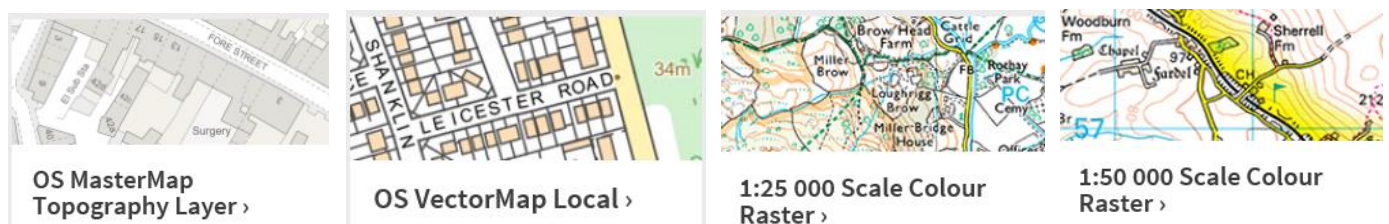# Publishing PSMA Data

by David Crowther

Whilst we know that there are many options for your basemapping within either your web or desktop GIS e.g. Open Street Map, Google etc.. many of our clients still like to maximise their use of the **Public Service Mapping Agreement** (PSMA) to utilise the many and varied basemaps that the Ordnance survey provide to them. These include:-

OS MasterMap Topography Layer › OS VectorMap Local › 1:25 000 Scale Colour Raster › 1:50 000 Scale Colour Raster ›

This blog describes one day in the life of a Cadline 'GIS Application Engineer', utilsiing Open Source geospatial applications, such as GeoServer and PostGIS, to publish these datasets into our webGIS (MapThat) and a well-known Open Source desktop GIS (QGIS).

Much of the detailed steps for publishing **OS Mastermap as a Web Map Service (WMS)** are covered in this previous Cadline White Paper:

https://www.cadlinecommunity.co.uk/hc/en-us/articles/115003309789-Publishing-OS-MasterMap-WMS-Layers

A - Publishing Large Scale Raster Data:

Before I decided to tackle accessing OS MasterMap, I decided to start by looking at the larger scale OS raster datasets such as 1:250K, MiniScale, 1:50K, 1:25K and VectorMap Local.
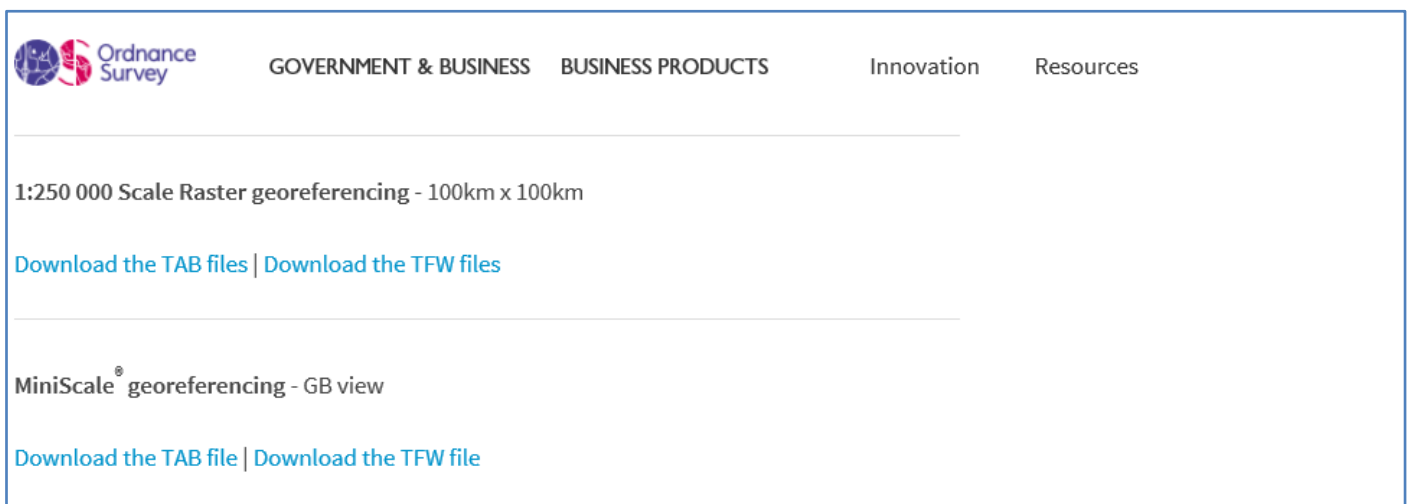
A1 – Downloading the required Raster Data:

Using your **OS Order** details, it is very simple to find and then download the relevant raster tiles for your Local Authority:
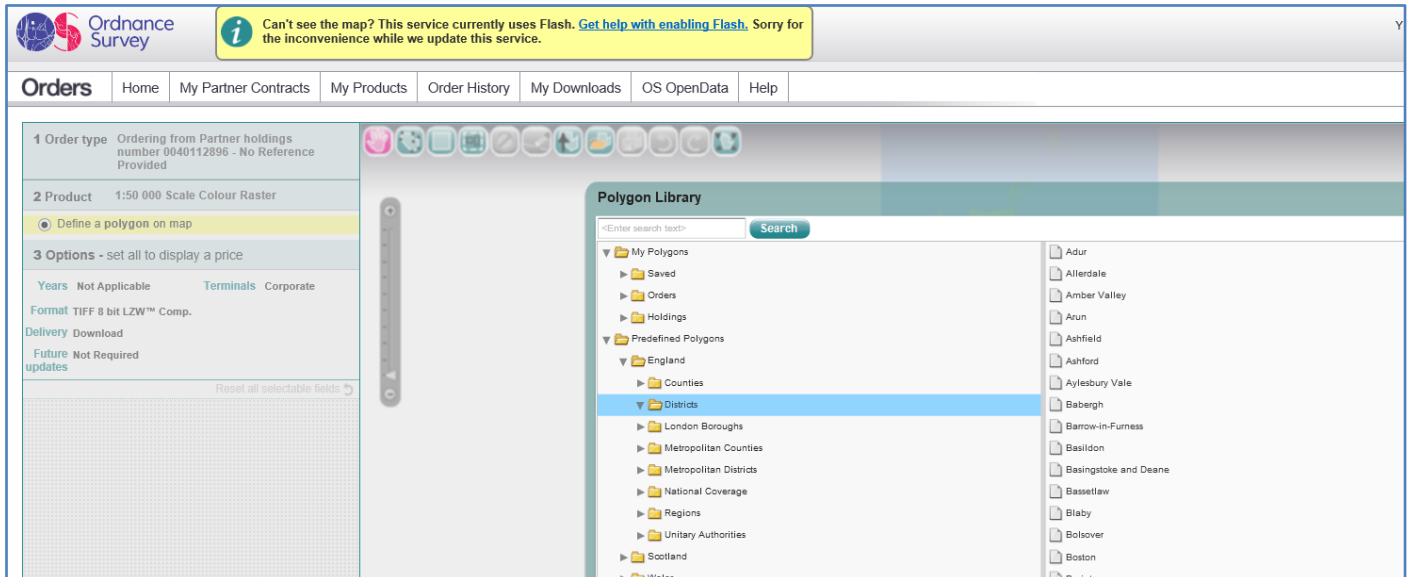
Having downloaded the relevant image files for each dataset, I needed to then source the corresponding georeference files (e.g. TFW).… and a quick google search led me here:

https://www.ordnancesurvey.co.uk/business-and-government/help-and-support/products/georeferencing-files-land-sea-tiles.html

