



GeoServer - Creating MapThat Label layers

by David Crowther

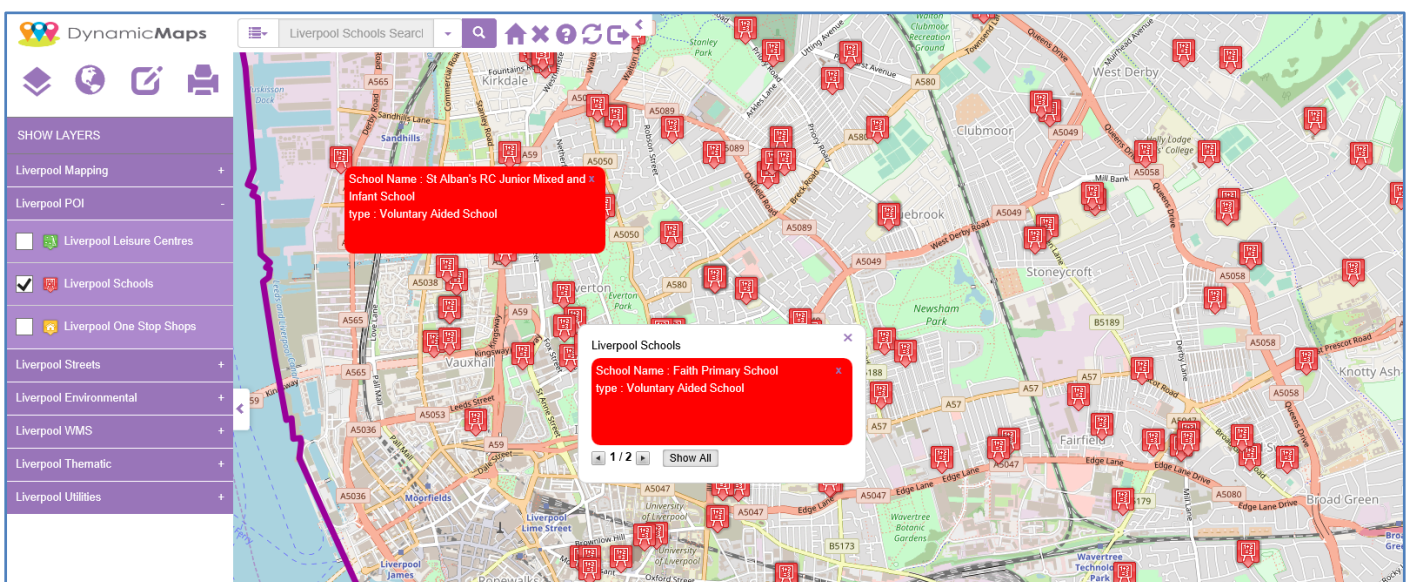
In this blog I would like to share my experience and some tips and tricks of how you can utilise **GeoServer** to create high impact Data Layers in your MapThat Instance. Utilising the power of Style Layer Descriptors (**SLD** files) we can create layers with options that;

- Reference a chosen image file – e.g. School Icon
- Label those points with a chosen attribute field
- Apply label options such as label placement, label font, label size and wrapping options.



Background to Requirement

Using the MapThat Admin Forms, we can create Point layers that utilise any given icon and then use the Information Bubble assigned to that layer to allow users to show information/attributes. For example, in our [TryMapThat](#) website the Schools layer uses a School icon and the Information bubble then reveals the School Name and School Type.





However, if you wish to show a Point layer that automatically has labels or maybe you wish to utilise a different icon for each individual school, then creating a WMS layer through GeoServer is the best way to do this. In this blog we will explore the steps that I undertook to create the example layer shown below.



Creating a WMS Layer in GeoServer

Using GeoServer I firstly created a new **Data Store** to the source data. In this case the data was a Shapefile, but it could have been in a SQL or PostGIS Database Table.

Edit Vector Data Source

Edit an existing vector data source

Shapefile
ESRI(tm) Shapefiles (*.shp)

Basic Store Info

Workspace *

Data Source Name *

Description

Enabled

Connection Parameters

Shapefile location *

[Browse...](#)





Once the Data Store was created I then published the **Layer**. Choosing a relevant **Name** and **Title**, and in the **Data Tab** choosing the **Coordinate Reference System** and then calculating the **Bounding Box** of the source data.

Coordinate Reference Systems

Native SRS
EPSG:27700 [EPSG:OSGB 1936 / British National Grid...](#)

Declared SRS
EPSG:27700 [EPSG:OSGB 1936 / British National Grid...](#)

SRS handling
Force declared

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
-84,702.61914736	-9,272.577651805	676,223.72419000	1,242,876.667023

[Compute from data](#)
[Compute from SRS bounds](#)

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
-10.918439432869	49.621782323681	3.1041356269080	61.067511008163

[Compute from native bounds](#)

Then in the **Publishing Tab** I ensured that the Style option was the default **Point** style.

