

Table of Contents

1 Getting started with Maria Map Maker.....	1
1.1 Concepts.....	1
1.2 Step by step instructions.....	2
2 Vector tiles accuracy.....	5
2.1 Maximum accuracy versus zoom levels.....	5
3 Color map layers.....	7
3.1 Palette.....	7
3.2 Data source.....	10
4 Maria Map Maker CLI.....	12
4.1 Commands.....	12
4.2 Preparation parameters.....	16
4.3 Exit codes.....	17
4.4 Examples.....	17
5 Maria Map Maker Map tools.....	19
5.1 Map right-click.....	19
5.2 Tiles export.....	19
5.3 Feature query.....	20
5.4 Distance / elevation tool.....	20
5.5 Elevation shading.....	21
5.6 Undock map.....	21
6 Tiles export.....	22
6.1 Usage.....	22
7 Maria Map Maker release notes.....	24
7.1 4.6.0.....	24
7.2 4.5.0.....	24
7.3 4.4.0.....	24
8 Maria Map Maker settings.....	25
8.1 Workspaces.....	25
8.2 Map.....	25
8.3 Product.....	26
8.4 Sources.....	26
8.5 Options.....	27
9 Setting files.....	28
9.1 MariaMapMakerConfig.xml.....	28
9.2 MariaMapMakerUserSettings.xml.....	30
9.3 MariaMapMakerAutoCompleteConfig.xml.....	31

Table of Contents

9 Setting files	
9.4 Settings.json.....	32
9.5 Version control.....	32
10 Products.....	33
10.1 Product organization.....	33
10.2 Versioning.....	34
10.3 Contents.....	34
10.4 Dependencies.....	35
10.5 Product metadata.....	36
11 Basemaps and overlays.....	38
11.1 Managing layers.....	38
12 Importing data.....	41
12.1 Common options.....	41
12.2 Map Tasks.....	42
12.3 Placeholder layer.....	43
12.4 Raster data (version 4.3 and older).....	43
12.5 Raster data (version 4.4 and newer).....	44
12.6 Elevation/Coverage data.....	45
12.7 Vector data.....	46
12.8 Location data.....	47
12.9 Routing databases.....	47
12.10 3D Tiles.....	49
13 Supported formats.....	50
14 Mapbox vector datasets.....	52
14.1 Import.....	52
14.2 Style layers.....	53
14.3 Duplicating datasets.....	60
15 MARIA GDK vector datasets.....	62
15.1 Importing data.....	62
15.2 Styling.....	65
16 Raster layers.....	74
17 Web layers.....	75
17.1 WM(T)S.....	75
17.2 TMS.....	75

Table of Contents

18 Dynamic layers.....	77
18.1 In the overlay.....	77
18.2 In the basemap.....	78
19 3D Tiles.....	79
19.1 Import.....	79
19.2 Navigation.....	80
19.3 3D tilesets.....	80

1 Getting started with Maria Map Maker

Maria Map Maker (M3) is an application to create, edit and manage maps. These maps can be used in other applications built with the MARIA Geo Development Kit (GDK).

This guide will take you the shortest path from first-time startup to a finished map. You will also hopefully get a feel for the workflow in M3.

1.1 Concepts

We begin with a quick explanation of a few key concepts in M3. These terms will probably sound familiar, but we will explain what they mean in M3.

1.1.1 Workspace



A workspace is simply a folder containing one or more products. The workspace should only contain products created by M3 - no subfolders or loose files. Working with maps requires a lot of reading and writing from disk. This means that you should create your workspace on a fast storage device (such as a local SSD) if possible.

1.1.2 Product



The product is the basic container for map data and all related files. It also contains metadata describing the product and its contents. The product usually contains one or more datasets with the same theme, covering a specific geographical area.

The product consists of a folder on disk, with a specific naming scheme: *<Product name>_<Sequence number>_<Version ID>*

Example: *NorwayTopoRaster_v25_db95990c*

1.1.3 Basemap/overlay



The basemap and the overlay are lists of map layers, where each layer is a reference to an existing raster or vector dataset, or the address of a web map. Information about visibility, scales and draw order is also stored here.

Basemaps and overlays are the starting points for editing maps in M3. They are also the entry point to your maps when they are used in other MARIA GDK applications.

Example: *Norway Topo Raster N50 - N5000*

1.1.4 Dataset



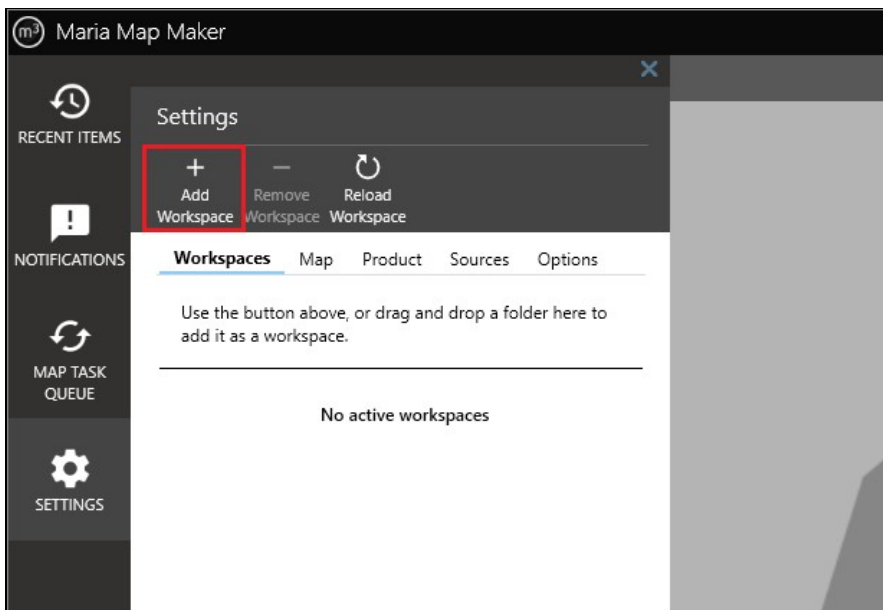
The dataset is the actual geographical data, with accompanying styling information and metadata. Datasets are always stored in a product. A dataset comes in a variety of types in M3:

- Vector
- Raster
- Elevation
- 3D
- Location
- Routing

1.2 Step by step instructions

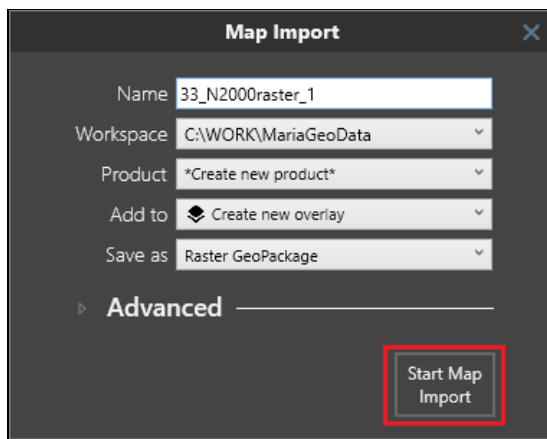
1.2.1 1. Set up workspace

When starting up Maria Map Maker for the first time, it's necessary to define a workspace. This could be a folder where you have some existing products, or an empty folder where you intend to create new products. Click the **SETTINGS** button on the left sidebar, and the **Workspaces** tab will be the first option. Each workspace you add gets its own button on the sidebar.



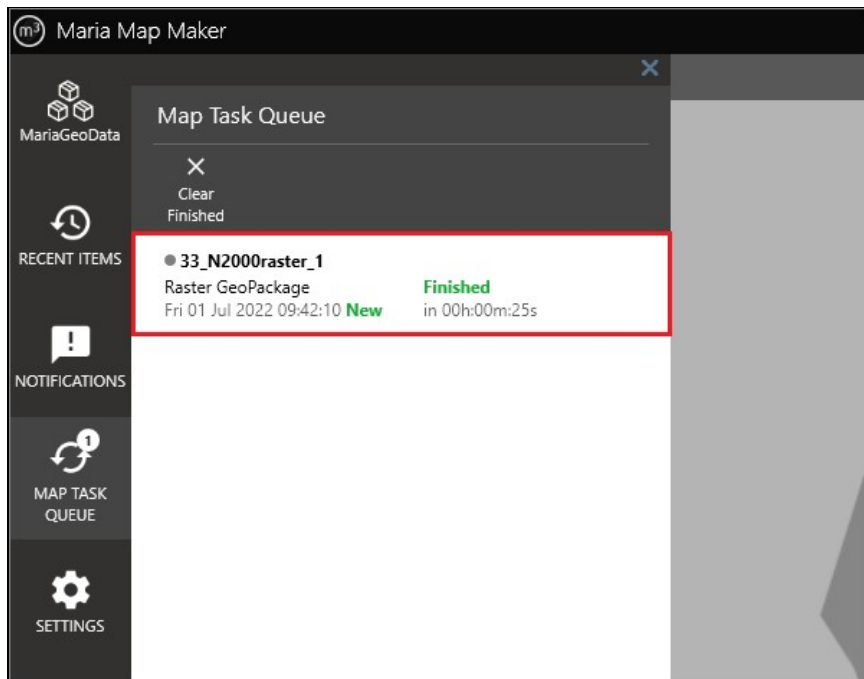
1.2.2 2. Import data

Now that you have a workspace configured, you can start importing data. In this example, we have a simple GeoTIFF file. The quickest way to import it is to simply drag and drop it on the map area. This will open the **Map Import** dialog. We will leave all the options as they are. This will automatically create a new *product* and a new *overlay* for the dataset we're importing.



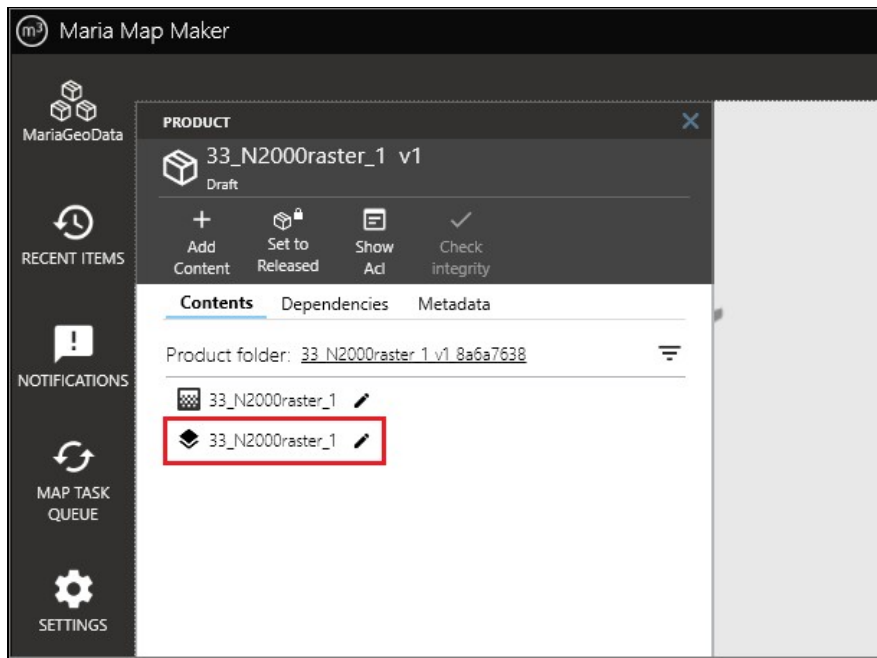
1.2.3 3. Map task

Starting the import creates a Map Task, which you can follow in the **Map Task Queue** window. When the Map Task is finished, you can double-click it to open the new product it was imported to.



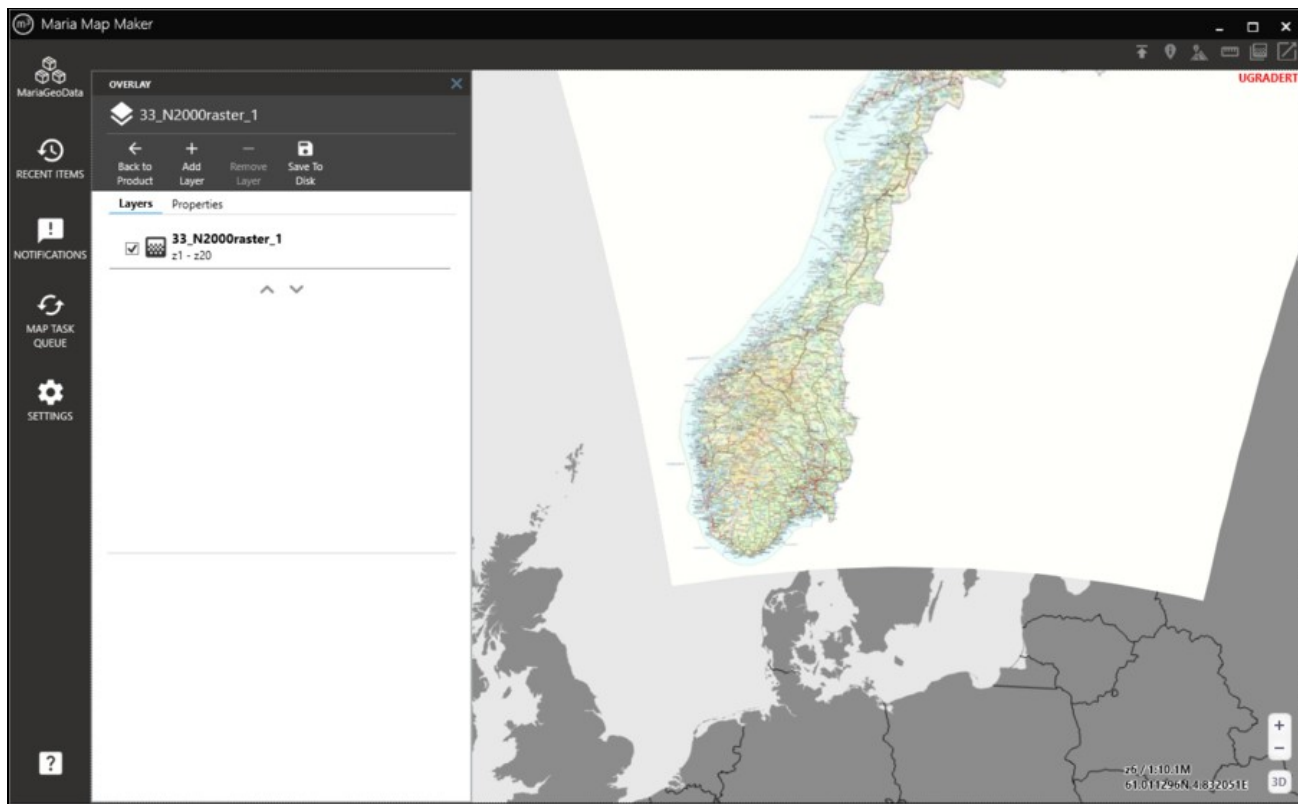
1.2.4 4. Open overlay

You now see the contents of the product - a raster dataset and an overlay. All editing and viewing of map datasets is done via the overlay, so double-click it to see your dataset in the map.



1.2.5 5. Finished

You are now looking at the overlay. It contains a single layer - a reference to the raster dataset you just imported. The dataset should be visible in the map area. Your product is now ready to be used in any application based on MARIA GDK.



2 Vector tiles accuracy

Vector tiles is the format used in Maria GDK to display vector data. This format is slightly different from other vector formats because coordinates are expressed in a local tile coordinate system, consisting of 4096x4096 units. This means that the vertexes of source data are "snapped" to the nearest unit in a tile, introducing a small amount of error. For each increasing zoom level, each tile is divided into four, reducing the error introduced. The vector tile format is intended for presentation, and the size of the tile coordinate system was chosen so that the error should always be smaller than the size of a pixel on the screen. Still, it can be useful to know the maximum accuracy of vector vector tiles at various zoom levels. This does not consider the accuracy of the source data in relation to the real-world objects they represent, which is not relevant in this context.

Note that these numbers describe the accuracy of the data stored in a vector tiles file. If the overzoom function is used in an application, the accuracy in meters remains that of the highest zoom level, but it decreases relative to the size of a pixel on screen.

2.1 Maximum accuracy versus zoom levels

This table is based on the tile matrix of a GeoPackage with projection *EPSG:3857* and *GoogleMapsCompatible* tiling. Each zoom level has a stated pixel size in meters, and each tile has a size of 256 x 256 pixels. In Maria GDK, a vector tile has the same size as a raster tile in pixels. We can calculate the "pixel size" of a vector tile by multiplying the raster tile pixel size with the ratio of 256 / 4096.

The maximum error introduced when converting to vector tiles is when the original vertex is located exactly in the middle of four tile units. In this case, the error can be calculated using Pythagoras:

```
max_error = sqrt((unit_size / 2)**2 + (unit_size / 2)**2)
```

Zoom level	Raster pixel size	Vector tile unit size	Vector tile max error
0	156543.0339	9783.9396	6918.2901
1	78271.517	4891.9698	3459.145
2	39135.7585	2445.9849	1729.5725
3	19567.8792	1222.9925	864.7863
4	9783.9396	611.4962	432.3931
5	4891.9698	305.7481	216.1966
6	2445.9849	152.8741	108.0983
7	1222.9925	76.437	54.0491
8	611.4962	38.2185	27.0246
9	305.7481	19.1093	13.5123
10	152.8741	9.5546	6.7561
11	76.437	4.7773	3.3781
12	38.2185	2.3887	1.689
13	19.1093	1.1943	0.8445
14	9.5546	0.5972	0.4223

Zoom level	Raster pixel size	Vector tile unit size	Vector tile max error
15	4.7773	0.2986	0.2111
16	2.3887	0.1493	0.1056
17	1.1943	0.0746	0.0528
18	0.5972	0.0373	0.0264
19	0.2986	0.0187	0.0132

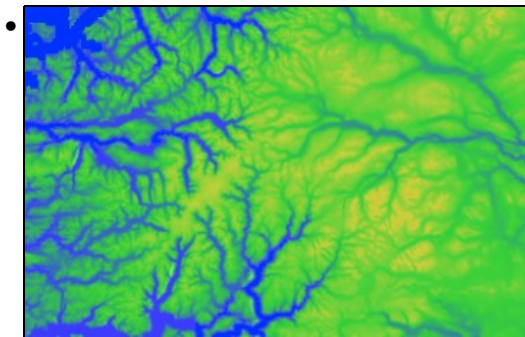
3 Color map layers



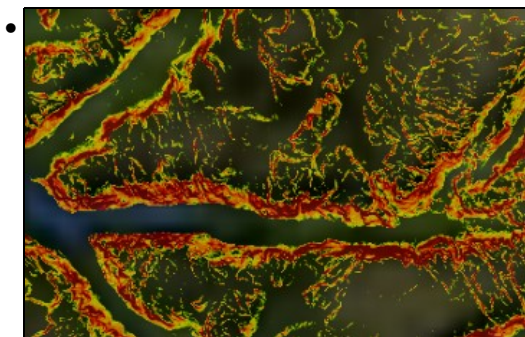
From version 4.4

Color map layers are defined in overlay/basemaps, and visualize **coverage datasets** such as elevation, census, land cover, etc. The values in the datasets are matched with colors in a palette in the color map layer. Colors may be mapped directly to data values, or there may be a calculation involved. It is for example possible to assign colors based on slope (gradient) or aspect (direction of slope) of elevation datasets.

In a basemap/overlay, these layers can be added by clicking **Add Layer ? Color map layer**.



Elevation values



Slope values



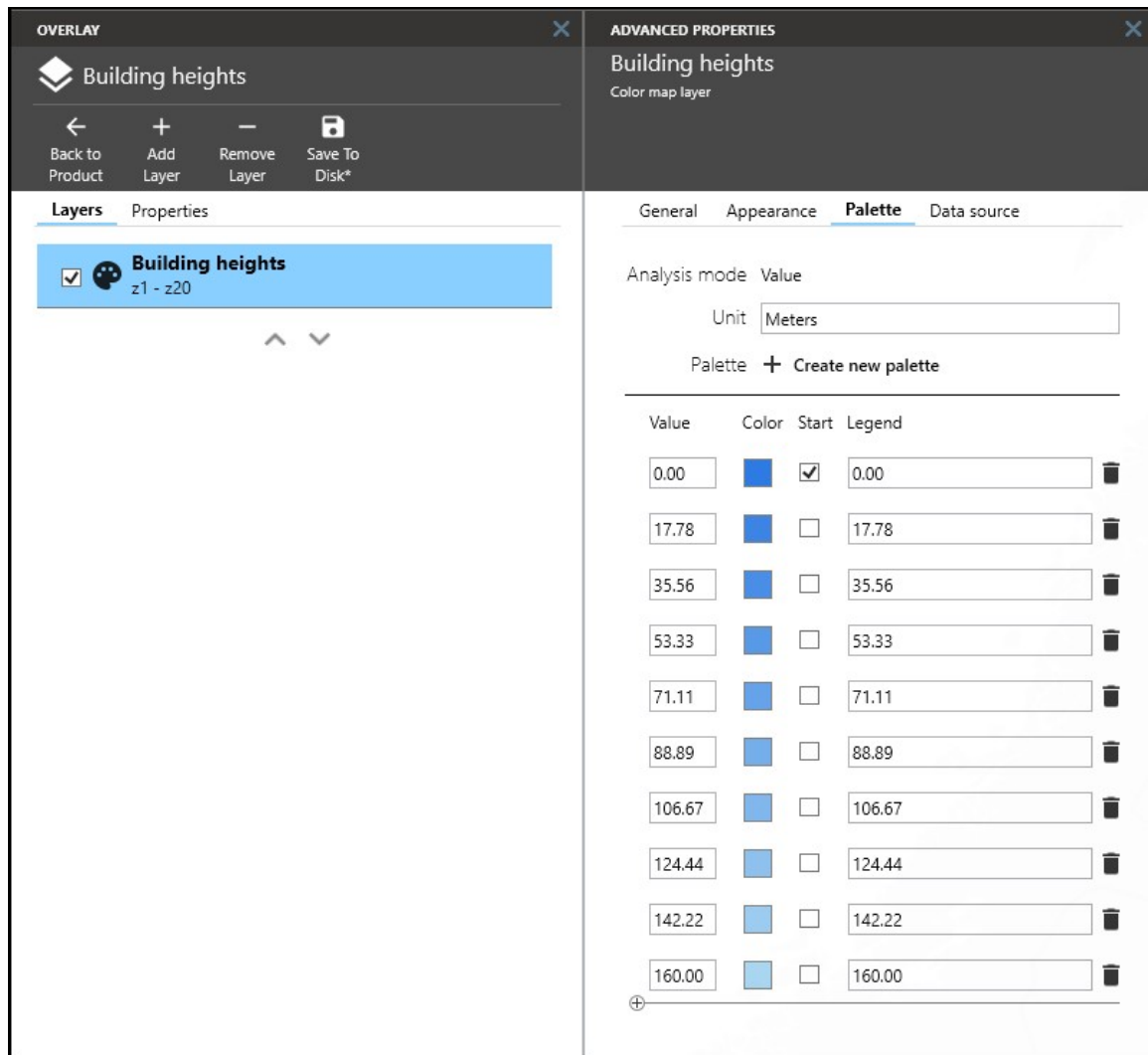
Aspect values

3.1 Palette

Color palettes are defined on the **Palette** tab, where either a new palette can be created or an existing palette edited. For each entry in the palette, you can edit **Value**, **Color** and **Start range**. You can delete an entry by clicking the **Trashcan**

button, or insert an entry at any point by mousing over the palette and using the **insert indicator**.

Start range can be used to indicate that an entry should define the start of a range. This means that there will be no color interpolation to the previous entry. If there are no previous entry, then any pixel with a lower value will be transparent.



3.1.1 Create new palette

To create a new palette, click the **Create new palette** button. This will open a new dialog with several options.

3.1.1.1 Palette

The type of palette to create:

- Custom, a user defined palette for color map.
- Elevation, a predefined palette for elevation color map.
- Bathymetry, a predefined palette for bathymetry color map.
- Slope, a predefined palette for slope color map.
- Aspect, a predefined palette for aspect color map.

When a custom palette is selected, several options are available.

3.1.1.2 Analysis mode

The function used to calculate the color of each pixel.

- *Elevation* where each pixel is colored according to its elevation value.
- *Slope* where each pixel is colored according to its gradient or slope value.
- *Aspect* where each pixel is colored according to the direction of slope.
- *Value* where each pixel is colored according to its elevation value (same as *Elevation*).

3.1.1.2.1 Unit

The unit of the data which is being visualized. This is only used for display in a legend.