Having now downloaded the images and the required georeference files, these were then copied to our Application Server ready to be published:
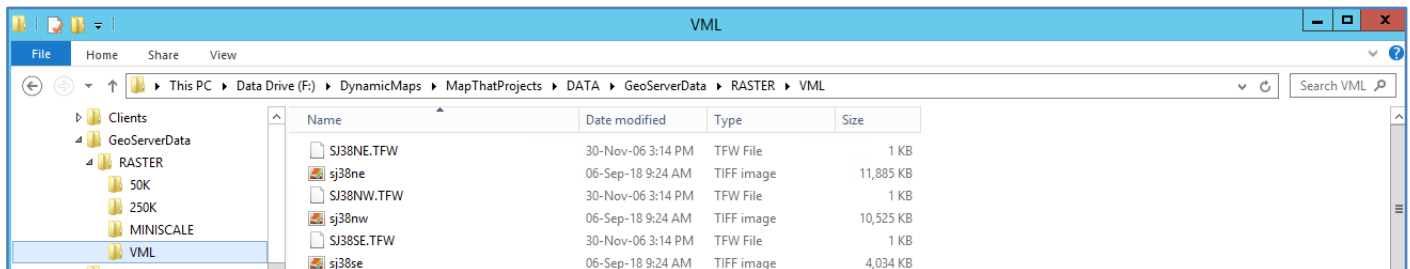

A2 – Publish the Raster Data using GeoServer:

Instead of publishing this raster data as individual tiles, which QGIS can happily consume, my aim was to use **GeoServer** to publish them as a **Web Map Service (WMS),** which meant that multiple tiles were mosaiced together into a tiled service.

Within GeoServer I created several new **Data Stores**. Firstly, a simple single **World Image** Data Store for the **MiniScale** tile.
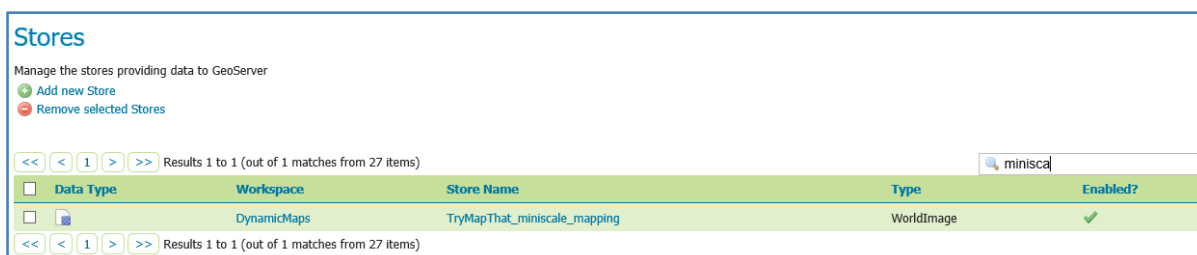
And then a series of **Image Mosaic** Data Stores; **250K, 50K, VectorMap Local**.
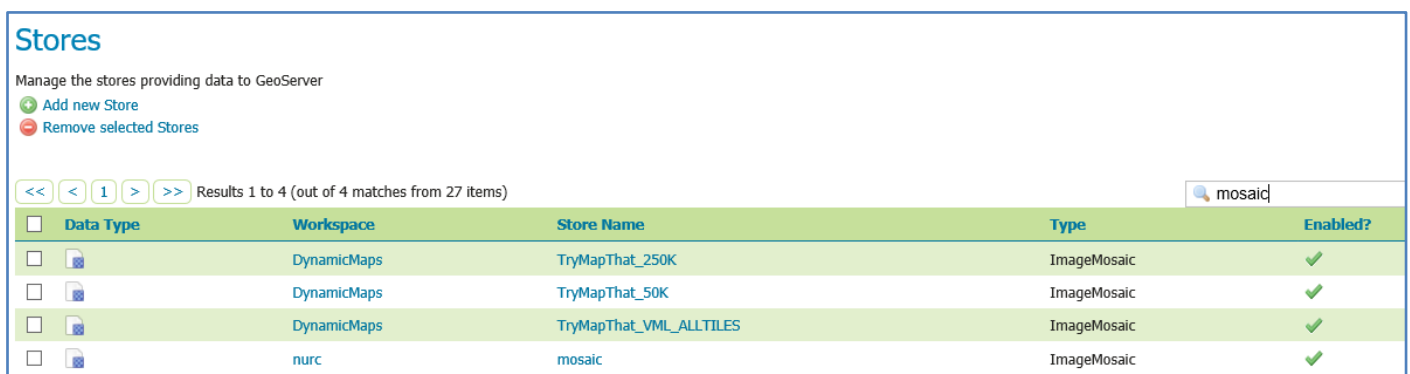
Once the Data Stores were created, I published the layers to create the required WMS. The key here was to ensure that the **CRS settings** were correct.
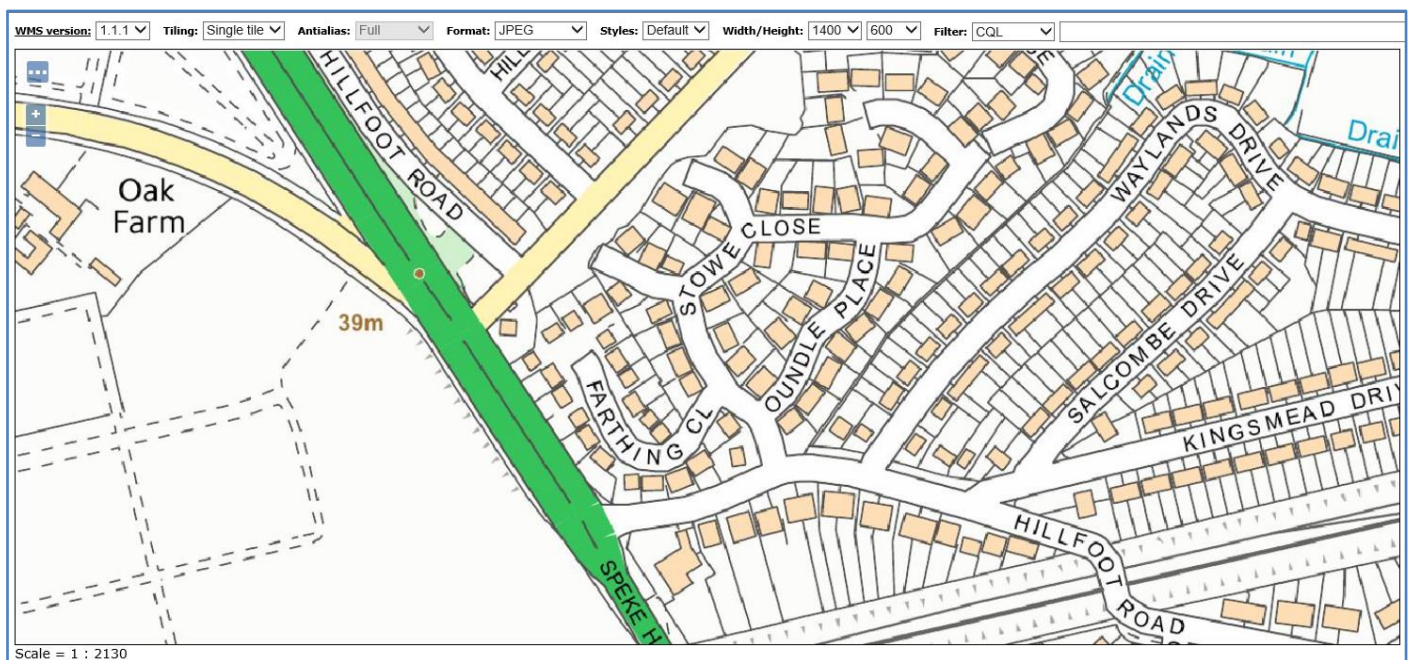

Any mistakes will mean that the WMS layer will not render correctly, which can be tested by using the GeoServer **Preview Layers** option to view the WMS in **OpenLayers**.

A3 – Create GeoServer Group Layers:

Having published each of the Raster datasets, I now had about 5 separate WMS layers that could be added into my desktop and webGIS. However, I wanted to group those Raster datasets into one WMS feed, where each different Raster dataset would auto display and hide at preset zoom levels.

