

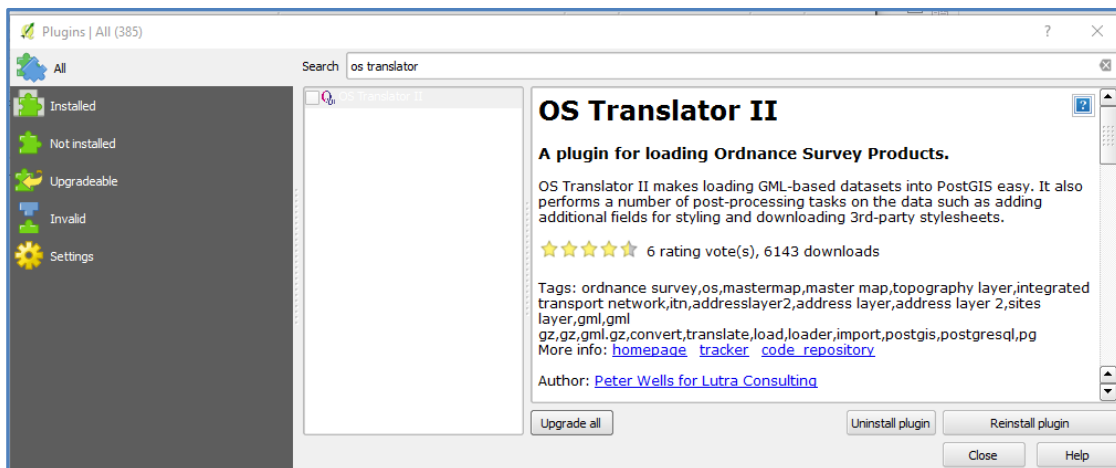
In this White Paper, we will outline how to easily import OS MasterMap (OSMM) data into a PostGIS database and publish that as a Web Map Service (WMS) with styling via GeoServer.

Objectives

- Import OS MasterMap into PostGIS
- Style OSMM using Style Layer Descriptors
- Publish OSMM as WMS via GeoServer

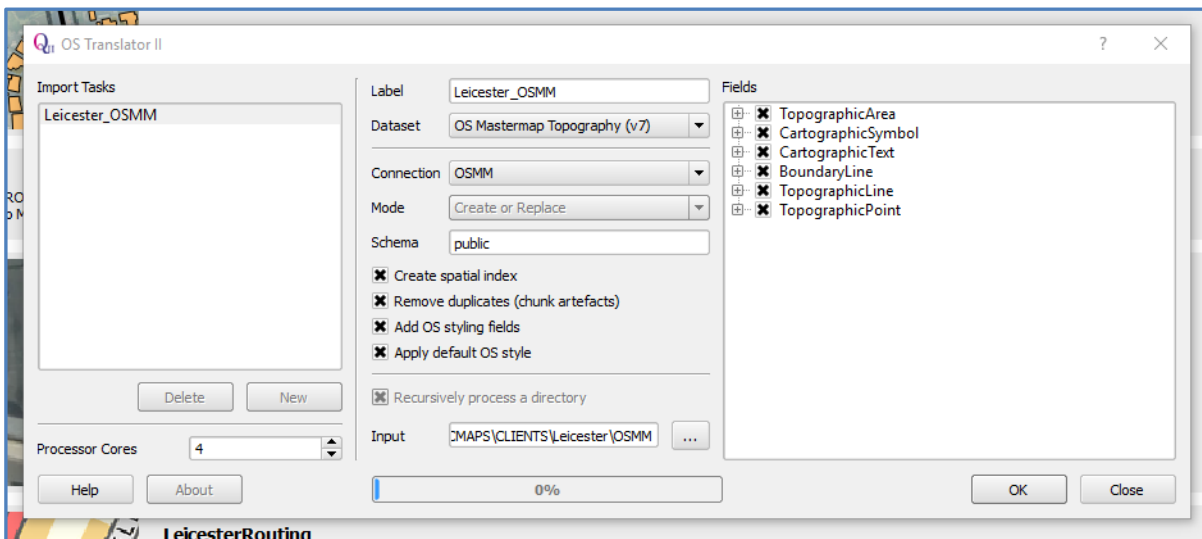
Install OS Translator Tool

- 1 In QGIS from the **Plugins** menu choose → **Manage and Install Plugins**.
- 2 Search for → **OS Translator** and then → **install** the plugin.