3.1.1.3 Min value

The minimum value of the color range.

3.1.1.4 Max value

The maximum value of the color range.

3.1.1.5 Entries

The number of entries in the palette.

3.1.1.5.1 Display precision

The number of decimals to use in the **Value** and **Legend** fields in the palette

3.1.1.6 Color ramp start

The start color of the color ramp. This color is mapped to the *Min value*.

3.1.1.7 Color ramp end

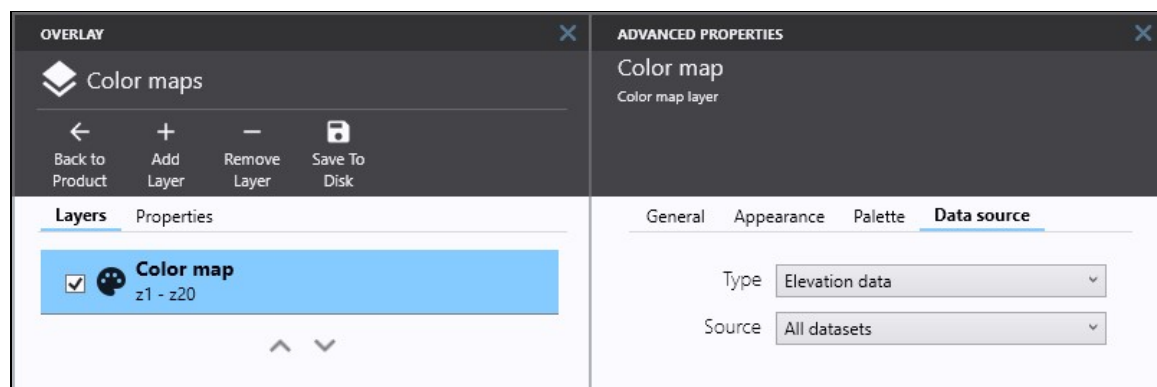
The end color of the color ramp. This color is mapped to the *Max value*.

3.2 Data source

Data sources are defined on the **Data source** tab. A data source is either elevation data or generic data (for example land cover, census, etc.) and can be selected from **Type**. The data source can be configured to use all available data sets, specific data sets or data sets based on tags.

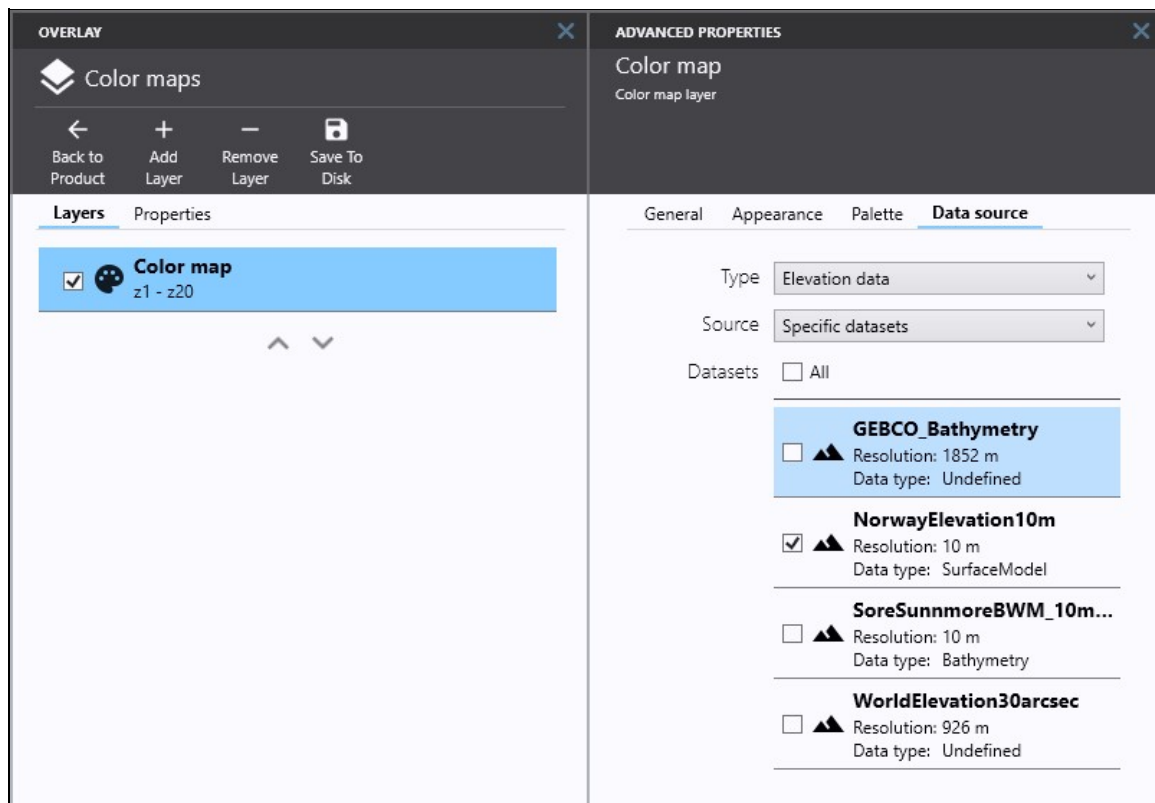
3.2.1 Data source from all datasets

To use all available datasets as data source select **All datasets** from **Source**.



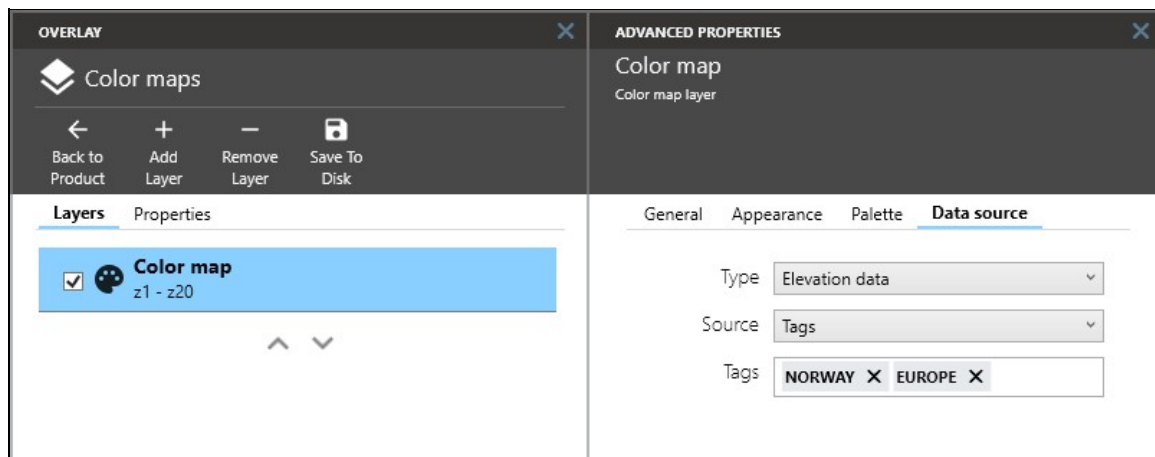
3.2.2 Data source from specific datasets

To use specific datasets as data source select **Specific datasets** from **Source** and then select the datasets to include from the list of available datasets.



3.2.3 Data source from tags

To use tagged datasets as data source select **Tags** from **Source** and then add the tags of the datasets to include.



4 Maria Map Maker CLI

From version 4.4

MariaMapMakerCLI (command line interface) is a tool to create and update products and datasets. It offers a subset of the functionality of Maria Map Maker. It can be used in scripts to automate the process of frequently updating datasets. It can also be integrated into other systems to produce Maria GDK compatible products and datasets.

The tool is distributed with the main Maria Map Maker application.

4.1 Commands

The tool divides functionality into 4 main commands. Each of these have a set of options, listed below. All commands and options are case insensitive. To see an overview of all commands, run `MariaMapMakerCLI.exe --help`

4.1.1 New-Product

Create a new product. This command should write the full path to the new product, as well as its product ID, to StdOut, so it can be piped to a subsequent command.

4.1.1.1 -n, --name

The name of the new product.

4.1.1.2 -p, --path

Path where the new product will be created. If not provided, the application will use the current directory.

4.1.1.3 -pmd, --productmetadata

Product metadata. Enter a valid product metadata key and value on the form `-pmd "key=value"`. Valid keys are:

- `displayname`
- `displayversion`
- `previewimage`
- `description`
- `virtualpath`

4.1.2 Update-Product

Update the metadata of a product, or change its status. The command should write the full path to the updated product, as well as its product ID, to StdOut, so it can be piped to a subsequent command.

4.1.2.1 `-p, --path`

Path to a valid product to update, or the path to a workspace containing products. If the path is a workspace, then option `--productid` must also be specified. If not provided, the application will use the current directory.

4.1.2.2 `-id, --productid`

The product ID to update. Requires that `--path` is specified to a workspace containing the specified product ID.

4.1.2.3 `-pmd, --productmetadata`

Product metadata. Enter a valid product metadata key and value on the form `-pmd "key=value"`. Valid keys are:

- `displayname`
- `displayversion`
- `source`
- `previewimage`
- `description`
- `virtualpath`

4.1.2.4 `-s, --setstatus`

Set the status of the product to `draft` or `released`.

4.1.2.5 `-m, --generatemanifest`

Generate a new manifest in the product. This can be done on a released product without changing its status. Valid values are `true` or `false`. Default `true`.

4.1.2.6 `-acl, --generateacl`

Generate a new accompanying letter with a summary of the product content. This can be done on a released product without changing its status. Valid values are `true` or `false`. Default `true`.

4.1.3 Query-Product

From version 4.6

Print information about products and datasets in a folder.

`-p, --path`

Path to product or a folder containing products. If not provided, the application will use the current directory.

4.1.4 New-Dataset

Create a new dataset by importing a georeferenced dataset. The MapPreparation add-in will be used to convert the source data to the selected output format. An overlay referencing the dataset will also be created.

4.1.4.1 `-n, --name`

The name of the new dataset.

4.1.4.2 `-p, --path`

Path to a valid product to create the dataset in, or the path to a workspace containing products. If the path is a workspace, then option `--productid` must also be specified. If not provided, the application will use the current directory.

4.1.4.3 `-id, --productid`

The product ID to create a new dataset in. Requires that `--path` is specified to a workspace containing the specified product ID.

4.1.4.4 `-s, --sourcedata`

File or directory containing supported georeferenced data to import.

4.1.4.5 `-dmd, --datasetmetadata`

Dataset metadata. Enter a metadata key and value on the form `-dmd "key=value"`.

4.1.4.6 `-ps, --prepscript`

The MapPreparation script to use when importing the data. The names are found in the `displayname` attribute in the MapPreparation script XML's. Default behaviour is to use the first valid script for the chosen source data.

4.1.4.7 `-pp, --prepparam`

The value of a parameter for the chosen `--prepscript`. Should be on the form `--prepparam "key=value"`.

4.1.4.8 `-i, --includesourcedata`

Include the source data in a sub-folder "sourcedata" in the dataset folder. This folder will always be cleaned out before updating a dataset. Valid values are `true` or `false`. Default is `false`.

4.1.4.9 `--addto`

The type of template to create to reference the dataset. Valid values are `overlay`, `basemap` and `none`. Default is `overlay`.

4.1.4.10 **--elevationdatatype**

The elevation data type. Valid values are `undefined`, `terrainmodel`, `surfacemodel` and `bathymetry`. Default is `terrainmodel`.

4.1.4.11 **--tags**

Comma separated list of text tags applicable to elevation data only.

4.1.4.12 **--epsg**

EPSG code of the source dataset if this information is not embedded.

4.1.5 **Update-Dataset**

Import source data, overwriting the data file in an existing dataset. This command shares many of the options of the `New-Dataset` command.

4.1.5.1 **--signature**

The unique map signature of the dataset to update.

4.1.5.2 **-p, --path**

Path to a valid product where the specified dataset can be found. If the path is a workspace, then option `--productid` must also be specified. If not provided, the application will use the current directory.

4.1.5.3 **-id, --productid**

The product ID where the specified dataset can be found. Requires that `--path` is specified to a workspace containing the specified product ID.

4.1.5.4 **-s, --sourcedata**

File or directory containing supported georeferenced data to import.

4.1.5.5 **-dmd, --datasetmetadata**

Dataset metadata. Enter a metadata key and value on the form `-dmd key=value`.

4.1.5.6 **--prepscript**

The MapPreparation script to use when importing the data. The names are found in the `displayname` attribute in the MapPreparation script XML's. Default behaviour is to use the first valid script for the chosen source data.

4.1.5.7 --prepparam

The value of a parameter for the chosen --prepscript. Should be on the form --prepparam "key=value". For an overview of the most common parameters, see [#Preparation parameters](#)

4.1.5.8 --includesourcedata

Include the source data in a sub-folder "sourcedata" in the dataset folder. This folder will always be cleaned out before updating a dataset. Valid values are true or false. Default is false.

4.1.5.9 --elevationdatatype

The elevation data type. Valid values are undefined, terrainmodel, surfacemodel and bathymetry. Default is terrainmodel.

4.1.5.10 --tags

Comma separated list of text tags applicable to elevation data only.

4.1.5.11 --epsg

EPSG code of the source dataset if this information is not embedded.

4.2 Preparation parameters

Preparation parameters (--prepparam) are key/value pairs which affect the result of the map import with the New-Dataset and Update-Dataset commands. The available parameters depend on the type of source data being imported. They are defined in the various map preparation scripts, which can be found at

{MariaMapMakerInstallDirectory}\ServiceHost\AddInRoot\AddIns\Preparation\scripts

The table below lists some of the most commonly used parameters along with valid values. For a more detailed explanation of these parameters, see [Importing data](#)

Import type	Parameter	Valid values
vector	minzoom	0-18
vector	maxzoom	0-18
vector	selectedlayerlist	Comma-separated list of layers to include
raster	srcnodata	"R G B" values to make transparent in imported data
raster	tile_format	auto, png, png8
raster	quality	10-100
coverage	datatype	16, 32
*coverage/elevation/raster	proccount	light, moderate, aggressive, extreme
*coverage/elevation/raster	clippingdataset	Path to vector dataset to clip imported data
*coverage/elevation/raster	resampling	Average, Nearest, Bilinear, Cubic, Lanczos, Gauss, CubicSpline

** Available for coverage/elevation imports from version 4.6*

4.3 Exit codes

Exit codes are output from MariaMapMakerCLI to indicate the result of the command.

Code	Description
0	OK
1	Command options error
2	Unknown product
3	Service error
4	Service settings error
5	Illegal operation on released product
6	Missing source data
7	Unknown error

4.4 Examples

Create a new product and then import a dataset into it:

```
MariaMapMakerCLI.exe New-Product --name "AIRAC export" -pmd "displayversion=2023-04" -pmd "virtualpath=NorwayTopoRaster_v32_859f2d06"
MariaMapMakerCLI.exe New-Dataset --path "c:\myproduct\AIRAC_export_v1_asf45345" --sourcedata "c:\srcfiles\N5000 Raster TIFF-format";
```

Set a released product to "Draft", update a dataset, and set to "Released":

```
MariaMapMakerCLI.exe Update-Product --path "c:\products\" --productid "9075c2fb-ef9f-4300-9008-83f4ecef0ec3" --setstatus "draft"
MariaMapMakerCLI.exe Update-Dataset --path "c:\products\" --productid "9075c2fb-ef9f-4300-9008-83f4ecef0ec3" --sourcedata "c:\srcfiles\N5000 Raster TIFF-format";
MariaMapMakerCLI.exe Update-Product --path "c:\products\" --productid "9075c2fb-ef9f-4300-9008-83f4ecef0ec3" --setstatus "released"
```

The following example is a Powershell script which sets a specified product to "Draft", updates each dataset with new data, updates metadata, and finally sets the product to "Released".

```
$m3cli = "MariaMapMakerCLI.exe"
$productpath = "F:\workspace\NorwayTopo-update-test\NorwayTopoRaster_v32_859f2d06"

# Hashtable of map signatures in the product and corresponding source data folder
$updates = @{
    N5000 = "F:\Geotiff_hovedkartserien\N5000 Raster TIFF-format";
    N2000 = "F:\Geotiff_hovedkartserien\N2000 Raster TIFF-format";
    N1000 = "F:\Geotiff_hovedkartserien\N1000 Raster TIFF-format";
    N500 = "F:\Geotiff_hovedkartserien\N500 Raster TIFF-format";
    N250 = "F:\Geotiff_hovedkartserien\N250 Raster TIFF-format";
}

# Set product to draft. This command returns two values:
# the new product path [0] and the product ID [1].
[array]$product = . $m3cli Update-Product --path $productpath --setstatus "draft"

# Update each dataset in turn
foreach($signature in $updates.Keys)
{
    $m3cli Update-Dataset --path $productpath --productid $product[1] --sourcedata $signature
}
```

```

Write-host "Updating $signature with data from $($updates[$signature])"
. $m3cli Update-Dataset `
    --path $product[0] `
    --signature $signature
    --sourcedata $($updates[$signature]) `
    --prepparam "srcnodata=255 255 252" `
    --prepparam "tile_format=PNG8" `
    --datasetmetadata "datadate=2023-11-01" `
    --datasetmetadata "metadatadate=2023-11-15"

}

# Update product metadata and release
. $m3cli Update-Product --path $product[0] --productmetadata "displayversion=2023-11"
. $m3cli Update-Product --path $product[0] --setstatus "released"

```

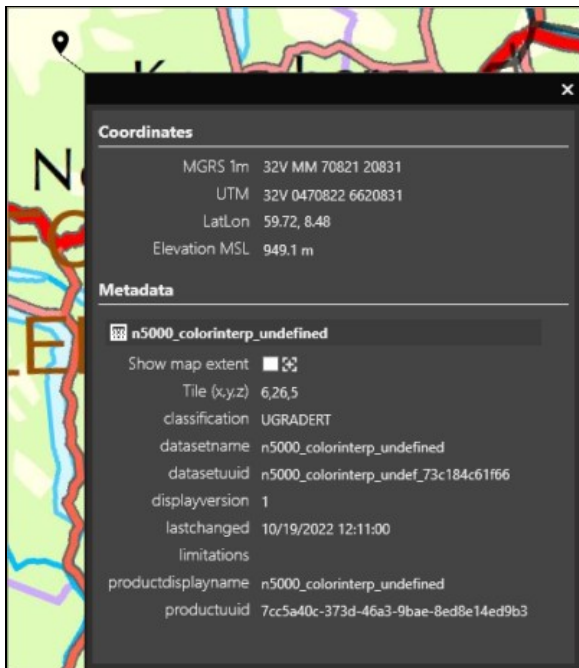
5 Maria Map Maker Map tools

Maria Map Maker includes a few tools for changing or getting information from the map. Most of these tools can be reached from the top-right toolbar. Some tools will not be available in 3D mode.



5.1 Map right-click

Right clicking the map will bring up a context menu, showing information on the exact geographic point which was clicked. Position in several common coordinate system is shown, in addition to a list of datasets which cover the clicked point. For each of these datasets, you can activate a coverage rectangle, and zoom to the extents.



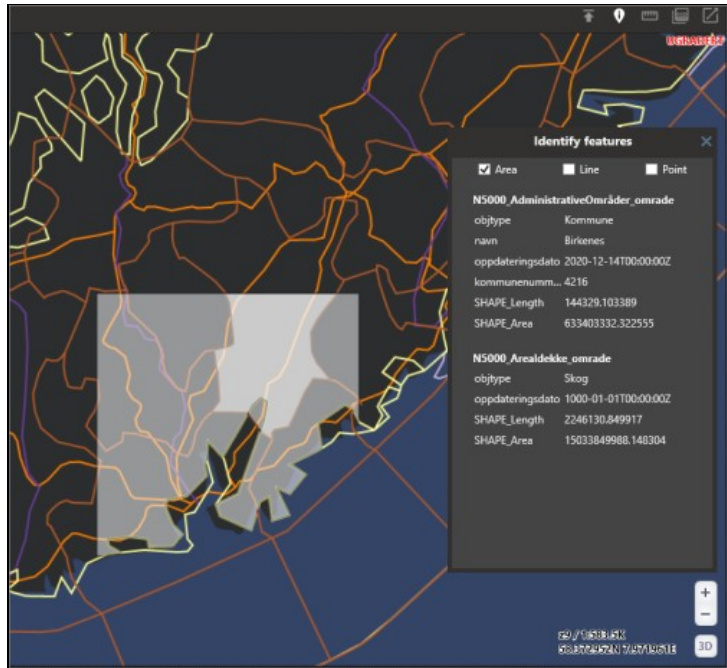
5.2 Tiles export

The Tiles export functionality is documented in a separate article: [Tiles export](#)

5.3 Feature query

Feature query is used to display the attributes of features in a vector dataset. When the tool is active, you can click on features to display their attribute data in a separate window. The geometry type can be filtered, so that only the specified type (points, lines, or areas) will return attributes.

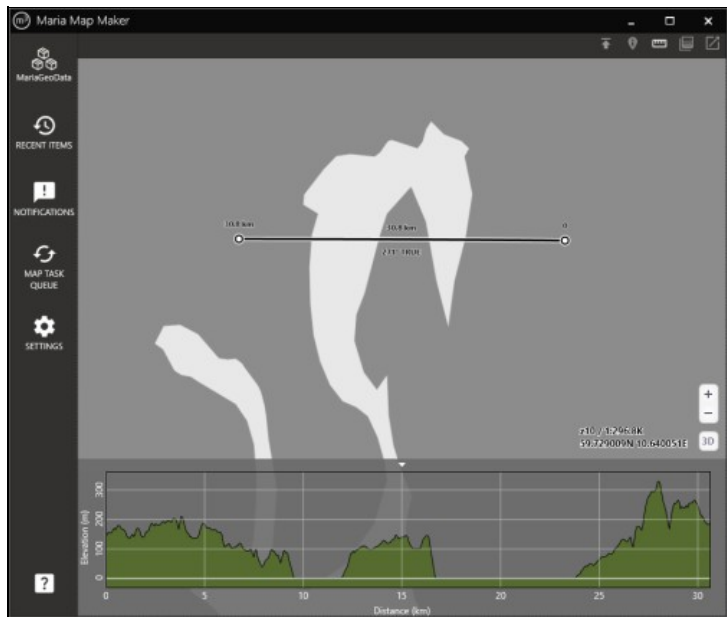
This tool is only enabled when a basemap/overlay is open.



5.4 Distance / elevation tool

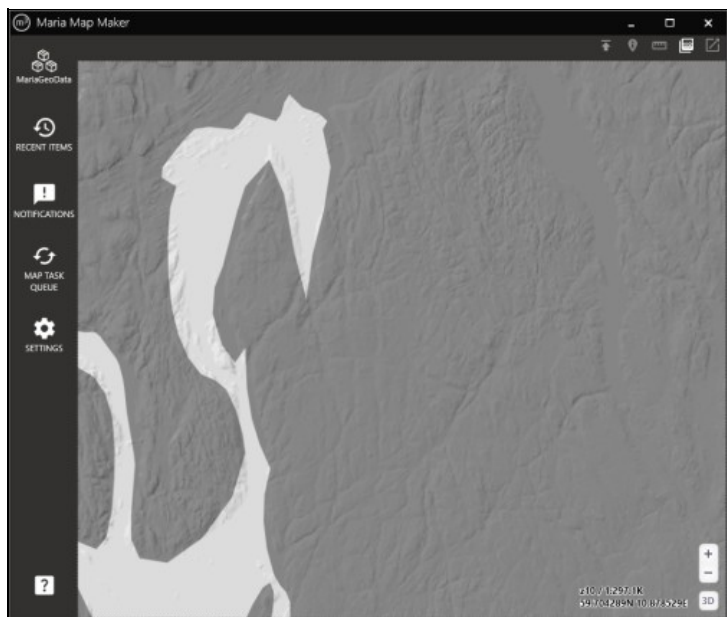
When the distance / elevation tool is active, you can left-click anywhere in the map to create the start point for a length measurement. Subsequent clicks creates waypoints, and a double-click completes the line. Each leg is labeled with its length, as well as the total length on the final point.

An elevation profile is also calculated along the line. The elevation is based on the highest available elevation datasets in your workspace.



5.5 Elevation shading

Elevation shading will be generated based on the highest available elevation datasets in your workspace. It will blend with any other map data which is currently displayed.