To achieve this, I firstly created new **STYLE** files for each of the Raster layers. Using the **Max/Min Scale Denominator XML** to define when that layer would display.

Style Editor

```
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>raster</Name>
7       <UserStyle>
8         <Name>raster</Name>
9         <Title>Raster</Title>
10        <Abstract>A sample style for rasters, good for displaying imagery</Abstract>
11        <FeatureTypeStyle>
12          <FeatureTypeName>Feature</FeatureTypeName>
13          <Rule>
14            <MinScaleDenominator>25000</MinScaleDenominator>
15            <MaxScaleDenominator>100000</MaxScaleDenominator>
16            <RasterSymbolizer>
17              <Opacity>1.0</Opacity>
18            </RasterSymbolizer>
19          </Rule>
20        </FeatureTypeStyle>
21      </UserStyle>
22    </NamedLayer>
23 </StyledLayerDescriptor>
24
```

Now that each Raster WMS auto turned on and off at different zoom levels, I could then Group these layers into one **GROUP Layer**.
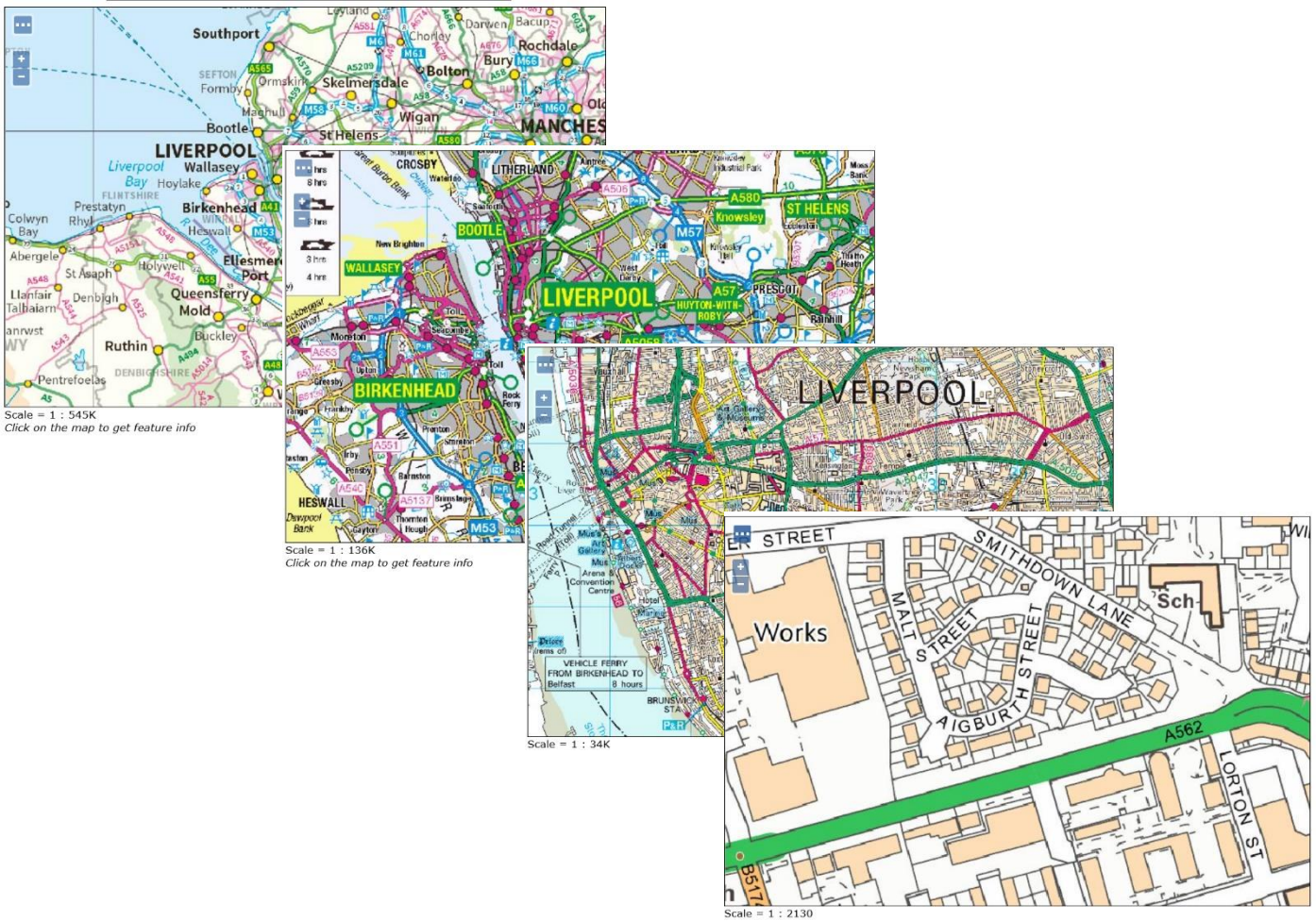
Layers

Add Layer...
Add Layer Group...
Add Style Group...

|	Drawing order	Type	Layer	Default Style	Style	Remove
1	↓	*Layer*	DynamicMaps:TryMapThat_VML_ALLTILES	☐	TryMapThat_VML	⛔
2	↑ ↓	*Layer*	DynamicMaps:TryMapThat-Raster50K-SJ48	☐	TryMapThat_50K	⛔
3	↑ ↓	*Layer*	DynamicMaps:TryMapThat_250K	☐	TryMapThat_250K	⛔
4	↑	*Layer*	DynamicMaps:MiniScale	☐	TryMapThat_Miniscale	⛔

<< < 1 > >> Results 1 to 9 (out of 9 items)

This now meant that I had one WMS layer that would auto turn off each different PSMA Raster layer as I zoomed into and out from the map:

Scale = 1 : 545K
Click on the map to get feature info

Scale = 1 : 136K
Click on the map to get feature info

Scale = 1 : 34K

Scale = 1 : 2130

A4 – Access Raster WMS within desktop GIS and CAD

Within **QGIS**, creating a **WMS connection** to my GeoServer instance then enabled me to select the **PSMA Group Layer** and **Add** it into the map.
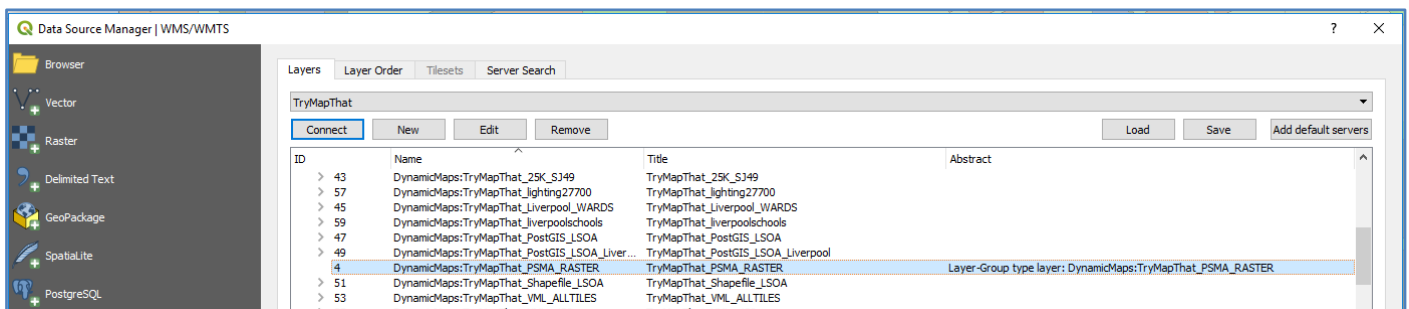
This then provided me with one WMS layer that displayed all the PSMA Raster tiles, and auto transitioned through the layers, displaying the correct tiles at each zoom level.
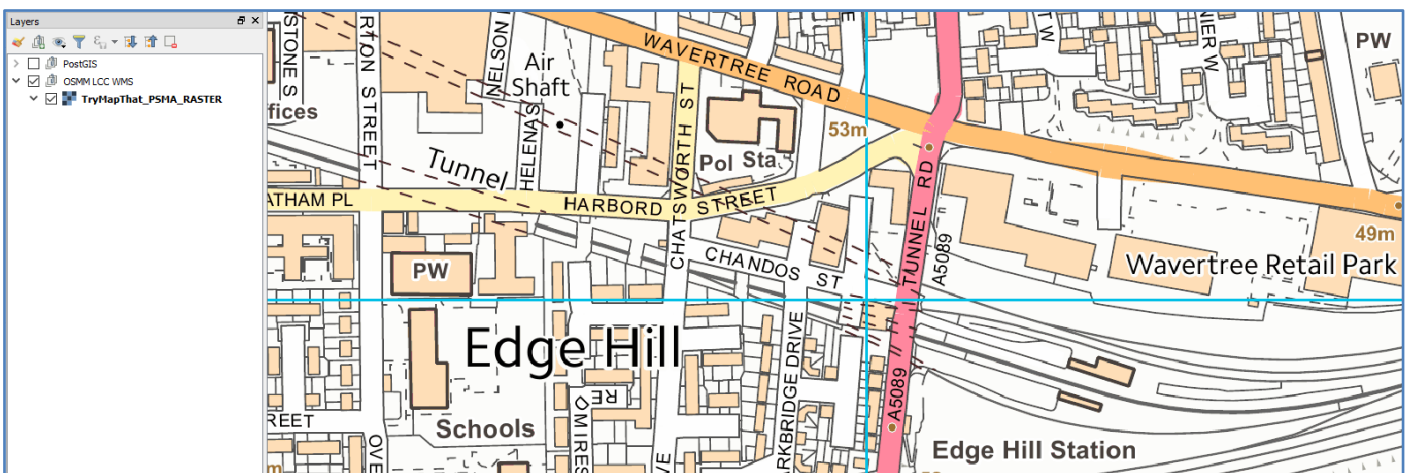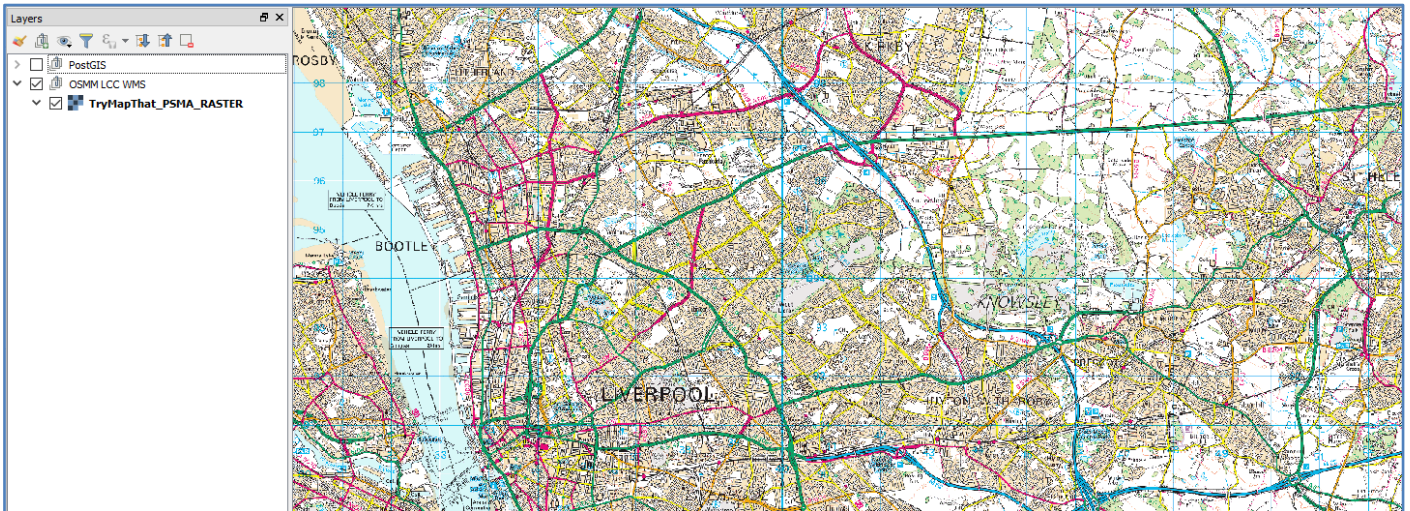
Implementing this within your GIS environment will mean that you can very easily publish all your PSMA Raster layers in one single URL, which your GIS users can access. In addition using the **WMS Data Connection** options in **Autodesk**, your CAD users can also make use of the same Raster datasets.

The image below shows how I used the WMS Connection in **Map3D** to access my PSMA Raster WMS to provide background mapping with my CAD application.
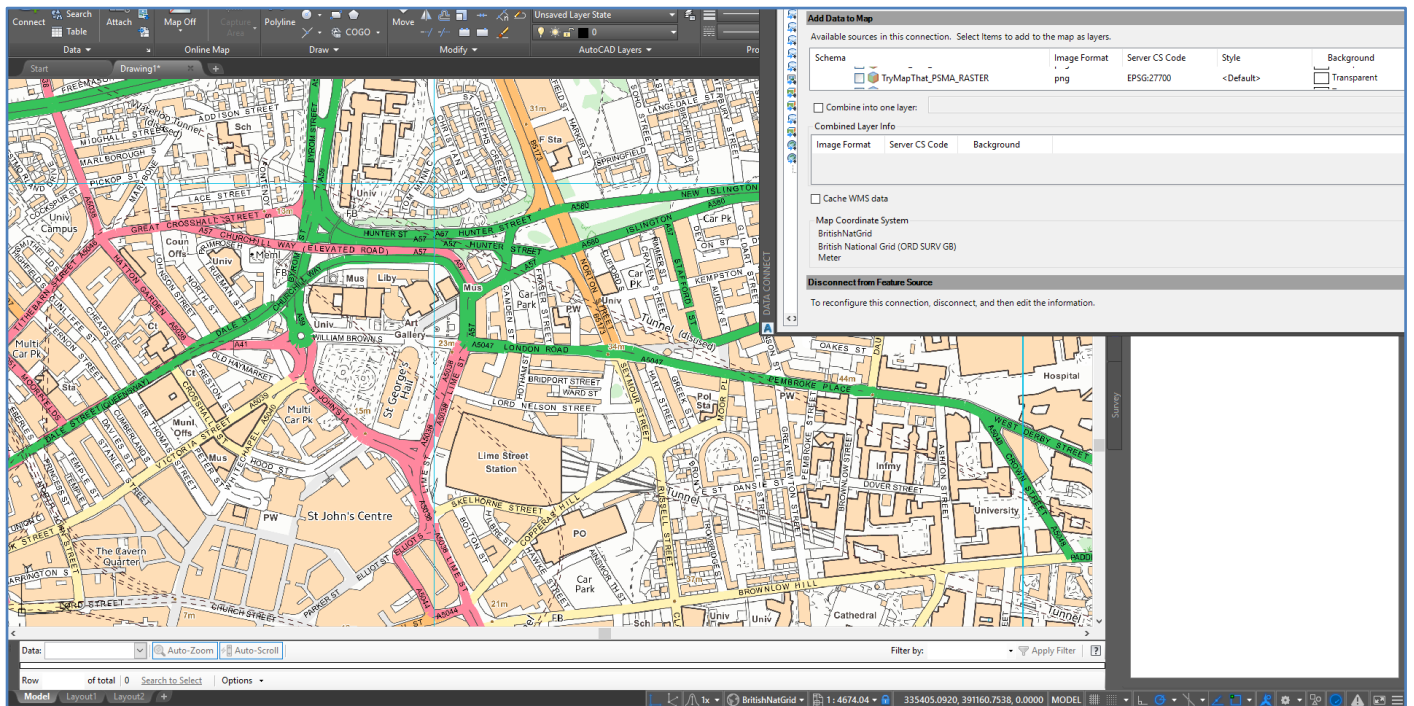
A5 – Access Raster WMS within webGIS (MapThat)

Having now created one single URL for your PSMA Raster data, you can also publish this within your webGIS as either an alternative to or replacement for any existing external basemaps. This means that you have full control over the basemapping that you present to your users.