- 3 The Plugin will now be accessible as a new tool under the **Plugin** menu and as an icon within Q





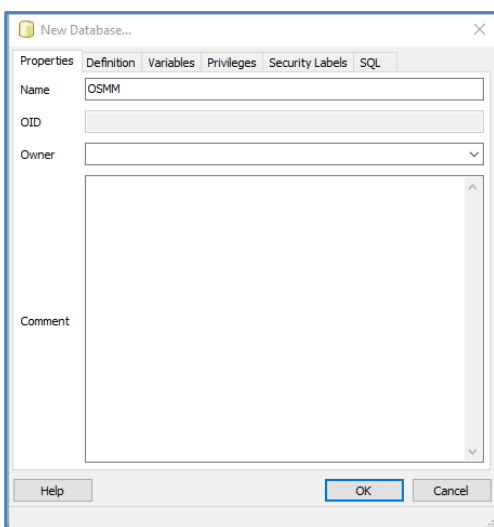
Create a PostGIS Database

1 Open → **PGAdmin**



2 In the → **Object Browser** select the → **Databases** → right click and choose → **New Database**.

3 Give the New Database a → **Name** e.g. OSMM and set the → **Privileges** to be → **ALL**.



- 4 To make the new database a spatial database run the following → SQL:



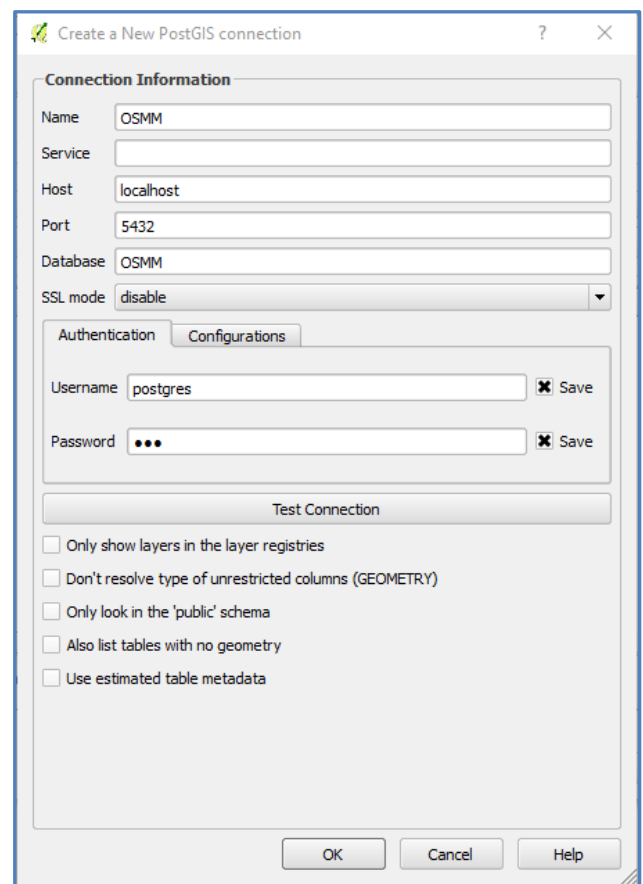
CREATE EXTENSION postgis;

Upload OSMM into PostGIS

- 1 In → **QGIS** choose to → **Connect to PostGIS**.



- 2 Choose a → **New Connection** and set the connection parameters as per the below.