5.6 Undock map

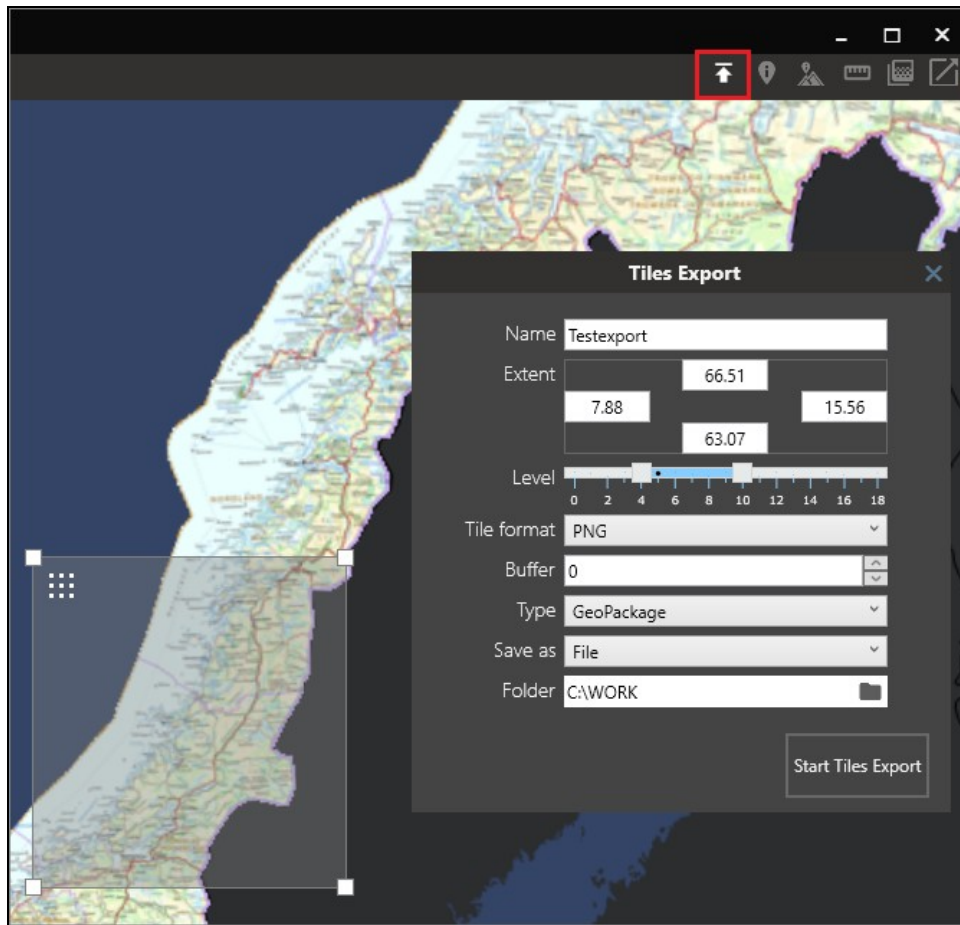
The map can be undocked into a new window, separate from the rest of the application. The same button will dock the map back into the application.

6 Tiles export

From version 4.1

The **Tiles export** tool allows you to extract a region of the 2D map in a basemap or overlay, and store it in a GeoPackage or MBTiles file. Everything displayed is exported, with the exception of the M3 background map. The resulting files will have web mercator (EPSG:3857) projection and *GoogleMapsCompatible* tiling scheme.

6.1 Usage



The tool can be started when a basemap or overlay is open, using the button on the top right toolbar.

6.1.1 Extent

You can drag a bounding box in the map to select the area to export. You can also type geographic coordinates for the extents of the area.

6.1.2 Min/max level

The minimum and maximum zoom levels to fetch tiles from within the selected extent. For reference, your current zoom level and map scale is always displayed in the bottom right corner of the map. Remember that for each zoom level increment, the number of tiles are quadrupled.

6.1.3 Tile format

Format to use when saving tiles:

- Auto - Mix of PNG and JPEG. PNG is used for tiles with transparency information, JPEG is used for fully opaque tiles.
- PNG - Lossless format, larger size. Best suited for thematic raster data with few unique colors.
- JPEG - Lossy format, smaller size. Best suited for imagery and maps with shading. Any transparency information in the source tiles will be lost. Transparent areas will be filled with black (0, 0, 0).

6.1.4 Buffer

Extra pixels to include around each tile. Use this for vector maps with labels, if the labels are cut off at tile borders.

6.1.5 Type

File type for the resulting database.

6.1.6 Save as

Create a single file in a chosen folder, or create an M3 product (parameters for creating the product are explained [here](#))

6.1.7 Elevation shading

Elevation shading can be included in the output file, generated from the highest resolution elevation dataset available in your workspace. Simply make sure that elevation shading is active before starting the tiles export. Use the button on the top right toolbar to activate elevation shading.

7 Maria Map Maker release notes

7.1 4.6.0

- The engine for importing raster GeoPackages (*massivegeopackage*) is now also implemented for importing coverage and elevation data. This means faster imports and the option to clip the output using a vector dataset.
- New option *Tiling scheme* which allows *GoogleMapsCompatible* tiling for raster, coverage and elevation datasets. This entails that the source dataset is reprojected to Web Mercator and resampled to the global GoogleMapsCompatible tiling scheme. This will increase processing time and may lower the resolution of the dataset, but it will also give an increase in performance in the output dataset. It can also be useful if the source dataset uses a spatial reference system which is not supported by the Maria GDK core components.
- New setting to allow the application to remember map position and scale between runs.
- Elevation datasets are now previewed with a color map when opened.
- New command *Query-Product* in *MariaMapMakerCLI* lists information on products and datasets in a folder.
- Improved help messages in *MariaMapMakerCLI*.
- UI enhancements
- Bugfixes

7.2 4.5.0

- In the *Sources* tab, a source can be toggled on or off without removing it
- Bugfixes

7.3 4.4.0

- New engine for importing 8 bit raster GeoPackages (*massivegeopackage*). Large input datasets consisting of multiple files will be imported using multiple logical processors, potentially increasing performance by several factors.
- New type of GeoPackage dataset: *Coverage GeoPackage*. This type can be chosen when importing 16, 32 or 64 bit regular gridded data. It can represent any geographical phenomenon, such as land cover, temperature, heatmap, etc. These datasets can be visualized with *Color map layers* in overlays and templates.
- Mapbox format vector styling: new function *attribute fanout* can create a set of layers based on unique values of an attribute. It can also create layers based on set intervals in numerical attributes.
- Mapbox format vector styling: new logical operator *OR* can now be used in combination with *AND* to create more advanced filters.
- Right-click anywhere in the map to get geographical info about the clicked position, as well as information about visible datasets covering it.
- Separate command line tool to create and update products and datasets (*MariaMapMakerCLI*). Can be used in scripts on datasets which require frequent updates.
- New function to duplicate vector datasets within a product. The MBTiles data file is not duplicated, only referenced, so multiple datasets can be styled independently without multiple copies of the data.

8 Maria Map Maker settings

This article describes the settings window in Maria Map Maker, which can be opened at any time using the button on the left vertical bar. As with most options and settings in Maria Map Maker, each setting has a help text which is displayed by hovering the mouse over the title of the specific setting.

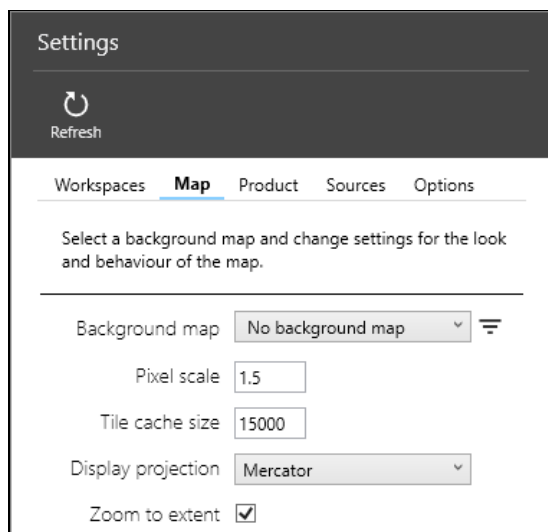
8.1 Workspaces

Add or remove workspaces. Each active workspace gets its own button on the left vertical bar. Note that adding or removing a workspace will automatically restart all the built-in map services, so you should save your work and make sure that there are no active map tasks before you do this.

If you have copied a product into your workspace or otherwise made changes outside Maria Map Maker, you can use the "Reload Workspace" button to restart the built-in map services. They will then be aware of the changes that you made.

8.2 Map

Settings related to how maps behave in Maria Map Maker. These settings applies to your M3 client only - they do not have any effect on the [products](#) you produce.



8.2.1 Background map

Background map which is always displayed at the bottom of your maps. Two simple maps are included with Maria Map Maker, but you can use any map as a background through an external source.

8.2.2 Pixel scale

When zooming in, this determines how much a tile is enlarged before it is re-drawn at a higher resolution. A smaller number has the effect of making the map look sharper, at the cost of performance.

8.2.3 Tile cache size

The amount of memory in megabytes allowed for storing tiles for re-use. A higher number can increase performance when panning and zooming in a limited geographical area.

8.2.4 Display projection

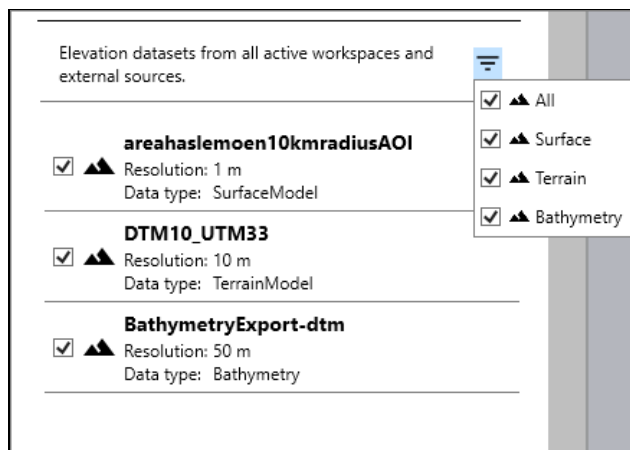
The projection to use when displaying maps. The default option **Mercator** is equivalent to the *Web mercator* (EPSG:3857) projection.

8.2.5 Zoom to extent

Automatically zoom to extents when opening basemaps, overlays and elevation datasets.

8.2.6 Elevation datasets

List of all elevation datasets available in your active workspaces and external sources. Datasets can be manually turned on or off, or filtered based on its type. These datasets are used as the basis for the elevation profile tool and shading when the map is in 2D mode. In 3D, they are used to model the terrain.



8.3 Product

Settings related to the **Accompanying letter**, which describes the contents of a product. The accompanying letter is created on the root of the product when it is released. It is an aggregation of metadata on each dataset, as well as the product itself.

8.4 Sources

Add an external Maria GDK server as a source. Any map data on the server will be available as background maps and elevation data in the `sources` section. The source URI should be on the form `http://ip:port`

8.5 Options

Settings related to the look and feel of the *fixed* windows in Maria Map Maker, such as the product window and basemap/overlay window. You can also change the path to use for temporary files when importing data. This setting requires the **Apply Settings** button to be used before it takes effect.

9 Setting files

This section documents the setting files used by Maria Map Maker. It is primarily intended for developers and advanced users.

Making invalid changes to the setting files can result in breaking the application, so be sure that you know what you are doing, and back up your files! If a setting is not explained here, it is controlled by the application and not intended to be changed by the user.

The files which M3 uses are located at %localappdata%\TeleplanGlobe\MariaMapMaker

9.1 MariaMapMakerConfig.xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<MariaMapMakerconfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  <version>1.0.0</version>
  <clientconfig>
    <productexporttypes>
      <exporttype>
        <type>Zip</type>
        <ext>zip</ext>
        <displaytype>Zip</displaytype>
      </exporttype>
      <exporttype>
        <type>ISO</type>
        <ext>iso</ext>
        <displaytype>ISO</displaytype>
      </exporttype>
      <exporttype>
        <type>Directory</type>
        <ext>*</ext>
        <displaytype>Directory</displaytype>
      </exporttype>
    </productexporttypes>
    <servicehoster internal="true" showwindow="false" />
    <lastworkspacefolder>C:\Data\MariaGeoData</lastworkspacefolder>
    <lastpreviewtilefolder>C:\Data\previewtiles</lastpreviewtilefolder>
    <tempdeviceminsize>5 GB</tempdeviceminsize>
    <lastwindowsize>1314,919.2</lastwindowsize>
    <windowstate>Maximized</windowstate>
  </clientconfig>
  <metadataconfig>
    <genericmetadata>
      <metadata key="classification" defaultvalue="UGRADERT" embedded="true">
        <legalvalues>
          <legalvalue value="NATO UNCLASSIFIED" />
          <legalvalue value="UGRADERT" />
          <legalvalue value="NATO RESTRICTED" />
          <legalvalue value="BEGRENSET" />
          <legalvalue value="NATO CONFIDENTIAL" />
          <legalvalue value="KONFIDENSIELT" />
          <legalvalue value="NATO SECRET" />
          <legalvalue value="HEMMELIG" />
        </legalvalues>
      </metadata>
    </genericmetadata>
  </metadataconfig>
</MariaMapMakerconfig>
```

```

        </legalvalues>
    </metadata>
    <metadata key="limitations" embedded="true" />
    <metadata key="designation" />
    <metadata key="datadate" date="true" />
    <metadata key="edition" />
    <metadata key="datasource" />
    <metadata key="name" />
    <metadata key="description" />
    <metadata key="expirationdate" />
    <metadata key="metadatadate" date="true" />
    <metadata key="metadataeditor" />
</genericmetadata>
<embeddedmetadata>
    <metadata key="productuuid" valuefrom="productuuid" hidden="true" />
    <metadata key="productdisplayname" valuefrom="productdisplayname" hidden="true" />
    <metadata key="displayversion" valuefrom="displayversion" hidden="true"/>
    <metadata key="datasetuuid" valuefrom="datasetuuid" hidden="true"/>
    <metadata key="datasetname" valuefrom="datasetname" hidden="true"/>
    <metadata key="lastchanged" valuefrom="lastchanged" hidden="true"/>
</embeddedmetadata>
</metadataconfig>
</MariaMapMakerconfig>

```

9.1.1 version

Semantic version used to compare config files. See [Version control](#)

9.1.2 productexporttypes

The child element `<exporttype>` defines a choice for the "Export Product" button in the Workspace window.

9.1.3 servicehoster

Attribute `internal="true/false"` indicates whether or not to use the service hoster which is built into M3. If the internal service hoster is not used, the application will attempt to connect to an external Windows service. This is not recommended outside debugging scenarios. Attribute `showwindow="true/false"` indicates whether or not to show the output window from the internal service hoster. This is also used for debugging purposes.

9.1.4 tempdeviceminsize

This number denotes the minimum amount of available storage space on the device which is used for temporary storage. If there is less space than this, it will not be possible to start new map imports.

9.1.5 genericmetadata

The `<metadata>` elements control the default metadata keys which appear in the Metadata tab on each dataset in M3. If the user fills them with a value, they will be written to the `<generic>` metadata of the dataset. See [Basemaps/overlays](#). There are a few ways to customize how each key appears to the user. Adding child element `<legalvalues>` with child elements `<legalvalue value="">` turns the key into a combo box, where the user is can choose from one of the legal

values, in addition to typing a value.

Adding attribute `<metadata date="true">` gives the key a date selector.

Adding attribute `<metadata embedded="true">` will add the key/value pair as child elements to the `<embedded>` element in the metadata file which follows the dataset.

9.1.6 embeddedmetadata

These metadata keys are intended to be automatically written to the `<embedded>` metadata on the dataset, and as such, these keys are not visible to the user in M3. The "valuefrom" attribute is used to automatically fetch values from the given variable. The available variable names to use with the "valuefrom" attribute are

- productname
- productsequence
- displayversion
- productdisplayname
- datasetuuid
- datasetname
- lastchanged
- releasedate
- maxclassification

Keys cannot contain spaces

9.2 MariaMapMakerUserSettings.xml

This file contains less consequential settings which are more related to a user's preferences. The settings in this file are mostly managed from the application.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<MariaMapMakerUserSettingsconfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <version>1.0.0</version>
  <usersettingsconfig>
    <openrepositories>
      <repository>C:\Data\MariaGeoData</repository>
    </openrepositories>
    <accompanyinglettersettings>
      <logoimagepath>C:\OneDrive\OneDrive - Teleplan AS\My Screen Shots\FMGT_logo_2017\original\logo_etatsv
      <supportinformation>
        <support name="Teleplan Globe" address="Fornebuveien 31" mail="info@teleplan.no" phonenumber="" />
      </supportinformation>
      <systemrequirements />
      <title>
        <metadatakey key="classification" type="highest" />
        <metadatakey key="limitations" type="unique" />
      </title>
    </accompanyinglettersettings>
    <recentlyimportedmaps>
      <importedmap jobid="c120d3a6-a5a9-4f99-99b5-e857f2b822ae" name="Buildings_Frogn" timestamp="2019-10-29T10:29:10.0000000" />
      <importedmap jobid="954598b8-e2f2-42d6-9e78-d02057e70313" name="finland_tactical_test" timestamp="2019-10-29T10:29:10.0000000" />
      <importedmap jobid="2c1b2efb-2e51-450a-b526-6bf688befa97" name="drobak_orto" timestamp="2019-10-29T10:29:10.0000000" />
    </recentlyimportedmaps>
  </usersettingsconfig>
</MariaMapMakerUserSettingsconfig>
```

```

        <importedmap jobid="63cf04be-4438-4ac0-9c8e-d4c63c3ac472" name="n50_akershus" timestamp="2019-10-29T
        <importedmap jobid="4ba995fd-488a-4001-93ff-e54ea351209b" name="20190821arctico" timestamp="2019-10-
    </recentlyimportedmaps>
    <zoomtoextent>false</zoomtoextent>
    <backgroundmap>
        <mapsignature>M3_background_05046c502cc3</mapsignature>
        <Visible>true</Visible>
    </backgroundmap>
</usersettingsconfig>
</MariaMapMakerUserSettingsconfig>

```

9.2.1 accompanyinglettersettings

The child element `<title>` defines the content of the sub-title in accompanying letters. Contains child elements `<metadatakey>`. Attribute `key` references a metadata key in the `<metadataconfig>` section in .

Attribute `type` decides how multiple datasets with the same key are handled. If `type="highest"`, then the highest value for that key is used. The ordering of the values is decided by the order in which they appear in a `<legalvalues>` element. The bottom value is the highest. See `<metadata key="classification">` for an example. Here, the classification value **HEMMELIG** is on the bottom, and thus it is the value which will appear in the accompanying letter subtitle.

If `<type="unique">`, then all unique values from the specified key will be listed in the accompanying letter subtitle.

9.3 MariaMapMakerAutoCompleteConfig.xml

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<MariaMapMakerAutoCompleteConfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
    <version>1.0.0</version>
    <dynamictags allowotherkeys="false">
        <tag key="IMAGERY-MAP"/>
        <tag key="IMAGERY-DETAIL"/>
        <tag key="TOPO-MAP"/>
        <tag key="TOPO-DETAIL"/>
        <tag key="SEA-MAP"/>
        <tag key="SEA-DETAIL"/>
        <tag key="AIR-MAP"/>
        <tag key="AIR-DETAIL"/>
    </dynamictags>
    <foldernames>
        <tag key="ELEVATION"/>
        <tag key="IMAGERY"/>
        <tag key="TOPO"/>
        <tag key="SEA"/>
        <tag key="DETAIL"/>
        <tag key="DEPTH"/>
        <tag key="LOCATION"/>
        <tag key="NAUTICAL"/>
        <tag key="OVERLAY"/>
        <tag key="BOUNDARIES"/>
        <tag key="ADMINISTRATIVE BOUNDARIES"/>
        <tag key="POLAND"/>
        <tag key="SWEDEN"/>
    </foldernames>

```