I used the MapThat web Admin forms to create a new WMS layer within our **TryMapThat** Instance.
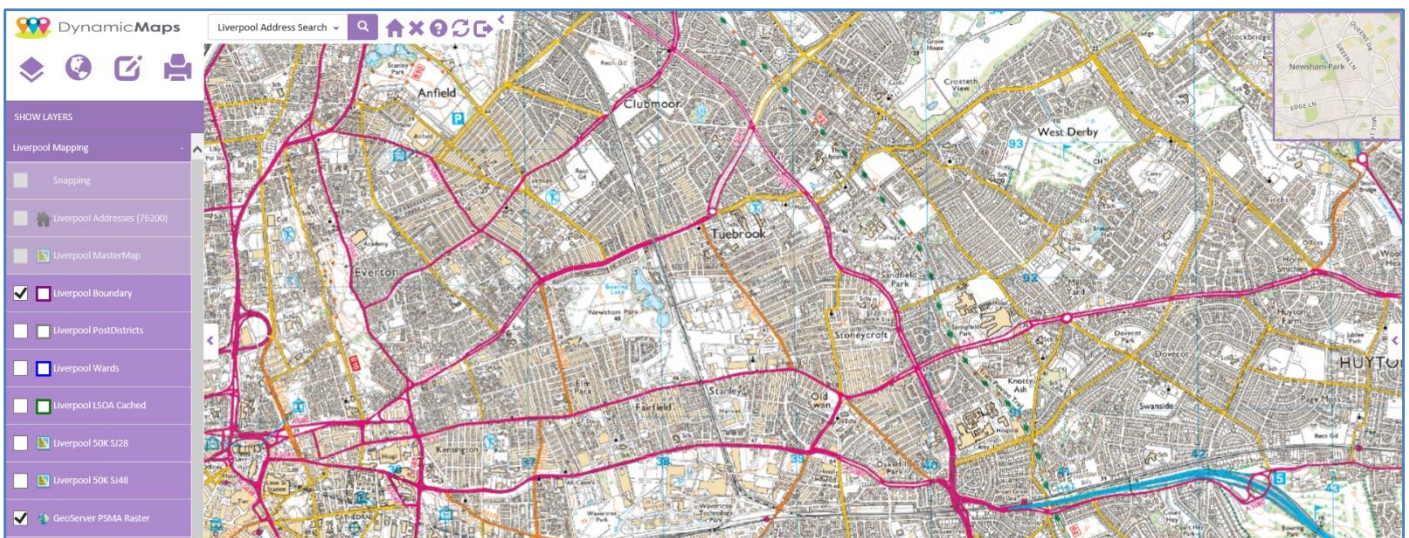
B - Publishing OS MasterMap:

Having now successfully published the Large-Scale Raster mapping, next I explored options for publishing **OS MasterMap** within both a webGIS and desktop GIS.
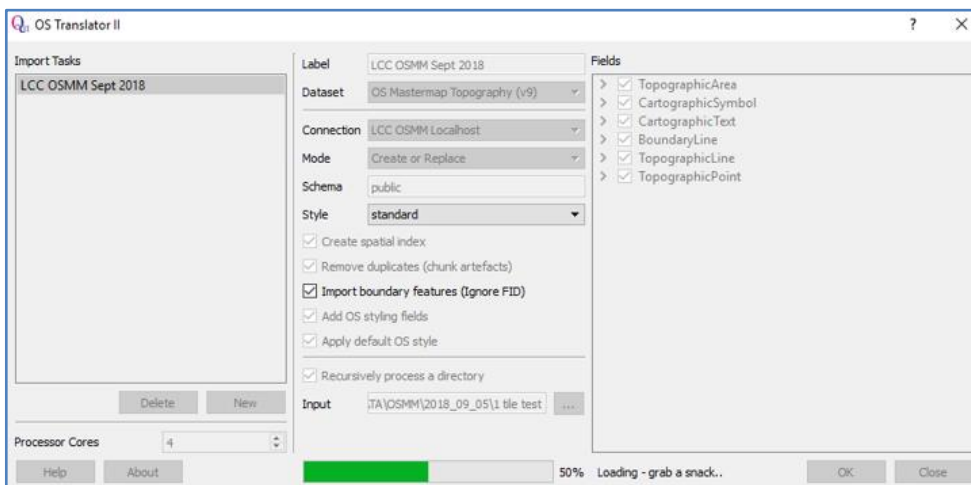
B1 – Importing the required OS MasterMap Data:

Using your **OS Order** details it is very simple to find and then download the relevant OS MasterMap GZ tiles for your Local Authority:

Once the GZ tiles have downloaded I used the **OS Translator 2** plugin within **QGIS** to import the MasterMap data into the individual tables within my PostGIS Database. The steps to do this and then create the WMS layers via GeoServer are fully described in this previous white paper - https://www.cadlinecommunity.co.uk/hc/en-us/articles/115003309789-Publishing-OS-MasterMap-WMS-Layers - although a summary of these steps are provided below.

The OS Translator tool took about 1 hour to import the 13 tiles for my chosen area.

After which I then had the required OS MasterMap tables in my PostGIS database.

B2 – Accessing the OS MasterMap Data in QGIS and CAD:

Using the **PostGIS connection** options within QGIS, it was very simple to add the OS MasterMap PostGIS tables into my QGIS project, and the styling was auto generated as the style options are stored in the PostGIS database for you.

