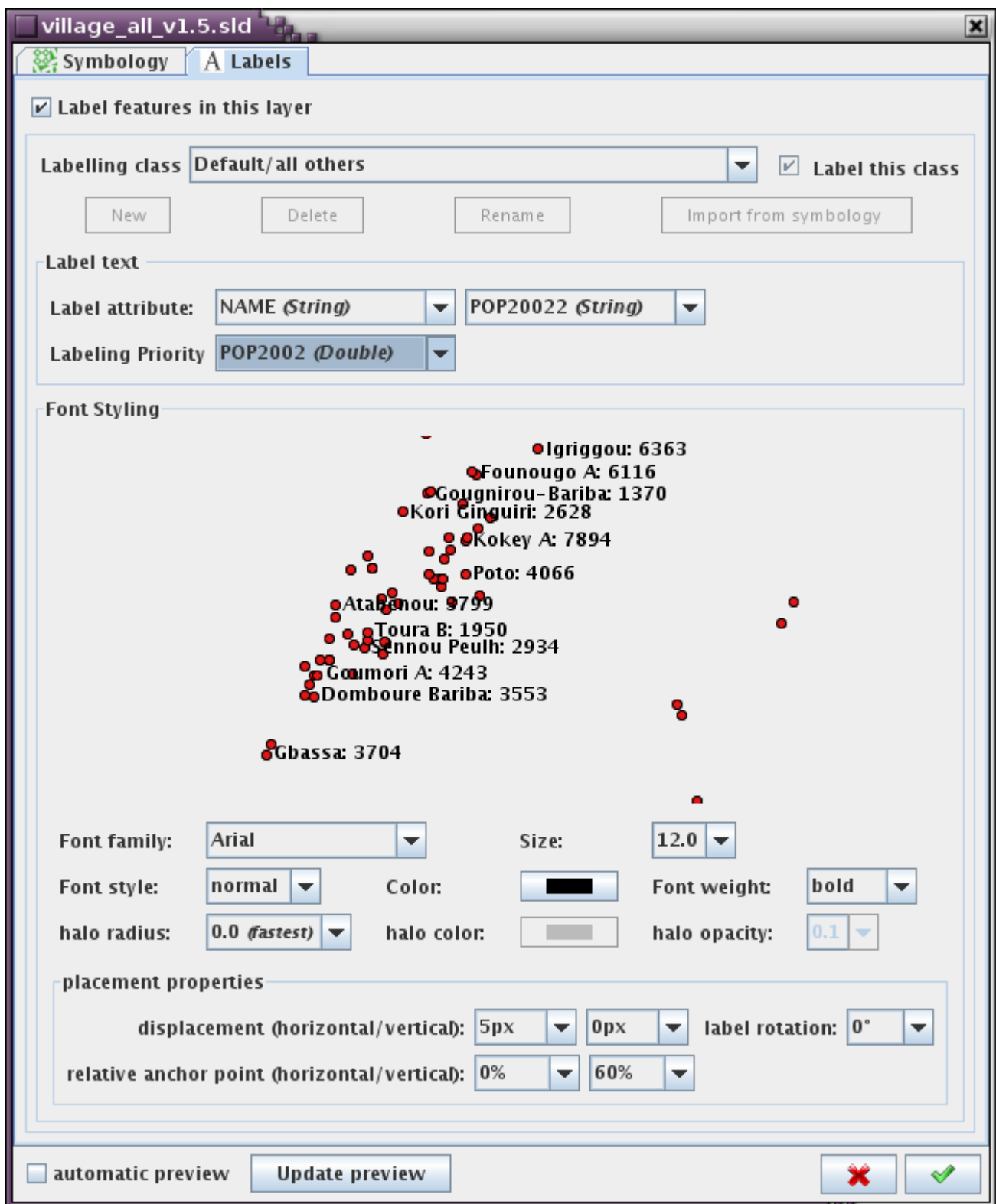
Tabelle 1 / 1 Standard STD Summe=7101182