WMS Settings

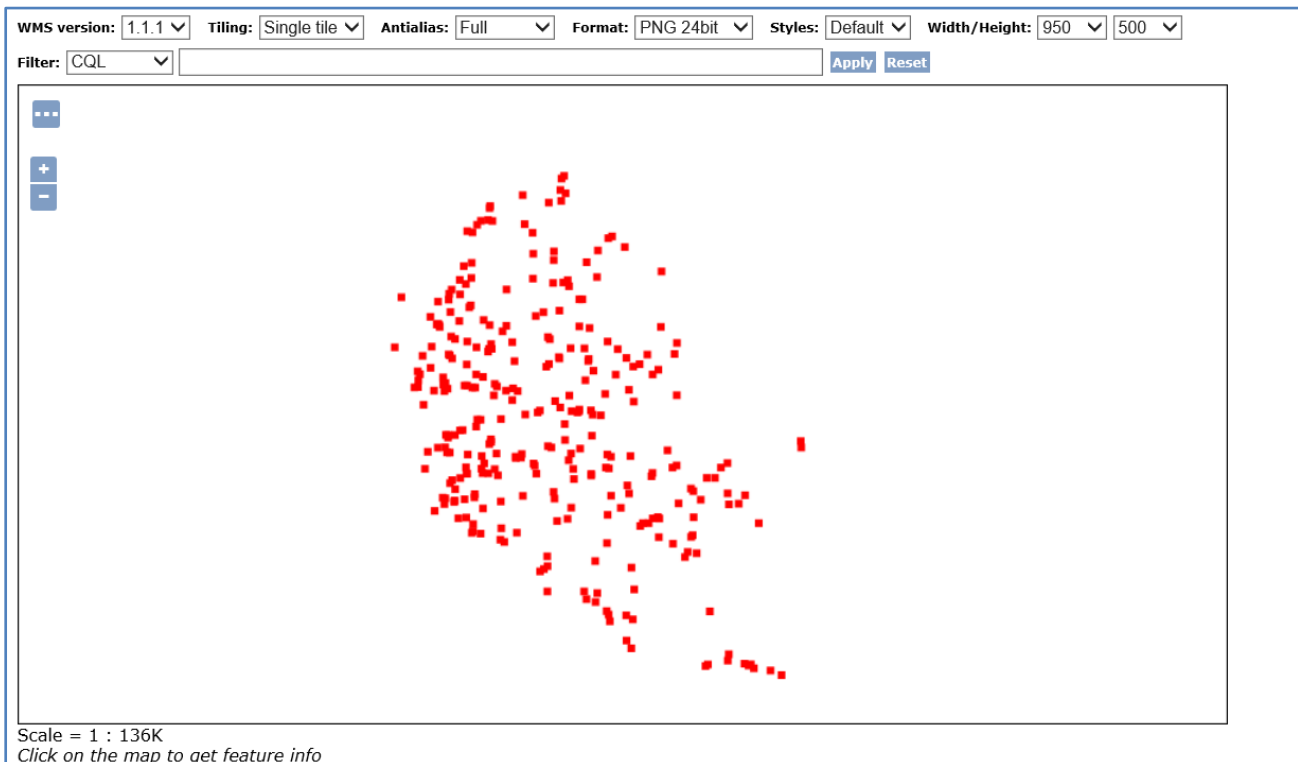
Layer Settings

Queryable
 Opaque

Default Style
point

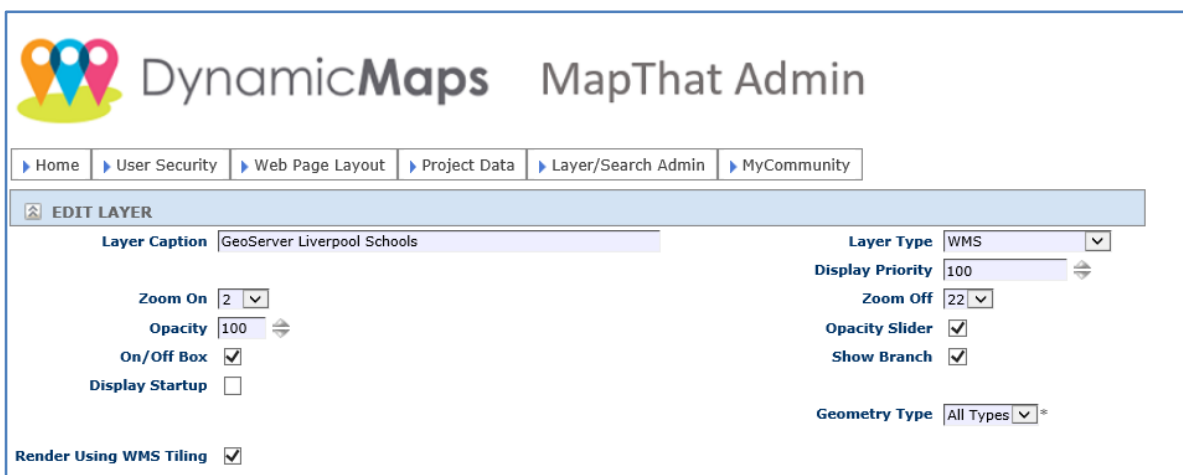
Once the layer was created I used the **Layer Preview** option to test this new WMS layer within **OpenLayers**.





Accessing the WMS Layer in MapThat

Now that I had a WMS layer being published from GeoServer, I could then create a layer to display in MapThat. Using the MapThat Admin Forms I created the following layer, setting the **Layer Type** to be **WMS**.





Setting the SRID to be **BNG (27700)** and copying in the WMS URL from OpenLayers into the **URL LINK** window. In order for the layer to render within MapThat, you simply edit the end of the OpenLayers URL to use the value of **format=image/png**.