Using the **PostGIS Data Connection** options in Autodesk, your CAD users can also access the OS MasterMap directly from the same PostGIS database.
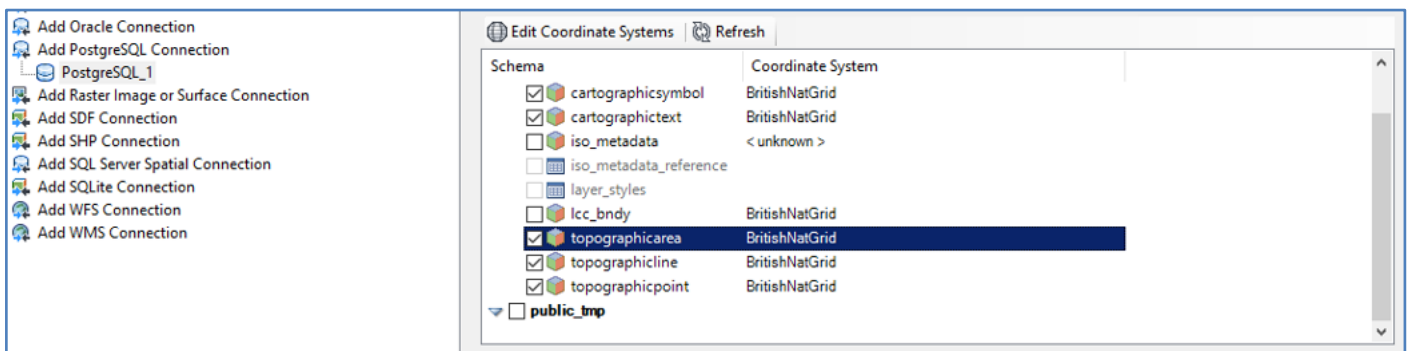
Once opened the PostGIS tables were added to my map canvas:-
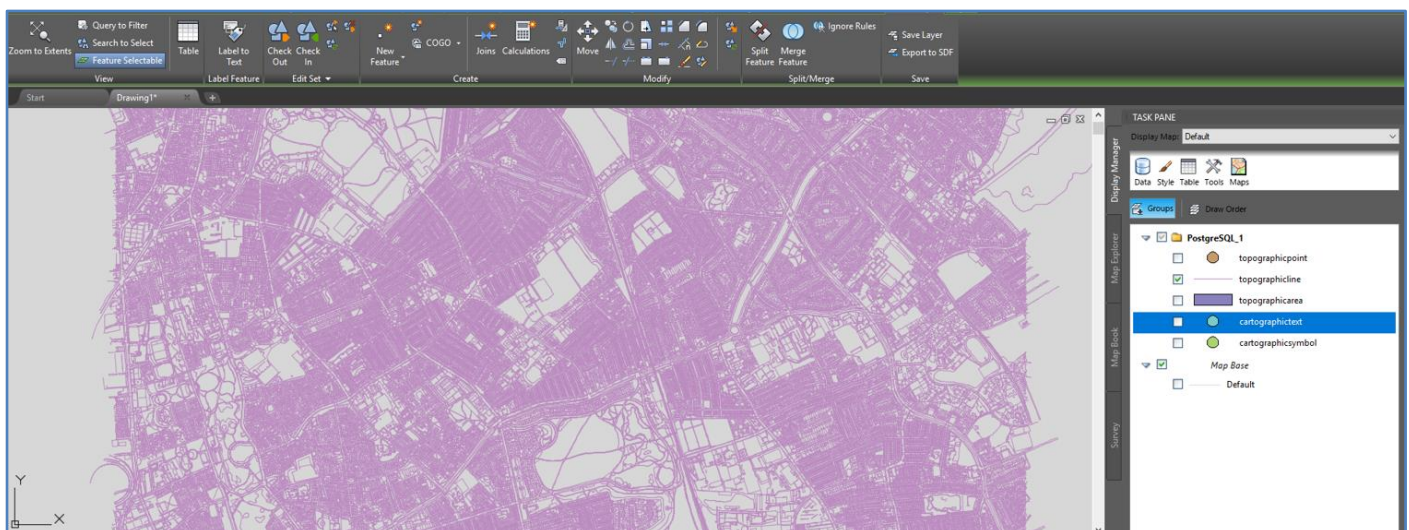
B3 – Publish the OS MasterMap data using GeoServer:

Instead of publishing this MasterMap data directly from PostGIS, I explored how to use **GeoServer** to publish them as a **Web Map Service (WMS),** which meant that I could simply publish one WMS instead of the individual tables.

Within GeoServer I created a new **Data Store** to the PostGIS database, which gave me the connection to the individual tables in the PostGIS database.

Add layer from DynamicMaps:TryMapThatPostGIS

You can create a new feature type by manually configuring the attribute names and types. **Create new feature type...**
On databases you can also create a new feature type by configuring a native SQL statement. **Configure new SQL view...**
Here is a list of resources contained in the store 'TryMapThatPostGIS'. Click on the layer you wish to configure

| << | < | 1 | 2 | > | >> | Results 0 to 0 (out of 0 items) | Search |

Published	Layer name	Action
✔	cartographicsymbol	Publish again
✔	cartographictext	Publish again
✔	lsoa_extra	Publish again
✔	topographicarea	Publish again
✔	topographicline	Publish again
✔	topographicpoint	Publish again

Once the Data Store was created, I published the layers to create the required WMS and then tested these layers using the GeoServer **Preview Layers** option to view the WMS in **OpenLayers.**

Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

| << | < | 1 | > | >> | Results 1 to 5 (out of 5 matches from 44 items) | lcc |

| Type	Title	Name	Common Formats	All Formats
●	LCC_cartographicsymbol	DynamicMaps:LCC_cartographicsymbol	OpenLayers KML GML	Select one
●	LCC_cartographictext	DynamicMaps:LCC_cartographictext	OpenLayers KML GML	Select one
▣	LCC_topographicarea	DynamicMaps:LCC_topographicarea	OpenLayers KML GML	Select one
⋈	LCC_topographicline	DynamicMaps:LCC_topographicline	OpenLayers KML GML	Select one
●	LCC_topographicpoint	DynamicMaps:LCC_topographicpoint	OpenLayers KML GML	Select one

TopoPoint –

CartoArea –

B4 – Create GeoServer Group Layers:

Having published each of the PostGIS MasterMap tables I now had about 5 separate WMS layers that could be added into my desktop and webGIS. However, I wanted to group those datasets into one WMS feed, where each different MasterMap table would be fully styled based on their **Style Codes.**

To achieve this, I firstly created new **STYLE** files for each of the PostGIS tables. Using the **SLD files** that are available from this GitHub link – https://github.com/OrdnanceSurvey/OSMM-Topography-Layer-stylesheets

Now that I had the SLD style files I created new Styles for each of the Mastermap layers, using the **upload SLD** to create the XML needed for each Style.
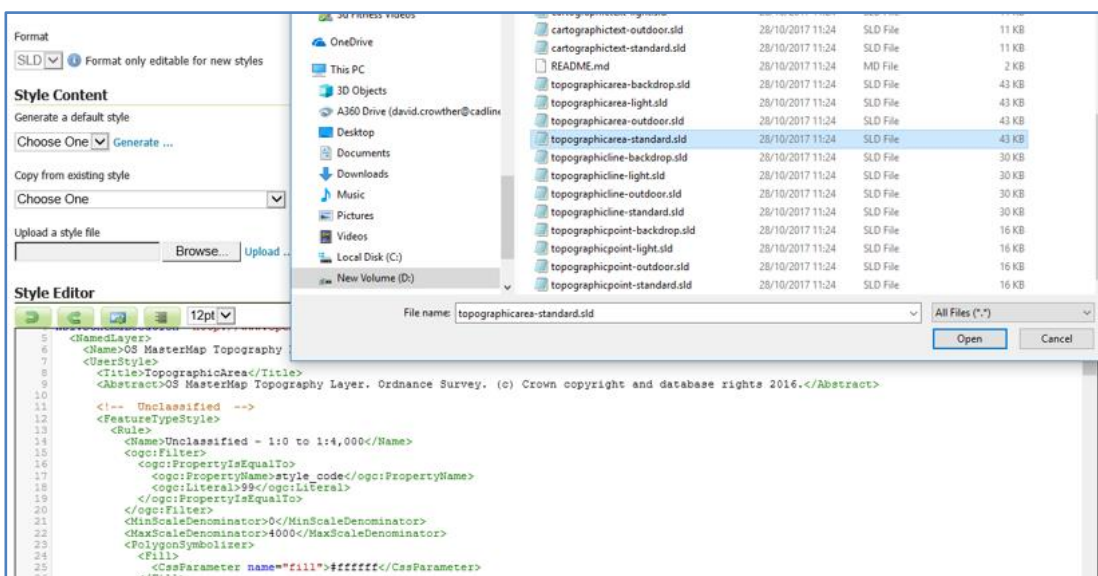
I then applied the new STYLE to each OS MasterMap layer e.g. **CARTO SYMBOL.**
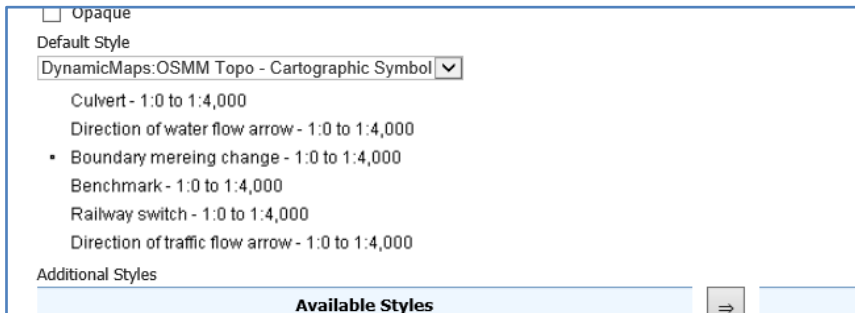
Then using the **Open Layers Preview** I could test that the styles were correctly being applied to the layers.