After reloading the changed Shapefile in AtlasStyler, the population is shown correctly without a decimal digit:

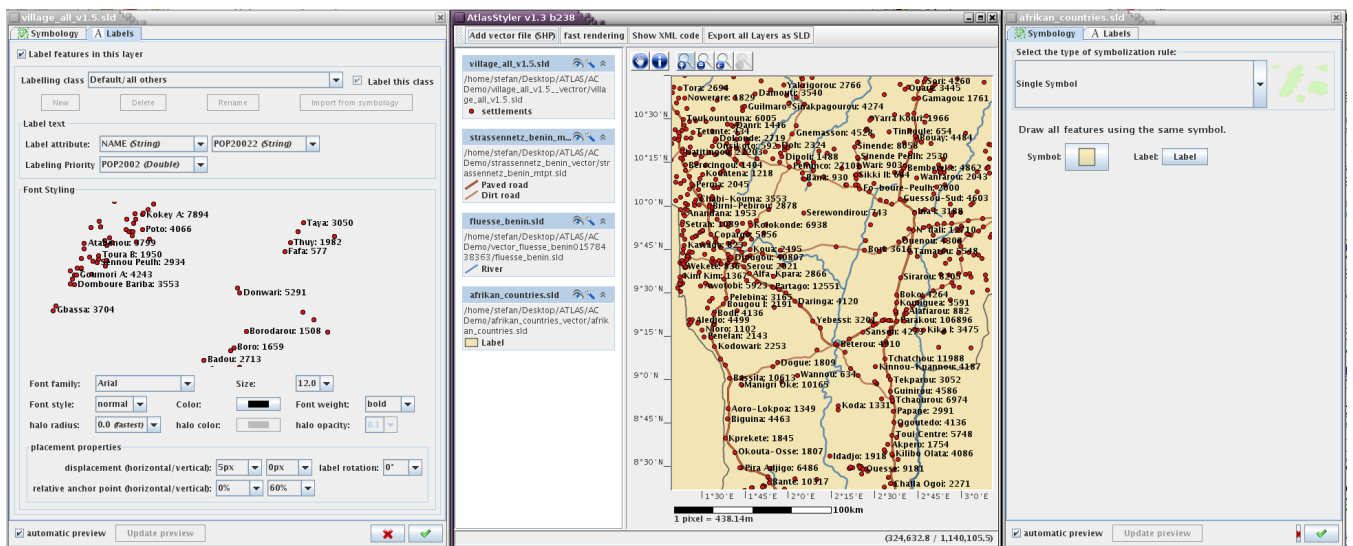


Labeling priority:

The third attribute selector is called *Labeling priority* and can be seen as a generalization hint. As we can clearly see in the screenshot not every point has a label. This is the trade-off to ensure that no labels collapse and hide each other. The rendering algorithm tries to make the best choice in which features to label, and which features to leave unlabeled. To influence this algorithm, a *labeling priority* column can be selected. Compare the previous and the next screenshot to see the different results. In the second map, settlements with higher population are labeled in favor of smaller settlements:

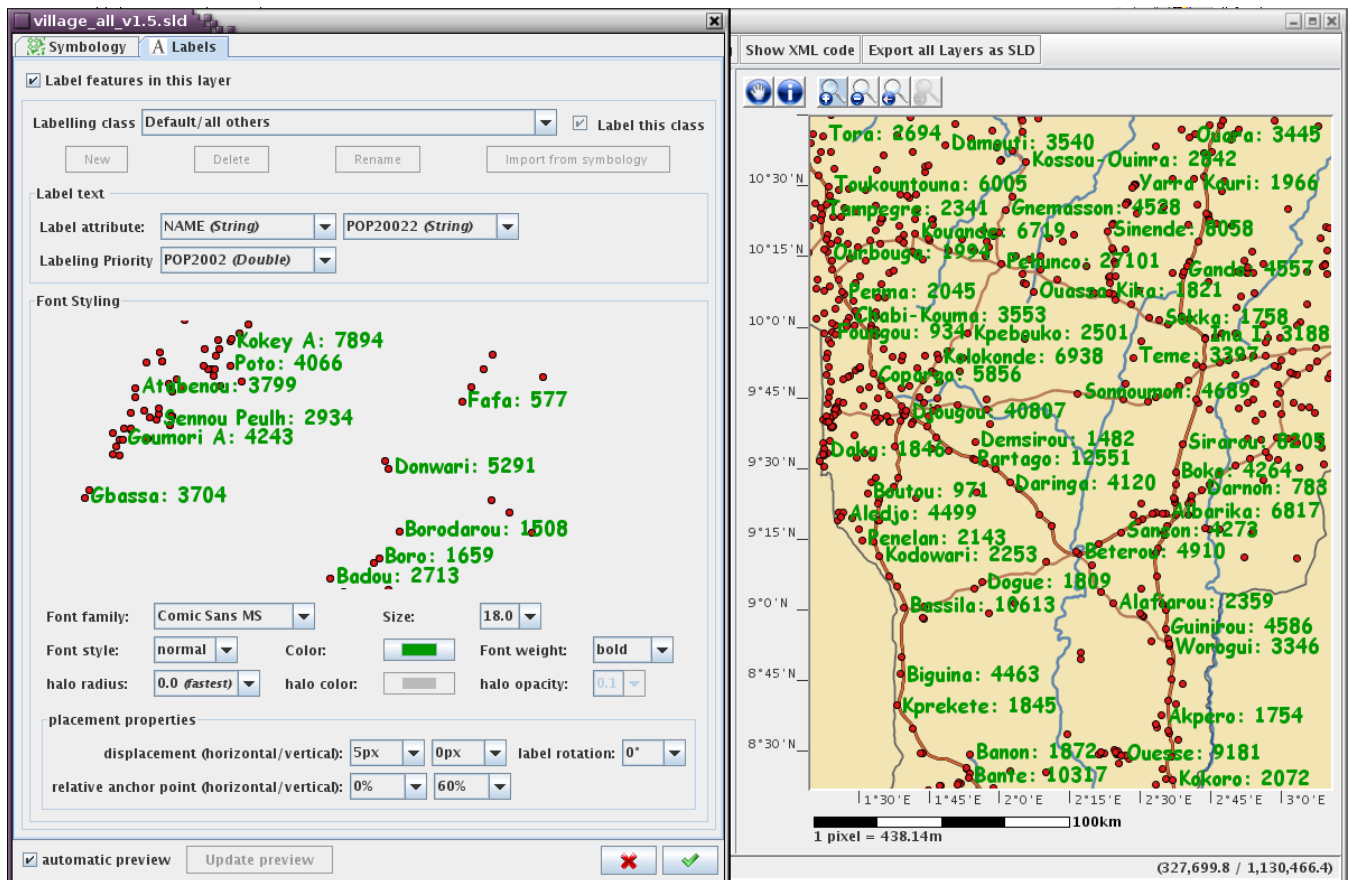


The last feature we will look at is the *halo effect*. To show its powers, let's add some more layers to the map. This can be done without closing the styling dialog for the settlements' layer. The next screenshot shows how to open styling dialogs which are open at the same time. If you activate the *automatic preview* checkbox, every change in the styling dialogs will automatically be reflected in the main map view. (Note: This should only be done if the datasets aren't too big.)



Halo-Effect:

As more layers are added to the map, reading the labels becomes harder again. Because SLD-styled maps are usually interactive, it's hard for the digital cartographer to select a font color: at some zoom level, a dark area might appear behind a text label, in another the background might be very bright. This makes choosing a good font color difficult. One option to ensure readability is to use a very bright font color and increase the font size - but this is usually not looking very good ;-):