Layer Icon 1 **** Please Select ****

Bubble on click

SRID 27700 - OSGB 1936 / British National Grid

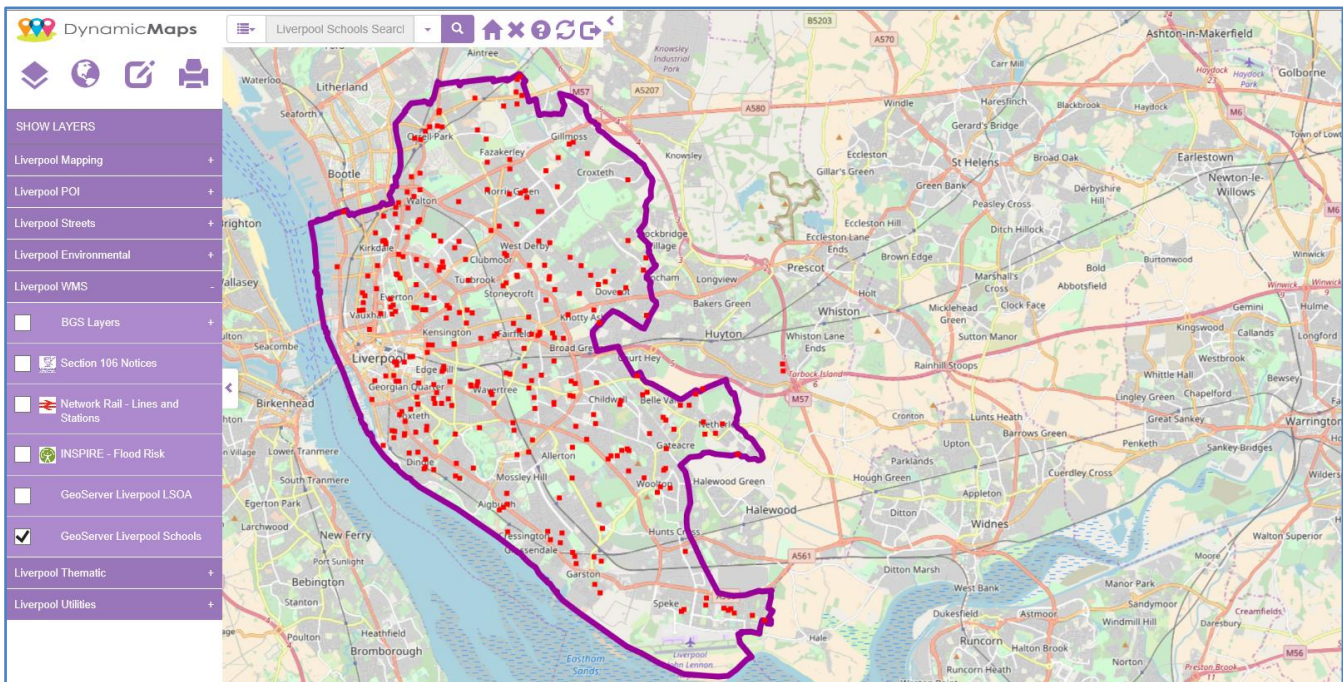
OS Data Source

URL Link `http://try.mapthat.co.uk:8080/geoserver/DynamicMaps/wms?service=WMS&version=1.1.0&request=GetMap&layers=DynamicMaps:TryMapThat_liverpoolschools&styles=&bbox=-84702.61914736108,-9272.577651805477,676223.7241900009,1242876.667023777&width=466&height=768&srs=EPSG:27700&format=image/png`

Get Capabilities

Mapthat Compatible

Having now added this new WMS layer to be revealed in the relevant **Tree Header**, I then had a new WMS layer within my MapThat.





Using SLD – to choose an Icon

Next, I started to explore using Style Layer Descriptors (SLD) to apply styling to this new WMS layer. To do this, in GeoServer I created a new **STYLE** file. Giving the Style a relevant name and inserting it into our DynamicMaps **Workspace**.

New style

Type a new style definition, or use an existing one as a template, or upload a ready made style from your file system. The editor can provide syntax highlighting and automatic formatting. Click on the "validate" button to verify the style is a valid style document.

Data

Style Data

Name

Workspace

Format

Style Content

Generate a default style

Copy from existing style

Upload a style file

Legend

Legend

[Preview legend](#)

There are many sources that you can utilise to create SLD's but as a starting point I would recommend the online **GeoServer SLD Cookbook** –

<http://docs.geoserver.org/stable/en/user/styling/sld/cookbook/>

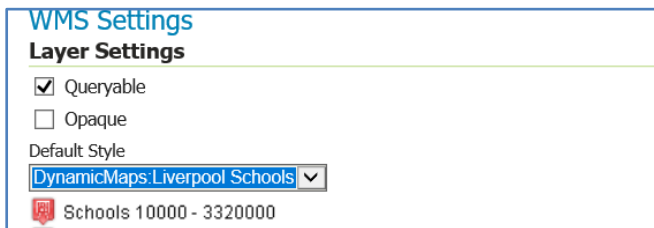
To display the School points using an icon from your server (Schools.png), you simply need to add a **Location** into the SLD, using a **xlink** reference;

```
14 <Title>Fire</Title>
15 <MinScaleDenominator>10000</MinScaleDenominator>
16 <MaxScaleDenominator>3320000</MaxScaleDenominator>
17 <PointSymbolizer>
18 <Graphic>
19 <ExternalGraphic>
20 <OnlineResource
21 <xlink:type="simple"
22 <xlink:href="file:///F:/DynamicMaps/MapThatProjects/MapThatIcons/PooleIcons/school.png"/>
23 </OnlineResource>
24 </ExternalGraphic>
25 <Size>24</Size>
26 </Graphic>
27
```

