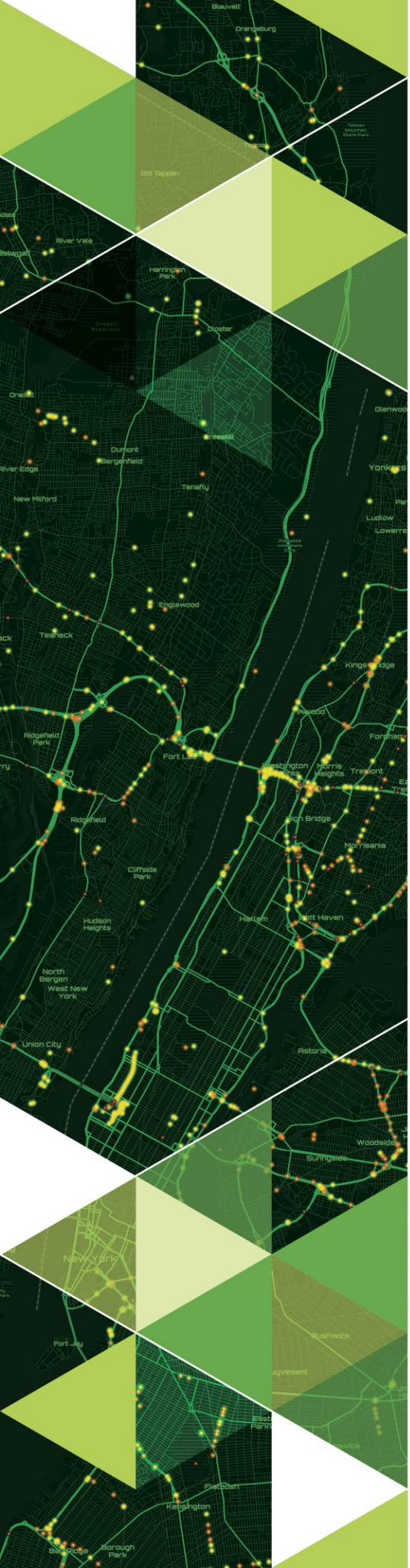

AN ESRI
TECHNICAL PAPER

July 2026

Mapping with Vector Tiles

380 New York Street
Redlands, California 92373-8100 USA
909 793 2853
info@esri.com
esri.com

esri | THE SCIENCE OF WHERE®

Table of Contents

Mapping with Vector Tiles

Executive Summary

From making to sharing, vector tile users have many options in ArcGIS Pro, supported by detailed help pages and guidance. However, the sequence of setting up a map and generating vector tiles with the right combination of settings is not always obvious, particularly when what appears on screen as input does not always match the published result. This paper explores some of the most common sticking points and outlines a practical, end-to-end workflow.

ArcGIS Pro is a mature, professional GIS with an extensive set of tools and configuration options designed to support a wide range of mapping workflows. This breadth of functionality can make it challenging to focus on the specific requirements of authoring vector tiles. Vector tile authors must be able to define scale-dependent styling behavior and understand precisely what will be shown, how it will be rendered, and why. The ArcGIS Pro interface does not always make these vector-tile-specific behaviors explicit, contributing to a gap between authoring intent and published outcomes.

The aim of this paper is to provide a practical bridge between intended map styling and real-world results. It addresses the more common questions and frustrations encountered in practice, shedding light on how vector tiles work and how to best prepare and publish them using ArcGIS Pro and the wider ESRI ecosystem.

Creating Maps for Vector Tiles

Step 1: Prepare source data

Is it updateable and compatible?

Like any workflow, it helps if any updates to source data can be adopted seamlessly by your map and already be appropriately generalized without duplicates or geometry errors.

Though consider that not all data types are supported for creating vector tile packages or for sharing as web layers. Such data will be excluded from the output unless converted to/resupplied as a supported feature layer.

Only point, line, polygon and multipoint features are supported. Sharing as a web layer has further restrictions on file or data type—see *Sharing as a web layer: Supported data types*, p24.

Also see *Editing Vector Tiles*, p24.

Step 2: Know your target platform

Is the final map going to be published and viewed online or shared in Pro? Is it going to be viewed with other layers or basemaps?

Online or browser-based renderers such as Map Viewer typically display maps only at fixed, predefined scales. If a standard web map is your primary target, it is best to work within those same fixed scales when authoring your map in Pro. This means ensuring that all minimum and maximum scales on layers, symbols, and labeling exactly match one of those defined scale levels. Doing so makes it much easier to understand what is visible at each scale, and when transitions occur.

Being intentional about where scale-dependent transitions—such as visibility, display filters, sizing, and symbol changes—take place ensures that your vector tile map can be consumed consistently across platforms and paired effectively with appropriate basemaps. It also helps avoid unnecessarily storing geometry at levels where it will never be used. *See Step 4: Work out your scale ranges.*

If instead you are targeting use in Pro, or another renderer that supports continuous zoom, the map can be designed accordingly. Changes in content and symbology can occur at whatever scales work and look best—whether aligned to other products or introduced as frequently as needed. However, it is uncommon for vector tiles to be viewed exclusively in Pro, so ensuring that transitions also behave well at standard “online” levels of detail (LODs) is often the more practical priority.

Good continuous zoom behavior comes from carefully choosing when features appear or disappear (scale thresholds), while remaining mindful of the tiling scheme being targeted and ensuring consistent behavior across platforms. The tiling scheme should not dictate the design, but it provides a useful framework that helps guide those scale decisions.

In practice, a tiling scheme provides a set of structural reference points:

- Discrete zoom levels (LOD0, LOD1, …)
- Associated map scales and resolutions

Even in a continuous zoom renderer such as Pro, these discrete levels still matter because:

- Web maps (raster and vector tiles) are built around them
- Digital users tend to think in terms of familiar zoom levels ("level 10", "street level", etc.)
- Filters as well as other platforms often snap to those predefined scales

The LOD concept is introduced in more detail in *Step 4*, p8, and explained on p26.

Step 3: Work out your map scales and set up your map document

Setting up Pro to respect your map scales

a) Check your coordinate system

Before anything, check that the coordinate system of your map in Pro is the one you wish to publish in. This well help ensure you get the scales and scale ranges set up correctly. See *Set the coordinate system of your map*, p19.

b) Turn off inclusive scale ranges

Right-click on the **Map** in the Contents pane and select **Properties**.
Under the **General** tab, ensure the checkbox for **Draw up to and including the maximum scale in scale ranges** is unchecked.

This allows us to define feature or symbol rules for ranges such as 1:10,000–1:5,000 and 1:5,000–1:1,000 without concern that both will draw at 1:5,000. We don't need to force artificial values such as 1:5,001 or 1:4,999.

It also ensures that layer scale ranges, scale-based symbol classes, and the resulting vector tiles behave consistently: the feature or symbol disappears cleanly once we reach the maximum scale of the range.

c) Set view scales

We recommend making any fixed or default scales the final map will use, the only ones you see in Pro, at least initially.

Click the **map scale** dropdown beneath the map window and click on **Customize** to open the Scale Properties window. **Delete All**, then either **Load** "ArcGIS Online / Bing Maps / Google" if using standard Mercator web map scales, or else load the values from a csv or text file—see next paragraph for details.

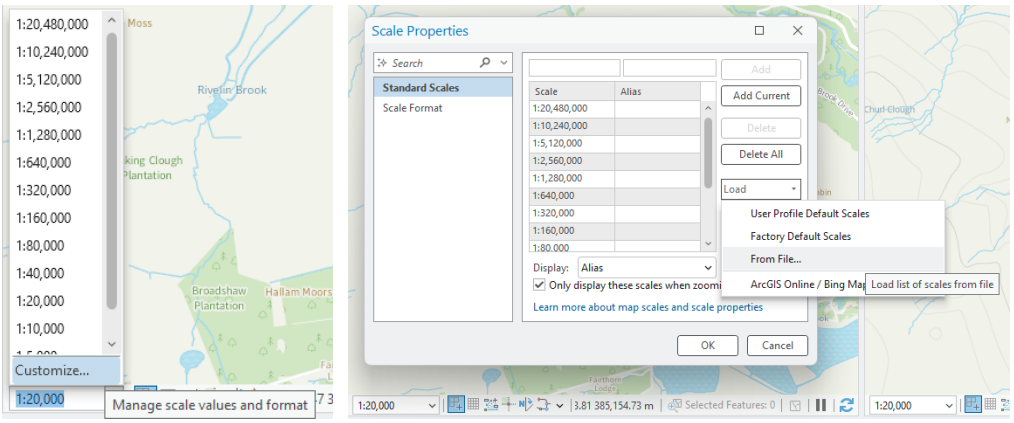

Figure 1: Scales and scale properties

Using a file, you are free to use any scales you wish so long as each scale is twice (or half) the previous. However, to avoid any issues, you are generally best served by using the scales already provided by Pro.