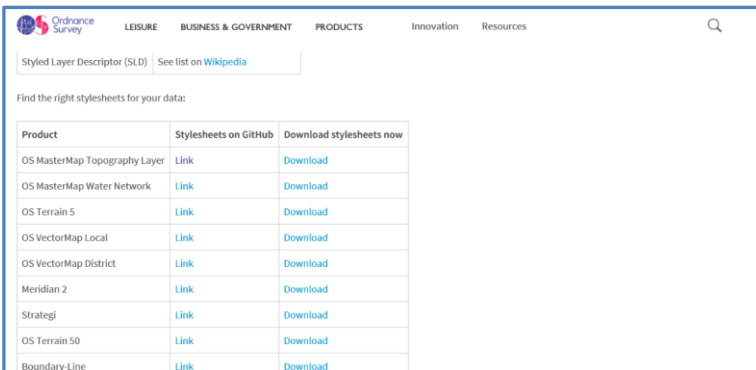


Create GeoServer Style Files



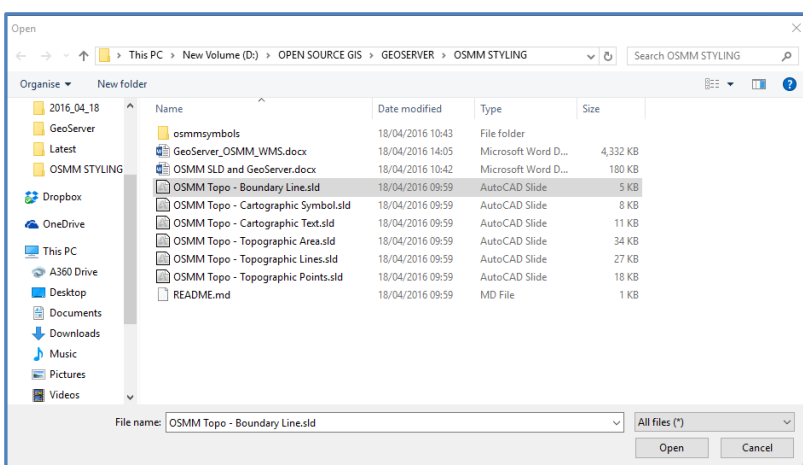
- 1 Open GeoServer → and Choose → **Data** → **Styles**.
- 2 Choose to → **Add a new Style**.
- 3 The SLD style files for OSMM are available to download from the link below. Open the link and save these locally:

<https://www.ordnancesurvey.co.uk/resources/carto-design/cartographic-stylesheets.html>



Product	Stylesheets on GitHub	Download stylesheets now
OS MasterMap Topography Layer	Link	Download
OS MasterMap Water Network	Link	Download
OS Terrain 5	Link	Download
OS VectorMap Local	Link	Download
OS VectorMap District	Link	Download
Meridian 2	Link	Download
Stratagi	Link	Download
OS Terrain 50	Link	Download
Boundary-Line	Link	Download

- 4 Within the new Style in GeoServer → scroll to the bottom of the Style page and choose → **Browse** to find the relevant SLD e.g. **BoundaryLine**



- Open the SLD → and then chose → **Upload** and the contents of the SLD will be written into the new Style file.

New style

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

Name

Workspace

Format
 SLD ▾

Generate a default style
 Choose One ▾ [Generate ...](#)

Copy from existing style
 Choose One ▾ [Copy ...](#)

12pt ▾

```

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>OS MasterMap Topography Layer (backdrop style)</Name>
7     <UserStyle>
8       <Title>BoundaryLine</Title>
9       <Abstract>OS MasterMap Topography Layer. Ordnance Survey. (c) Crown copyright and database rights 2015.</Abstract>
10
11     <FeatureTypeStyle>
12       <Rule>
13         <Name>Parish boundary - 1:0 to 1:4,000</Name>
14         <Ogc:Filter>
15           <ogc:PropertyIsEqualTo>
16             <ogc:PropertyName>style_code</ogc:PropertyName>
17             <ogc:Literal>134</ogc:Literal>

```

- Before saving/submitted ensure you change the → **Name** of the Style file or you will encounter the following GeoServer message because the Name of the SLD is currently the name and physical location of the uploaded SLD file.

Oops, something went wrong...

Sorry, something unexpected happened on the server.