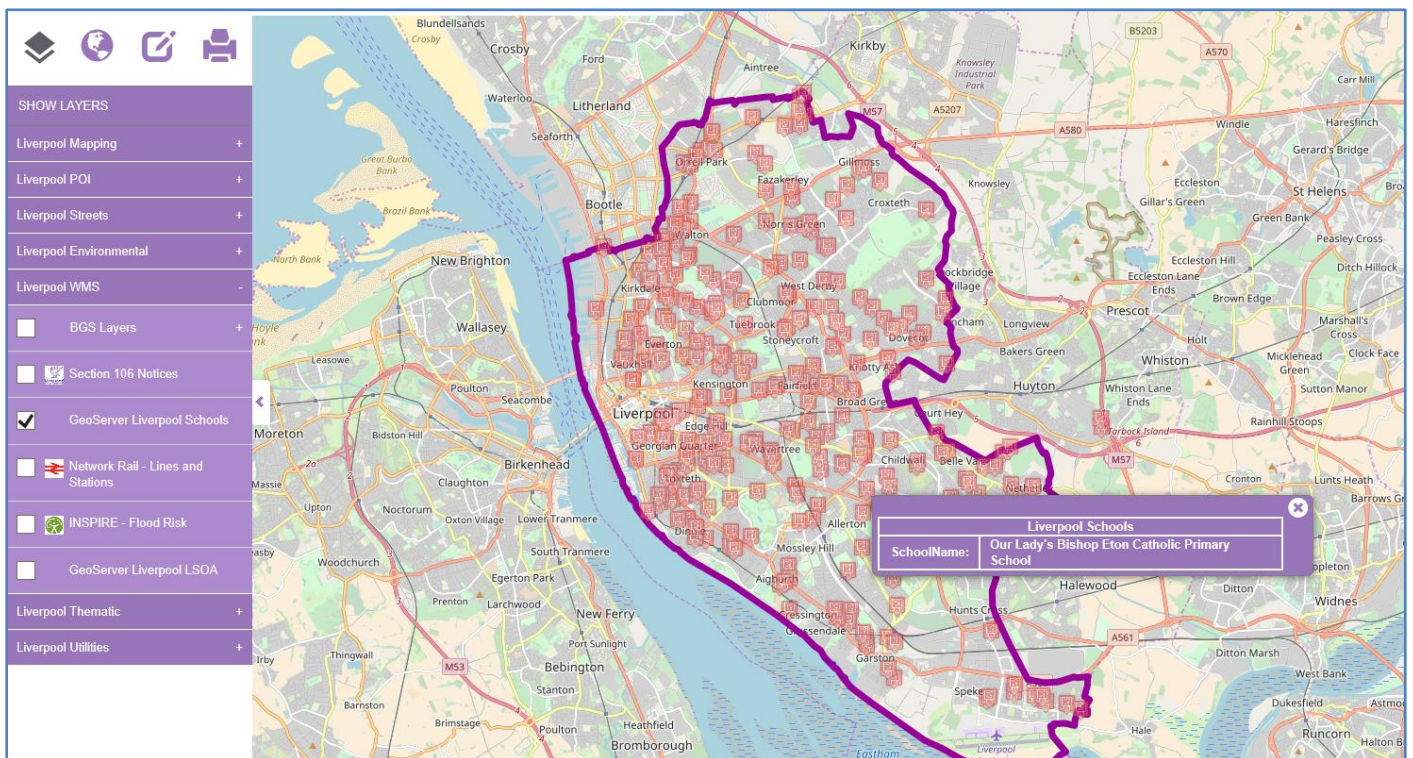




Having created this new STYLE file, I then updated the **Publishing** options for the Schools layer, to now use this new **Style**.



As soon as the new STYLE was applied to my WMS layer, the layer within MapThat now updated and used the Schools icon to represent the location of each School.



Using SLD – to apply a label

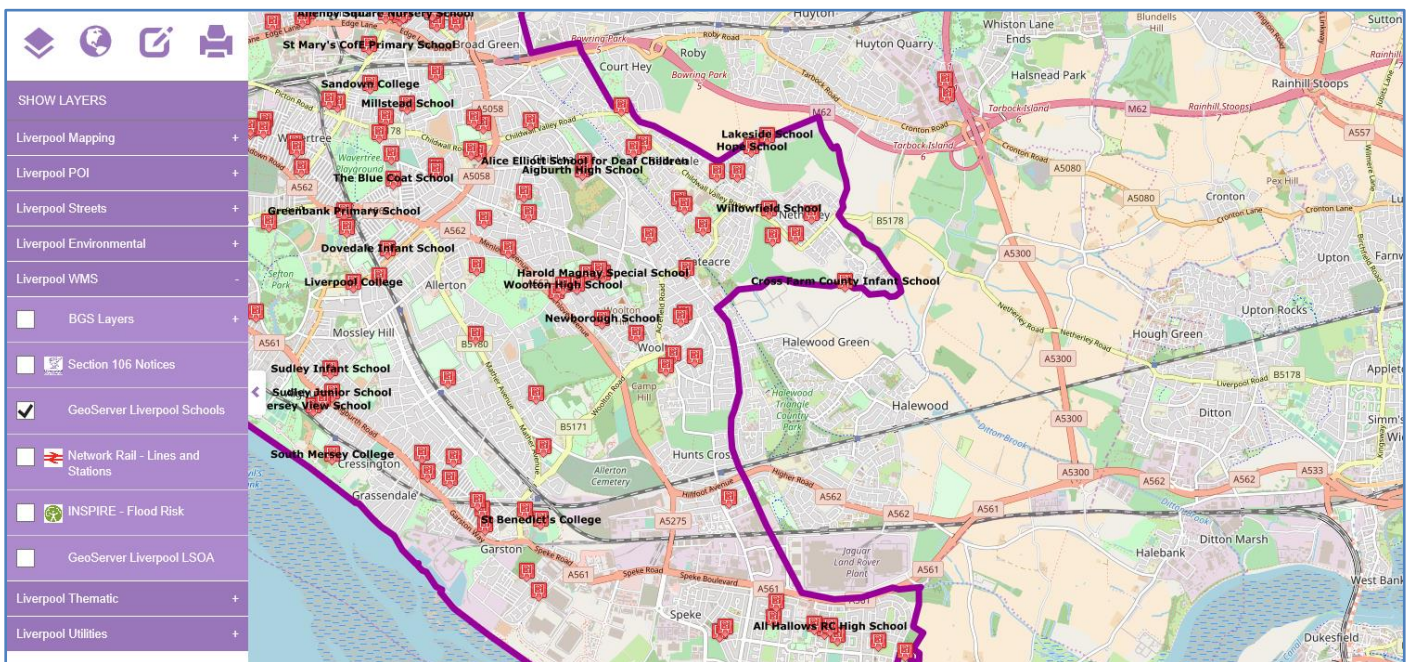
Having successfully got the WMS layer to display using the referenced icons, I then wanted to use the power of SLD files to apply **labels** automatically to each feature. In my first test I managed to create a simple black label using the **School Name** (establishm) field.