However, when I previewed the TopoArea Layer I noticed that some symbols were missing and being represented as **grey diamonds**.

The fix for this was quite simple. The issue was due to the fact that the SLD being used was not correctly referencing the **OSMMSYMOLS** library that you should also download from GitHub.

This library contains the **SVG files** which QGIS and GeoServer need to reference in order to display the benchmark symbols, woodland fills etc…

So, I re-visited each new **STYLE** file and edited any value which was referencing the **OSMMSYMBOLS** folder so that the link was now correct.

Now that each MasterMap WMS had the correct styling I could then **Group** these layers into one **GROUP Layer**.

	Drawing order	Type	Layer	Default Style	Style	Remove
1	⬇	Layer	DynamicMaps:LCC_cartographicsymbol	☐	cartographicsymbol-standard	⊖
2	⬆ ⬇	Layer	DynamicMaps:LCC_cartographictext	☐	cartographictext-standard	⊖
3	⬆ ⬇	Layer	DynamicMaps:LCC_topographicarea	☐	topographicarea-standard	⊖
4	⬆ ⬇	Layer	DynamicMaps:LCC_topographicline	☐	topographicline-standard	⊖
5	⬆	Layer	DynamicMaps:LCC_topographicpoint	☐	topographicpoint-standard	⊖

`<<` `<` `1` `>` `>>` Results 1 to 5 (out of 5 items)

This now meant that I had one WMS layer that contained all the pre-styled OS MasterMap layers.
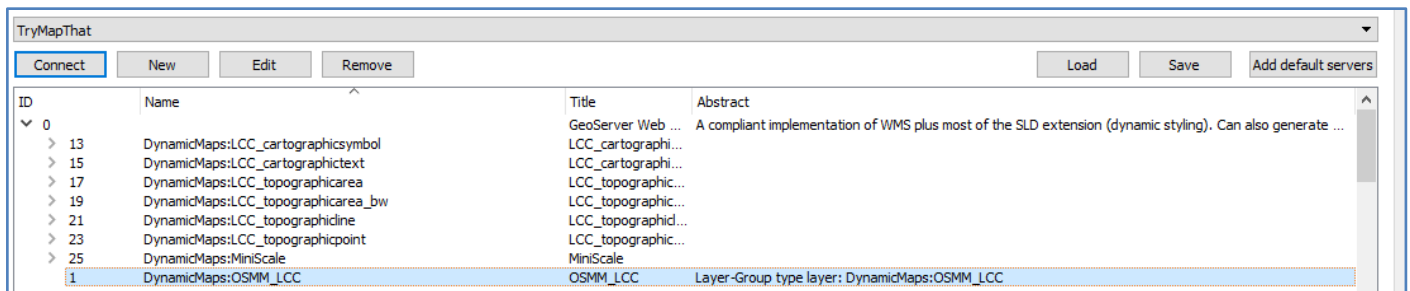

Scale = 1 : 1065
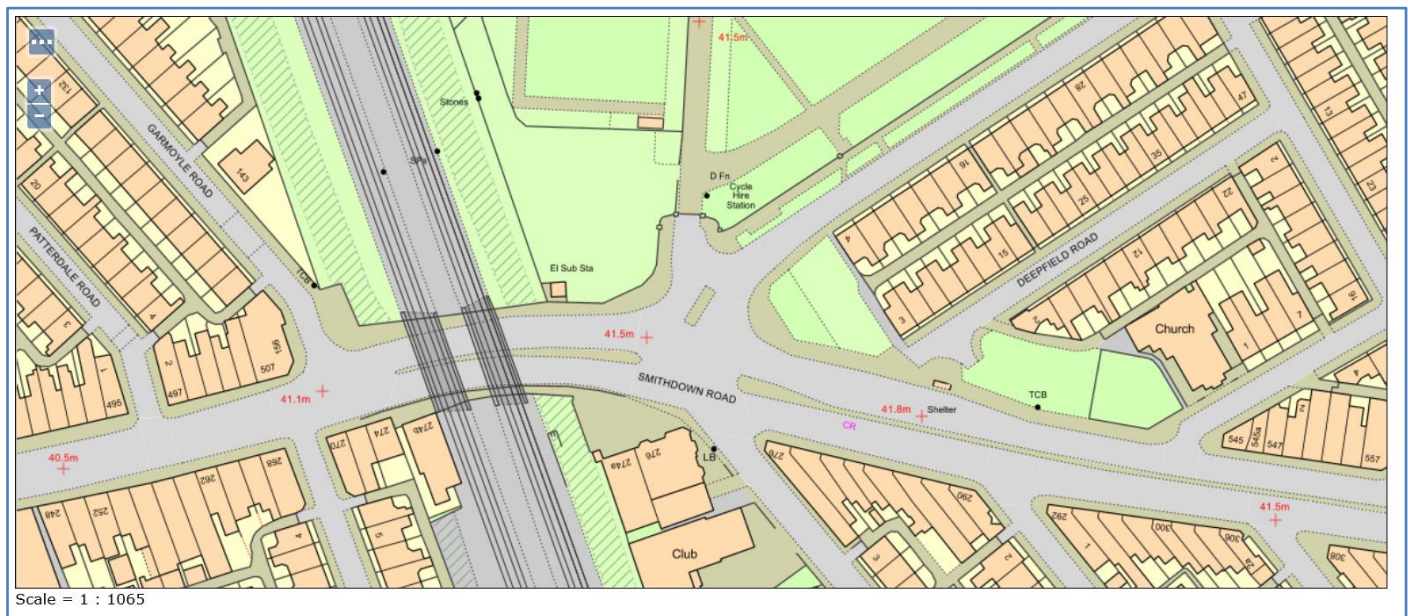
A5 – Access OS Mastermap WMS within desktop GIS and CAD

Within **QGIS**, using the same **WMS connection** to my GeoServer instance then enabled me to select the new OS Mastermap Group layer and **Add** that to the map.

This then provided me with one WMS layer that displayed all the OS MasterMap tables in one layer, pre-styled with the correct colours and line styles.

Tip! – you may find that the rendering of OS MasterMap as a WMS feed may be slow to begin with. However, by increasing the **Max Rendering Memory** within your GeoServer WMS settings you can increase rendering performance in client application. Here I have multiplied the default value x 4.

Having loaded the WMS layer into my GIS, you can also use the **WMS Data Connection** options in Autodesk, so that your CAD users can also make use of the same MasterMap data.