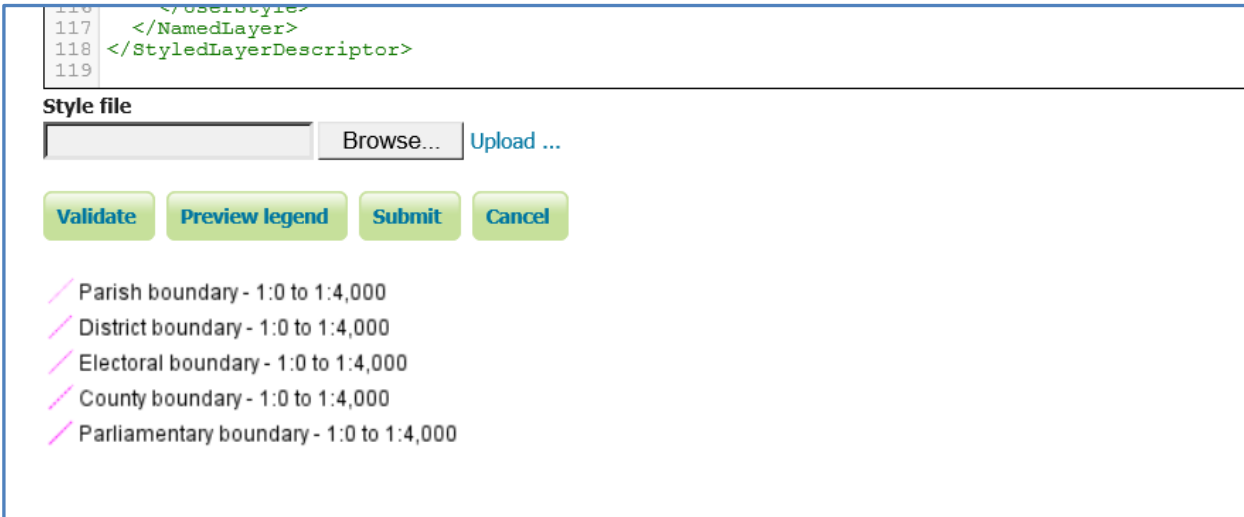
Here's an error report you can include in a [JIRA bug report](#) about this issue:

```

org.apache.wicket.WicketRuntimeException: Method onFormSubmitted of interface org.apache.wicket.markup.html.form.IFormSubmitListener targeted at component [MarkupContainer [Con
at org.apache.wicket.RequestListenerInterface.invoke(RequestListenerInterface.java:193)
at org.apache.wicket.request.target.component.Listener.ListenerInterfaceRequestTarget.processEvents(ListenerInterfaceRequestTarget.java:73)
at org.apache.wicket.request.AbstractRequestCycleProcessor.processEvents(AbstractRequestCycleProcessor.java:92)
at org.apache.wicket.RequestCycle.processEventsAndRespond(RequestCycle.java:1250)
at org.apache.wicket.RequestCycle.step(RequestCycle.java:1329)
at org.apache.wicket.RequestCycle.steps(RequestCycle.java:1436)
at org.apache.wicket.RequestCycle.request(RequestCycle.java:545)
at org.apache.wicket.protocol.http.WicketFilter.doGet(WicketFilter.java:484)
at org.apache.wicket.protocol.http.WicketServlet.doPost(WicketServlet.java:160)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
at org.springframework.web.servlet.mvc.ServletWrappingController.handleRequestInternal(ServletWrappingController.java:159)
at org.springframework.web.servlet.mvc.AbstractController.handleRequest(AbstractController.java:153)
at org.springframework.web.servlet.mvc.SimpleControllerHandlerAdapter.handle(SimpleControllerHandlerAdapter.java:48)
at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:923)
at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:852)

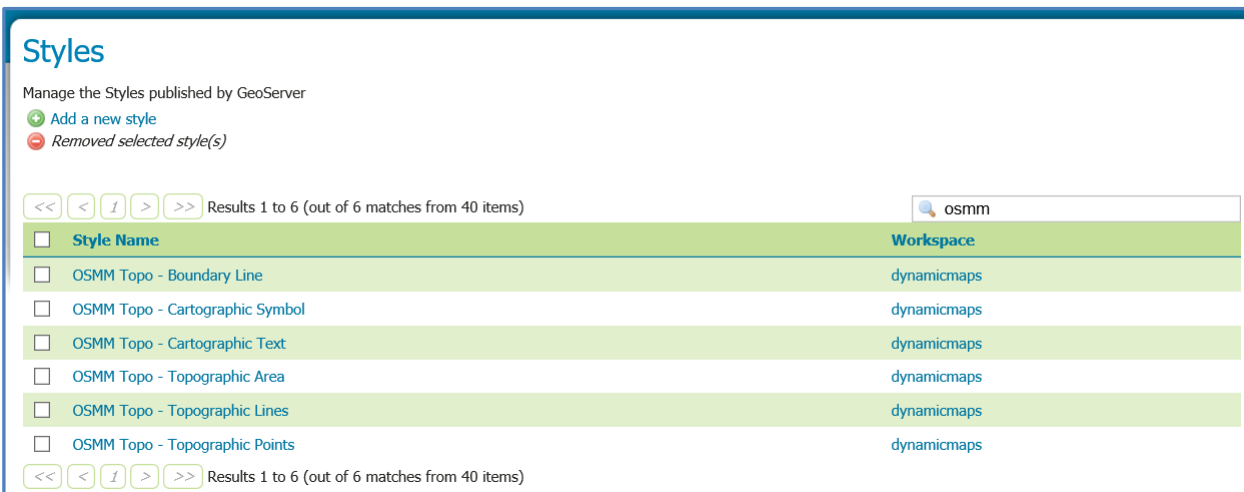
```