To do this, first find the scales relating to your coordinate system and tiling scheme, and then create the CSV or TXT file to use. This can be done in one of two ways:

1. Open the **Create Vector Tile Index** tool, uncheck **Package for ArcGIS Online etc.**, open the XML file that appears in the **Tiling Scheme** box. Manually copy and paste these scales into a csv or txt file, or

2. Open up the dialog for the **Create Vector Tile Package** tool and populate with the items you plan to use. Then open the cached scale dropdown and manually type these scales into a csv or txt file.

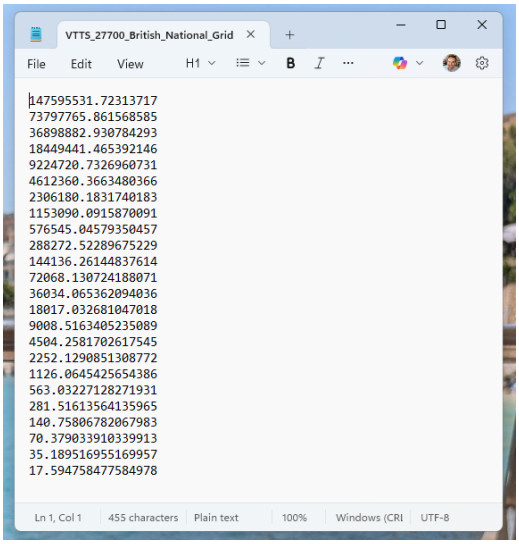

Figure 2: Example of one such text file, for British National Grid

Note: For a text file, the scales must be a list on separate lines, any second value on the same line will show as an alias.

Step 4: Work out your scale ranges

When will features, their symbols, or labels come and go?

When working with scale ranges in Pro, it helps to imagine zooming in on the map. The minimum (or smallest) scale is when we want something to first appear, then as we zoom in further, the maximum (or largest) scale is when we want it to disappear.

Browser and web maps use a concept called Levels of Detail (LOD) and Zoom Levels which we will discuss later but in general they are the amount of detail shown at various map scales.

So, say for example you want forests to appear at the “Cities” zoom level and then disappear at the “Neighborhood” level; then, in the default ArcGIS Online setup, that corresponds to a minimum scale of 1:144,448 and a maximum scale of 1:18,055.

By selecting the layer in **Contents** pane, and opening the **Feature Layer** ribbon, you can populate these scales in the **Visibility Range** section. Rather than type these, you can choose them from the dropdown. Alternatively, this can be done via *Layer Properties > General*.

If you are using a different coordinate system or tiling scheme, you should see these in your dropdown of map scales instead (if not, follow *Step 3*, above).

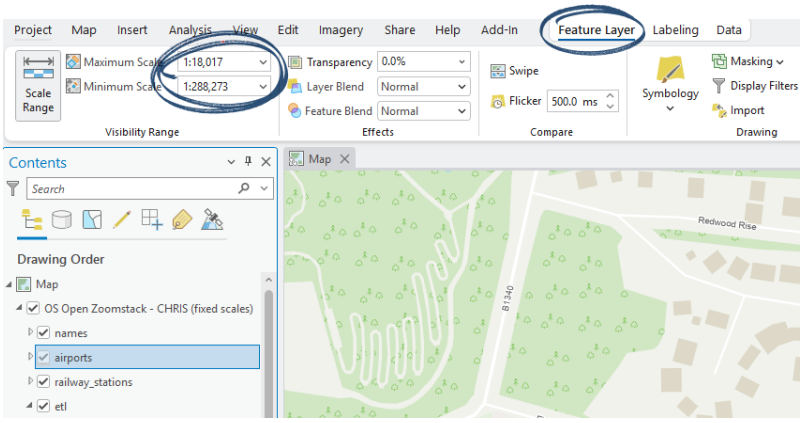

Figure 3: Setting feature layer level scale ranges

If designing for continuous scales, you are likely to experience a bigger difference between (1) what you see in Pro before tile creation, (2) as tiles in Map Viewer, and (3) as tiles in Pro. The secret here is to trust in the scales you are setting up for each feature and accept that in a fixed LOD viewer they may jump to the nearest LOD. This may cause mismatches between scale thresholds and rounding differences may also occur.

Also be aware that straddling scales means storing the same features in two LODs, which will increase the combined file size of the tiles.

Instead, stick to published scales whenever you can, and make use of multi-scale layers using the sliders in the “Scale Range” view. This allows you to set different symbol variants through the required scales and removes the need for duplicate

layers. Duplicating content is less efficient, also resulting in greater total file size. See *Symbol scales, Symbol classes and Display filters*, p13, for further detail.

For further explanation of zoom levels and levels of detail (LOD), see *Understanding cooking, caching, leaf tiles, zoom levels and LODs*, p26.

Step 5: Prepare your map styles

Make or refine your symbology and labeling with vector tiles in mind. Prepare any pop-ups.

When it comes to map symbology and labeling, not everything in Pro is compatible with vector tiles.

If you are only sharing the vector tiles and not the original styled data from Pro, then there is no point setting symbology rules that make things look nice in Pro if they're going to be ignored in vector tile creation. It also makes what you see further from what you get, thus making it harder to work with.

What follows is a detailed description of what does port to vector tiles and examples of how some style appearances change.

Symbology

Supported symbology includes Pro's range of classification methods, most symbol properties, and scale-based display controls.

Uniting expectations and results for positive outcomes across various scenarios

Symbol complexity beyond that of the standard vector tile specification, is usually either simplified to a more basic appearance or is rasterized into a PNG sprite. As a simple example, a wave effect on a polygon outline is not possible to show with vector tile symbol specification. Instead, the outline will be output as a basic line with the same color and weight.

Further examples

Boundary example

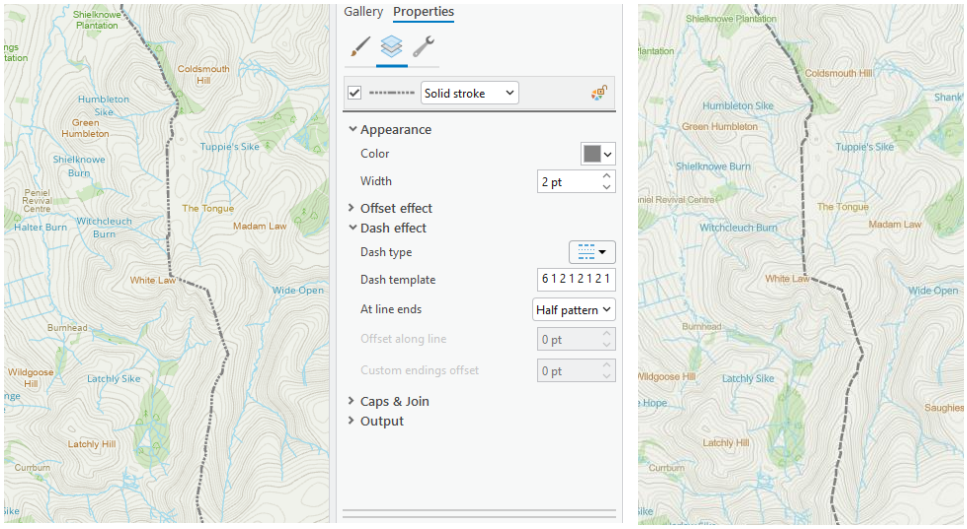

Figure 4: Complex dash template such as boundary lines. Features styled in ArcGIS Pro (left and center) versus the output vector tiles as viewed in Map Viewer (right)

Complex dashed lines created in Pro are simplified to the length of the first dash and following gap. Preceding gaps and any further dash and gap information is ignored as the vector tile currently only handles a simple 2-value dash array.

Choosing the more complex boundary from the ArcGIS 2D symbol gallery, the dash template **6-1**-2-1-2-1-2-1 will be read into vector tiles as **6-1**, the first two characters, thus completely changing the appearance of the boundary line.

So, for vector tiles, there are two options:

1) Create or choose a simpler boundary symbol that is a simple dash array, in figure 5 this is a 4px dash and 4px gap template. Note that the half-pattern ends will create a slight shift; to avoid this set **At line ends** to *No constraint*.

2) For Online or Enterprise, you can correct the **line-dasharray** property to the more advanced template in the Vector Tile Style Editor after publication. The complex array is then honored by ArcGIS Online, ArcGIS Enterprise, and the JavaScript API.