Once added to the map canvas, I then had OS MasterMap pre-styled and available as a background map within my CAD application. No need to translate GIS data from one format to a CAD compliant format. Instead we now have **One Source of Truth** for that data, which is easily styled through GeoServer and then available for both GIS and CAD users!
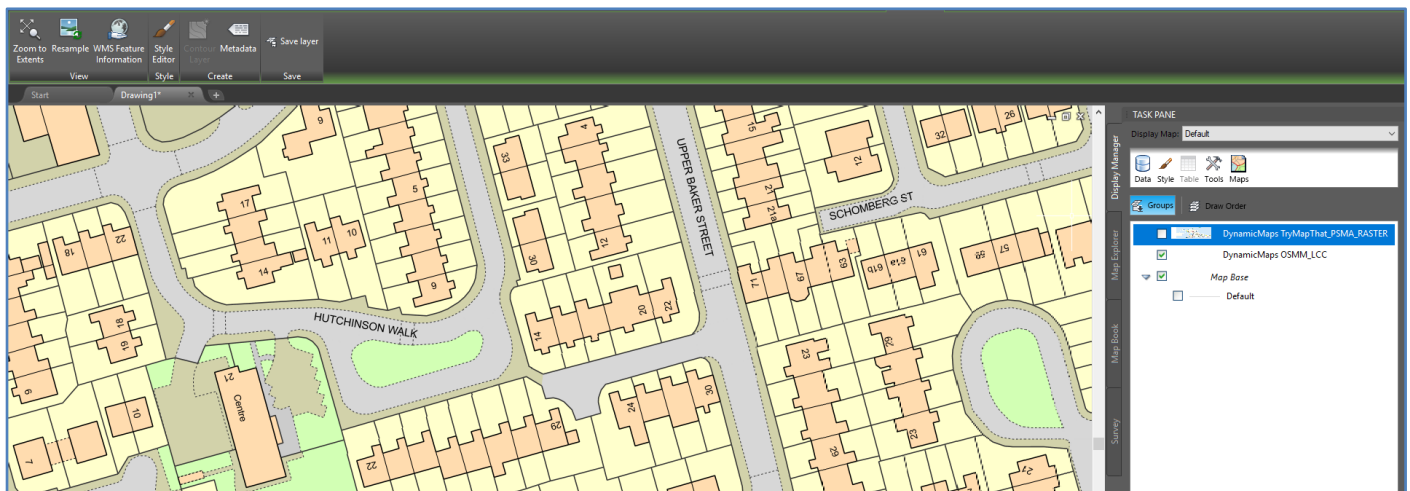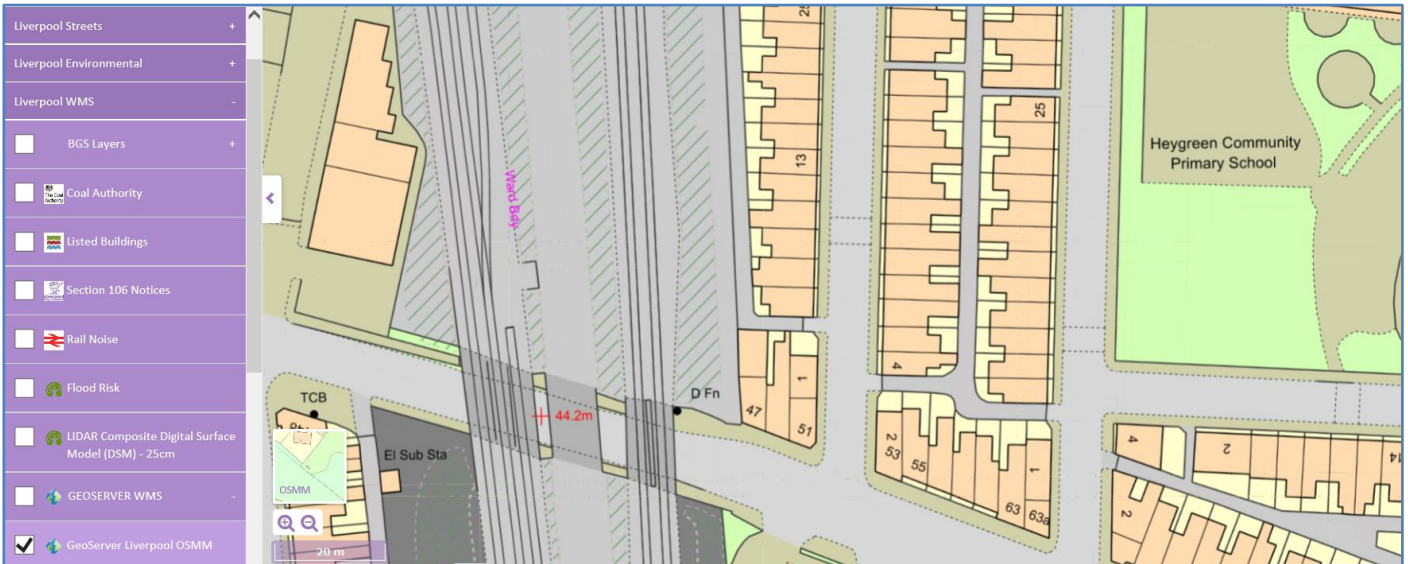

A6 – Access OS Mastermap WMS within webGIS (MapThat)

Having now created one single URL for your OS MasterMap data, you can publish this within your webGIS as either an alternative to or replacement for other tiles services.

Using the **MapThat Admin** forms I was able to easily create a new layer in MapThat which used the same URL to access my new OS MasterMap WMS feed.

Finally, I decided to make use of the additional SLD style files that are supplied on the GitHub page to create **Outdoor** and **Light** versions of my OS MasterMap WMS.

https://github.com/OrdnanceSurvey/OSMM-Topography-Layer-stylesheets

📁 images	New styles added (Outdoor & Light)	2 years ago
📁 osmmsymbols	New styles added (Outdoor & Light)	2 years ago
📄 README.md	Note about GeoServer SLDs with data in Oracle	11 months ago
📄 boundaryline-backdrop.sld	Rename boundaryline-backdrop.sld.sld to boundaryline-backdrop.sld	2 years ago
📄 boundaryline-light.sld	New styles added (Outdoor & Light)	2 years ago
📄 boundaryline-outdoor.sld	New styles added (Outdoor & Light)	2 years ago
📄 boundaryline-standard.sld	Rename boundaryline-standard.sld.sld to boundaryline-standard.sld	2 years ago
📄 cartographicsymbol-backdrop.sld	Rename cartographicsymbol-backdrop.sld.sld to cartographicsymbol-back...	2 years ago
📄 cartographicsymbol-light.sld	New styles added (Outdoor & Light)	2 years ago

To do this, I simply created **New STYLE** files to create an Outdoor and Light version of each of the OS MasterMap Layers – CartoText, TopoArea, CartoSymbol etc….

	Style Name	Workspace
☐	cartographicsymbol-outdoor	DynamicMaps
☐	cartographictext-outdoor	DynamicMaps
☐	topographicarea-outdoor	DynamicMaps
☐	topographicline-outdoor	DynamicMaps
☐	topographicpoint-outdoor	DynamicMaps

<< < 1 > >> Results 1 to 5 (out of 5 matches from 45 items) 🔍 outdoor ✕

<< < 1 > >> Results 1 to 5 (out of 5 matches from 45 items)

Then I simply created **2 new GROUPS** for the OS MasterMap layers.

Layer Groups

Define and manage layer groupings

➕ Add new layer group

➖ Remove selected layer group(s)

<< < 1 > >> Results 1 to 3 (out of 3 matches from 7 items) 🔍 osmm ✕

	Layer Group	Workspace
☐	OSMM_LCC_Light	
☐	OSMM_LCC_Outdoor	
☐	OSMM_LCC	DynamicMaps

<< < 1 > >> Results 1 to 3 (out of 3 matches from 7 items)

And applied the corresponding styles (Outdoor or Light) to each layer in that group.

	Drawing order	Type	Layer	Default Style	Style	Remove
1	⬇	Layer	DynamicMaps:LCC_topographicarea	☐	topographicarea-outdoor	➖
2	⬆⬇	Layer	DynamicMaps:LCC_cartographicsymbol	☐	cartographicsymbol-outdoor	➖
3	⬆⬇	Layer	DynamicMaps:LCC_cartographictext	☐	cartographictext-outdoor	➖
4	⬆⬇	Layer	DynamicMaps:LCC_topographicline	☐	topographicline-outdoor	➖
5	⬆	Layer	DynamicMaps:LCC_topographicpoint	☐	topographicpoint-outdoor	➖

<< < 1 > >> Results 1 to 5 (out of 5 items)

I now had **3 different WMS** versions of my OS MasterMap data, which could then easily be added into QGIS, CAD and webGIS.

OS MasterMap WMS - Standard

OS MasterMap WMS - Outdoor

OS MasterMap WMS – Light

www.cadline.co.uk   01784 419 922   sales@cadline.co.uk