```
<tag key="DENMARK"/>
</foldernames>
</MariaMapMakerAutoCompleteConfig>
```

This configuration file allows you to define a set of values which will show up in an auto-complete list as the user types. Below is the text input fields which can be used with auto-complete.

9.3.1 dynamictags

Tags in dynamic layers and overlays. Attribute `allowotherkeys="false"` prevents the user from entering a value which is not pre-defined.

9.3.2 foldernames

Folder names for products.

9.4 Settings.json

This file is used by the built-in service hoster component of Maria Map Maker and the GDK WCF service hoster. The file is documented at [Service configuration](#). The settings.json file used by M3 is controlled by the application and there is no need for the user to change it.

9.5 Version control

The config files `MariaMapMakerConfig.xml` and `settings.json` both have a version number. On startup, M3 will check `%programdata%\TeleplanGlobe\MariaMapMaker` for config files with a higher version number. If a higher version exists, the file will be copied into `%localappdata%\TeleplanGlobe\MariaMapMaker` and it will be used as the active config file. This allows the application to be distributed with custom configuration settings.

10 Products



In M3, the product is a mandatory container for maps. It is a folder on the file system whose name consists of a user-defined name, sequence number, and a version ID. For example

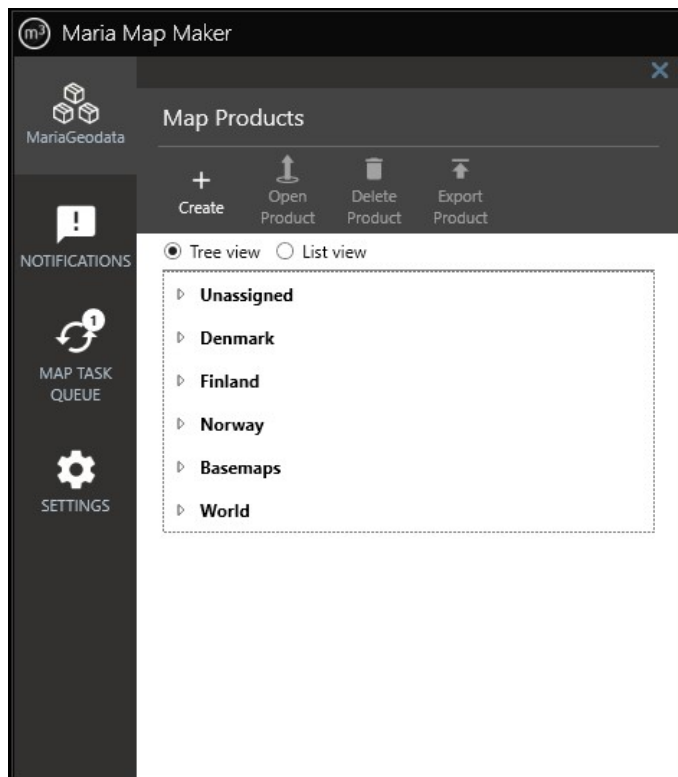
WorldNauticalDNC_v1_d04495f7

Every product contains a metadata file with extension `.productmetadata.xml`. Among other information, it contains the Product ID, which uniquely identifies the product. It is a randomly generated string of characters (**GUID**). The Product ID will remain unchanged even as the product is updated.

10.1 Product organization

Open a workspace to view a list of products. There are two ways to display this list. The **Tree view** lets you organize your products in "virtual" folders. You can create folders with the **Create** button, or right-clicking on an empty spot below the tree. Sub-folders are created by right-clicking existing folders. They can then be filled with products by dragging and dropping (You can't change the folder of a released product, however).

Organizing the products in virtual folders does not actually move them around in the file system. The virtual path is just a string which is stored in the product metadata file.



Products organized by coverage area

Conversely, the **List view** ignores the virtual path and displays all products in a flat list.

You can create new products, and delete or open existing ones.

You can delete multiple products by selecting them using the Ctrl key, and clicking **Delete Product**. Note that if a product is open, you can't delete it.

You can create a copy of a product by right-clicking it, and then assigning a new name. A copy of the product will be created, where all its contents (overlays, basemaps, datasets) will keep their names, but receive new unique id's.

10.2 Versioning

All products have three metadata elements which combined define the product version.

Status Either **Draft** or **Released**.

Version ID A shortened GUID which is re-generated every time the product changes status to **Draft**.

Sequence number Starts at 1 in new products, incremented by 1 every time a product changes status to **Draft**.

Changing the status of a product is done by opening the product and clicking **Set to Draft** or **Set to Released**

10.2.1 Set to Draft



You will be prompted to choose whether to change the status on a *copy* of the product, or to change the status on the product directly. The sequence number will be incremented and a new version ID will be generated.

The product and its contents are unlocked for editing.

10.2.2 Set to Released



You will be prompted to choose whether you want to create an **Accompanying letter**, which will be stored as an HTML file in the released product. You can edit some aspects of the accompanying letter from the vertical toolbar, with **SETTINGS ? Product**.

A file manifest `ProductName_v1_290d104e.productmanifest.xml` will also be generated, which allows the application using the product to verify that no files have been changed since release of the product.

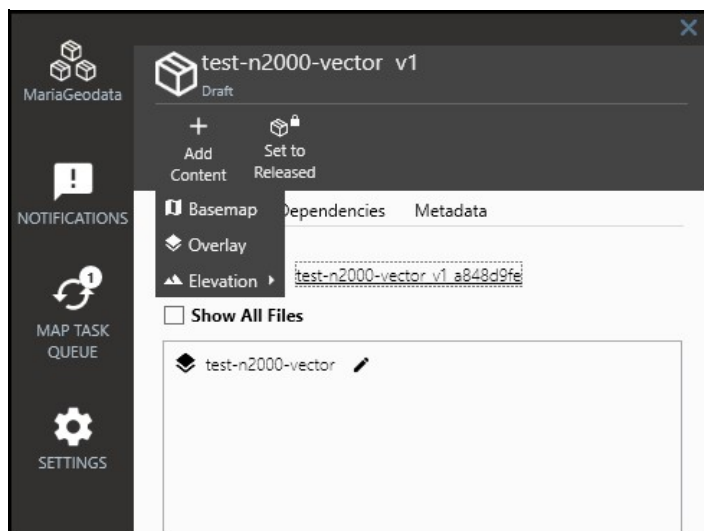
The product and all its contents are locked from editing. The only allowed change is setting the status back to **Draft** (It's still possible to open basemaps and overlays to view the maps).

10.3 Contents

The contents of a product can be viewed by opening it (it's the default tab). Here, any basemaps, overlays and datasets will be shown. They can in turn be opened by double-clicking or left-clicking the "edit"-symbol.

You can create new content in the product with **Add Content**. Choosing **Basemap** or **Overlay** will create the file and automatically open it. Choosing **Elevation** lets you select a file or folder of **elevation source data**, which will be added to

the product via a Map Import.



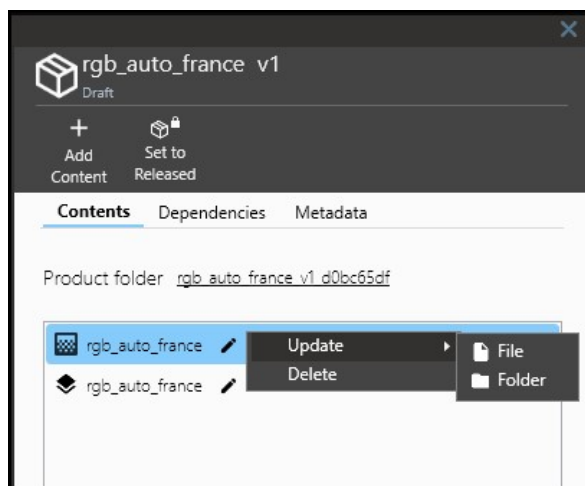
Adding new content to a product

10.3.1 Updating datasets

From version 4.2

Datasets can be updated by right-click -> Update dataset. You can then select the source data with which to update - a single file or a folder. A **Map import** will start, where the destination options (Name, Workspace, Product) are already fixed and cannot be changed. The map import will replace the data file in the dataset, leaving everything else - name, map signature and metadata, as it were. This means that basemaps and overlays referencing the dataset will still work, and will now display the updated dataset.

Currently, updates are supported for all dataset types except for vector datasets.



10.4 Dependencies

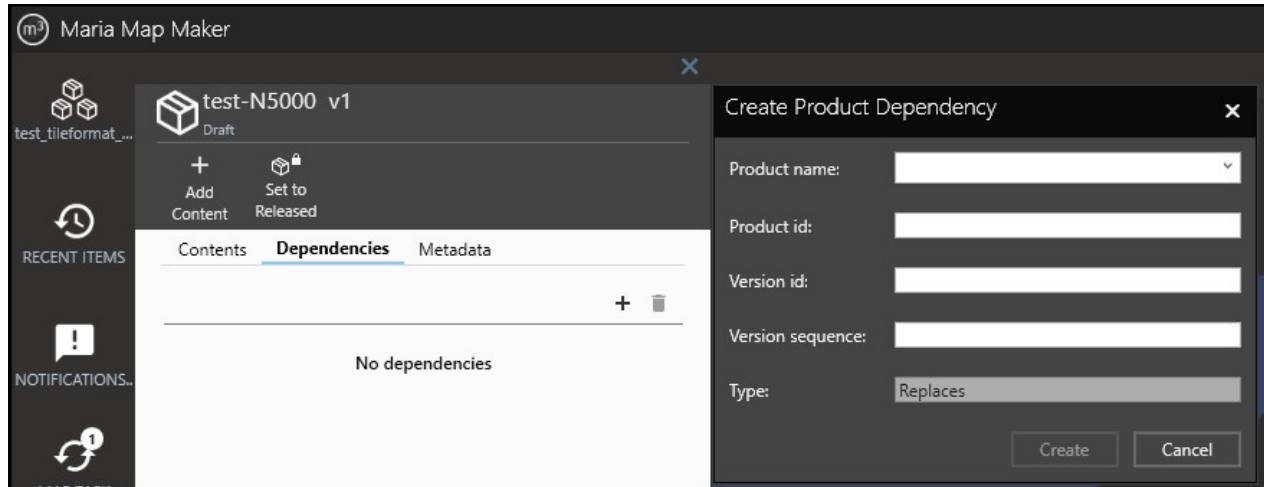
The dependency is a concept which allows a product to remove other products when it is installed. The main usecase for this functionality is if you have two products covering two areas - for instance counties. If these counties should be merged, then in the next version you'll want to create a new product, and you also want to remove the two older products

when it is installed.

Click the + to add a dependency to another product in your workspace.

From version 4.2:

You can also drag and drop products from Windows File Explorer onto the Dependencies window. You can drag-drop product folders, or a folder containing product folders.



10.5 Product metadata

Some metadata fields can be edited by the user if the product has status **Draft**. Others are automatically set when the product changes status. You can see the product metadata in the **Metadata** tab in an open product. In addition to text fields, you can upload a preview image to the product. The image is intended to be a small representative preview of the maps in the product. The image can be uploaded as a PNG file, or from the clipboard.

When you edit a field, the change is saved automatically.

MariaGeodata

NOTIFICATIONS

MAP TASK QUEUE

SETTINGS

Norway Administrative Units v8

Released

+ Add Content

Set to Draft

Contents Dependencies **Metadata**

Preview

From File

From Clipboard

Display name

Description

Source

Display version

Name

ID

Status

Last Changed Date

Last Changed By

This product is released, so the metadata fields can't be edited

11 Basemaps and overlays



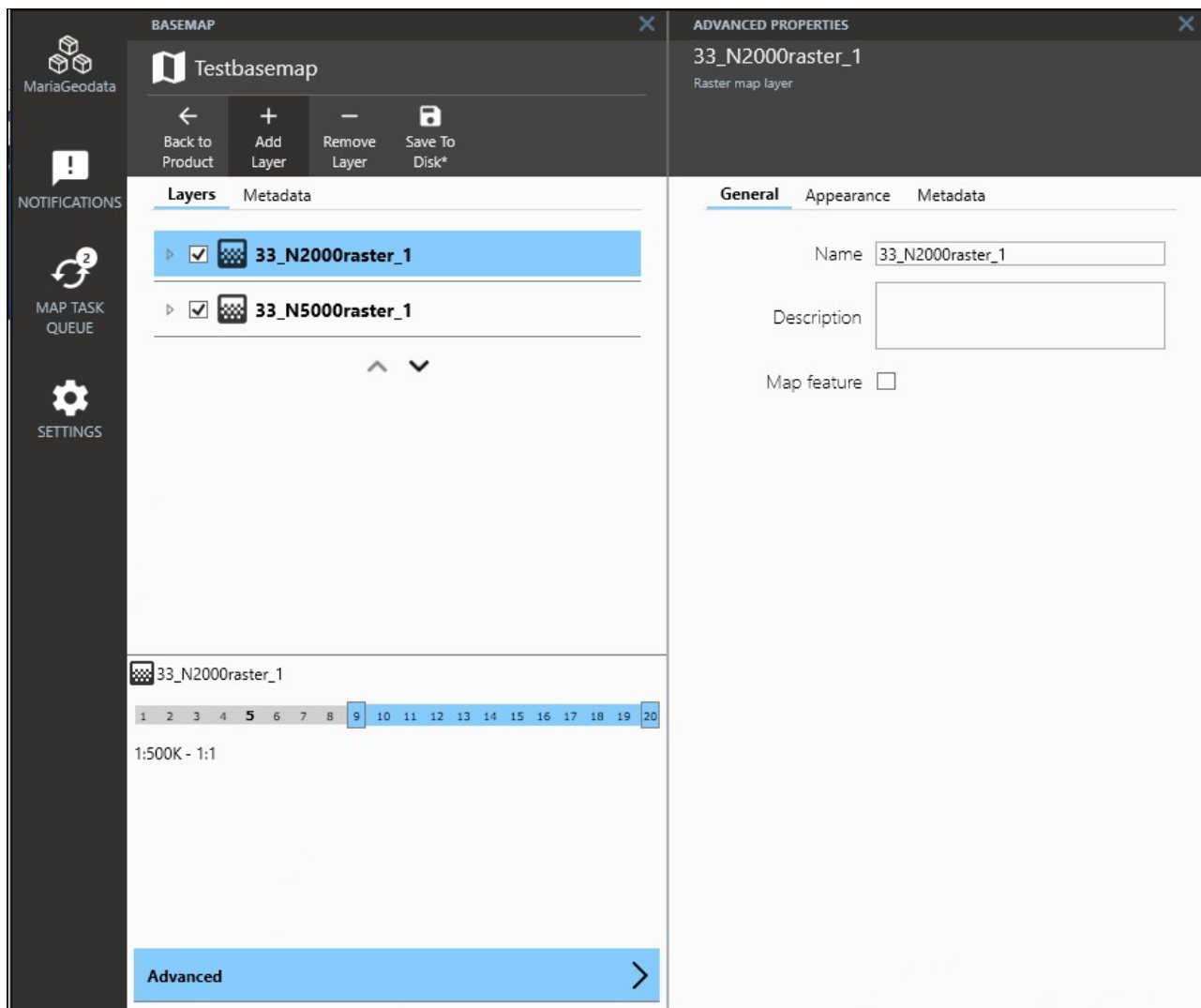
The basemap and overlay are collections of map layers, and the starting point for editing and styling maps in Maria Map Maker. They reside in products and can be created and opened from the product window. The difference between a basemap and an overlay in M3 is that the basemap can have special **Dynamic layers**, which can reference entire overlays using tags.

At the very basic level, the basemap and overlay are XML files which contain lists of references to datasets or web maps. In the Maria GDK core documentation, the basemap and overlay are collectively referred to as **templates**.

There are several different types of layers which can be added to a basemap/overlay:

- Vector
- Raster
- Web map
- Dynamic (basemap only)
- Colormap

11.1 Managing layers



Layers in a basemap. Selecting a layer lets you set a scale limit. Double-click the layer for advanced properties. Managing layers is done with **Add Layer** and **Remove Layer**. The checkbox on each layer controls visibility. The top layer in the list is drawn on top in the map. You can change the drawing order by dragging and dropping layers. Clicking a layer reveals the scale selector on the bottom pane. This lets you limit the zoom level range in which the layer is visible.

To open **Advanced properties** for the layer, use the button at the bottom, or simply double-click the layer. The **General** and **Appearance** tabs allow you to change how the map signature appears in this particular basemap or overlay. In other words, changes made here are only applied to the basemap/overlay. The map signature itself is not changed.

Raster and vector layers have a **Metadata** tab. Here you can type in information about the map signature and/or the underlying data source. You can include additional fields and delete them again with the buttons on the top right in this tab. The default fields cannot be deleted here, this must be done in the [settings file](#). The metadata is stored in a file next to the map signature, and they will have the same base name.

Maria Map Maker

MariaGeodata

NOTIFICATIONS

MAP TASK QUEUE

SETTINGS

BASEMAP

Testbasemap

←

 Back to Product

+

 Add Layer

−

 Remove Layer

💾

 Save To Disk*

Layers

Metadata

▶ ☒ 33_N2000raster_1

▶ ☒ 33_N5000raster_1

⬆

⬇

33_N2000raster_1

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

1:500K - 1:1

Advanced

➤

ADVANCED PROPERTIES

33_N2000raster_1

Raster map layer

General

Appearance

Metadata

Metadata

+

🗑

Key	Value
name	: N2000 raster
classification	: UGRADERT
limitations	:
designation	:
datadate	: 2020-04-01 <div>15</div>
edition	:
datasource	: Kartverket
description	: 1:2M Topographic raster
expirationdate	:
metadatadate	: 2020-04-06 <div>15</div>
metadataeditor	: TPG

Filling out the default metadata fields.

12 Importing data

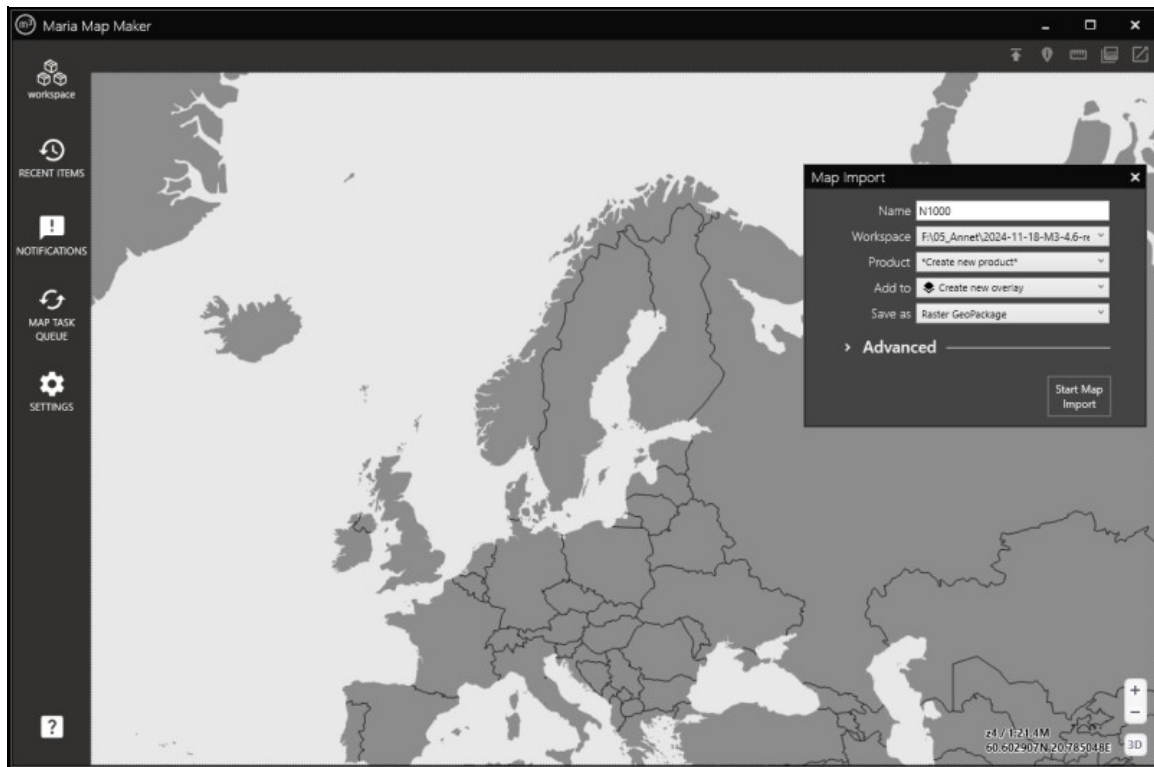
To use georeferenced data in M3, it needs to be optimized and imported into a product in your workspace. There are several ways to start an import:

- Drag and drop **any supported** files or folders from Windows File Explorer onto the map area in M3. You can do this at any time regardless of where you are in your workflow. The only prerequisite is that you have an active workspace.
- When you have a basemap or overlay open, click **Add Layer ? Import from file/folder**
- For elevation data, in addition to the two methods above, you can also use **Add content ? Elevation** in an open product.

All of these will result in a Map Import window on the right side of the map area. This is where you set your options and start the import process. These options will differ depending on of both the source and destination data formats. There are a few common options, however.

12.1 Common options

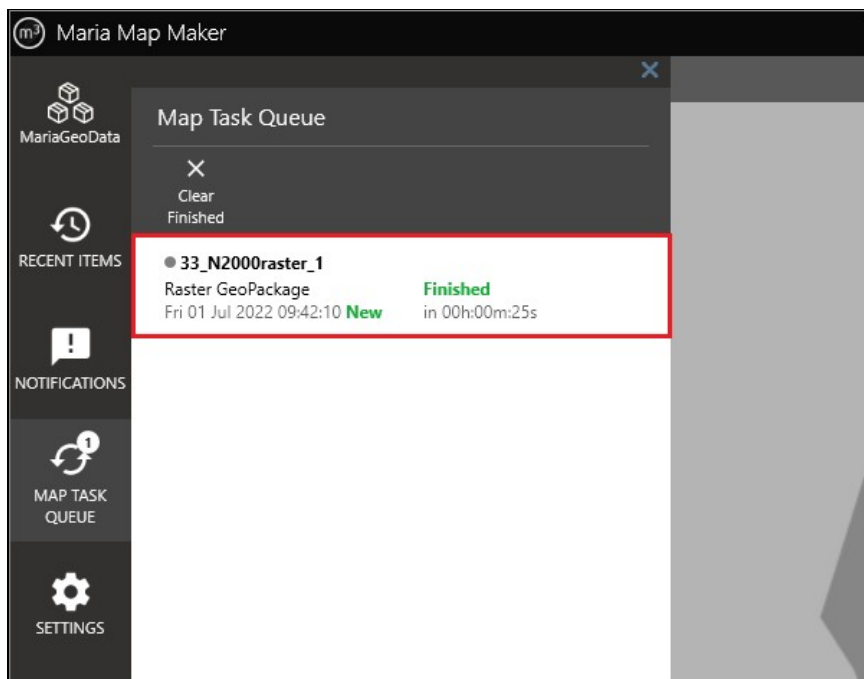
Name	Used to generate the map signature, which uniquely identifies this map when adding it to basemaps and overlays. It is also used as the product name, if you choose to import it into a new product.
Workspace	Lists all your active workspaces. Choose which of them to import the map to.
Product	Lists all the available (Draft) products in the chosen workspace. If you have a product, basemap or overlay open, this product will be the default choice. You can also choose to Create new product specifically for the map you are importing.
Add to	Automatically add the map as a layer to a new or existing overlay/basemap in the selected product. You can also choose to not add it as layer.
Save as	Lists the available destination formats for the source data you are importing.
Advanced	These options vary according to the input data and the chose destination format, and can often be left to their default values. See the pages for raster and vector imports for a description of these options.



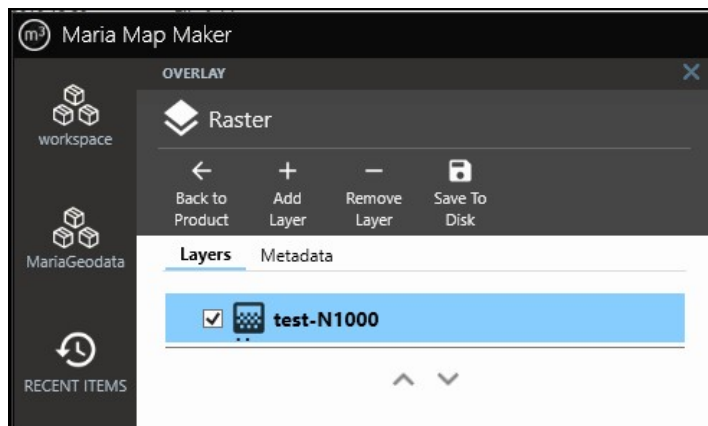
12.2 Map Tasks

When you start the Map Import process, a **Map Task** will be created. You can monitor it in the **Map Task Queue** window. You can have several map tasks active at the same time.

From version 4.2: Double-click a finished map task to open the product which it was imported to.