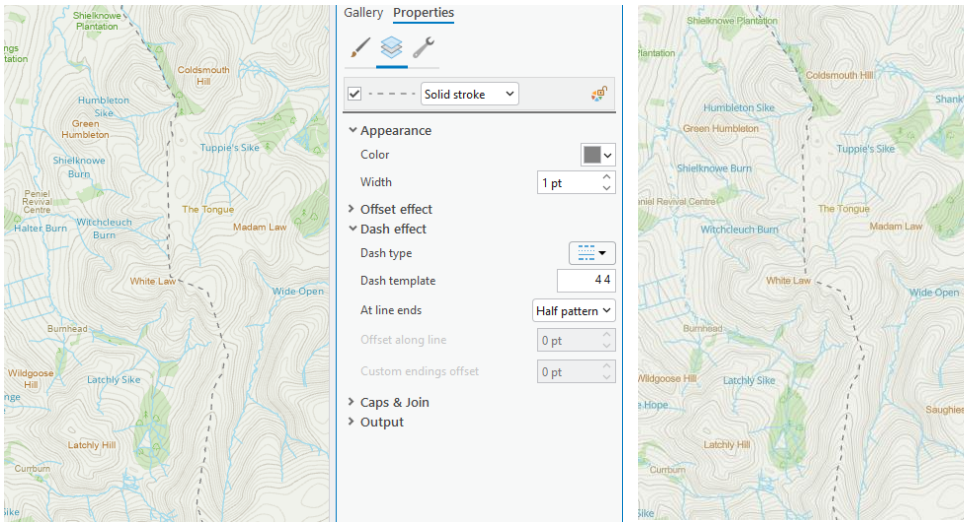

Figure 5: Improved dash consistency with the simpler array

Forest or woodland, and Functional sites examples

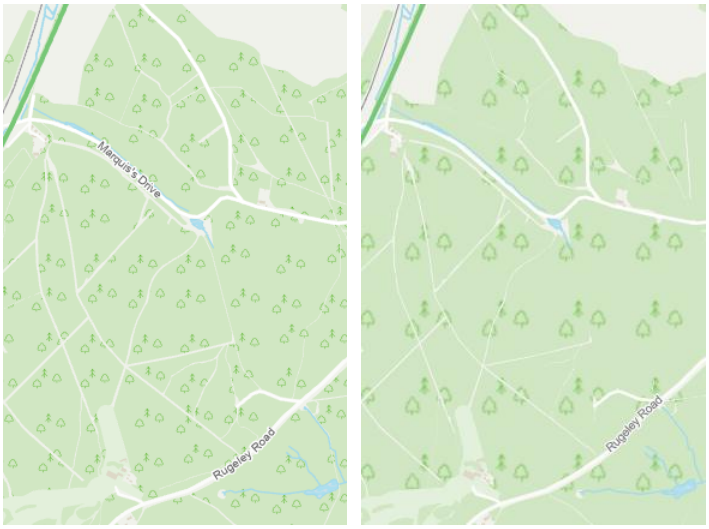

Figure 6: Size of marker fill symbols in polygons; styled features in ArcGIS Pro (left) versus vector tiles (right)

Polygon marker fills, e.g., forests or woodland with vegetation marker symbols, are converted to a seamless raster, i.e., a repeated pattern without visible seams. The style JSON in the vector tile package will refer to a "pattern" which is essentially this raster stored in the Sprites folder. You will notice the scale or size of the pattern may not look identical to the original marker placement, particularly when opened in a different application such as Map Viewer. This is because the sprite is saved at a set resolution of 192ppi. If your screen or application resolution is less than 192ppi then the pattern will look enlarged, if your resolution is greater than 192ppi then it will look reduced in size. As sprites are raster, despite being high quality, they will degrade the more you "zoom in" and are not going to scale to higher resolutions for export or printing.

Hatched fills are currently converted to solid fills using the same color value. A workaround is to use raster hatched patterns but with the same sprite limitations as described above.

Figure 7: Hatches. Original views in ArcGIS Pro (left) versus the output vector tile as seen in Map Viewer (right); with vector hatches shown top and raster hatching shown bottom.

Road shield (number) example

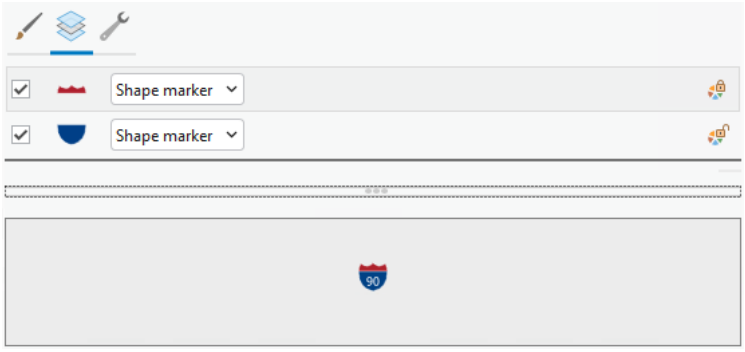

Figure 8: Road shield symbol composition in ArcGIS Pro

The technique used to convert Road (highway) shields and non-simple-shape point symbols from Pro to vector tiles depends on how they have been created, i.e. what the underlying symbol type(s) is. See *Appendix A.*

US highway shields, specifically “Shield 1” from the symbol gallery, comprise of the text label and an embedded point symbol made from two shape markers. These two shapes are saved as a single rasterized sprite. The differences seen between the original vectors in Pro and the sprite in vector tiles may include pixel shifts and loss of quality (resolution).

See *Appendix A* for a handy table of what is and is not compatible with vector tiles at the time of writing.

Symbol scales, Symbol classes, and Display filters

We strongly recommend using these controls in the Symbology pane as they provide a great way to control both feature visibility (what features on a layer are shown, and when) as well as to symbolize them differently at different scales. They help avoid multiple scale-specific copies of the same layer, improving map performance and maintainability.

For example, scale-based symbology is handy to change the size of point symbols or widths of line feature symbols between scales; symbol class scales are very useful for changing the symbology of features like roads (beyond just size) between scales; and a display filter could be used to show less contours (or a larger contour interval) at smaller scales.

Varying symbology size by scale

If the only variance needed between scales is the symbol size, then scale-based sizing is the most efficient way to change the size of a symbol, e.g., width of road lines, between the different zoom levels.

From either the **Symbology** pane or the **Contents** pane, click on the graphical preview of the symbol to open the **Format Symbol** options. Change from the **Gallery** to the **Properties** tab.

Under **Size** or **Line Width**, check the box **Enable scale-based sizing**.

Here, using airports as an example, we start with a symbol size of 14 pt. Once "Enable scale-based sizing" is turned on, we can adjust the symbol size at each end of the visible scale range by dragging the corresponding handles on the slider.

For example, leaving the smallest visible scale at 14 pt and setting the largest visible scale to 24 pt causes the symbol size to interpolate linearly between those values as we zoom in on the map. The number of intermediate size steps depends on how many zoom levels fall within the visible scale range.

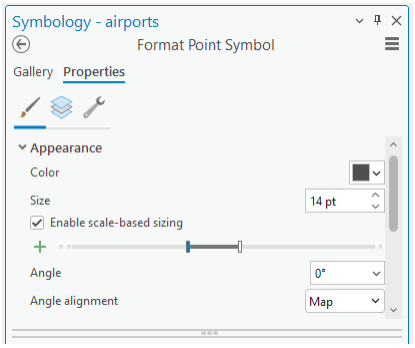

Figure 9: Airports. A simple example of the scale-based sizing active within the UI

Note: This can also be used for stacked symbols, e.g., roads with casing or centerlines, if styled as one line on top of another.

Symbol class scales

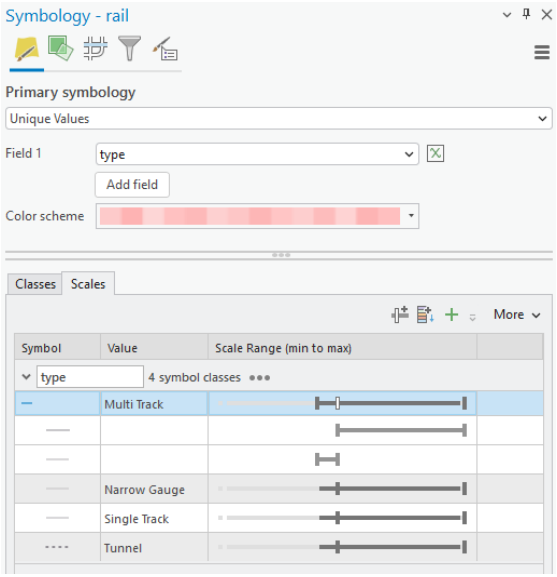

Figure 10: Railways. A simple example of scale-based symbology, but it is also very useful for features with many different types, some examples will follow.