7 Having changed the Name of the Style file, choose → **Validate** and → Preview to check the SLD is correct.



8 Now choose → **Submit** and the new Style file will be created.

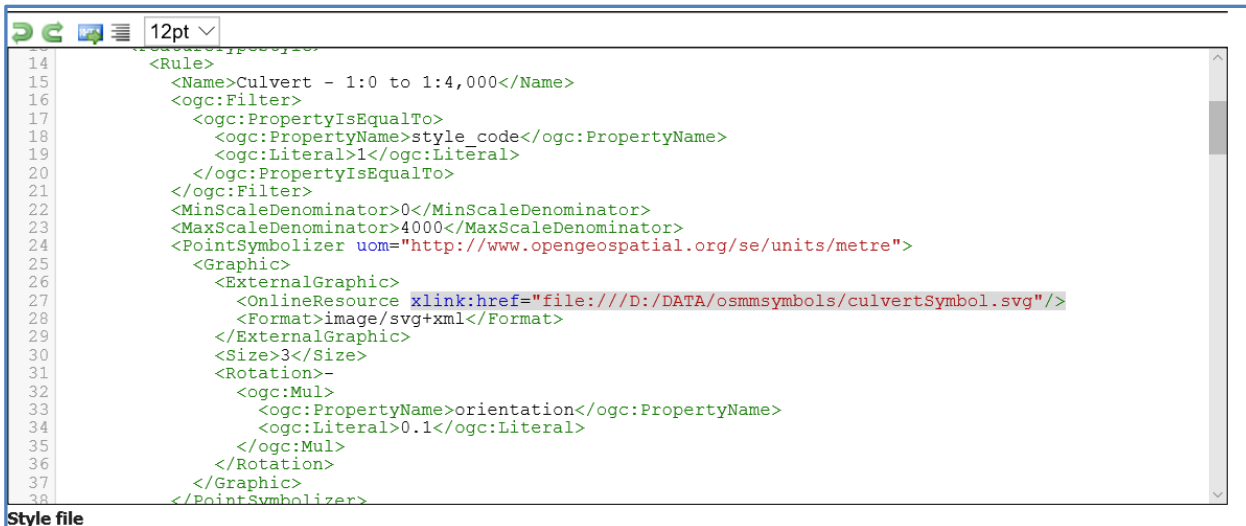
9 Repeat the above for all the → **SLD's**.



<input type="checkbox"/>	Style Name	Workspace
<input type="checkbox"/>	OSMM Topo - Boundary Line	dynamictimaps
<input type="checkbox"/>	OSMM Topo - Cartographic Symbol	dynamictimaps
<input type="checkbox"/>	OSMM Topo - Cartographic Text	dynamictimaps
<input type="checkbox"/>	OSMM Topo - Topographic Area	dynamictimaps
<input type="checkbox"/>	OSMM Topo - Topographic Lines	dynamictimaps
<input type="checkbox"/>	OSMM Topo - Topographic Points	dynamictimaps

10 The only edit that you will need to make is for the **Cartographic Symbol SLD** as this utilises image files stored as **SVG** in a folder. When you downloaded the SLD's from the link above, you will also have downloaded a **osmmsymbols** folder which contains all the symbols used for the Cartographic Symbols. **Save** → this folder to a relevant location e.g. → **D:/Data**.