12.3 Placeholder layer



The dataset test-N1000 is still processing and will be visible when it is finished

When you choose to add your vector or raster dataset as a layer in an overlay/basemap using the **Add to** option, a

placeholder layer will be created there. The animated icon  indicates that the dataset referenced by the layer is still processing. When the map task is finished, it will turn into a normal layer, and the dataset will be visible in the map. This happens automatically, regardless of what you are doing in the application.

12.4 Raster data (version 4.3 and older)

Importing raster data is usually quite simple. However, you have to make sure that if you're importing multiple files into a single dataset, they have to be homogenous (same file format, pixel size, projection, band types)

Depending on the source file format you can either choose to save as GeoPackage, or use the source format without translation. With some formats (ECW and MrSID), it's recommended to use the source format. These are formats which already have good performance, and translating them to GeoPackage leads to larger file size and loss of image quality.

12.4.1 Parameters

Different source formats will have different parameters.

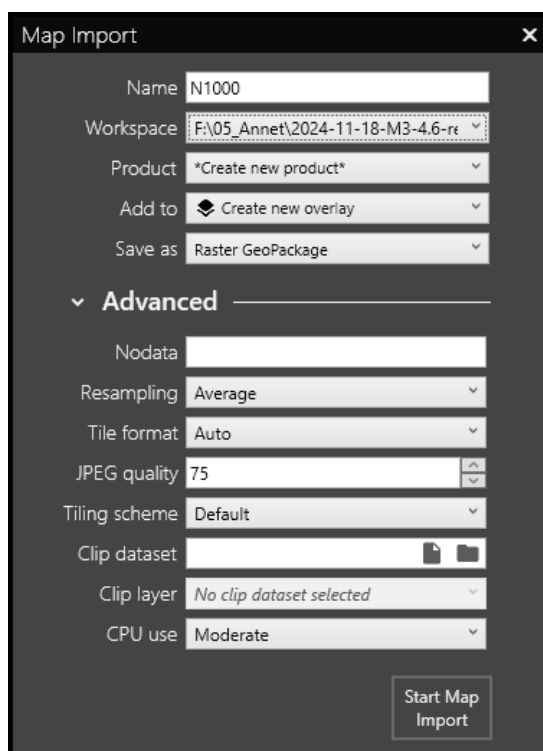
- **Nodata** - If the source data is an RGB raster, the nodata parameter should be a space-separated list of RGB values to make transparent - for example 0 0 0. If the source data uses a color palette, then the parameter is single number - the color index to make transparent.
- **Projection/Proj4** - These parameters only appear if projection information cannot be found in the source files. A few common projections are provided in the dropdown menu, or you can type in a projection in Proj4 format.
- **Resampling** - Resampling method to use when creating downsampled overviews.
- **Jpeg quality** - the level of compression to use when creating jpeg tiles. A larger number results in better quality and larger file size. 75-90 is usually a good balance between size and quality.
- **Tile format** - *From version 4.2* - Image format for tiles in the output GeoPackage. "Auto" uses JPEG on opaque tiles and PNG on tiles where transparency is needed. "PNG" uses 32 bit RGBA PNG format on all tiles. "PNG8" calculates a color table with maximum 256 different colors for each tile. The "PNG8" option can reduce file size significantly when the input data has few unique colors.

12.4.2 Raster data with more than 8 bit

If a file has bands with more than 8 bit, it is usually elevation data. But in some cases, it can also be imagery from a satellite. Because Maria GDK and the GeoPackage format only supports 8 bit bands for maps and imagery, these datasets must be scaled down in the import process. In M3, this is handled as follows:

Number of bands	Bits per band	Comment
1	> 8	Use Save as GeoPackage (Scale to 8 bit) . Values are automatically scaled using default parameters as described in the gdal_translate documentation
3	> 8	Use Save as Raster GeoPackage . You can change the scale interval with the parameter Scale factor . This will affect the contrast of the resulting GeoPackge - a lower number results in a brighter image.

12.5 Raster data (version 4.4 and newer)



12.5.1 Parameters

- **Projection/EPSSG code** - These parameters only appear if projection information cannot be found in the source files. A few common projections are provided in the dropdown menu, or you can type in a projection in EPSG format.
- **Nodata** - If the source data is an RGB raster, the nodata parameter should be a space-separated list of RGB values to make transparent - for example 0 0 0. If the source data uses a color palette, then the parameter is single number - the color index to make transparent.
- **Resampling** - Resampling method to use when creating downsampled overviews.
- **Tile format** - Image format for tiles in the output GeoPackage:
 - ♦ *Auto* uses JPEG on opaque tiles and PNG on tiles where transparency is needed.
 - ♦ *PNG* uses 32 bit RGBA PNG format on all tiles.

- ◆ **PNG8** calculates a color table with maximum 256 different colors for each tile. This option can reduce file size significantly when the input data has few unique colors.
- **JPEG quality** - the level of JPEG compression to use when Tile format = "Auto" is used. A larger number results in better quality and larger file size. 75-90 is usually a good balance between size and quality.
- **Tiling scheme** (from version 4.6) - Tiling scheme to use when building the output GeoPackage:
 - ◆ **Default** - Use the existing spatial reference system of the source, and use a tiling which ensures that the georeferencing remains unchanged.
 - ◆ **GoogleMapsCompatible** - Source dataset is reprojected to Web Mercator and resampled to the global GoogleMapsCompatible tiling scheme. This will increase processing time and may lower the resolution of the dataset, but it will also give an increase in performance in the output dataset. It can also be useful if the source dataset uses a spatial reference system which is not supported by the Maria GDK core components.
- **Scale factor** - Factor to use when it is necessary to scale down to 8 bit. This parameter becomes available if the input dataset has bands with more than 8 bit, and **Save as** is set to *Raster GeoPackage*. The range of values to use from each source band is calculated as $(\text{mean} \pm \text{stddev} \times \text{scalefactor})$, and then scaled to 8 bit (values 0-255).
- **CPU use** - the number of logical processors to use. When the input dataset consists of multiple files, Maria Map Maker will split the work across the logical processors of the CPU.
- **Clip dataset** - a vector dataset to use for clipping the input raster dataset.
- **Clip layer** - a named layer from the clip dataset to use for clipping. If there are multiple area features in the layer, only the first feature will be used. If the raster dataset falls completely outside the clip layer, an error will be thrown, and no output GeoPackage will be created.

12.6 Elevation/Coverage data

Map Import

Name: 33-125-131

Workspace: F:\05_Annet\2024-11-18-M3-4.6-re

Product: *Create new product*

Save as: Elevation GeoPackage

Elevation Type: Please Select

Tags:

Advanced

Resampling: Average

Tile format: PNG (16 bit)

Tiling scheme: Default

Clip dataset:

Clip layer: No clip dataset selected

CPU use: Moderate

Start Map Import

For any regular gridded input data of 16 bit or more, the result will be a **tilled gridded coverage GeoPackage**. In Maria Map Maker, these are classified as either an *Elevation dataset*, or a *Coverage dataset* (chosen in the **Save as** drop-down)

12.6.1 Elevation GeoPackage

Datasets which represent the heights above a reference. Importing data as an elevation dataset requires that you classify the elevation type as one of the following:

- **Terrain Model** - Models the ground terrain, where trees and man-made structures have been removed.
- **Surface Model** - Models the surface of the earth, where trees and man-made structures are present.
- **Bathymetry** - Models the sea bed

12.6.2 Coverage GeoPackage

Datasets which can represent any geographical phenomenon. It's recommended that you create a **Tag** which describes the dataset for instance "landcover", "trafficedensity", "temperature", etc. This allows the dataset to be visualized in a **Color map layer**.

12.6.3 Common parameters

Parameters which are shared for Elevation and Coverage dataset imports.

- **Tags** - Allows the dataset to be used in **Color maps** with matching tags
- **Tile format**
 - ♦ **PNG** - Tiles are saved as PNG with data type Int16. If the input is Float32, a scale and offset will be calculated to achieve an effective precision of around 0.01 - 0.001 meters (less varied source data results in higher precision). This option is the most space efficient, and usually sufficient for elevation data with 10 meter resolution or lower.
 - ♦ **TIFF** - Tiles are saved as TIFF with data type Float32. This option will retain the precision of the source data at the cost of larger file size.
- **Resampling** - Resampling method to use when creating overviews.
- **Tiling scheme** (*from version 4.6*) - Tiling scheme to use when building the output GeoPackage:
 - ♦ **Default** - Use the existing spatial reference system of the source dataset, and use a tiling which ensures that the georeferencing remains unchanged.
 - ♦ **GoogleMapsCompatible** - Source dataset is reprojected to Web Mercator and resampled to the global GoogleMapsCompatible tiling scheme. This will increase processing time and may lower the resolution of the dataset, but it will also give an increase in performance in the output dataset. It can also be useful if the source dataset uses a spatial reference system which is not supported by the Maria GDK core components.
- **CPU use** (*from version 4.6*) - the number of logical processors to use. When the input dataset consists of multiple files, Maria Map Maker will split the work across the logical processors of the CPU.
- **Clip dataset** (*from version 4.6*) - a vector dataset to use for clipping the input raster dataset.
- **Clip layer** (*from version 4.6*) - a named layer from the clip dataset to use for clipping. If there are multiple area features in the layer, only the first feature will be used. If the raster dataset falls completely outside the clip layer, an error will be thrown, and no output GeoPackage will be created.

12.7 Vector data

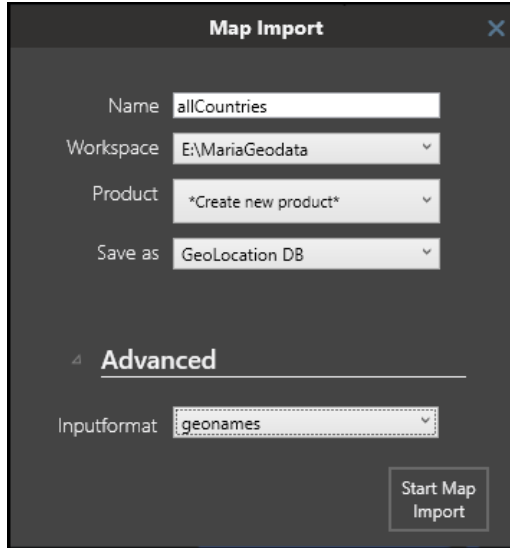
For importing and styling MARIA GDK vector data, see **MARIA GDK vector datasets** (version 4.2 and older).

For importing and styling Mapbox vector data, see [Mapbox vector datasets](#) (version 4.3 and newer).

12.8 Location data

M3 can create geolocation databases from text files. The text files must be formatted according to one of several well established schemas/sources:

- GNS - [NGA GEOnet Names Server - Non-US placenames](#)
- GNS_US - [US placenames](#)
- Geonames - [Global placenames](#)



The screenshot shows the 'Map Import' dialog box. It has a title bar with a close button. The main area contains several input fields and dropdown menus. The 'Name' field is set to 'allCountries'. The 'Workspace' dropdown is set to 'E:\MariaGeodata'. The 'Product' dropdown is set to '*Create new product*'. The 'Save as' dropdown is set to 'GeoLocation DB'. Below these is an 'Advanced' section with a dropdown for 'Inputformat' set to 'geonames'. At the bottom right is a 'Start Map Import' button.

12.9 Routing databases

From version 4.1

M3 can create routing databases based on Openstreetmap raw data (pbk files). These files can be exported from the map at openstreetmap.org, or they can be downloaded from services such as [Geofabrik](#). The routing databases can be used by a Maria GDK-based application to calculate the fastest route between two points.

12.9.1 Profiles

When importing the data, you can select which profiles to include in the output database. A profile describes a vehicle and its behaviour when travelling the road network. Note that the profiles cannot be changed after the database has been created. Hover the mouse over each profile for a short description.

Map Import

Name

test-routingdb

Workspace

E:\MariaGeodata

Product

Create new product

Save as

Routing database

Advanced

Choose which profiles to include in the routing database.

☒

Car Classifications

☐

Car Shortest

☐

Car Fastest

☐

Bicycle Networks

☐

Bicycle Balanced

☐

Bicycle Shortest

☐

Bicycle Fastest

☐

Pedestrian Fastest

☐

Pedestrian Shortest

☐

Motorcycle Classifications

☐

Motorcycle Shortest

☐

Motorcycle Fastest

☐

Moped Shortest

☐

Moped Fastest

☐

Small Truck Classifications

☐

Small Truck Shortest

☐

Small Truck Fastest

☐

Big Truck Classifications

☐

Big Truck Shortest

☐

Big Truck Fastest

☐

Bus Shortest

☐

Bus Fastest

Start Map Import

12.10 3D Tiles

From version 4.2

Map Import

Name: Hvittingfoss Skole_simplified_3d

Workspace: C:\WORK\MariaGeoData

Product: 3D models (v1_14ba6df1)

3D Tileset: Create new 3D Tileset

Model: Top level model

Save as: 3D Tiles

Advanced

Compress Textures ☒

Auto place ☒

Latitude: 65

Longitude: 10

Elevation: 50

Unit select: m

Start Map Import