Click on the Add alternate symbol button, then a new row will appear as here for Multi Track. Drag the slider handles to create alternate symbols at different scales.

Let's begin with an example of railway lines, where we want to have a particular railway type show with two different symbols depending on map scale. This may be as simple as two different line widths.

Rather than duplicate the Multi Track symbol class, we can use the Scales tab to add an alternate symbol, i.e., make the same symbol class have two or more different symbols depending on map scale. This method is more efficient.

To create symbol class scale ranges, right-click on a layer in the **Contents** pane and open the **Symbology** pane. Ensure the Primary symbology dropdown is set to **Unique Values** (or a grouped option such as Graduated Colors).

Ensure there are symbol classes for every symbol differentiated by a field before moving on to differentiate by scale. You can remove "All other values" if you are confident it is not needed.

Next, change from the Classes tab to the **Scales** tab. You'll notice that by default each symbol class has min and max scale ranges of none, which we treat as being shown at all scales. However, any scales beyond the visible scale range of the layer will show as a grayed out portion on the slider bar.

For each value (class), move the min scale and max scale handles to where you want them to first appear and first disappear.

Now, highlight a particular value and click **Add alternate symbol**.
Drag the new handle to the scale where you want the symbology to first change and then restyle the alternate symbol. Do this as many times as required.

Contours example

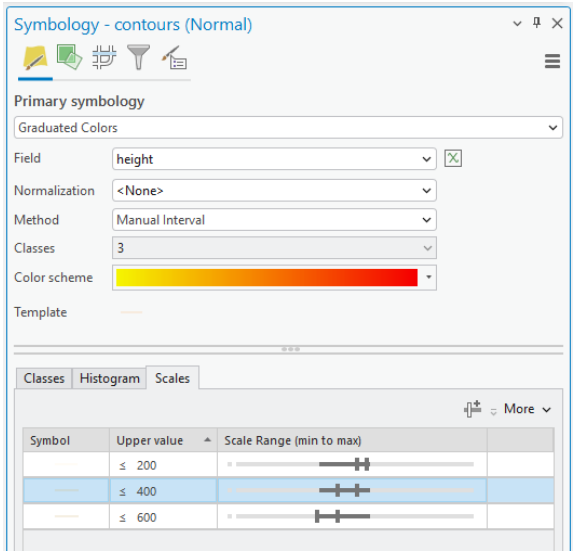

Figure 11: Creation of symbol class scale ranges

Say you want to style contours differently at three different height ranges, each constrained to its own visible scale range. We usually begin by ensuring there are symbol classes for every symbol differentiated by a field.

Then, simply go to the **Scales** tab and drag the sliders to set the visible ranges as required.

Road example

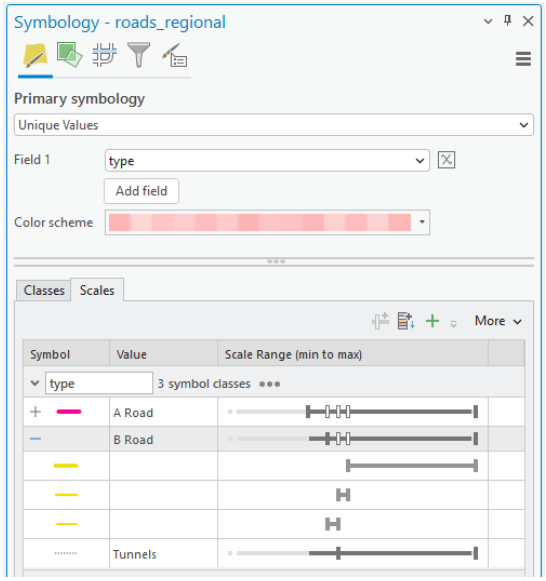

Figure 12: Symbol class scale ranges for roads. Notice how we have expanded the B Road type and added alternative symbols.

In a more advanced example of roads, we can use scale range to set the maximum and minimum visibility ranges of different symbol classes ***as well as*** splitting each type into alternative symbols based on map scale (ranges).

If you are comfortable with editing the style JSON of a vector tile package, then post processing the style in JSON is another option that is even more efficient—allowing the varying symbology to be set on a single layer.

Display Filters *(Contours example)*

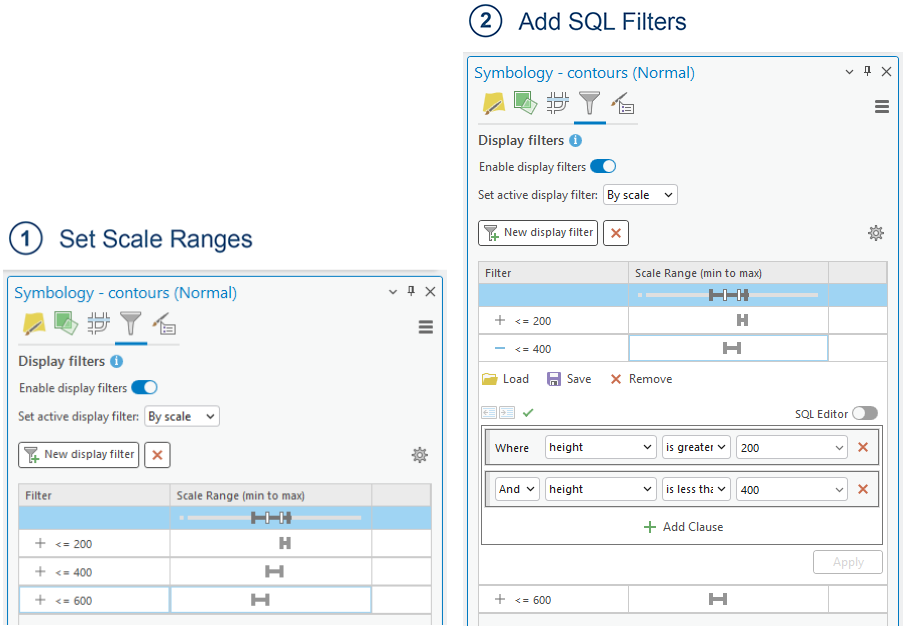

Figure 13: Contour visible scale ranges configured using Display Filters. Although the interface appears similar, this example is defined under the Display Filters tab rather than the Primary Symbology tab used in earlier scale examples.

Returning to the contours example. Say we wanted the different scale ranges for each height or elevation range, but all would share the same symbology. Then there is an even more efficient way: Creating manual display filters using SQL queries or the filter prompt UI.

First ensure the visibility scale range of the layer is how you want it. Right-click on the layer in the **Contents** pane, go to **Properties** > **General**, and ensure either *Show layer at all scales* or **Show layer between these scales only** is set correctly.

Next, in **Symbology** pane, choose your line symbol (**Single Symbol**), then change from the Primary Symbology tab to the **Display Filters** tab. Here, you may choose to move the outer handles to the ends of the visible range for tidiness but it's functionally indifferent to leaving them at the far left and right.

Then, click on **New display filter** to add as many sub ranges as needed. Move the new white handles to the required scales. Then, on each line of the table under **Filter**, click the **+** icon to expand the section and add a filter clause or SQL filter for each height range.

Note: Display filters will snap to the scale of the nearest (integer) vector tile LOD. A transition at say LOD1.8 or LOD2.2 will instead be cooked at LOD2.0 thus affecting "continuous" scales. Also, you may need to exit and revisit Symbology for the display filter UI to update scales, etc.

Labeling

Given the comparatively basic nature of the specification, not all ArcGIS Pro and Maplex options port across into vector tiles. However, there are quite a lot that do. Sticking to these will help keep consistency between what you see in Pro vectors and in vector tiles, whether viewed in ArcGIS Pro, Map Viewer, or elsewhere.

There are also some settings in Pro such as connect features (lines) and label largest part (polygons) which will happen in vector tiles by default regardless of the Maplex setting.

Label Priority

While Maplex will try to resolve overlapping or conflicting point labels, vector tiles will only show the one with highest priority, so it is important to set label priorities. To do this, right click on the map in the **Contents** pane, choose **Labeling > Priorities…**.

You will be presented with all label classes within the map and can reorder them so that the ones nearest the top are given a higher labeling priority. Labels within the same label class are considered equal so it will decide which ones are best to place or not show.

The range and subtlety of these behaviors make them less effectively conveyed through practical examples than through a simple table; accordingly, they are consolidated in *Appendix A*, where a structured presentation provides ease of reference.