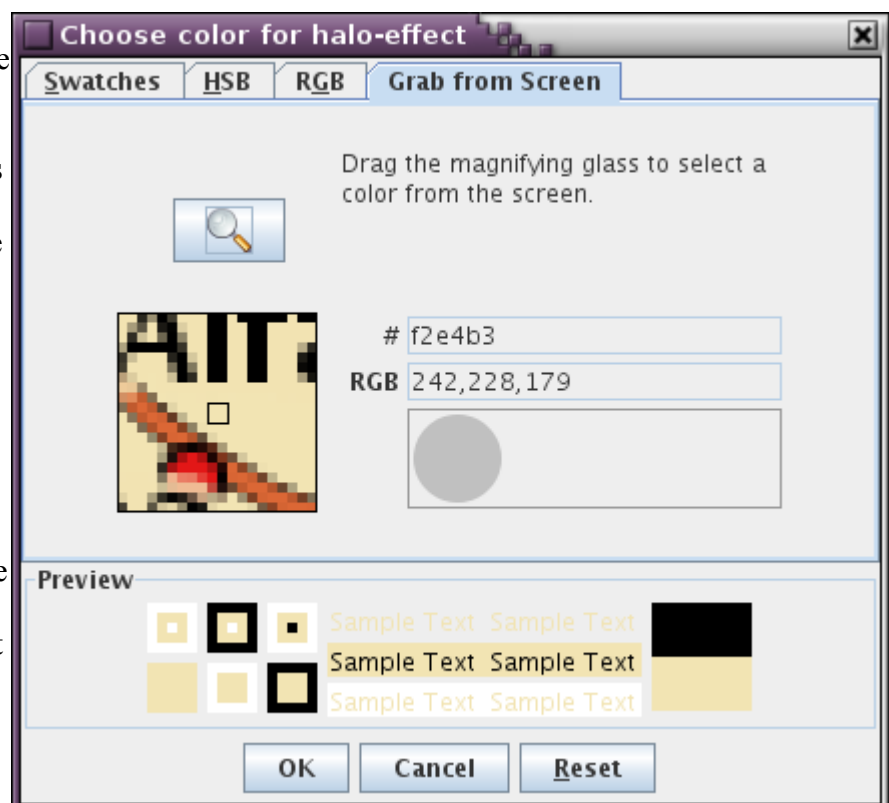


To avoid such an ugly labeling, the *halo-effect* draws an aura in a specific color around every text label. This ensures, that there is always a good contrast between the label and the background color.

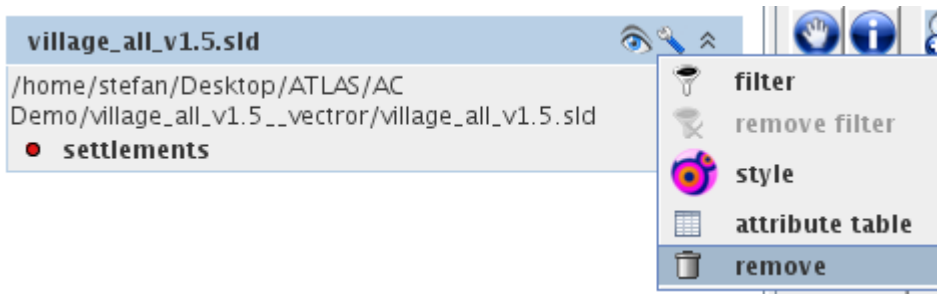
When a *halo-radius* greater than zero has been selected, the halo effect can be configured with color and opacity. An opacity of 0.8 makes the labels look smooth. When selecting the *halo-color*, you can use the new color picker: It allows to grab a color from the screen. Hint: Choose the map's main background color as the *halo-color*.

Save the SLD

To save the SLD to a file, close the dialog and choose *remove* from the tool menu. By default the *.sld* will be saved next to the data source *.shp* file. The