Importing models is handled by the library assimp. A full list of supported formats can be found at <https://github.com/assimp/assimp/blob/master/doc/Fileformats.md>, however many of the formats have not yet been tested in Maria Map Maker. The imported 3D tileset will consist of the models themselves in the format b3dm, as well as texture files, and several text files which define the position and relationships between the models.

12.10.1 Parameters

- **3D Tileset** - Choose an existing 3D tileset as a destination for the model, or create a new tileset.
- **Model** - Choose an existing model as a parent for the current import, or define it as a top-level model.
- **Compress textures** - Compress textures into a DirectDraw Surface container file. This reduces file size, but should not be used if you intend to use the tileset in the Cesium platform, as it's not supported there.
- **Auto place** - Attempt to find geographic position for the model in the source files.
- **Latitude / Longitude / Elevation** - Manually enter position and elevation for the placement of the model. Only used if "Auto place" is disabled or unable to find geographic position in the source files.
- **Unit select** - Unit in the source data - meters or feet.

13 Supported formats

Maria Map Maker can import and setup a wide variety of geodata formats. Some filetypes are imported into the workspace and read directly, while others are processed into optimized formats for greater performance. With some formats both options are available, though using the source format is recommended. The formats used for optimization are

- Vector Tiles
- Raster GeoPackage
- Elevation GeoPackage
- Geolocation SQLite database
- RouterDB
- 3D Tiles

The following formats can be imported. Formats in **bold** can be imported directly into the workspace without conversion or processing.

Source format	Type	Destination format
ESRI Shape	vector	Vector Tiles
ESRI File Geodatabase	vector	Vector Tiles
GeoJSON	vector	Vector Tiles
GML	vector	Vector Tiles
KML	vector	Vector Tiles
GeoPackage vector	vector	Vector Tiles
MBTiles vector	vector	Source
GeoTIFF	raster	Raster GeoPackage
JPEG	raster	Raster GeoPackage
JPEG2000	raster	Raster GeoPackage
PNG	raster	Raster GeoPackage
NITF	raster	Raster GeoPackage
MrSID	raster	Source or Raster GeoPackage
ECW	raster	Source or Raster GeoPackage
MBTiles raster	raster	Source
GeoPackage raster	raster	Source
Teleplan QM2	raster	Source
GeoTIFF	elevation	Elevation GeoPackage
USGS DEM	elevation	Elevation GeoPackage
Arc/Info ASCII Grid	elevation	Elevation GeoPackage
SRTM HGT	elevation	Elevation GeoPackage
DTED	elevation	Elevation GeoPackage
GNS	placenames	Geolocation database
GNS_US	placenames	Geolocation database
Geonames	placenames	Geolocation database

Source format	Type	Destination format
Openstreetmap PBF	Routing database	RouterDB
Various 3D model formats	3D	3D Tiles (<i>From version 4.2</i>)

14 Mapbox vector datasets

From version 4.3

Maria Map Maker supports the [Mapbox vector tiles format](#), as well as a subset of the [Mapbox JSON style specification](#). This article explains the process of importing and styling vector data with these formats.

14.1 Import

14.1.1 Format



When Maria Map Maker imports vector data, it is translated to Mapbox vector tiles (.mbtiles). This is a very high-performant format, but the translation process requires some tuning and know-how to get the desired result.

The MBTiles format splits the vector data into tiles and zoom levels. Each tile contains 4096 x 4096 nodes, which are roughly equivalent to a pixel in a raster image. When vector data is converted to MBTiles, all vertices are snapped to the nearest node in a tile, losing some precision in the process.

Each tile belongs to a zoom level. Zoom level 0 is a single tile which covers the entire Earth (in Web Mercator projection). With every increasing zoom level, each tile is divided into four. As an example, zoom level 15 roughly equals map scale 1:15 000 at the equator. At this level, the entire Earth consists of around one billion tiles.

When importing vector data to MBTiles, you need to decide which zoom levels (z) to create. This zoom extent is not possible to change later. The vector data simply does not exist on the levels outside the zoom extent you choose. This means that you have to be familiar with your source data and how they are going to be used. Using zoom levels which are not appropriate for the source data can have some negative effects:

Using too low zoom level - too much data is stored in each tile, which would lead to poor performance. To prevent this, after a tile reaches a threshold of 500 KB or 200 000 features, additional features destined for that tile will either be simplified or dropped. Using too high zoom level - large, simple features are split across an exaggerated number of small tiles, which leads to unnecessary large files and long processing time.

For information about the accuracy of vector tiles at a given zoom level, see the article [Vector tiles accuracy](#)

14.1.2 Import dialog

The vector data import dialog lists all the layers in the source dataset, with a zoom selector for each. The checkboxes controls whether or not the layer is included in the resulting MBTiles dataset. You can use the top **All layers** control to set a common zoom extent.

You can choose the style format for the output dataset - the Mapbox JSON format described in this article, or the legacy MARIA GDK XML format. The MARIA GDK style will be phased out starting with version 4.3.

Map Import

Name: UK_London_OpenMap

Workspace: CAWORK\MariaGeoData

Product: *Create new product*

Add to: Create new overlay

Save as: Mapbox vector tiles

Advanced

Style type: Mapbox

Vector tiles require a zoom extent to be set. Choose an extent which is suited for the intended use of the tileset.

☒ **All layers**

☒ **TQ_Building**

☒ **TQ_CarChargingPoint**

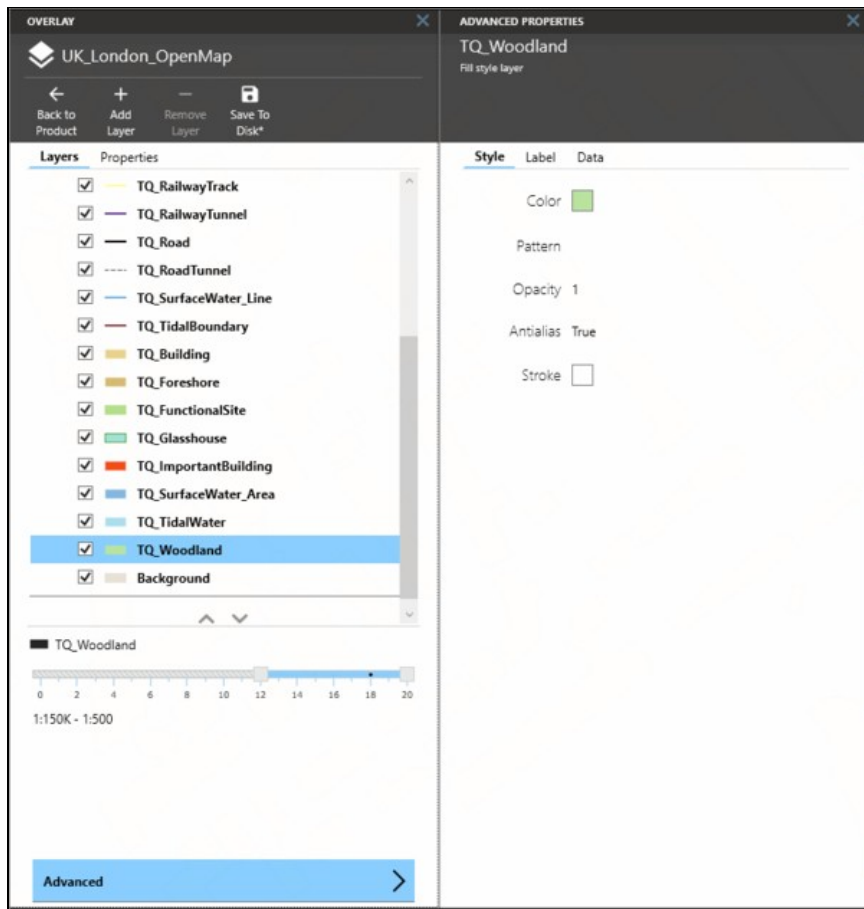
☒ **TQ_ElectricityTransmissionLine**

☒ **TQ_Foreshore**

☒ **TQ_FunctionalSite**

Start Map Import

14.2 Style layers



To work with the vector dataset, open the overlay that it was added to (**Add to** in the previous screenshot).

Style layers are listed by expanding the dataset in a basemap/overlay. The list has several functions:

- Drag-drop layers to change the drawing order (the top layer in the list will be drawn on top in the map).
- Show or hide layers with the checkbox.
- Single-click a layer twice (slow double-click) to change its name

14.2.1 Create new layer

To create a new layer, simply right-click anywhere in the Style layers list, and then select the layer type from the **New layer** submenu. A new layer will be created at that position. Finally, go to the **Data** tab to select a Source layer.

14.2.2 Style tab

Styles are configured by double-clicking a layer and selecting the **Style** tab. Style properties are documented by mousing over the property name. The style properties vary by the type of layer:

14.2.2.1 Fill

A fill style layer renders one or more filled (and optionally stroked) polygons on a map.

14.2.2.2 Line

A line style layer renders one or more stroked polylines on the map.

14.2.2.3 Symbol

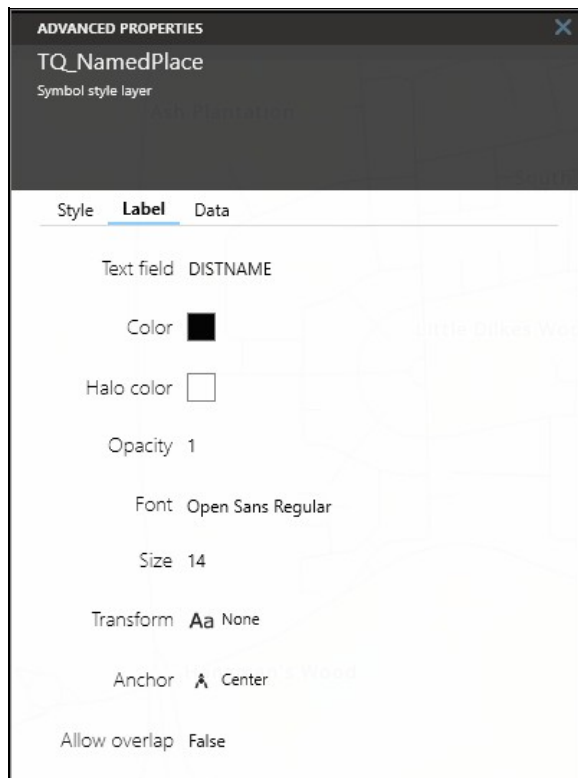
A symbol style layer renders icon and text labels at points or along lines on a map. The **Image** property allows you to select a PNG or SVG file to use as an icon. Additional icons can be imported into the dataset with the **Add symbols** button. This will open the **Symbol libraries** browser. Here, you can import symbols from any installed symbol library. Some SVG symbols can also be styled before they are imported.

14.2.2.4 Background

A layer which covers the entire dataset. Useful if your content does not cover everything and you want to configure a background color.

14.2.3 Label

A text label can be configured on any type of layer. Click the **Label** tab and select an attribute for the **Text field** property to activate the label. You can then configure the appearance of the label with various properties.



14.2.4 Data

ADVANCED PROPERTIES

Road name

Line style layer

Style

Label

Data

Layer name

Road name

Source layer

tq_road

Layer type

Line

CLASSIFICA

Equal (==)

A Road

DRAWLEVEL

Equal (==)

0

+

Add filter

DISTNAME

Equal (==)

King's Scholar's Passage

+

Add filter

+

Add filter group

AND

OR

The **Data** tab contains settings related to the data source of the style layer.

14.2.4.1 Source layer

Source layers are the named layers in the MBTiles data file. A style layer must always reference a source layer (except for the background layer). The source layer type is indicated with an icon in the list (either Line, Fill, or Symbol).

14.2.4.2 Layer type

Layer type for this Style layer. This does not have to match the type of source layer it is referencing. By using a different type, you get some additional style options:

Source layer type	Style layer type	Result
Polygon	Line	Outline of polygons are drawn
Polygon	Symbol	Text labels or icons are drawn within the polygon
Line	Symbol	Text labels or icons are drawn along the line

14.2.4.3 Filters

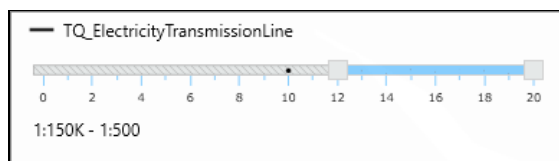
A filter is a requirement that a feature must meet to be drawn in the map. When you create a new filter, you must select an attribute, an operator and a value. The value menu contains a small sample of values from the selected attribute, but you can also type in any value. Only features which match the filter will be drawn.

You can create multiple filters on a layer, and set the logical operator to connect them (**AND/OR**). You can also **Add filter group** to create groups of filters. The result (either true or false) is first evaluated for each group using the inner AND/OR, then between the groups using the outer AND/OR, to finally determine if an object should be drawn or not.

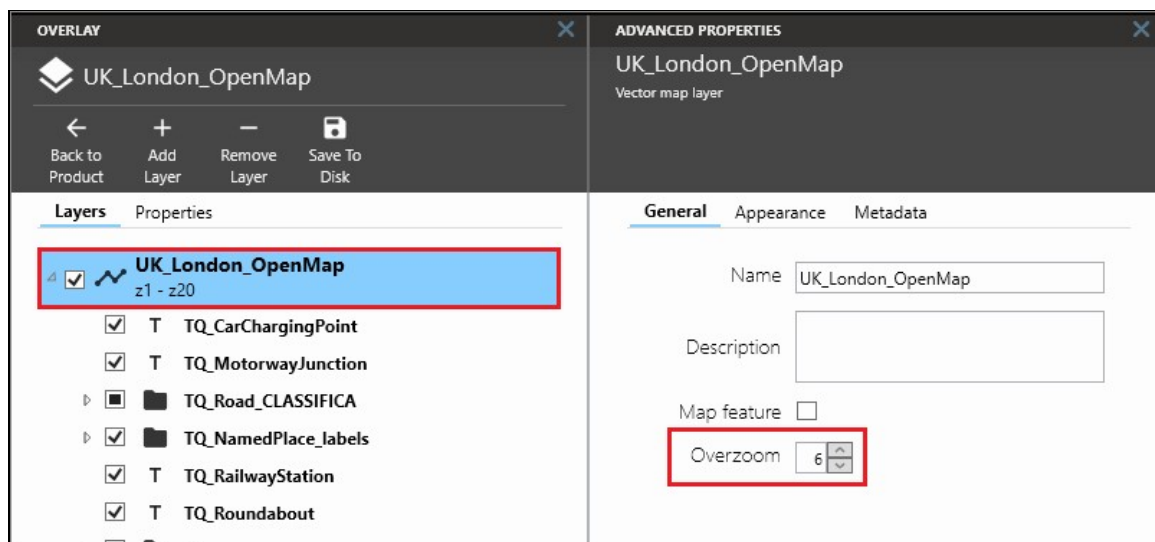
14.2.5 Zoom selector

Each style layer's visibility can be controlled with the Zoom selector at the bottom of the Layers list. The Zoom selector consists of two sliders which represent the max and minimum zoom levels where the layer is visible. If the map is zoomed outside the area between the sliders, the layer will be hidden. Setting the maxzoom to z20 effectively means that the layer will not disappear while zooming in.

The zoom selector is automatically adjusted so that its not possible to set a minzoom where the source layer does not exist. This is represented as a scribbled area. The "Valid" area is the interval between minzoom and (maxzoom + overzoom).



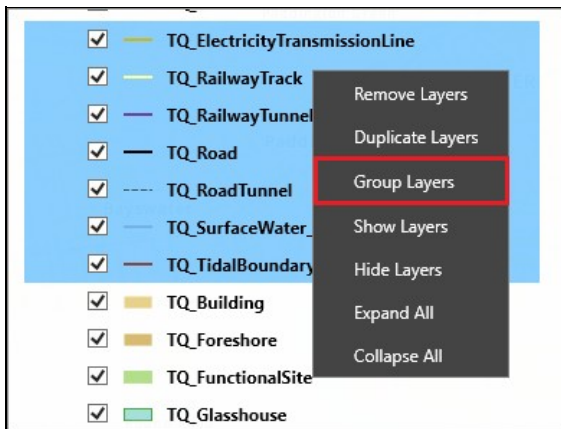
Layers can still be shown when zooming in past the max level that exists in the data file. This is called **Overzoom**. The number of levels to overzoom can be set by double-clicking the root Vector layer. The default is 6 levels.



14.2.6 Advanced styling

14.2.6.1 Grouping

Layers can be grouped together by selecting multiple layers with Ctrl+left click. Right-click the selection, and click **Group layers**. This allows the layers to be moved and turned on and off as a single unit.



14.2.6.2 Duplicate layer

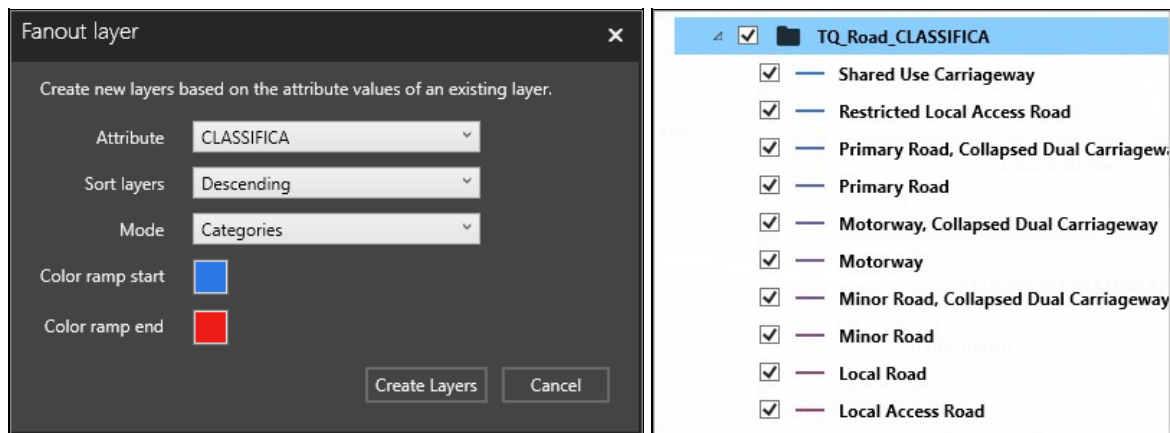
Any layer can be duplicated, where the only difference from the original is the name. This is useful if you for example want to style a line with an inner color and a different outline color. You would then place the duplicate below the original in the layer list, and set a slightly larger width.

14.2.6.3 Fanout

From version 4.4

M3 can automatically set up a new set of layers based on the attribute values in a layer. Right click any layer and click **Fanout** to open this dialog. The **Color ramp** options control the colors of the first and last layer created, and a color ramp is calculated linearly for all the layers in between. There are two modes when using fanout:

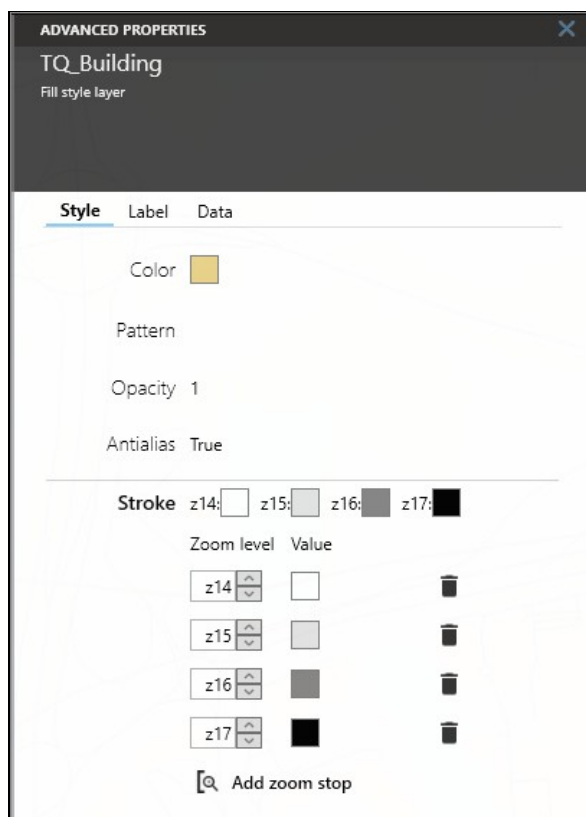
- **Category** - A new layer is created for each unique value in the selected attribute. Both string and numerical attributes are supported.
- **Graduated** - The full range of numerical values in the selected attribute is split into classes. The number of classes is controlled by the **Classes** option. The method to use when calculating these classes can be further controlled with the **Graduated type** option:
 - ♦ **Equal interval** - The full range of values is split into equally sized classes.
 - ♦ **Equal count** - Classes are calculated such that there will be an equal number of objects in each class. This requires the application to scan through all features in the dataset, so it can take a long time. Suitable for attributes where the values strongly deviate from a normal distribution.