- 11 Now open the → **OSMM Topo – Cartographic Symbol Style** file and edit the references to the SVG files to be → D:/Data/



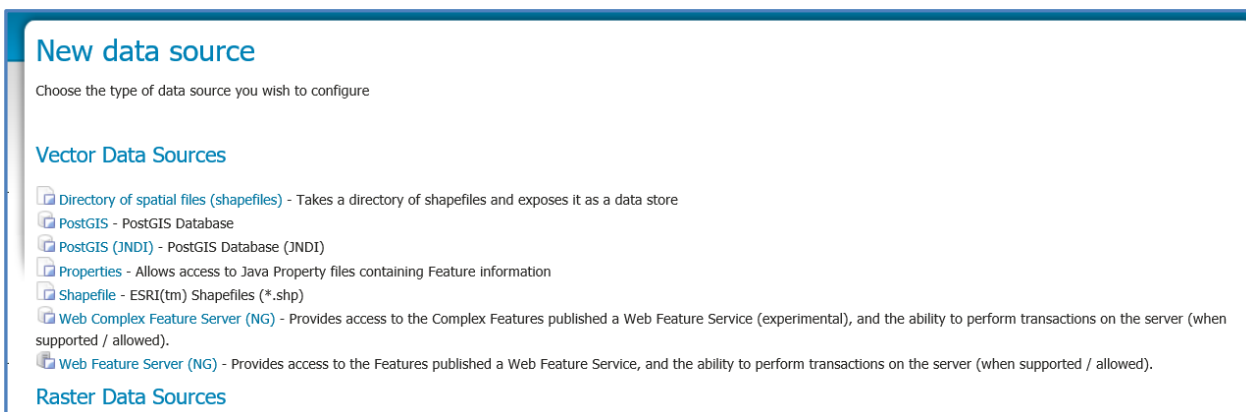
```

14 <Rule>
15 <Name>Culvert - 1:0 to 1:4,000</Name>
16 <ogc:Filter>
17 <ogc:PropertyIsEqualTo>
18 <ogc:PropertyName>style_code</ogc:PropertyName>
19 <ogc:Literal>1</ogc:Literal>
20 </ogc:PropertyIsEqualTo>
21 </ogc:Filter>
22 <MinScaleDenominator>0</MinScaleDenominator>
23 <MaxScaleDenominator>4000</MaxScaleDenominator>
24 <PointSymbolizer uom="http://www.opengeospatial.org/se/units/metre">
25 <Graphic>
26 <ExternalGraphic>
27 <OnlineResource xlink:href="file:///D:/DATA/osmmsymbols/culvertSymbol.svg"/>
28 <Format>image/svg+xml</Format>
29 </ExternalGraphic>
30 <Size>3</Size>
31 <Rotation>-
32 <ogc:Mul>
33 <ogc:PropertyName>orientation</ogc:PropertyName>
34 <ogc:Literal>0.1</ogc:Literal>
35 </ogc:Mul>
36 </Rotation>
37 </Graphic>
38 </PointSymbolizer>
    
```

- 12 Having made the change to all the Xlink references ensure you → **Submit** the changes.

Create PostGIS Data Store

- 1 In GeoServer → choose → **Data** → **Stores** → **Add New Store**.

New data source

Choose the type of data source you wish to configure

Vector Data Sources

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- PostGIS - PostGIS Database
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Complex Feature Server (NG) - Provides access to the Complex Features published a Web Feature Service (experimental), and the ability to perform transactions on the server (when supported / allowed).
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the ability to perform transactions on the server (when supported / allowed).

Raster Data Sources

2 Choose to → **Add a new PostGIS** Data Store and enter the details to connect to your PostGIS database.

New Vector Data Source

Add a new vector data source

PostGIS
PostGIS Database

Basic Store Info

Workspace *
dynamicmaps

Data Source Name *
OSMM

Description
OSMM

Enabled

Connection Parameters

dbtype *
postgis

host *
localhost

port *
5432

database
OSMM

schema
public

user *
postgres

passwd

Now you have created a New PostGIS data store you will have a list of the Database Tables within your OSMM PostGIS Database.