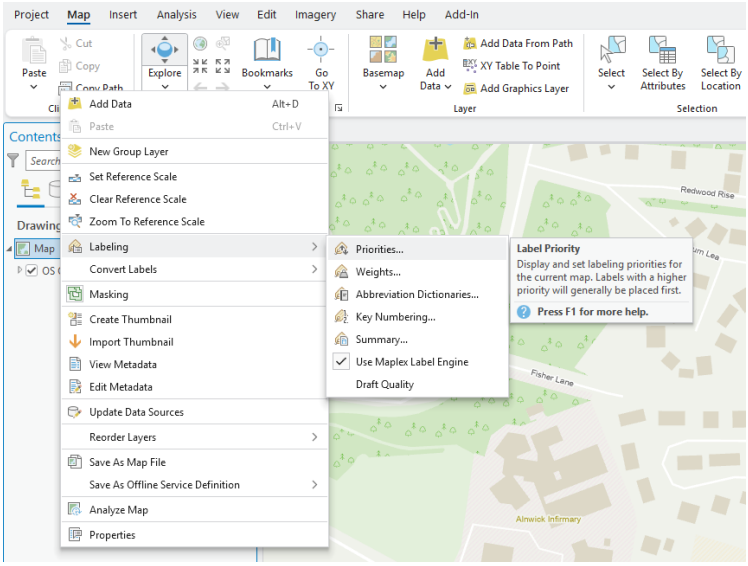

Figure 14: Navigating through ArcGIS Pro to the Label Priorities

See *Appendix A* for a handy table of what is and is not compatible with vector tiles.

See *Appendix B* for a list of additional notes for common questions.

Creating Vector Tiles

These steps for converting your styled map data into actual vector tiles are sometimes referred to as “cooking” or “baking”.

1 Set the coordinate system of your map

A lot of online maps use the WGS 1984 Web Mercator (auxiliary sphere) projection and coordinate system [3857], sometimes known as Pseudo Mercator. If the web platform you are using prefers this, then it’s likely the way to go. It also means your vector tiles can be used with Esri basemaps. Otherwise, we encourage you to use the best projection and coordinate system for your map and its geography, and/or the tiling scheme defined by your organizations basemaps if this vector tile layer will be paired or viewed with them.

To set the coordinate system, right-click on the Map title in the **Contents** pane, open (Map) **Properties**, and navigate to the **Coordinate Systems** tab on the side of the dialog window. Either directly search by name or code, or navigate the newly improved folder structure, e.g., in our example to choose Projected > Europe > Ireland and United Kingdom > British National Grid.

2 Set the scale range of your map

This step will significantly speed up processing on larger datasets. You can set the scale range for each data layer, and we would encourage this also, but a quick way to set it for the whole map is to select all your layers, right-click and choose **Group** to create a group layer from them. That way you can open the group layer **Properties** (again, right-click) and set the section **Show layer between these scales only**.

You can also speed things up by specifying a LOD range on cooking, see *step 4*.

Tile Cache and Tiling Scheme

The steps below will automatically generate a tiling scheme for your data. While you can use the Generate Tile Cache tool to create a custom or edited tiling scheme, or to create tiles to a custom pixel size, this is a raster tool and is generally not recommended for vector tiles. While it can be used to customize the map scales at each zoom level of your tiling scheme, this comes with a huge warning that the transition scales of the vector tile package will likely mismatch those of the input due to rounding and conversion errors, etc. Tile sizes different to 512px also cause a difference in LOD level numbering. So, we advise sticking to the default scheme assigned by ArcGIS Pro. If you really want to use your own, then be advised that each scale must be precisely double (or half) the previous scale.

3 Create Vector Tile Index (optional step)

This is a good step to run because it will give you a preview of the tiles and their density.

For many users the default maximum vertex count will suffice, but we occasionally increase this to reduce tile splitting and decrease it to help optimize performance.

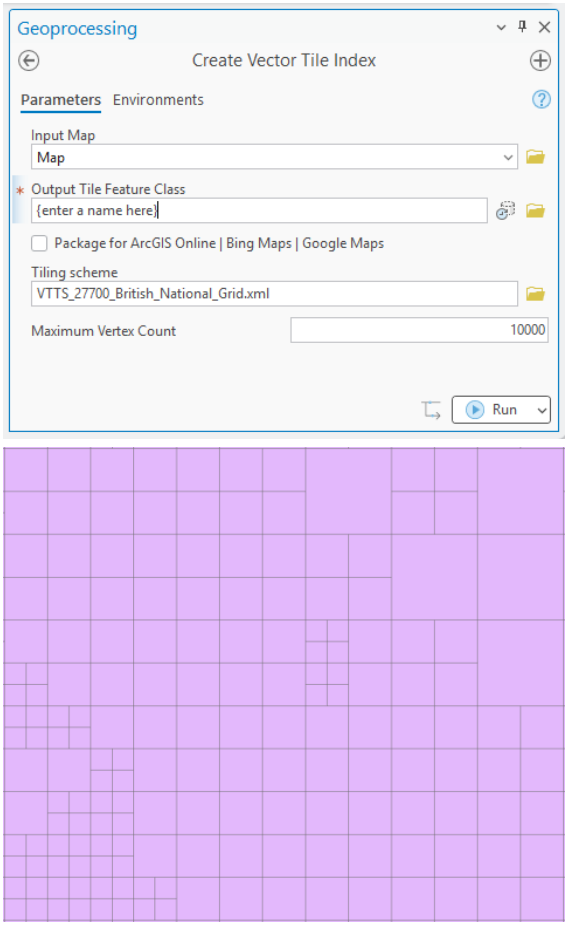

Figure 15: Vector tile index: the tool UI and a visual example of an index

Although you can skip this step if you are using Pseudo Mercator and plan to use our default tiling option for ArcGIS Online/Bing Maps/Google Maps, pre-creating the vector tile index is a more efficient workflow if you plan to run Create Vector Tile Package more than once on the same tiles. Create Vector Tile Package will create the index if you don't have one, but that tool discards it afterwards, whereas this step will give you a saved copy you can use with Create Vector Tile Package to speed up future runs of the tool.

Note: If using a projected coordinate system from Pro, then the correct tiling scheme will be prepopulated for you.

4 Create Vector Tile Package Vs. Share As Web Layer

The next step is vector tile creation (aka “cooking”). Using the **Create Vector Tile Package** tool, you can create a standalone vector tile package which is saved as a file that can be added to Pro, Enterprise, Online or potentially elsewhere. Alternatively, using **Share As Web Layer** you can create the tiles directly on an Enterprise or Online portal but with a few limitations discussed further below.

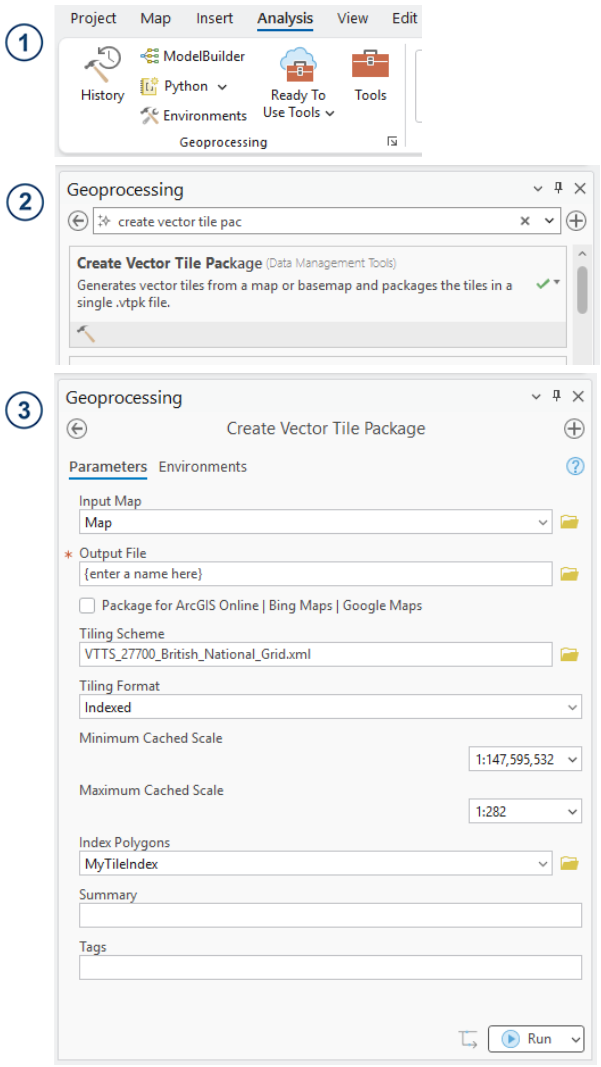

Figure 16: Create vector tile package, UI steps

To create vector tile package, go to the **Analysis** ribbon, select **Tools**, and find the appropriately named Geoprocessing tool.