 - ♦ **Display precision** - Number of decimal points to use for the layer names.



14.2.6.4 Style across zoom ranges

Many style properties have the ability to change according to the current map zoom level. For instance, this allows a smooth transition of line width or color as you zoom in and out.

Click **Style across zoom range** on a property to activate the feature. You can then set up a list of zoom stops with corresponding value for the property. **Rate of change** controls the exponential change between stops. The default is 1.0, which increases values linearly. To disable this function, delete all zoom stops.



14.2.6.5 Creating repeating fill patterns

To create fill patterns which repeat seamlessly across tiles in the map, you need to first create the repeating pattern in a graphics software, for instance [Inkscape](#). You can create a pattern in Inkscape by clicking **New ? New from Template ? Other ? Seamless Pattern**). The image should be exported to PNG with dimensions of powers of 2, up to the size of a

tile (i.e. 16x16, 32x32, 64x64, 128x128, 256x256).

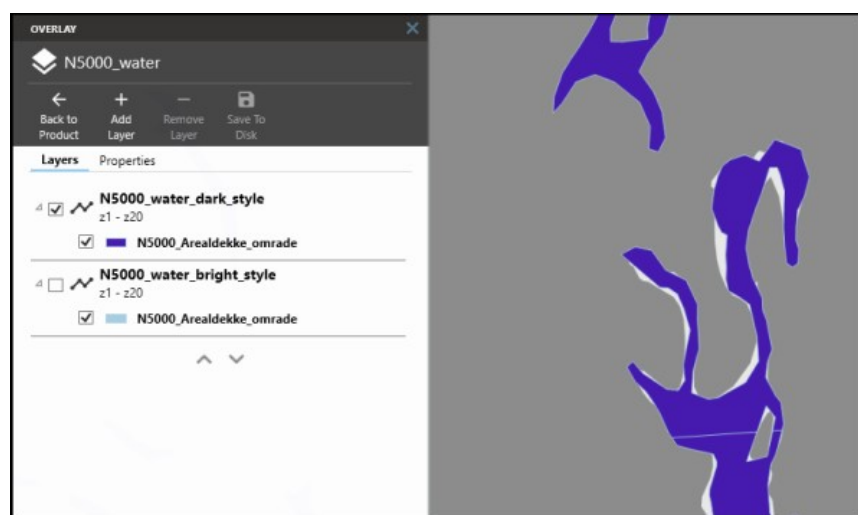
To add it to your dataset in M3, open the properties of a fill style layer, click **Style ? Pattern ? Add symbols from file**.

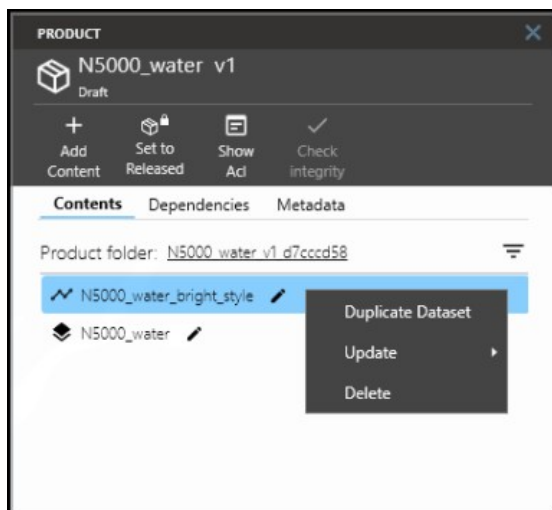


14.3 Duplicating datasets

From version 4.4

In some cases you may want to create multiple styles which the user can choose from, without duplicating the data file (.mbtiles). This can be achieved by first importing the data as described [in this article](#), and then creating a style. In the product window, you can right-click the dataset and select **Duplicate Dataset**. This results in a new dataset which is identical to the original, except that it references the original's data file instead of duplicating it. The duplicate dataset can then be added to an overlay and styled independently.





Note that using the Update function on any dataset with a shared data file will affect all the datasets using that file. Deleting a dataset which contains a shared data file will not delete the actual data file. The data file is only deleted when the last dataset using it is deleted.

15 MARIA GDK vector datasets

This article is valid for version 4.2 and older. For version 4.3, see [Mapbox vector datasets](#)

15.1 Importing data



Vector tiles are structured in a quadtree of tiles and zoom levels. The number of tiles quadruples for each increasing zoom level.

When Maria Map Maker imports vector data, it is translated to Vector Tiles (.vtiles.sqlite). This is a very high-performant format, but the translation process requires some tuning and know-how to get the result you want.

The Vector Tiles format splits the vector data into tiles, and puts each tile into a particular zoom level. Zoom level 0 is a single tile which covers the entire Earth (in Web Mercator projection). With every increasing zoom level, each tile is divided into four. As an example, zoom level 15 roughly equals map scale 1:15 000 at the equator. At this level the Earth is split into slightly more than a billion tiles.

When importing vector data to Vector Tiles, you need to decide which zoom levels (z) to create. This *zoom extent* is not possible to change later. The vector data simply does not exist on the levels outside the zoom extent you choose. This means that you have to be familiar with your source data and how they are going to be used. Using zoom levels which are not appropriate for the source data can have some negative effects:

- Using too low zoom level - too much data is stored in each tile, which leads to poor performance.
- Using too high zoom level - large, simple features are split across an exaggerated number of small tiles, which leads to unnecessary large files and long processing time.

An example could be some very detailed building data for a city. If you try to import these into z10-z12, all the data will probably be pushed into a single, huge tile at z10. This would cause very poor performance in the finished map.

For information about the accuracy of vector tiles at a given zoom level, see the article [Vector tiles accuracy](#)

15.1.1 Overzoom

The overzoom function eases the zoom extent choice somewhat. It lets you display vector features at higher zoom levels than what they were created for. If you for example create a dataset at z10-z14, the default use of overzoom in M3 (6 levels) lets the data be visible all the way down to z20.

The downside with using overzoom is that performance will decrease as overzoom increases. The precision of the vertices of the vector data also decreases.

15.1.2 Starting import

Map Import

Name:

Workspace:

Product:

Add to:

Save as:

Advanced

Vector tiles require a zoom extent to be set. Choose an extent which is suited for the intended use of the tileset.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1:62500 - 1:7813

Layers in the source data. You can choose which layers should be included in each zoom level in the tileset.

- ▶ Level 12
- ▶ Level 13
- ▶ Level 14
- ▶ Level 15

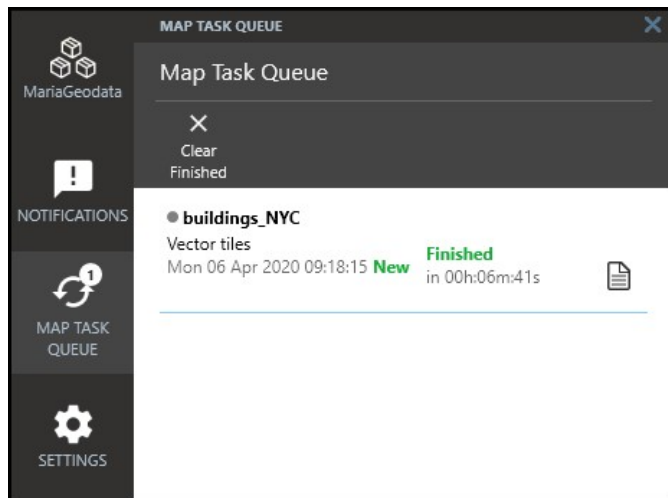
Review map import report ☐

Start Map Import

Choose the zoom extent with the two sliders. The zoom indicator shows the current zoom level in the map. Below, all the zoom levels in the zoom extent are listed. They can each be expanded, and you can uncheck layers that you don't need at that particular zoom level. By selecting **Review map import report** you can get an analysis of the file you created. Each zoom level will have a list of "concerns", indicating if there is too much data on that level. When you are satisfied

with your settings, start the map import.

A map task will start, and you can monitor its progress in the **Map Task Queue**. If you selected **Review map import report**, you can click the "Report" icon.

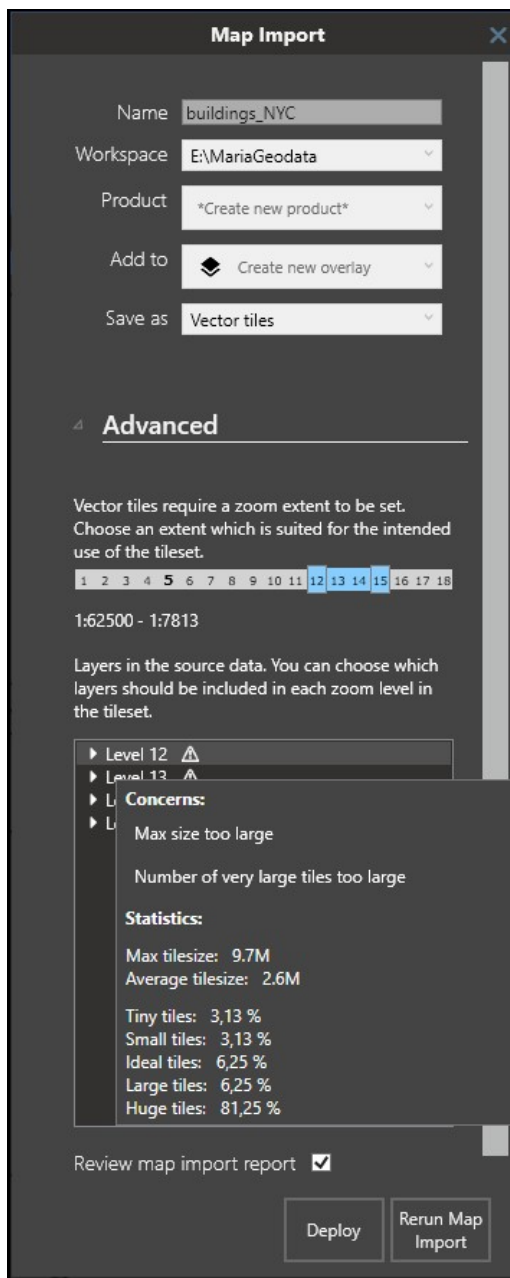


This will open the Map Import window again, but now an analysis has been performed on the vector tiles. You get the opportunity to re-run the translation process with different settings according to the analysis.

Mouse over each layer to read the result. Zoom levels with a ! have been determined to be sub-optimal. The messages from these will inform you of one of the following:

- *Max size too large and Fraction/number of large tiles too high* - You have put data which is too detailed into a zoom level which is too low. Each tile should ideally not be more than 0.1 MB. Depending on the size of the dataset, however, having a few tiles a little over 0.1 MB is not necessarily a problem. When many of them exceed 0.2 MB however, you should consider re-running the process with a higher minimum zoom level.
- *Tiles contain little data* - this is not as much of a problem as the opposite. The size of the data file will be larger than it could be, but performance will not be affected.

If you change any settings according to the analysis, you have to use **Rerun Map Import**, and the translation process will start over. If you use **Deploy** the map import will be finalized to your workspace as it is.



This dataset on buildings in NYC is too detailed for the zoom extent chosen. A better extent would be z15-z17

15.2 Styling



Setting up and styling vector data is by far the most complex operation when composing a map. When the source data is well described with attributes, the possibilities are quite endless.

You can reach the vector editing tools by fully expanding a vector layer. When a vector dataset is imported, there will be a one-to-one correspondence between the layers in the source and the layers in the finished map signature. Each layer will be assigned a random style by default.

Any changes to the vector styling needs to be saved before it is visible in the map.

15.2.1 Sub-layers

You can select multiple layers with the Ctrl key and toggle their visibility. As with the top-level layers, the layer on top of the list will be drawn on top in the map, and you can change drawing order by dragging and dropping. You can also control the scale limits of each layer. However, the scale limits of its parent will take precedence.

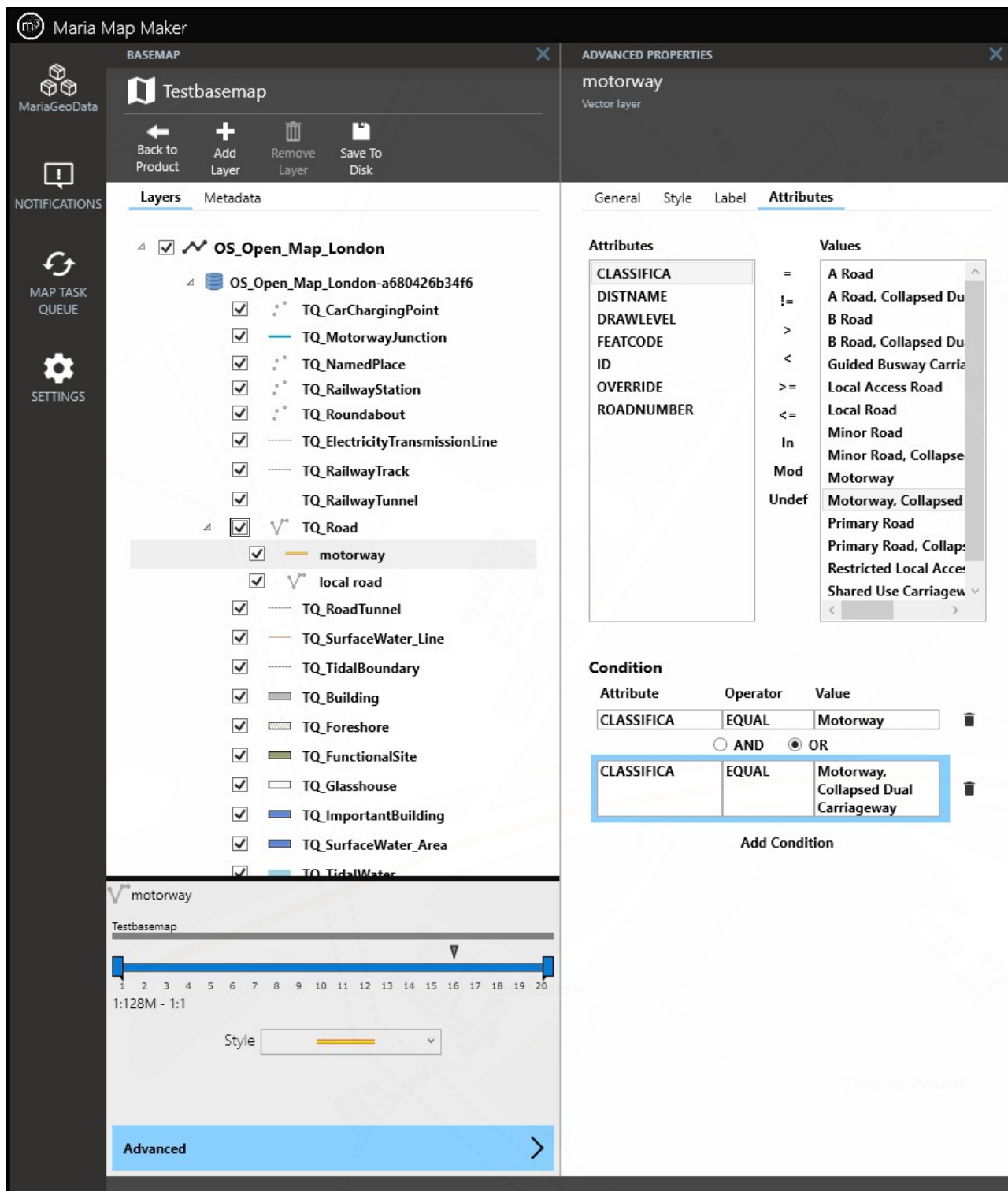
Right-clicking a layer lets you create new sub-layers or delete existing ones. You can also create new layers by right-clicking the dataset (the second level from the top).

15.2.2 Attributes

Open the advanced properties window and go to the **Attributes** tab. Here, you can get an overview of the attributes contained in the layer. You can also use the attributes to create conditions to filter geographic features.

The **Attributes** list shows all the attributes contained in the layer. Clicking one of them will show a sample of possible values for this attribute.

Below this, you can add one or more conditions. Make sure that the empty, default condition is selected, and you can then click a combination of attribute, operator and value. It's possible to manually type in a value if you can't find it in the **Values** list. If you add more than one condition, select either of the logical operators **AND/OR** to connect them. If you select the **Mod** (Modulo) operator, then the left part of the value field should have the divisor and the right should have the remainder. The **Undef** (Undefined) operator is satisfied if the attribute does not exist.



Adding two conditions where one of them must be satisfied for features to be drawn

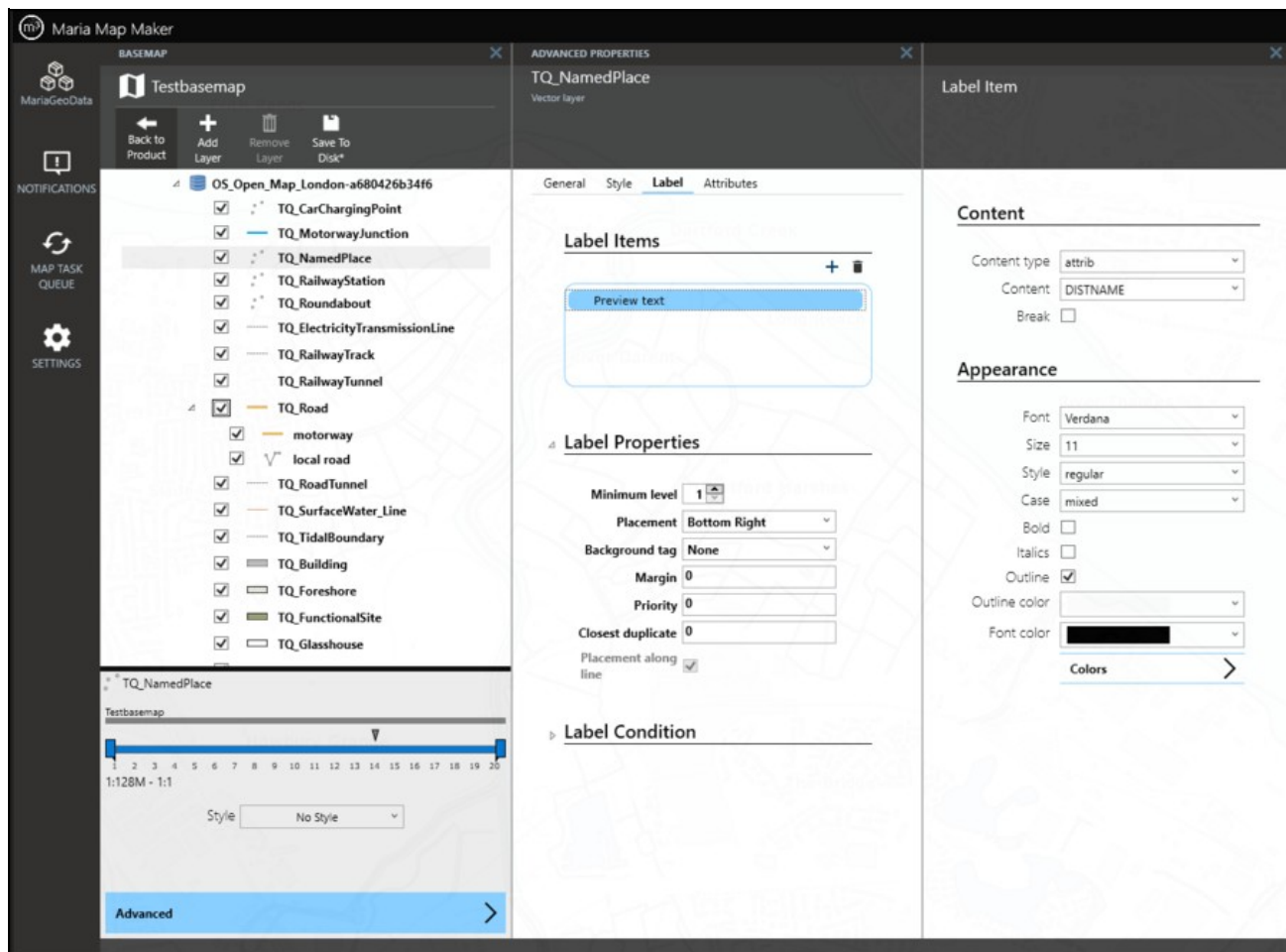
15.2.3 Labels

Each layer can have one or more labels. When you open the **Label** tab, use **+** to add a new label item. In the **Label Item** window, you start by selecting the content for the label, either an attribute or a static string. Below, in the **Appearance** section, there are a number of settings which control how the label text is drawn.

Back in the main **Label** tab, in the **Label Properties** section there are also a few settings. These will affect *all* the label items. The **Label Condition** section lets you set conditions similarly to the **Attributes** tab. These conditions decide if the

label is drawn or not.

It is possible to apply labels to all vector layer types (area, line, point). However, automatically placing labels on *area* features is hard to do well, especially with tiled vector data. For this reason, it is recommended that you make sure that the source data use *point* features for labelling if possible. This makes the placement of the labels much more predictable.

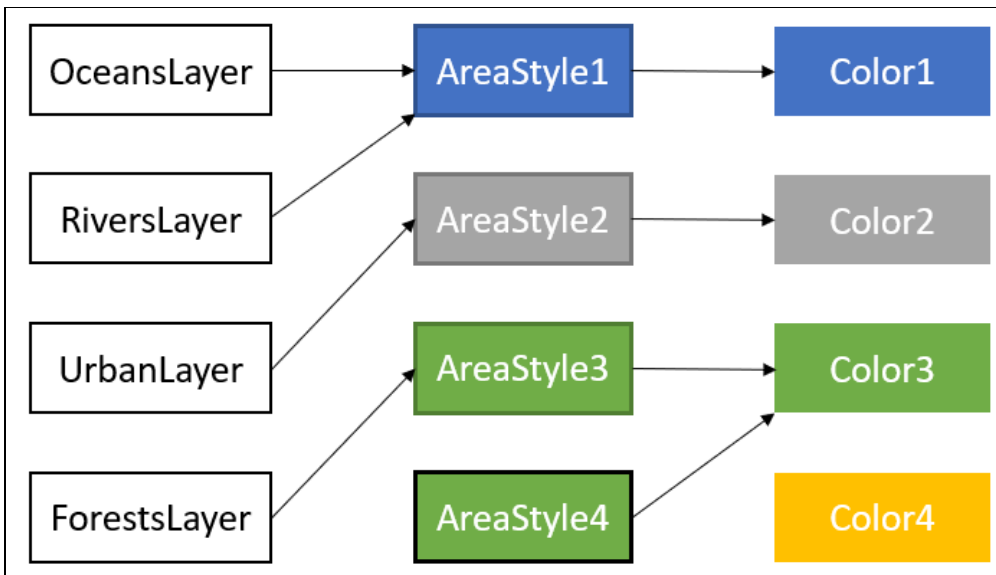


15.2.4 Style tab

The **Style** tab lets you create rules for how the vector data is displayed in the map.

Styling vector layers in M3 will be familiar to anyone who's used a GIS before, but there is one thing which might be different from what you're used to. We have a collection of layers and a collection of styles, and each layer can have a reference to one of the styles. Several layers can reference the same style. This means that if you change a style, all the layers referencing it will change appearance accordingly.

In the same way, there is a collection of colors which is referenced in the styles. If you change a color which is referenced in several styles, all the styles (and again all the layers) change accordingly.



Layers reference styles, which in turn reference colors. Changing Color1 to pink will result in pink oceans and pink rivers. This way of separating layers, styles and colors lets you re-use the combinations you create, and can save some work. If you find it confusing however, you can just create individual styles and colors for each layer, so that there is no overlap.

All layers can be set to "No style", and the layer data will not be drawn. Any labels will still be visible though.

Create descriptive names for all styles and colors to keep your collection tidy.

15.2.4.1 Points



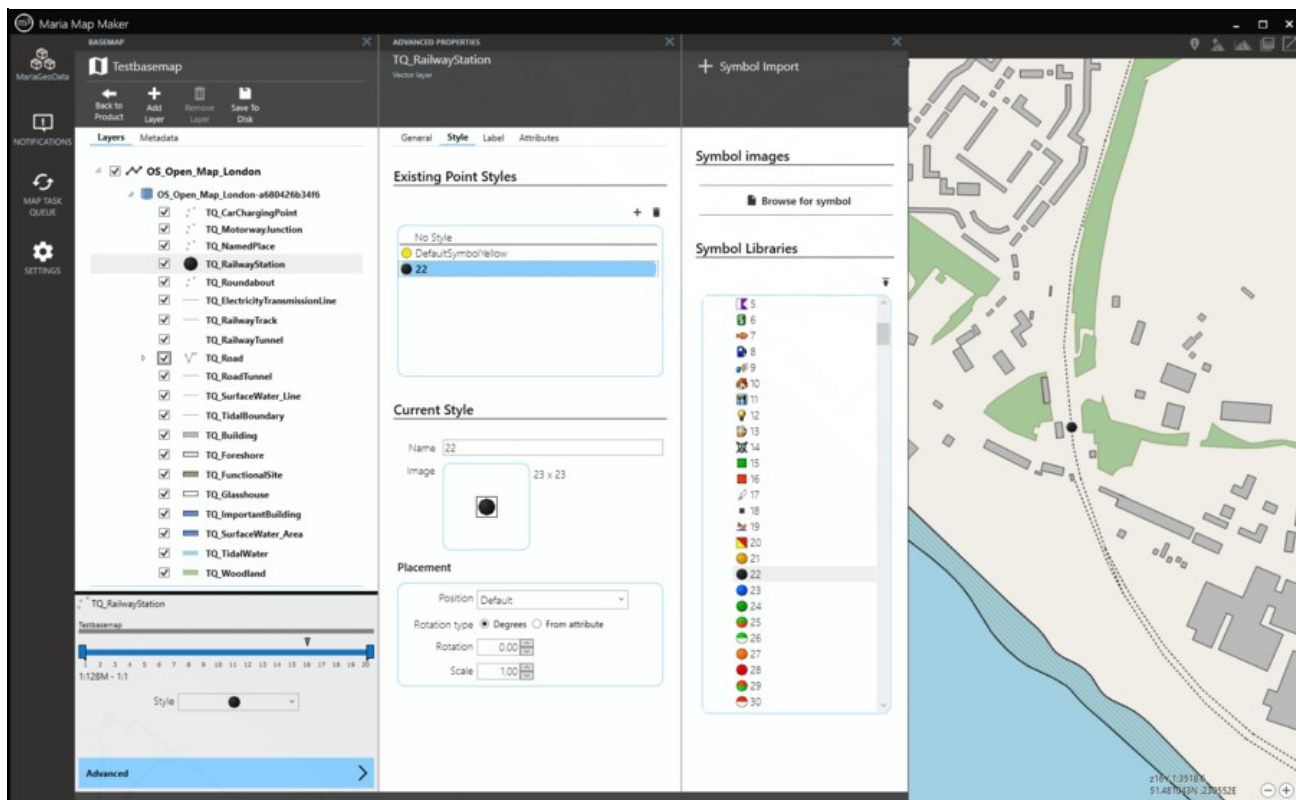
When vector datasets are created, a simple, default point symbol is included. This will be listed under **Existing Point Styles**. Double-click a symbol to select it for the current layer. Right-click to get additional options to duplicate or remove a symbol from the dataset.


Clicking an existing point style lets you adjust several placement options:

- Position - how to define the symbol position in relation to the point feature in the data.
- Rotation - rotate the symbol, either by a specified number of degrees, or by a number from an attribute.
- Scale - specify a decimal number with which to scale the symbol.