New Layer

Add a new layer

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 'OSMM'. Click on the layer you wish to configure

<< < | > >> Results 1 to 8 (out of 8 items)

Published	Layer name	Action
	Leicester_ITN	Publish
	boundaryline	Publish
	cartographicsymbol	Publish
	cartographictext	Publish
	layer_styles	Publish
	topographicarea	Publish
	topographicline	Publish
	topographicpoint	Publish

<< < | > >> Results 1 to 8 (out of 8 items)

Publish the PostGIS Layer

- 1 In GeoServer → choose → **Data** → **Layers** → **Add a New Resource**.



New Layer

Add a new layer

Add layer from

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 'PostGIS OSMM'. Click on the layer you wish to configure

Results 0 to 0 (out of 0 items)

Published	Layer name	Action
✓	boundaryline	Publish
✓	cartographicsymbol	Publish
✓	cartographictext	Publish
✓	topographicarea	Publish
✓	topographicline	Publish
✓	topographicpoint	Publish

- 2 One by one choose to Publish your PostGIS OSMM tables by clicking → **Publish**.
- 3 From the → **Publishing Tab** choose the Style to be the new **OSMM SLD** style you created earlier.

Default Style

- Default Line - 1:0 to 1:4,000
- Building Outline - 1:0 to 1:4,000
- Edge Line - 1:0 to 1:4,000
- Road Or Track Line - 1:0 to 1:4,000
- Building Division Line - 1:0 to 1:4,000
- Polygon Closing Line - 1:0 to 1:4,000
- Inland Water Line - 1:0 to 1:4,000
- Property Closing Line - 1:0 to 1:4,000


4 **Save** your new Layer and repeat these steps to publish all the OSMM tables as layers within GeoServer.

<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_topographicarea	✓	EPSG:27700
<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_cartographicsymbol	✓	EPSG:27700
<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_cartographictext	✓	EPSG:27700
<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_topographicline	✓	EPSG:27700
<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_topographicpoint	✓	EPSG:27700
<input type="checkbox"/>		dynamicmaps	PostGIS OSMM	PostGIS_OSMM_Leicester_boundaryline	✓	EPSG:27700

5 To test that the Layers are correct choose → **Data** → **Preview Layers** and preview each of your new layers in OpenLayers.

WMS version: 1.1.1 | Tiling: Single tile | Antialias: Full | Format: PNG 24bit | Styles: Default | Width/Height: 1000 / 500

Filter: CQL |



Scale = 1 : 533

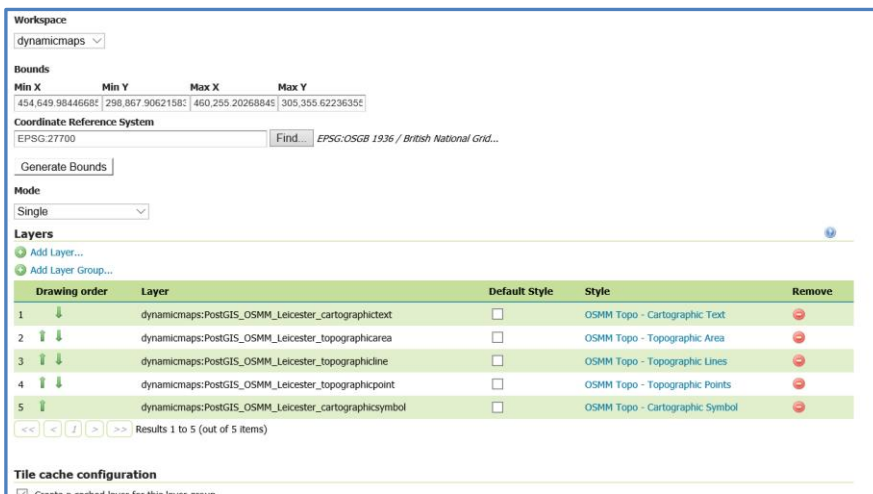
PostGIS_OSMM_Leicester_topographicarea

fid	fid	featurecode	version	versiondate	broken	calculatedareavalue	make	physicallevel	physicalpresence	styl
PostGIS_OSMM_Leicester_topographicarea.58367	osgb1000002097642928	10056	4	2015-09-10		36079.548126	Natural	50		Nat