This will create a vector tile package file (.vtpk) which is essentially a zipped folder that Esri software and applications recognize as being a set of vector tiles.

To then add these vector tiles to Enterprise or Online, navigate to **Content** in portal, click on **New Item**, and either drag your file to the drop zone or click on **Tile layer** and follow the prompts.

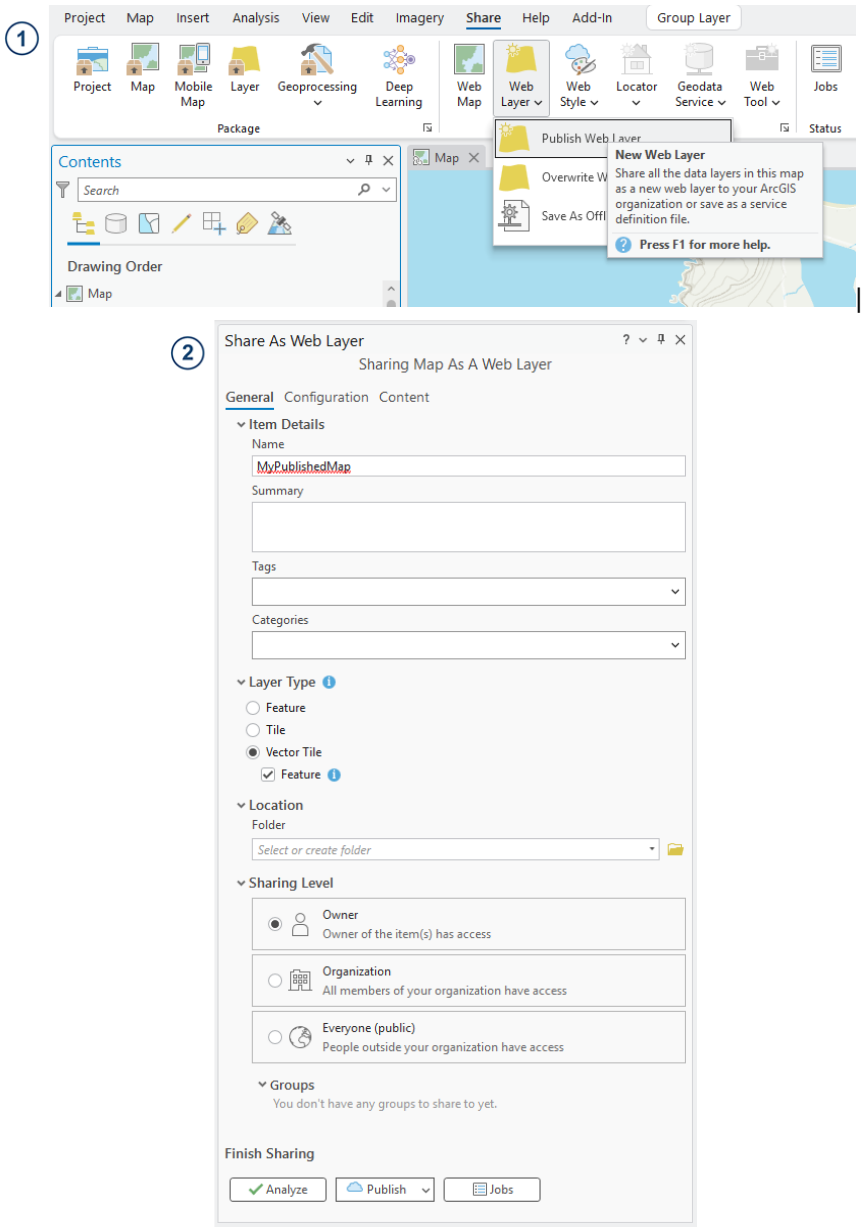

Figure 17: Share as web layer

To share directly to Enterprise or Online portal, go to the **Share** ribbon, select **Web Layer**, and **Publish Web Layer**. (Not to be confused with *Share as web map* which is a similar raster tile workflow).

Under **Layer Type**, choose **Vector Tile**.

Leave the Feature box checked if you want to be able to update or query the tiles, or if you need to retain any advanced symbology in Enterprise or Online. It essentially creates a second layer that contains feature information not available through the standard vector tile layer.

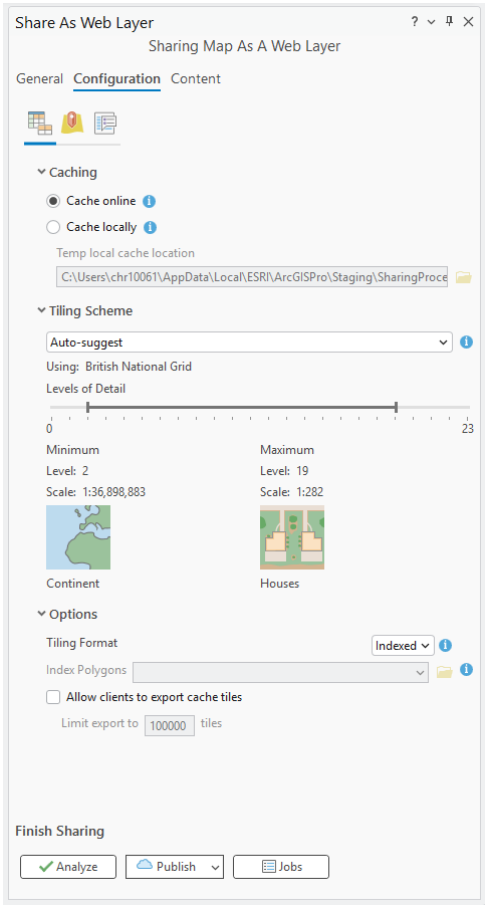

Figure 18: 'Sharing map as web layer' configuration

Under the **Configuration** tab, you have the option of caching the tiles during creation directly online or locally. There are also options to override the tiling scheme and LOD range. We suggest going here, if only to set the LODs to only what is needed.

Online is the default caching option, if you choose local then it basically creates the vector tile package above and imports that to online—if you choose local caching, this effectively creates a vector tile package. In this case, you may prefer to use Create Vector Tile Package directly to see and control the settings before adding it to portal manually.

Analyze is an optional step and way to check everything is ready for publishing. As well as commonly reminding us that a map needs a description, it may also highlight compatibility issues (see *Sharing as a web layer: Supported data types*).

After publishing, your new layers appear in your portal and can be used as hosted vector tiles in ArcGIS Pro and Map Viewer. Alternatively, they can be converted to .mbtiles format (using one of the many third-party tools available on GitHub or via Python packages), or extracted and served as PBF tiles using the Extract Package tool in Pro, for consumption by third-party applications.

We don't intend to cover third parties here, but we appreciate some customers require such routes, so know that you should be able to get our vector tiles into the likes of MapLibre, Mapbox, CARTO, Felt, Leaflet, QGIS, Deck.gl, and even Cesium. Usually this involves obtaining the REST URL for the hosted vector tile service, using an API key and any required third-party plugins.

Sharing as a web layer: Supported data types

However, not all data types are supported for sharing as web layers. Such data will be excluded from the output unless converted to/resupplied as a supported database feature class. One such example we see from major geodata providers is GeoPackage.

There are tools in ArcGIS Pro that can convert them. If you go this route, use either the **Create SQLite Database** tool to keep the data as close as possible to source or use the **Copy Features** tool to output the GeoPackage to an Esri file geodatabase.

However, we recommend that if source data is GeoPackage then keep it as that. You can run **Create Vector Tile Package** directly on GeoPackage data, save vector tiles locally, and then upload these to your portal afterwards.

Editing Vector Tiles

Vector tile content is cooked into read-only caches. To edit them you must edit the source data and update or rebuild the tile layer. There are three ways to do this.

1. For those looking to update or revise the input data to an existing, standalone, vector tile set: return to the source map in ArcGIS Pro, make any edits in Pro, and re-cook the vector tiles using either *Create Vector Tile Cache* or, from the **Share** tab, choose **Web Layer** > **Overwrite existing Web Layer**. To overwrite a local vtpk, just use the same name in the *Create Vector Tile Package* tool.

   Warning: Overwrite may break web map connections to tile layers.

2. For a tile layer published with an associated feature layer, you can edit the feature layer in ArcGIS Pro or Map Viewer, then go to the vector tile layer's item page, and choose **Rebuild Cache**.