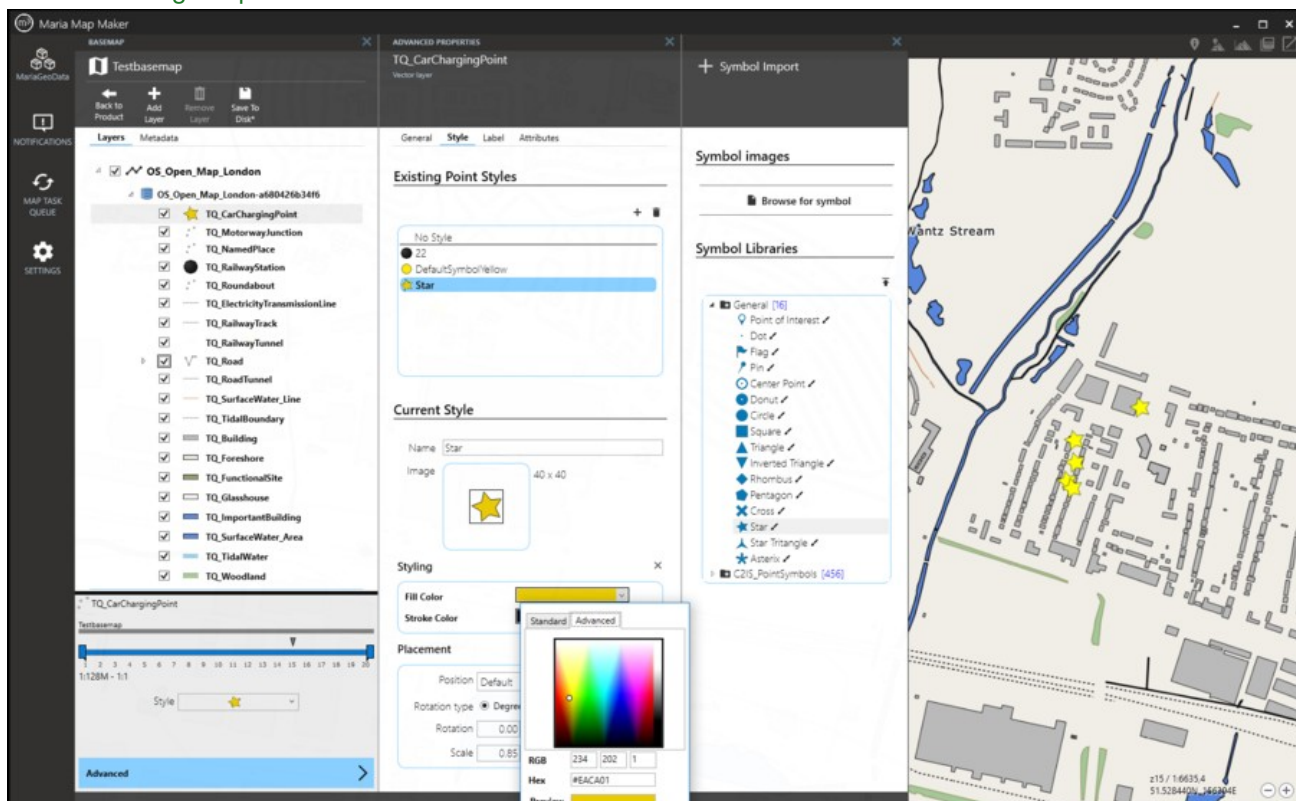
15.2.4.2 Symbol point libraries

You can import additional symbols from a **symbol point library**, which is included with the M3 installation. To open the **Symbol Import** window, click **+**. Here, you can import a library symbol into your dataset by double-clicking it. You can also browse to a PNG file on the computer.



In addition to PNG, point libraries also support the Scaled Vector Graphics (SVG) format. This format can be set up to have elements which can be styled by the user. These types of symbols will have a brush icon  next to it in the library list. When it's imported into the dataset, the style options will be available.

Don't worry if you end up with lots of symbols in your dataset. It can be automatically cleaned by deleting unused symbols when [releasing the product](#).



15.2.4.3 Lines

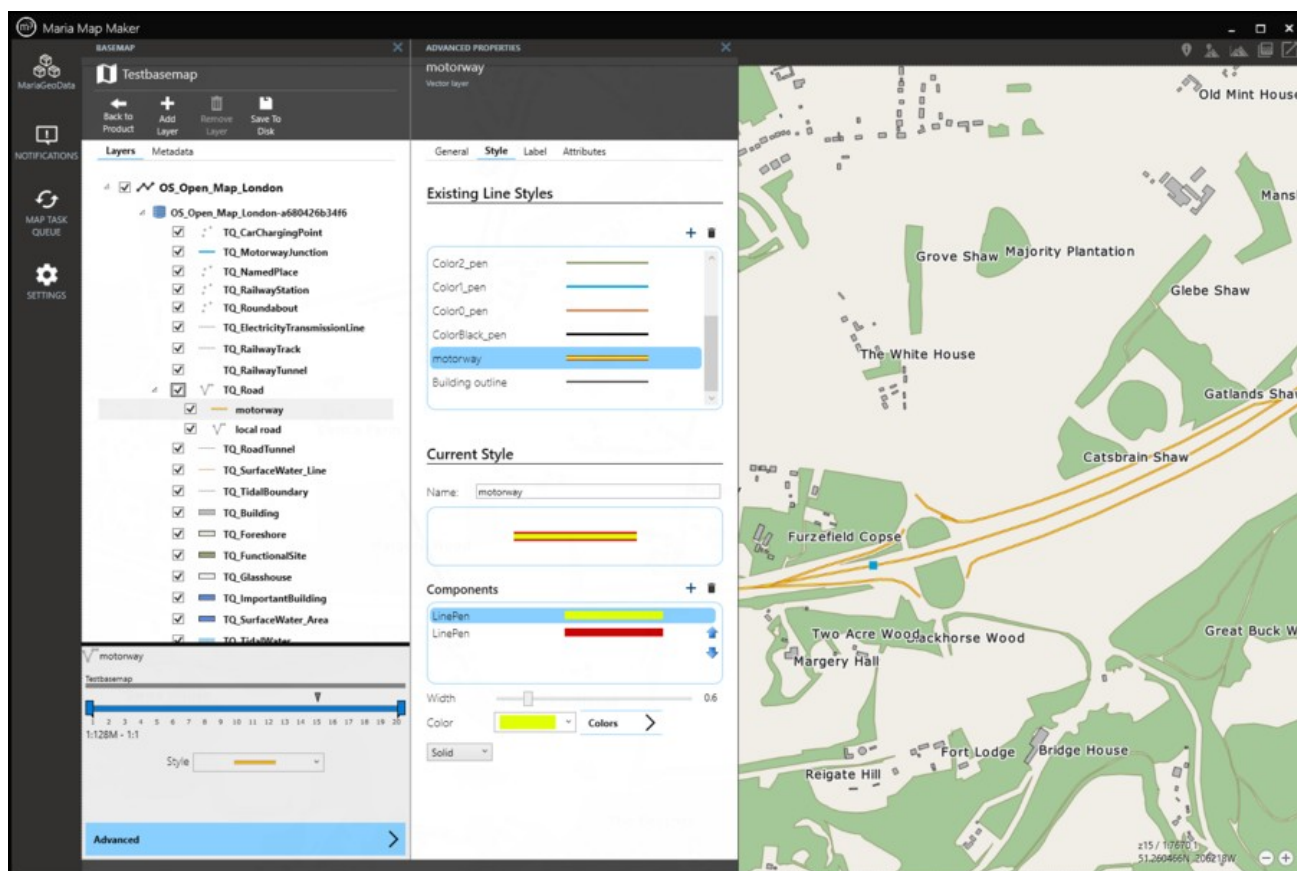


Existing Line Styles is the collection of line styles which are already defined. You can create and delete styles using the symbols to the top right of the list. Selecting one of the styles will make it the active style for the current layer. It also brings it up in the **Current Style** section below.

Each line style consists of one or more **Components**. You can add several lines of different **Width** and **Color**, which will be drawn on top of each other (use the arrows to change drawing order).

A line component can also have **Dashes**. In this case, a sliders will appear to let you control the length of the **Dashes** and the length of the **Gaps** between them. The actual appearance in the map will depend on the width of the line component itself, so you will have to experiment a bit to get the look you want.

For color you can either select one of the pre-defined colors from the drop-down menu, or you can use the **Colors** button to open a new window where you can create, delete and edit colors from the collection.



15.2.4.4 Areas

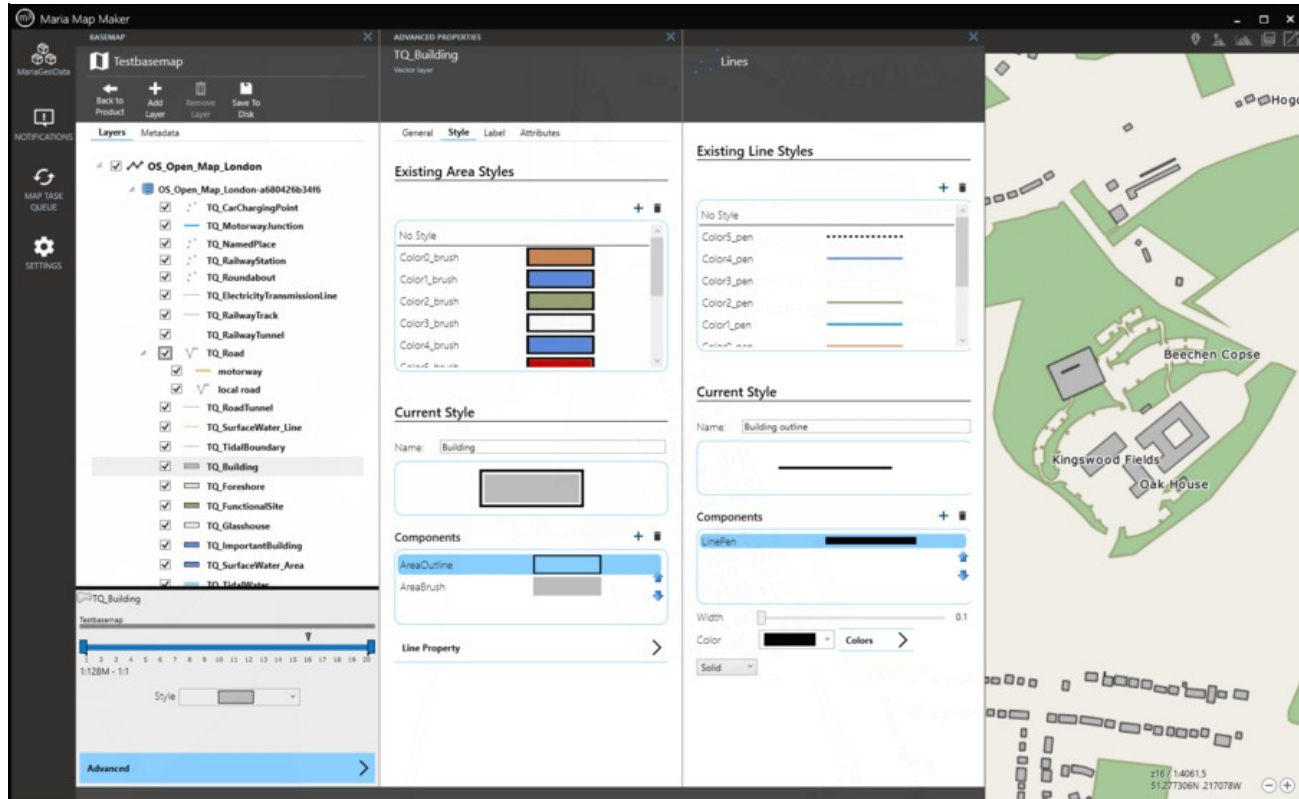


Styling areas is similar to lines. In each style you can add an **AreaBrush** (fill) component. The brush must be paired with one of the defined colors. You can also choose a **Pattern**, such as solid, horizontal or vertical lines, cross-hatch and more.

For the **AreaOutline** component, you choose a defined line style.

For the **Symbol** component you select one of the defined point styles. The map renderer will then attempt to draw this symbol at regular intervals on the area features.

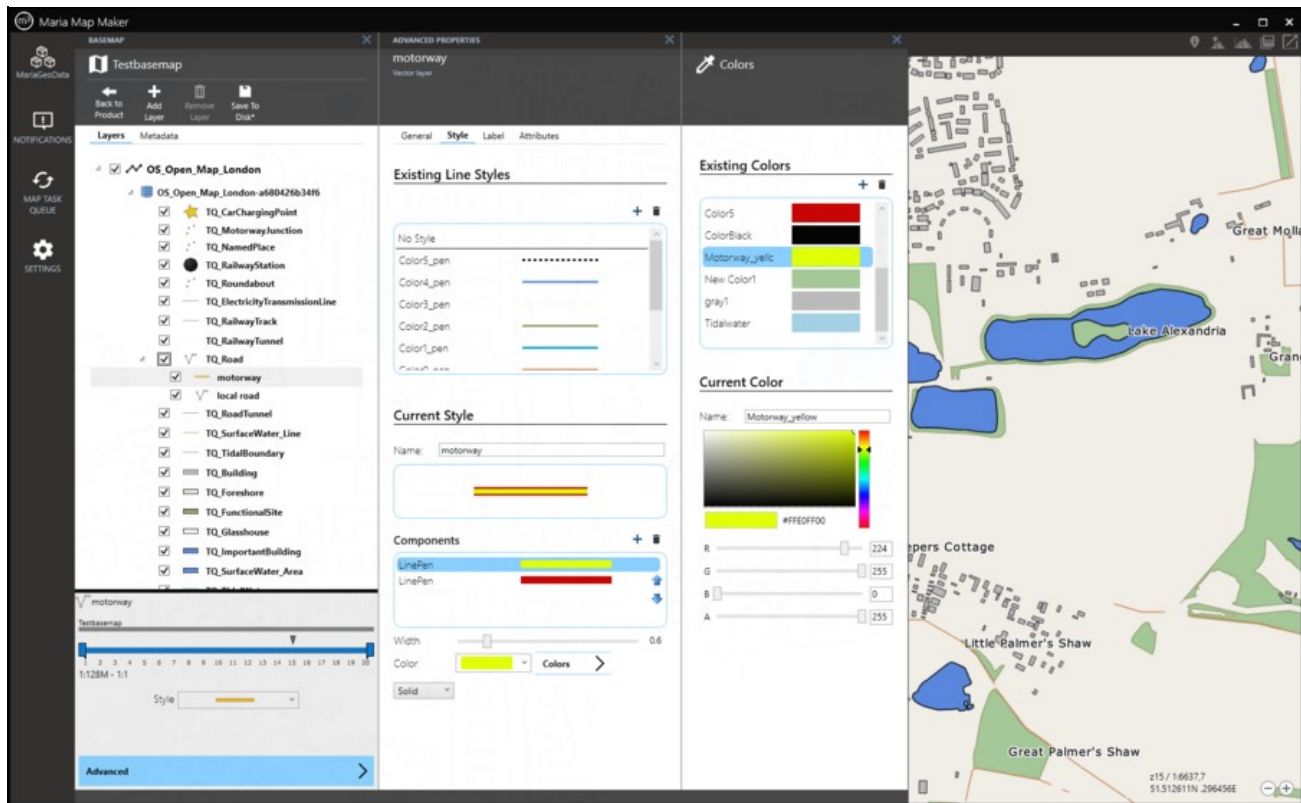
For the **Bitmap brush** component you use custom PNG files from the FillPatterns folder. The map renderer will then tile the area feature with this PNG file.



15.2.4.5 Colors

The **Colors** window is used when you want to edit the collection of defined colors. You reach it through the style tab of line and area layers, and in the label window.

You can create, delete and edit colors in the same way as with line and area styles. When a color is selected you can use the widget to visually select a color, or you can input hex or RGBA color codes.

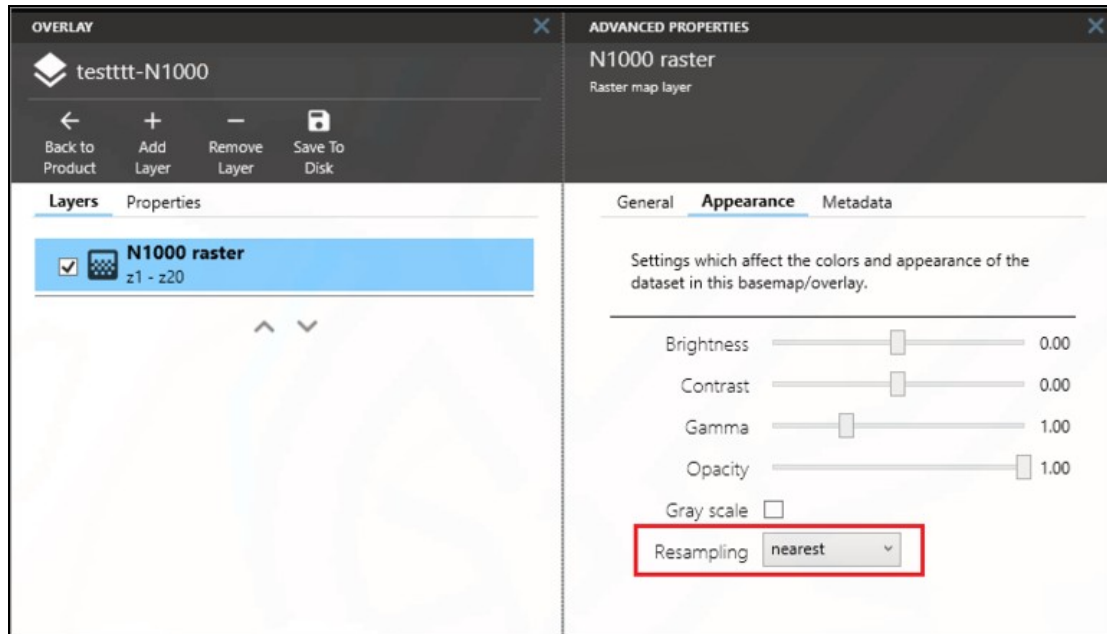


16 Raster layers



For raster layers there are few settings besides the visibility and zoom extents described in [Basemaps and overlays](#). In datasets which use GeoPackage files, most settings are input as parameters in the map import process, and can't be changed later.

If you have thematic raster data where the pixels should have precise values, you can change the rendering resampling method in the **Appearance** tab. Setting the **Resampling** parameter to *Nearest* will render each pixel as it is when zooming in past the native zoom level.



17 Web layers



There are three types of web layers available in Maria GDK and M3:

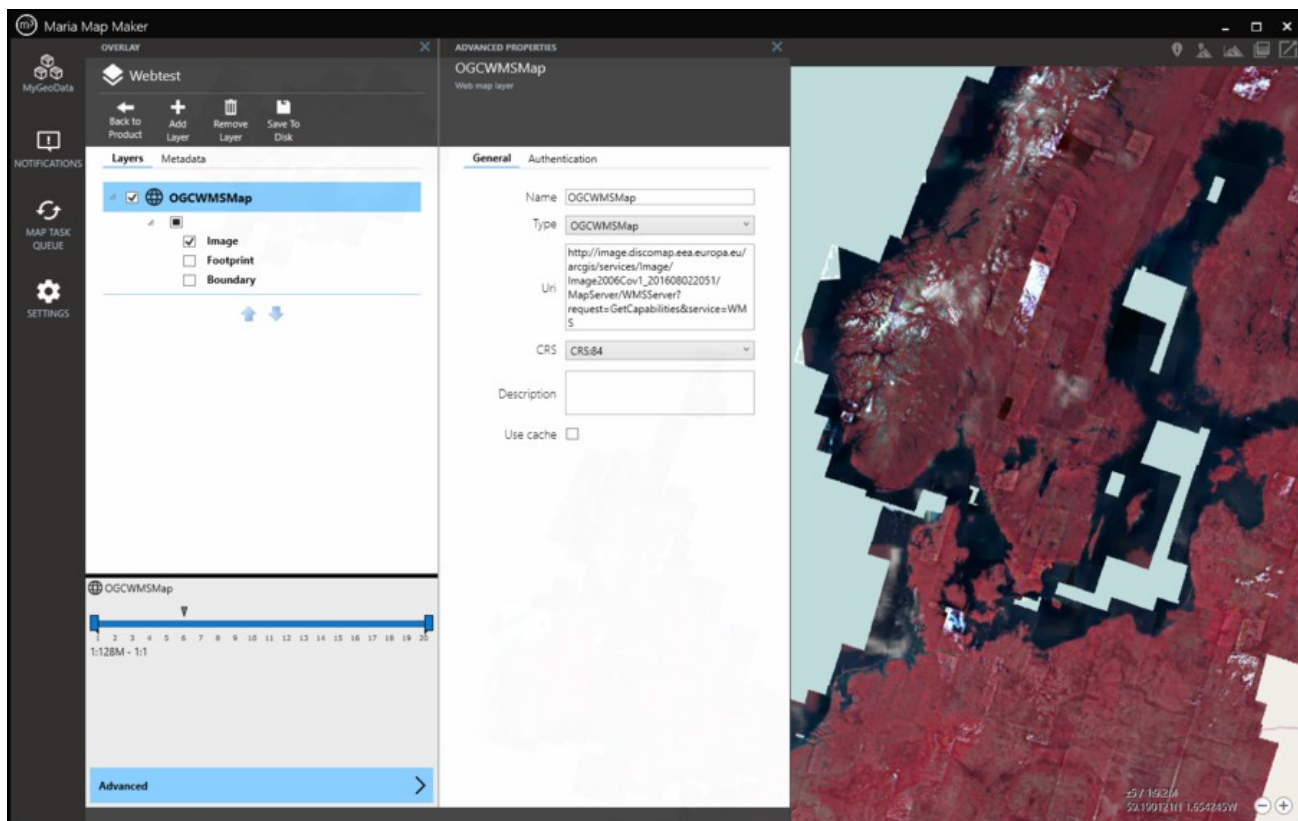
- Web Map Service
- Web Map Tile Service
- Tile Map Service

17.1 WM(T)S

The **WMS** and **WMTS** layers are similar in setup in M3. Both require you to set the address of the service (**Uri**) and the coordinate system (**CRS** for WMS, **Tile matrix set** for WMTS). The layer can then be expanded and you can view the sublayers provided by the service. For WMS you can select any number of sublayers, while in the WMTS layer you can only select one at a time.

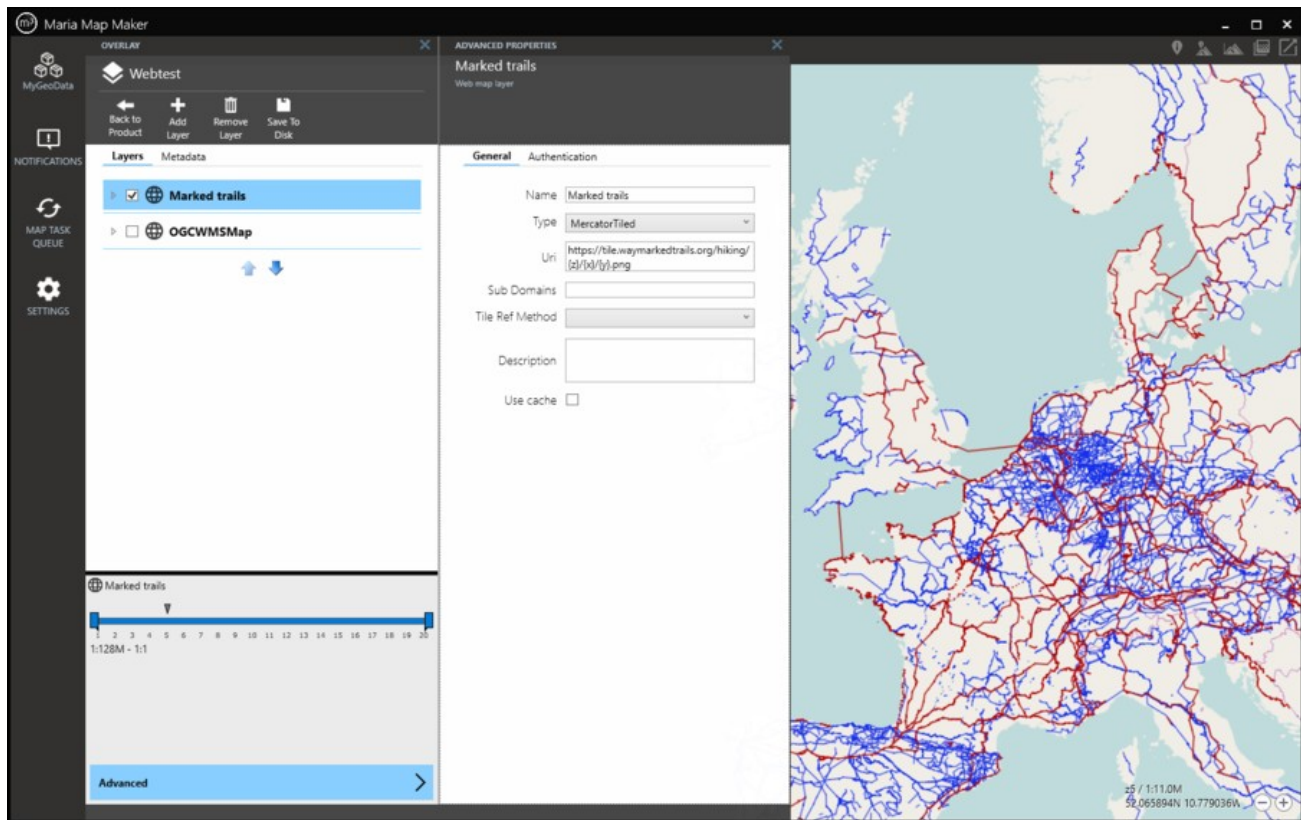
You need to use **Save to disk** for changes to be visible in the map.

The **Authentication** tab lets you input credentials if needed.



17.2 TMS

A Tile Map Service layer does not have sublayers, so setup is minimal. You need to provide a URL to the tiles with X,Y and zoom level placeholders, for instance <https://tile.waymarkedtrails.org/hiking/{z}/{x}/{y}.png>. If the server provides it, you can also enter a comma-separated list of subdomains (often "a,b,c")

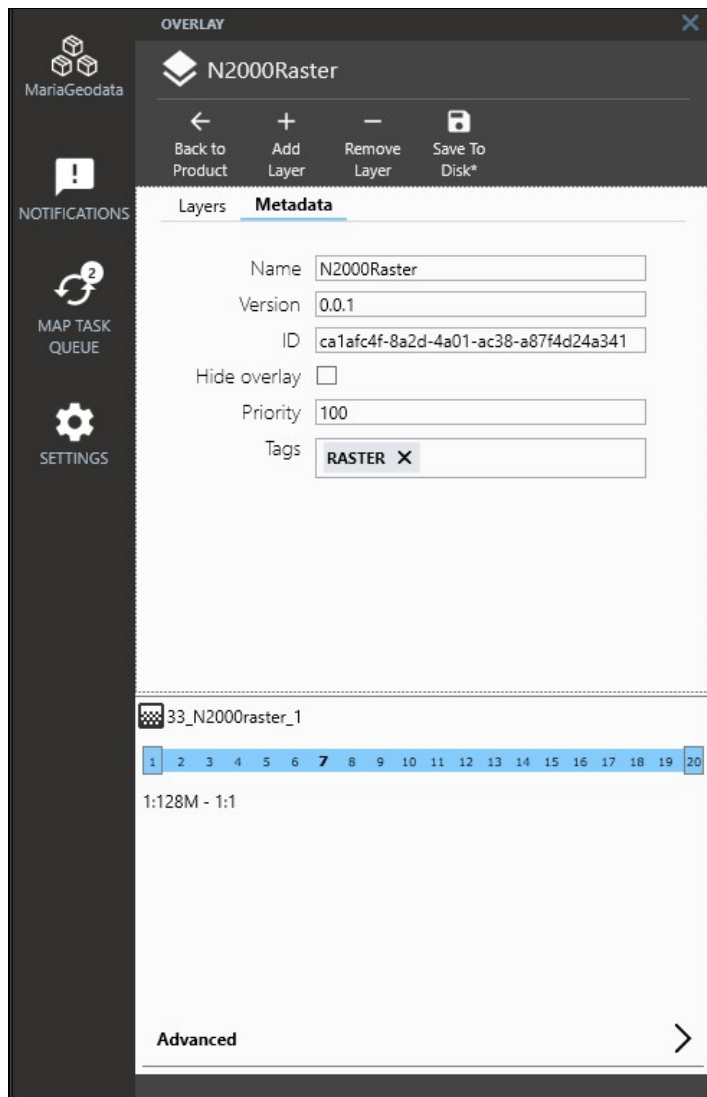


18 Dynamic layers

Dynamic layers are a special basemap layer type where instead of referencing a map signature, you add one or more tags. Any overlay (and its collection of layers) with the same tags will be displayed via the dynamic layer. The inner workings of dynamic layers and overlays is explained in the [Maria GDK core documentation](#)

18.1 In the overlay

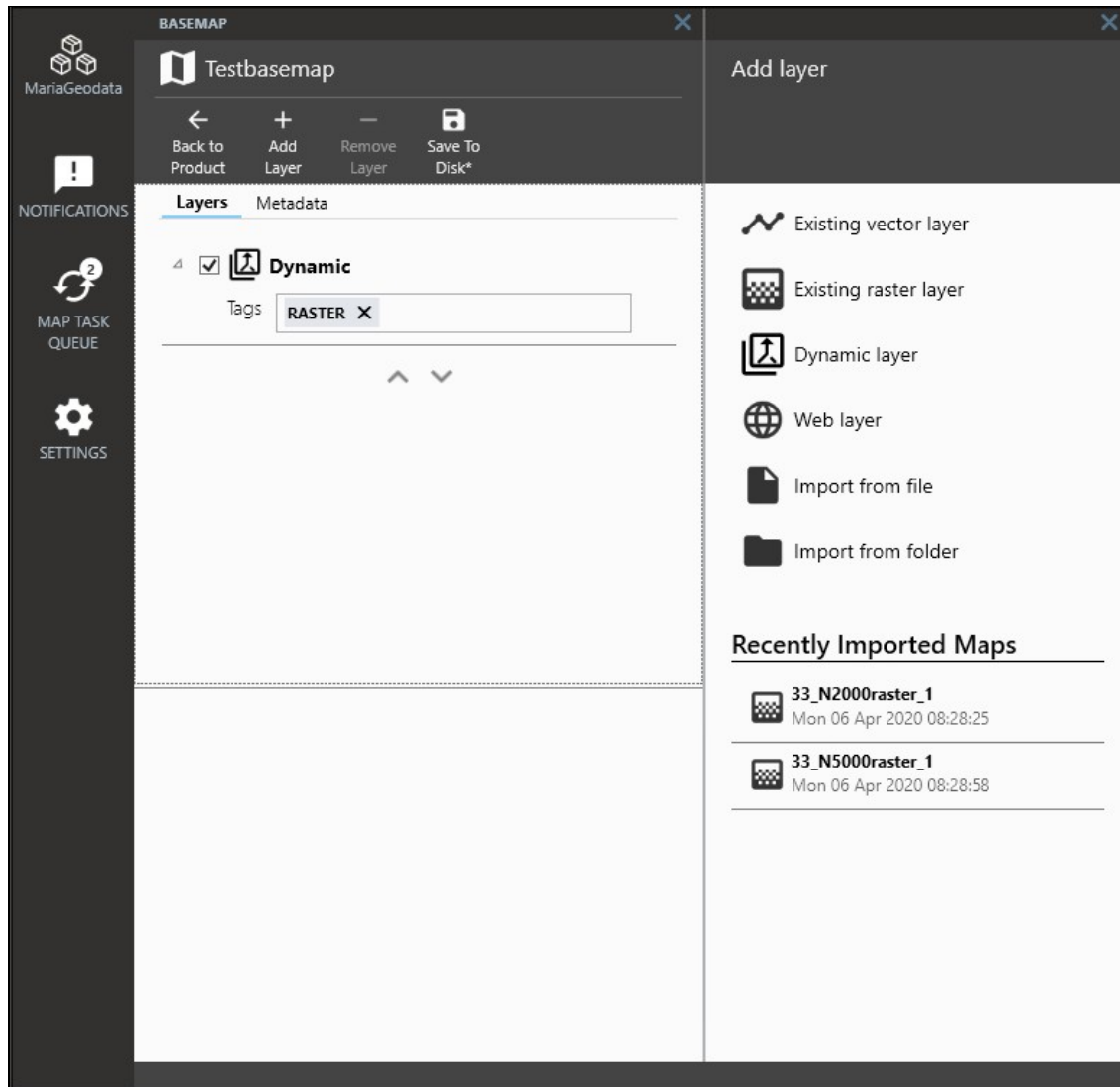
To tag an overlay, click the **Metadata** pane. Type a in a tag in the **Tags** field, and press Enter to complete it. You can add multiple tags, and you can remove them by clicking the "x" on each tag. The **Priority** field is a number which is used to decide the drawing order if there are multiple overlays with the same tag which also overlap. The overlay with the lowest number will be drawn on top. Finally, **Save to disk**.



Added the tag RASTER to an overlay.

18.2 In the basemap

In a basemap, use **Add Layer? Dynamic layer**. You can then add tags to the dynamic layer in the same way as in the overlay. The dynamic layer can be re-named and scale limited in the same way as other layer types. After clicking **Save to disk**, this basemap will display all overlays with the corresponding tag. If the overlays have multiple layers of their own, the internal drawing order and scale limits will be respected.



Added a new dynamic layer to display all overlays that are tagged RASTER.

19 3D Tiles



From version 4.2

Maria Map Maker is capable of importing, rendering and editing 3D models from a variety of sources. The format used for imported models is 3D Tiles (<https://www.ogc.org/standards/3DTiles>). A collection of models in the 3D Tiles format is called a 3D tileset, and contains all information about geographical position, textures and transformations to correctly place and render a model on a 3D globe.

19.1 Import

Importing models is handled by the library assimp. A full list of supported formats can be found at <https://github.com/assimp/assimp/blob/master/doc/Fileformats.md>, however many of the formats have not yet been tested in Maria Map Maker. The imported 3D tileset will consist of the models themselves in the format b3dm, as well as texture files, and several text files which define the position and relationships between the models.

19.1.1 Parameters

- **3D Tileset** - Choose an existing 3D tileset as a destination for the model, or create a new tileset.
- **Model** - Choose an existing model as a parent for the current import, or define it as a top-level model.
- **Compress textures** - Compress textures into a DirectDraw Surface container file. This reduces file size, but should not be used if you intend to use the tileset in the Cesium platform, as it's not supported there.
- **Auto place** - Attempt to find geographic position for the model in the source files.

- **Latitude / Longitude / Elevation** - Manually enter position and elevation for the placement of the model. Only used if "Auto place" is disabled or unable to find geographic position in the source files.
- **Unit select** - Unit in the source data - meters or feet.

19.2 Navigation

When you open a 3D tileset, Maria Map Maker will enable 3D mode, which will render the background map on top of the highest resolution elevation data available in your workspace (or external sources). You can also switch between 2D and 3D at any time using the button at the bottom right of the map window. There are several ways to navigate the 3D environment:

Movement

Move camera along the surface

Tilt and rotate camera around a point

Move along the camera direction

Mouse / keyboard

Left mouse button on the terrain / keyboard arrow keys / WASD keys. Hold Ctrl to move slower.

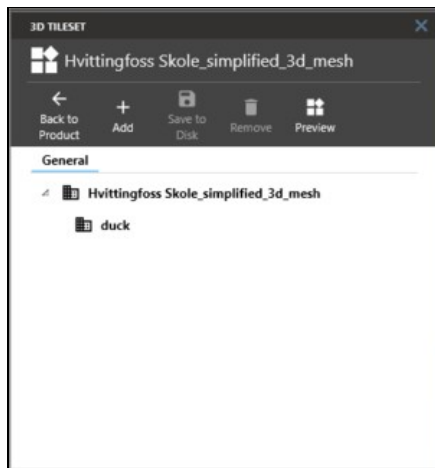
Right mouse button

Mouse wheel up/down

19.3 3D tilesets

3D tilesets are stored in products in the same way as other types of datasets. They can be opened by double-clicking them in a product's **Contents** tab.

19.3.1 Model hierarchy



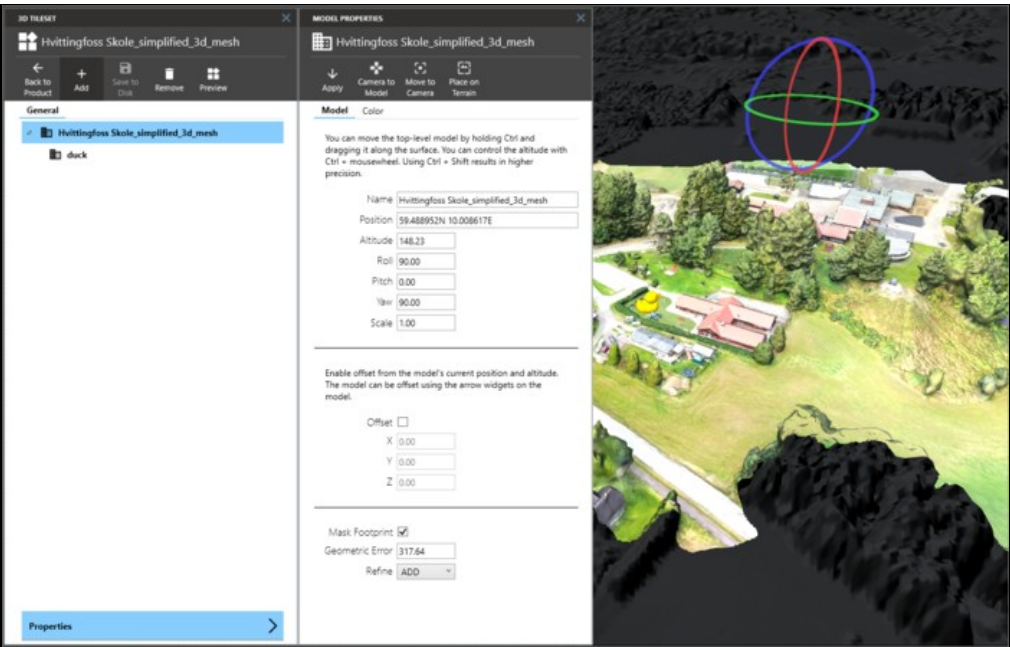
The 3D tileset is represented as a hierarchy of models. You can import a new top-level model with the **Add** button, or you can create a sub-model by right-clicking a specific model in the hierarchy and clicking **File model**. This will open a browser for any supported 3D model formats, and then initiate a Map import.

Double-click any model in the hierarchy to open its properties.

You can delete models from the tileset by right-clicking it and selecting **Remove**.

The **Preview** button hides the model widgets and renders the 3D environment with all 3D tilesets in the current product.

19.3.2 Model placement



An important feature of the 3D tileset is that any moving, rotation or scaling of a top-level model will also apply to all its sub-models. To adjust the placement of a model, double click any model to open its properties in a window to the left. Most changes must be put in effect using the **Apply** button at the top of this window.

A top-level model's placement on the globe is defined by geographical coordinates (latitude and longitude) and an altitude above the surface of the globe. Vertical datum is not considered.

The position and altitude of a sub-model is only defined as an offset from its parent model. It is also possible to enable offset on a top-level model, in which case the offset will be in relation to the values in its Position/Altitude fields.

The following table shows how models can be placed, offset and rotated in the 3D environment. The method varies according to whether the model is at the top level or a sub level.



Action	Model level	Mouse / keyboard
Move model in any direction along the surface	Top level	Ctrl + Left mouse button on the model. Holding Shift as well increases precision.
Move model up and down	Top level	Ctrl + mouse wheel up/down. Holding Shift as well increases precision.
Offset model along X,Y, or Z axis	Top level or sub-level	Left click the arrow components of the model widget
Rotate model about X,Y, or Z axis	Top level or sub-level	Left click the circular components of the model widget

The top bar of the **Model properties** window gives you a few convenient buttons related to the placement of a model:

- **Camera to Model** - Move the camera to the position of the model. Only the position of the camera is moved, not the altitude.
- **Model to Camera** - Move the model to the position of the camera.
- **Place on Terrain** - Calculates the average height of the model and places it on the terrain at the current position

19.3.3 Other properties

19.3.3.1 Scale

The model can be scaled by a user-defined scale factor. This is mostly relevant for instance models - for instance vehicles which are placed around the 3D environment for added realism.

19.3.3.2 Mask Footprint

Allow the model to mask any terrain that would otherwise clip through it.

19.3.3.3 Geometric Error

The maximum allowed error in meters before a model is refined and its sub-models are considered for rendering.

19.3.3.4 Refine

Refinement determines the process by which a lower resolution parent model renders when its higher resolution children are selected to be rendered. If the model has "REPLACE" refinement, the child models are rendered in place of the parent, that is, the parent model is no longer rendered. If the model has "ADD" refinement, the children are rendered in addition to the parent model.

19.3.3.5 Color

Ambient lighting. Changes the base color of the model texture. White (255,255,255) is default.

19.3.3.6 Highlight Color

Specular lighting. Color of the lightsource which hits the model. Black (0,0,0) is default.

19.3.3.7 Shininess

The amount of light which reflects off the model.