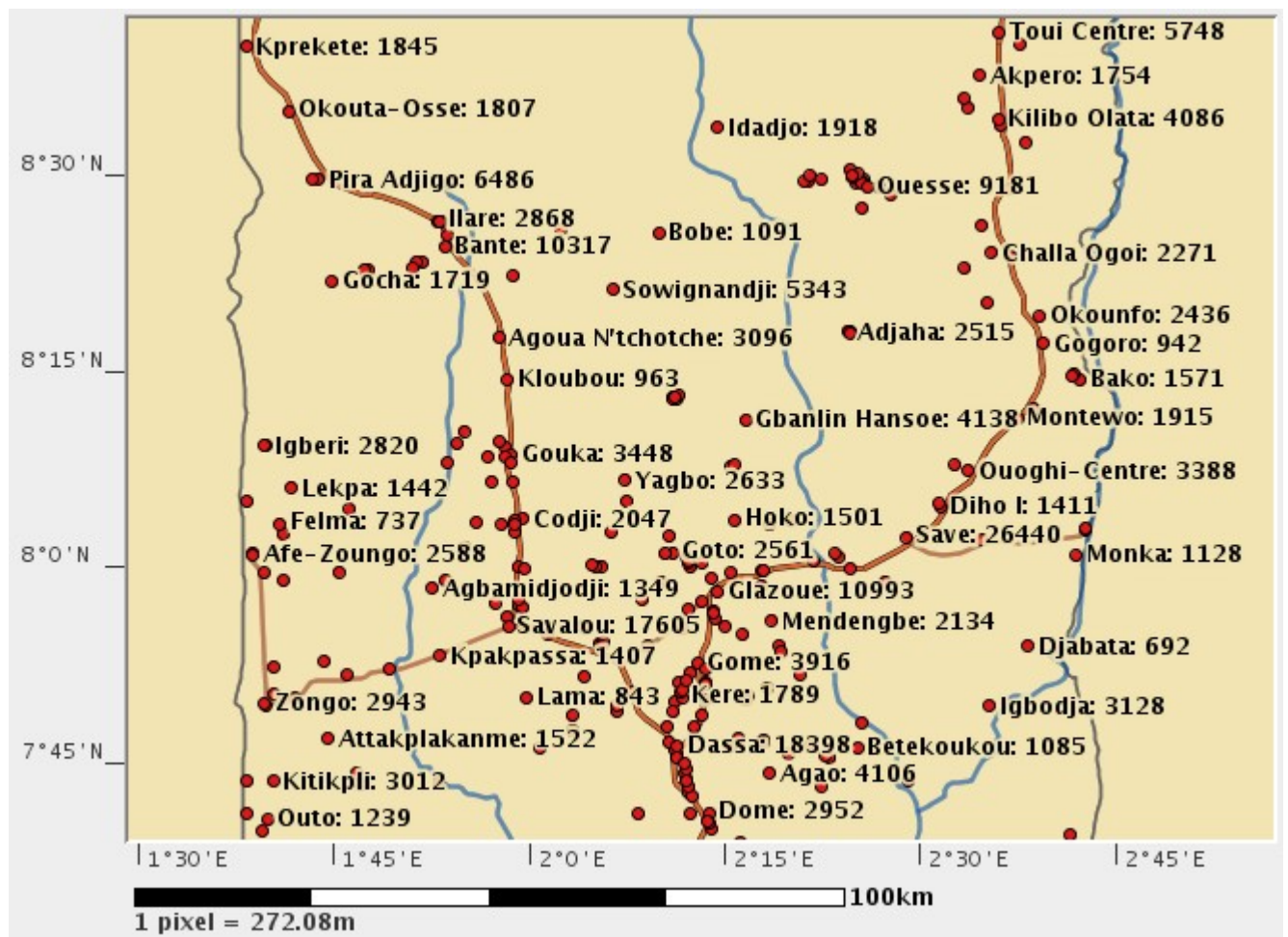


resulting.sld file for this project can be downloaded here: [benin_settlements.sld](#)



The end

This comes to the *end* of this [AtlasStyler/Geopublisher](#) tutorial about labeling. Of course the resulting map contains too many labels and should be further generalized with scale-dependent filters, but to demonstrate the automatic generalization capabilities this crowded layer was probably just right:



AtlasStyler & Geopublisher 1.3 will be released early October 2009. Feel free to comment on this blog or join our [mailinglist for Geopublisher and AtlasStyler users](#) with any questions or ideas.