To do this I updated the SLD file to add the following text using a **TextSymboliser**:

```
Style Editor
12pt
24 </ExternalGraphic>
25 <Size>24</Size>
26 </Graphic>
27
28 </PointSymbolizer>
29 <TextSymbolizer>
30 <Label>
31 <ogc:PropertyName>establishm</ogc:PropertyName>
32 </Label>
33 <Font>
34 <CssParameter name="font-family">Verdana</CssParameter>
35 <CssParameter name="font-size">11</CssParameter>
36 <CssParameter name="font-style">normal</CssParameter>
37 <CssParameter name="font-weight">bold</CssParameter>
38 </Font>
39 <LabelPlacement>
```

When this was displayed in the map I now had automatic labels for this WMS layer. However, as you can see many overlapping labels were not being shown and the label text stretched in one long line.



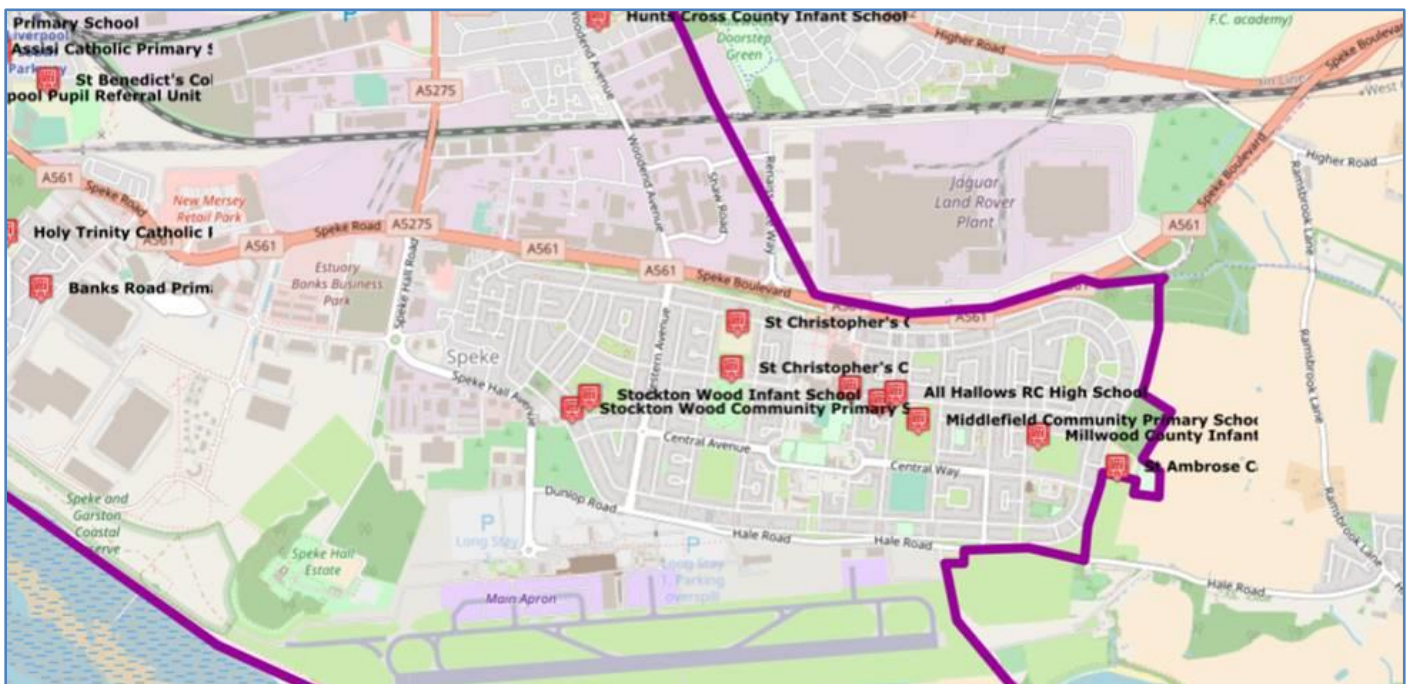
Using SLD – to Add More Labels

By amending the SLD using the **VendorOption - partials** I was then able to increase the number of labels that were being shown.



```
41 0.0
42 </AnchorPointX>
43 <AnchorPointY>
44 0.5
45 </AnchorPointY>
46 </AnchorPoint>
47 <Displacement>
48 <DisplacementX>
49 20
50 </DisplacementX>
51 <DisplacementY>
52 0
53 </DisplacementY>
54 </Displacement>
55 </PointPlacement>
56 </LabelPlacement>
57 <VendorOption name="partials">true</VendorOption>
58 </TextSymbolizer>
59 </Rule>
60
61 </FeatureTypeStyle>
62 </UserStyle>
```

The labels now showed for **even more** School locations.