3. You can also replace the hosted vector tile layer. This maintains the portal item ID's and REST endpoint, also allowing archiving. Navigate to the item page of the published layer in ArcGIS Online or Enterprise portal and choose **Replace Layer**.

However, as outlined by Matheson (2019), a major advantage of vector tiles is how easily you can modify their style to better suit your own needs. Because the store of data is separate from the styling information, we can modify a vector tile style without touching the vector tile data (Tiwari & Fauvell, 2019).

Restyling vector tiles from Online/Enterprise

Accessible either directly from the (vector) tile layer's ellipses (three-dot) menu in ArcGIS Online or Enterprise, or via the **Vector Tile Style Editor** app, this capability gives users complete control to restyle Esri basemaps and any vector tile layers published within their organization.

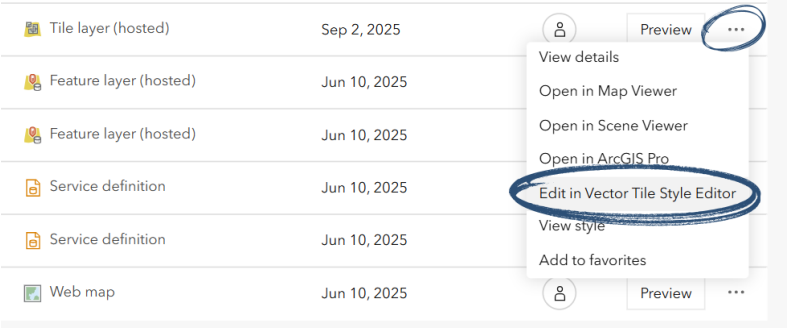

Figure 19: Straight to editing a specific vector tile set from Portal

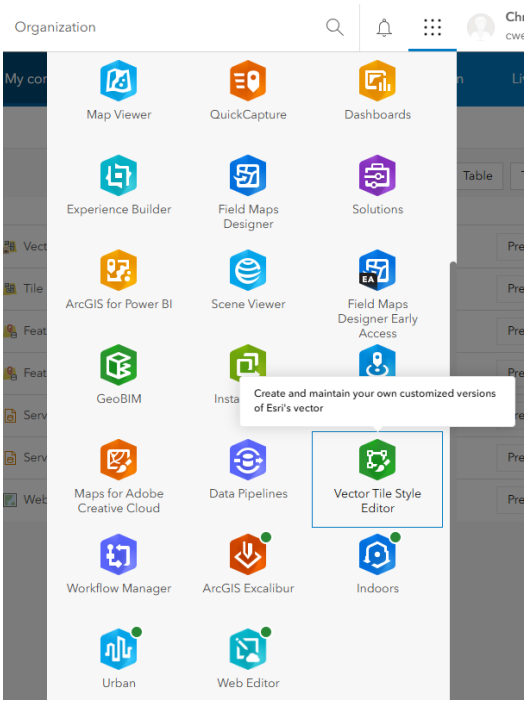

Figure 20: Opening the Vector Tile Style Editor app from Portal

Just like other Esri apps, you can also store the URL and sign-in from the app's welcome page. The URL for the style editor is www.arcgis.com/apps/vtseditor.

Once in the editor, users may choose to update the color scheme using high-level category groupings or fine-tune feature-level properties such as symbol colors, sizes, and line weights. Alternatively, they can bypass the graphical UI and edit the style's JSON directly for granular or programmatic control.

Restyling from the vector tile package

To make custom vector tile style changes from Pro, the usual workflow would be to publish your vector tiles to Online or Enterprise and then follow the steps above. However, it is also possible to create a vector tile package and edit the output without opening it in either Online or Enterprise. There are two ways to do this:

1. Copy and rename the vector tile package to a .zip extension and unzip the contents.

2. Within ArcGIS Pro, run the **Extract package** tool.

Then open **p12/resources/styles/root.json** with a suitable script editor, make any changes and repackage the folder.

In addition to editing style values, manually editing the JSON also allows you to do some more bespoke exercises such as filtering without editing the data files; combining layers from different vector tile services; and adding non-standard functions used by third-party vendors for further symbol control and refinement.

Style support

We support the styling options defined in the open standard originally published by Mapbox, as documented in the Mapbox Style Specification v8.0. While the version identifier in the JSON remains at v8, the specification itself has continued to evolve over time, with subsequent updates reflected in the official release notes and package repositories. As a result of this ongoing evolution—and of extensions introduced by Mapbox, MapLibre, and other contributors—there is no longer a single, fully canonical set of style definitions. In practice, our implementation targets full support for the original Mapbox v8.0 specification, while also incorporating additional capabilities that reflect more than a decade of continued development. We plan to document and share details of this expanded feature support in future publications.

Understanding cooking, caching, leaf tiles, zoom levels and LODs

Cooking, or baking, is simply a term used to describe the publishing of an area of the map to vector tiles. A particular feature can be described as being cooked into the tiles at a given set of levels of detail (LODs). In practice, the cooked tiles are protocol buffer (PBF) files, a standard and highly efficient way to package such data. They are accompanied by fonts stored in PBF as glyphs, sprites stored as PNG images, and map style instructions stored as a JSON file.

Caching is the process of then storing these files so they can be loaded quickly into a map.

Cooking and caching are commonly treated as co-dependent concepts when discussing the tile indexing described earlier and how the map from Pro is effectively "cut up" or subdivided into tiles at different LODs.

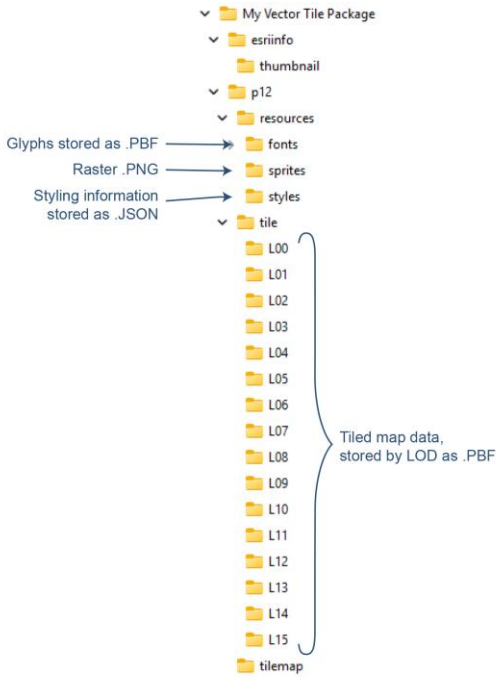

Figure 21: The folder structure within a vector tile package

A *"LEAF" tile* is created when there is no longer a need to subdivide the tile—for example, if there is no additional map detail or change in display as you zoom to larger scales. If the user zooms in further, then the same tile is shown at the new scale rather than being replaced with smaller, more detailed tiles. This concept, known as "overzooming", helps to prevent the total size of the package becoming unnecessarily large.

You generally don't need to worry about these directly, as Pro automatically calculates them for you. However, if you run *Create Vector Tile Index* before creating the vector tile package, then you can see from the resulting tile grids those tiles with a LEAF attribute of 1, indicating which tiles will be "leaf tiles", and at what LOD they occur. Manual modifications to the LEAF field can be made in the attribute table.

You shouldn't normally notice leaf tiles—they primarily improve efficiency behind the scenes. However, if you do observe an unexpected change at a particular LOD, leaf tiles are a useful first place to investigate. Remember that the LEAF value switches to 1 at the LOD where leaf behavior begins. An area that is not a leaf (i.e., LEAF = 0) at, say, LOD 7 may still become a leaf (LEAF = 1) at LOD 8 or higher.

A *zoom level* is a number or numeric value that corresponds to an index of scales used in a map viewer. Strictly speaking, it is a property of the vector tile layer rather than the viewer scale itself, but the term is commonly used to describe how "zoomed in or out" a web map is.

In contrast, a *level of detail (LOD)* is a defined scale within a tiling scheme that manages the amount of detail shown. This term is typically used when describing the generalization and content of a web map across scale ranges.
In many cases—such as in ArcGIS Online—these values align and are functionally equivalent, so the terms are often used interchangeably. In ArcGIS Online, levels range from 0 (most zoomed out, least detail) to 23 (most zoomed in, most detail). This preset is also the default in ArcGIS Enterprise.

The vector tiles will only include the LODs requested by the *Create Vector Tile Package* tool or configured when using *Share As Web Layer*. We typically make use of leaf tiles above LOD16 as part of the tiling implementation, unless feature complexity extends beyond that scale.

Are vector tiles just for basemaps?

Many use cases are background or reference mapping, but you can convert any kind of map whether it be thematic or analytical into vector tiles.