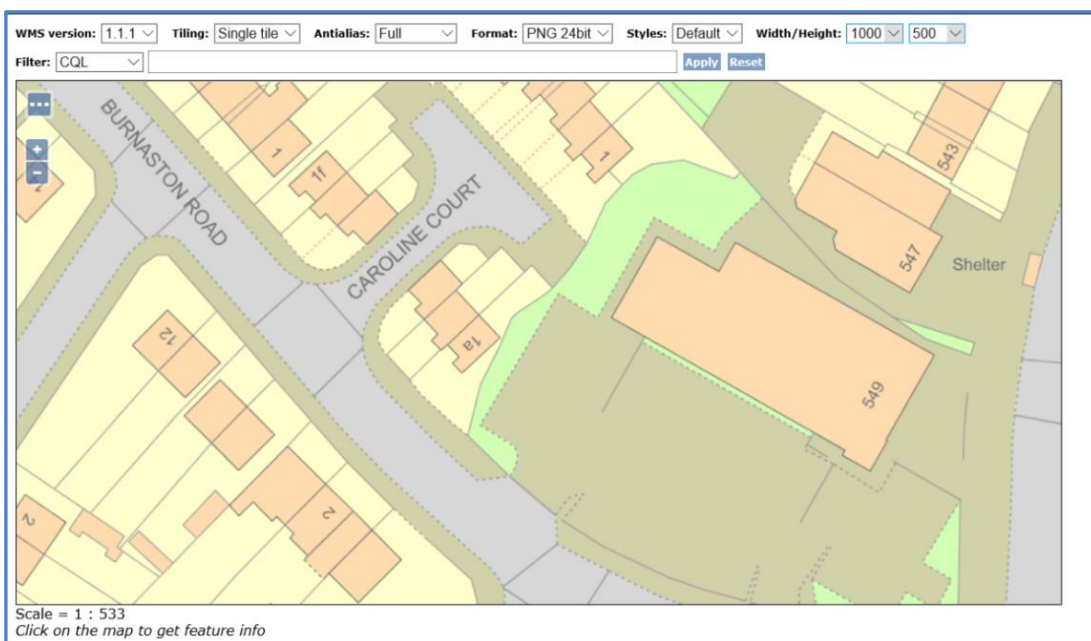
Create a Group Layer



- 1 In GeoServer → choose → **Data** → **Layer Groups** → **Add a New Layer Group Resource**.
- 2 Specify a → **Name** and → **Title** for the New Layer Group and then in the **Layers** section → add the OSMM Layers one by one.



- 3 Choose the required → **order** in which you would like the individual components of the Group Layer to be drawn.
- 4 Then finally update the → **Bounding Box** based on the layers you have added.
- 5 Save the New Layer Group and to test that the Layers are correct choose → Data → Preview Layers and preview the new Group Layer.



- 6 You can now use the WMS to add your styled OSMM data into the client application of your choice. For example, below, I have added the WMS layer into the **DynamicMaps webGIS – MapThat**.



- 7 If required, you can edit the default styles provided by OS to make changes. For example you may wish to change the building outlines from grey to black.
- 8 To do this, simply choose to edit the → corresponding **Style file**, find the → **Style Code** reference e.g. Building Outline is the value of **2**, and then edit the → **stroke colour** as required. E.g. the default light grey (#6a6a6a) could be changed to #000000 to represent a black line instead.

Copy from existing style

Choose One Copy ...

12pt

```

33 <FeatureTypeStyle>
34 <Rule>
35 <Name>Building Outline - 1:0 to 1:4,000</Name>
36 <ogc:Filter>
37 <ogc:PropertyIsEqualTo>
38 <ogc:PropertyName>style_code</ogc:PropertyName>
39 <ogc:Literal>2</ogc:Literal>
40 </ogc:PropertyIsEqualTo>
41 </ogc:Filter>
42 <MinScaleDenominator>0</MinScaleDenominator>
43 <MaxScaleDenominator>4000</MaxScaleDenominator>
44 <LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">
45 <Stroke>
46 <CssParameter name="stroke">#6a6a6a</CssParameter>
47 <CssParameter name="stroke-width">0.07</CssParameter>
48 </Stroke>
49 </LineSymbolizer>
50 </Rule>
51 </FeatureTypeStyle>
52
53 <!-- Edge Line -->
                    
```

9 Save the changes and → **preview** the Layer in **Open Layers**.