Using SLD – to Wrap Labels

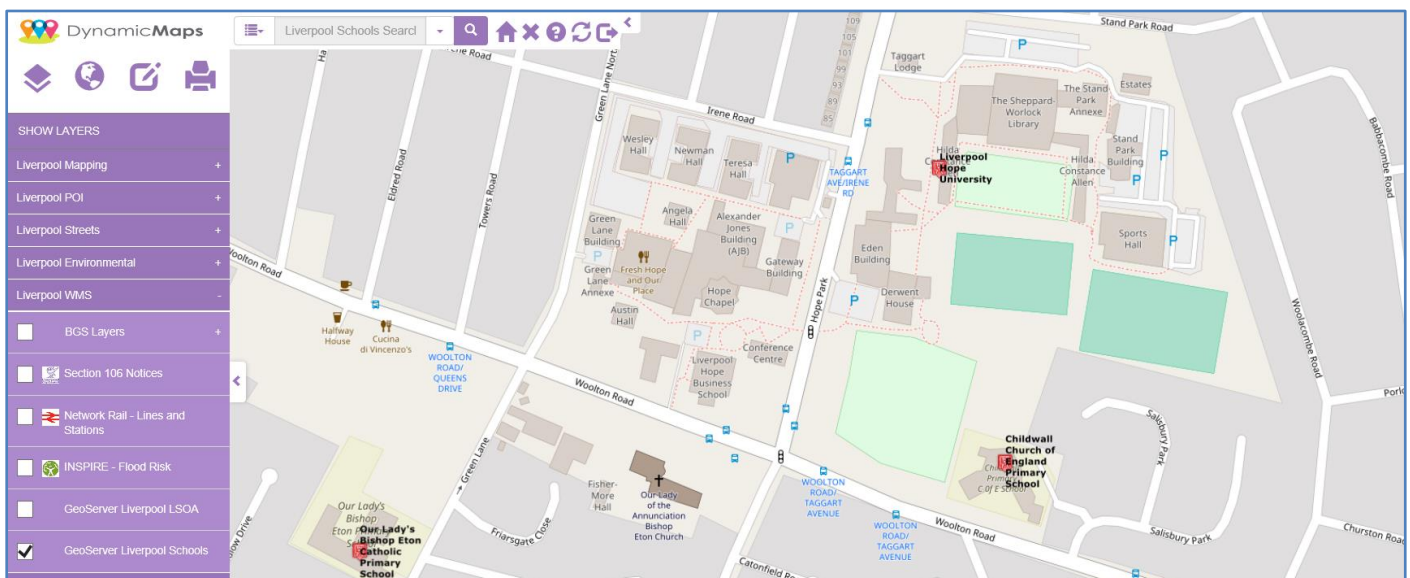
By amending the SLD further, this time using the **VendorOption - autowrap** I was then able to wrap long School Names onto separate lines. With the **value 75** indicating the maximum size for a word before that is cut off and wrapped.





```
51 <DisplacementY>  
52 0  
53 </DisplacementY>  
54 </Displacement>  
55 </PointPlacement>  
56 </LabelPlacement>  
57 <VendorOption name="partials">true</VendorOption>  
58 <VendorOption name="autoWrap">75</VendorOption>  
59 </TextSymbolizer>  
60 </Rule>  
61  
62 </FeatureTypeStyle>  
63 </UserStyle>  
64 </NamedLayer>  
65 </StyledLayerDescriptor>  
66
```

The labels now appeared **wrapped** for the School Names.



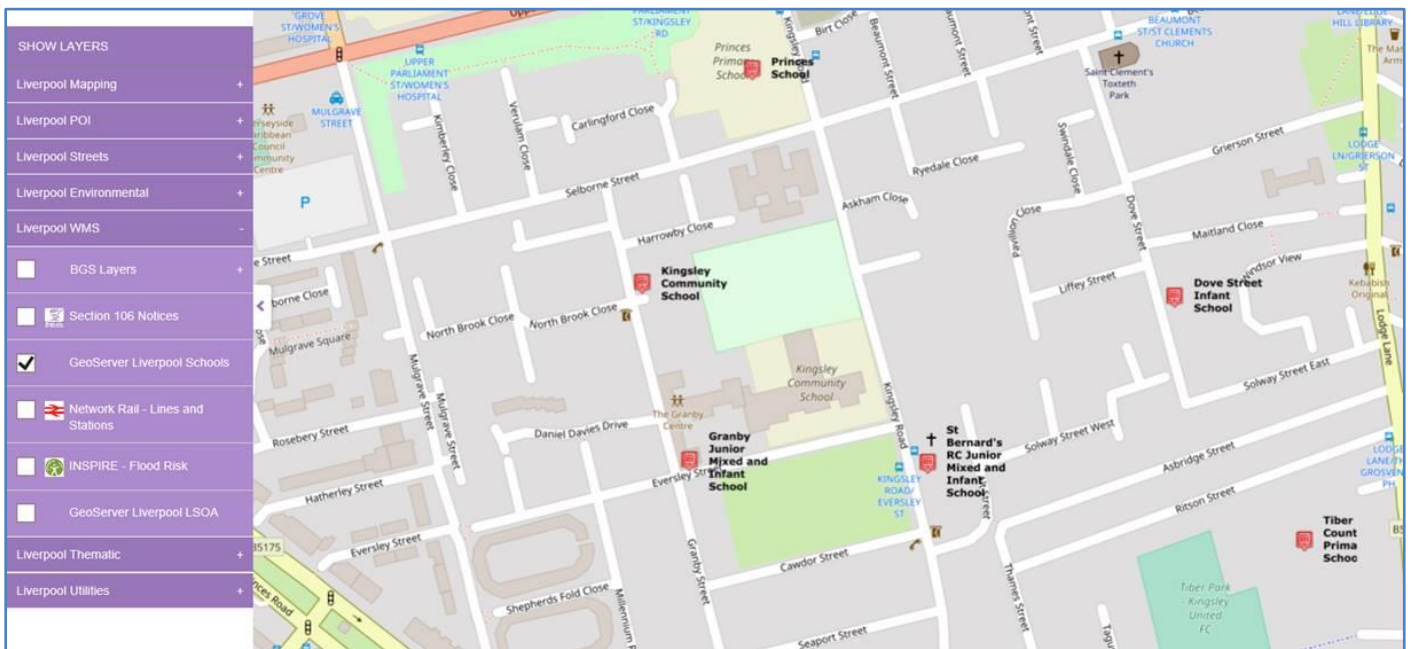
Using SLD – to Offset and Anchor Labels

As the labels were still centered over the School icon, the School was being obscured slightly, so I then decided that it would be better if the labels were offset and positioned away from the School point. In this case I updated the SLD to add **Label Placement** options for anchoring the label.



```
Style Editor
12pt
32 </Label>
33 <Font>
34 <CssParameter name="font-family">Verdana</CssParameter>
35 <CssParameter name="font-size">11</CssParameter>
36 <CssParameter name="font-style">normal</CssParameter>
37 <CssParameter name="font-weight">bold</CssParameter>
38 </Font>
39 <LabelPlacement>
40 <PointPlacement>
41 <AnchorPoint>
42 <AnchorPointX>
43 0.0
44 </AnchorPointX>
45 <AnchorPointY>
46 0.5
47 </AnchorPointY>
48 </AnchorPoint>
49 </PointPlacement>
50 </LabelPlacement>
51
52 </TextSymbolizer>
53 </Rule>
54 <Rule>
55 <Title>Schools below 10000</Title>
56 <MaxScaleDenominator>10000</MaxScaleDenominator>
```

In MapThat the labels then appeared **offset** from the School icon.



Using SLD – to implement zoom based labelling

Finally I decided that the School Name labels would look better if they were autowrapped at a low zoom level, but at higher zoom levels, the initial School Name (unwrapped) would look better. So I then edited the SLD using the **Min and Max Scale Denominator** settings. To do this I simply created



two RULES within the SLD, one that adds the style and labels between 1:10,000 and 1:332,000, which used a standard black label with no wrapping, but offset and placement options.

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
3   xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.opengis.net/sld http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd">
5   <NamedLayer>
6     <Name>Schools</Name>
7     <UserStyle>
8       <Title>Schools</Title>
9       <Abstract>A sample style that gives points a School icon</Abstract>
10
11     <FeatureTypeStyle>
12       <!--FeatureTypeName>Feature</FeatureTypeName-->
13       <Rule>
14         <Title>Schools</Title>
15         <MinScaleDenominator>10000</MinScaleDenominator>
16         <MaxScaleDenominator>332000</MaxScaleDenominator>
17         <PointSymbolizer>
18           <Graphic>
19             <ExternalGraphic>
20               <OnlineResource
21                 xlink:type="simple"
22                 xlink:href="file:///F:/DynamicMaps/MapThatProjects/MapThatIcons/PooleIcons/school.png"/>
23             </OnlineResource>
24             <Format>image/png</Format>
25           </ExternalGraphic>
26           <Size>24</Size>
27         </PointSymbolizer>
28       </Rule>
29     </FeatureTypeStyle>
30   </NamedLayer>
31 </StyledLayerDescriptor>
```

And then another Rule that applied the style and labels below 1:10,000,

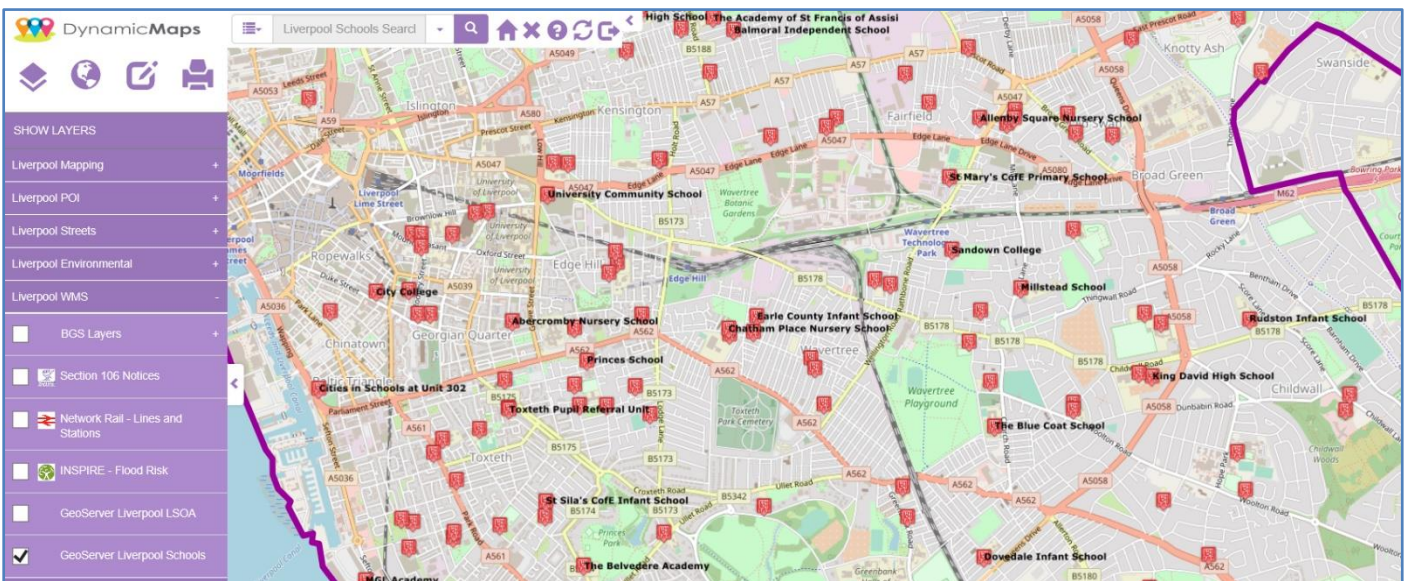
```
48 </AnchorPoint>
49 </PointPlacement>
50 </LabelPlacement>
51
52 </TextSymbolizer>
53 </Rule>
54 <Rule>
55   <Title>Schools below 10000</Title>
56   <MaxScaleDenominator>10000</MaxScaleDenominator>
57   <PointSymbolizer>
58     <Graphic>
59       <ExternalGraphic>
60         <OnlineResource
61           xlink:type="simple"
62           xlink:href="file:///F:/DynamicMaps/MapThatProjects/MapThatIcons/PooleIcons/school.png"/>
63         </OnlineResource>
64         <Format>image/png</Format>
65       </ExternalGraphic>
66     </Graphic>
67   </PointSymbolizer>
68 </Rule>
```

which still includes the settings we applied for offsetting, anchoring and wrapping labels.... Where I also changed the School Name label to be **Blue**.



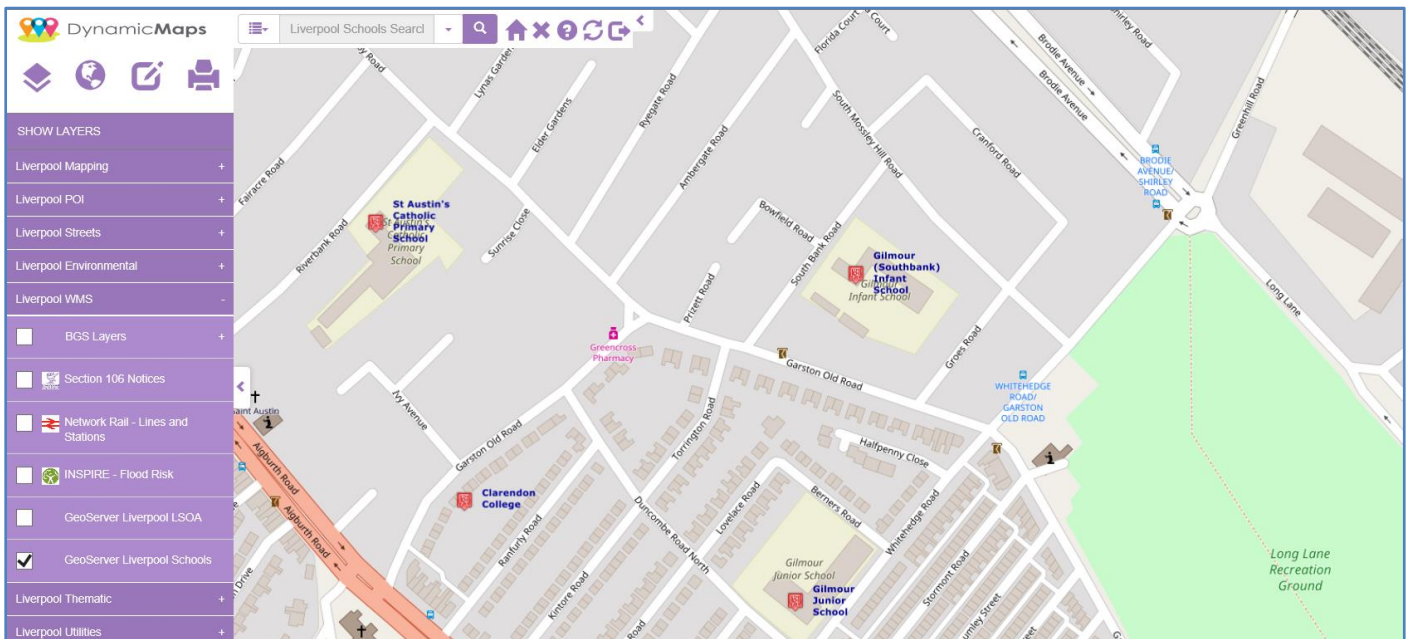
```
Style Editor
12pt
75 <CssParameter name="font-size">11</CssParameter>
76 <CssParameter name="font-style">normal</CssParameter>
77 <CssParameter name="font-weight">bold</CssParameter>
78 </Font>
79 <LabelPlacement>
80 <PointPlacement>
81 <AnchorPoint>
82 <AnchorPointX>
83 0.0
84 </AnchorPointX>
85 <AnchorPointY>
86 0.5
87 </AnchorPointY>
88 </AnchorPoint>
89 <Displacement>
90 <DisplacementX>
91 20
92 </DisplacementX>
93 <DisplacementY>
94 0
95 </DisplacementY>
96 </Displacement>
97 </PointPlacement>
98 </LabelPlacement>
99 <Fill>
```

Using **two Rules** within the SLD which apply a **Min/Max Scale Denominator**, the WMS layer now shows two different types of labels, one at the startup zoom level:



And another rule for labelling when the user zooms in below 1:10000 scale.





Why not have a go with using GeoServer to show how you can better display your layers, using the Styling and Labelling options available with SLDs.