The two main things to remember are that (1) the data will become tiled, which is highly efficient so long as you consider there will be tile edges, and (2) a vector tile package is an optimized rendering of the data. Meaning that direct analytical operations on vector tiles are not possible, but they do allow for the examination of data attribution from published or packaged vector tiles via pop-ups within ArcGIS Pro.

Pop-ups

For pop-ups to propagate from authoring to viewing in Pro, the pop-up fields must be marked to Highlight in Pro before cooking. To do so, right-click a layer in the **Contents** pane, select **Data Design** > **Fields** (or navigate there from the Feature Layer tab).

Check the required fields for the popup under the **Highlight** field of the table and then **Save**.

Once published and loaded into Pro, right-click on the vector tile layer in the **Contents** pane and choose **Enable Popups**.

Note: This won't work if the corresponding table is read only.

Cloud stores

For some thematic maps you may encounter very large file sizes. If so, then there is a workflow to serve these from a cloud store. You cannot simply copy an existing vector tile package because the format for Enterprise to read from the cloud needs to be .vtiles rather than .vtpk.

1. Use the **Extract package** tool in ArcGIS Pro to convert the *vtpk* to *vtiles* and move to your cloud.

2. Register your cloud store in ArcGIS Enterprise. There are various ways to do this but one is to login to **ArcGIS Server Manager**, navigate to **Data Stores** tab, click **Register Data Store** and go from there.
3. Publish the web layer directly from the data store item in the portal. The resulting tile layer will dynamically reference the cache content stored in your cloud environment.

Key Takeaways

- Design for the target platform(s) and only use symbology settings end users will see
- Stick to set scales where possible: easier to set up and avoids data unnecessarily spilling over into other LODs
- Let ArcGIS Pro recommend scales and calculate tile indexing
- Use symbol-based scale ranges rather than duplicating content

By authoring maps with a more complete understanding of how individual settings influence outcomes—and of how vector tile cooking, caching, and LODs interact— we are better positioned to produce vector tiles that behave as intended and perform effectively across platforms.

Links to Esri help topics

Author a map for vector tile creation

Author a multiscale map

Caching Maps and Vector Tile Layers: Best Practices

Edit vector tiles in style from Map Viewer

Introduction to sharing web layers

Introduction to vector tile services

Other ArcGIS Blog posts on vector tiles

Acknowledgement

Many thanks to Tommy Fauvell and Dan Stephen for their help in refining and shaping this document.

Appendix A: Symbology and labeling types and options in ArcGIS Pro and their compatibility with cooking or styling vector tiles

SYMBOLOGY	Compatible with vector tiles (as of ArcGIS Pro 3.7)	Notes
Single symbol	✓	
Unique values	✓	
Graduated colors	✓	
Bivariate colors	✓	
Unclassed colors	✓	
Graduated symbols	✓	
Proportional symbols	×	
Charts	×	
Heat map	×	
Dictionary	×	
Symbology classes & scales	✓	
Properties		
Color, size, width & outline	✓	
Scale-based sizing	✓	
Transparency	✓	
Rotation	✓	
Caps and joins	✓	
Overprint	×	
Effects	◆	Offset, Dash, and Move work; All others do not. Note effects may cause missing features when close to tile edges
Vary by attribute		
Transparency	✓	
Rotation	✓	
Color	×	Only for basic circle markers
Size	✓	
Other options		
Symbol layer drawing	✓	
Display filters	✓	
Other select scenarios		
Hatched picture fill with tinting	✓	May rescale depending on tile resolution
Thin lines	×	Depending on line type and opacity they either draw but are thickened to 1px (0.75pt); or they retain size and may not be visible in the resulting vector tiles due to pixel rendering
Unique value polygon outlines	✓	
LABELING		
Label Class: Expression		
Text case specified in label expression	✓	
Label formatting rules, e.g. line breaks, concatenation	✓	
Text formatting tags	×	E.g. overriding font color, style and size
Label Symbol: General		
Font name, style and size	✓	
Scale-based sizing	✓	
Color	✓	
Outline	×	
Transparency	✓	
Underline	×	
Strikethrough	×	
Text case	✓	Small caps does not work, other cases do
Position adjustment (subscript/superscript)	×	
Position offset	✓	
Rotation	✓	
Halo	✓	
Shadow	×	
Point callouts (e.g. for highway shields)	✓	Defaults to centrally anchored text and icon
Other callout types	×	

Label Symbol: Formatting		
Word spacing and letter spacing	✓	
Letter width	×	
Line spacing	×	
Internationalization	◆	We suggest you do not change the text direction from default
Label Position		
Placement: Point: Best position	◆	All labels will be placed in the preferred zone 1 position
Placement: Point: Specified	✓	
Placement: Line: Offset Straight	✓	Works but currently the greater the offset the more label characters overlap or disappear
Placement: Line: Centered Straight	✓	Vector tiles will join lines before labeling, whereas Maplex will label each part. In this regard Street or Contour placement is a closer match to vector tiles than Regular
Placement: Line: Centered Perpendicular	×	Gets placed as if curved along the line
Placement: Line: Offset Perpendicular	×	Offset works but gets placed as if curved along the line
Placement: Line: Constrain offset	✓	Above/below work
Placement: Measure offset from	◆	Works for lines not points
Placement: Polygon: Horizontal in Polygon	✓	
Placement: Polygon: Horizontal around Polygon	×	Defaults to horizontal within polygon
Placement: Polygon: Straight in Polygon	✓	Produces very different results in vector tiles to Maplex partly due to known cooking issues. Avoid it if you can.
Placement: Polygon: Curved in Polygon	✓	Produces very different results in vector tiles to Maplex (Regular), and is more akin to Maplex (River placement), partly due to known cooking issues. Avoid it if you can.
Placement: Polygon: Curved around Polygon	×	Defaults to horizontal within polygon
Placement: Polygon: Boundary Placement	✓	
Placement: Polygon: Avoid holes	×	Vector tiles ignores this setting. It tends to do this anyway, not explicitly but as a consequence of placing labels at the polygon centroid (eroded center).
Placement: Try horizontal position first	×	
Placement: May shift label upon fixed position	×	
Placement: Polygon: May place label outside polygon...	×	
Placement: Polygon: Place label at fixed position…	×	
Orientation: Align to graticules	×	
Rotation: Straight	✓	
Rotation: Perpendicular or Horizontal	×	
Rotation: Keep label upright	×	
Position: Fitting Strategy		
Stack label	◆	Works only with forced split. Line and character limits ignored
Prefer to stack long labels	×	
Reduce size	×	
Horizontal alignment	✓	
Preferred offset	✓	
Connect features	✓	Connect features, and Label largest feature part happen regardless of ON/OFF
Repeat minimum interval (lines and curved in polygon)	✓	May help in some cases
Never remove unplaced labels	✓	
Label styles from expression	×	Honors new line and concatenation but not font styling (size, color, etc.)
Text case specified in UI	✓	
Abbreviate	×	
Position: Conflict Resolution		
Remove duplicate labels	◆	Only for horizontal label placement
Remove ambiguous labels	×	
Minimum feature size	×	Although can be done as a label expression
Buffer	×	Vector tiles does its own label buffering
Feature weight	×	
Background labels	×	
Unplaced labels: Never remove	✓	
Labeling (Contents)		
Label priorities	✓	Works and seemingly stronger in vector tiles which ignores weights
Abbreviation dictionaries	✓	
Key numbering	×	

Appendix B: Additional notes for common questions

1. Very thin lines in Pro might not be visible in the resulting vector tiles due to pixel rendering. While we cannot offer a minimum width that works, as it depends on the feature's alignment to the pixel grid, as a rule of thumb we recommend keeping lines above 0.5 pt and if they need to appear fainter then do so by color. Be aware that if you use transparency the pixel rendering may further reduce the appearance of the line thickness, the more transparent the more of the line you're likely to use. This may be done deliberately depending on the desired visual outcome.
2. Minimum interval spacing only works for lines and is not honored across tile edges. For points or polygons, you can edit the vector tile style in the online vector tile style editor.
3. It is advisable to generalize complex geometries with lots of vertices before labeling especially when using offset or viewing at smaller scales.
4. Use contour placement for contour labels as results tend to be more reliable and similar to Maplex's regular placement.
5. Line label types such as river placement are processed as lines, and labeling will occur after initial cooking and thus update at each LOD based on the situation in terms of the shape of the line and conflicts. Using horizontal line placement (which processes as points) encourages a greater number of line labels to be retained. Alternatively, to retain specific positioning and control over labels, convert lines to points for labeling. For example, "Never remove" will force such labels to display.
6. Expressions for label line spacing are not honored by vector tiles.
7. Labeling of each part of multipart features is not supported in vector tiles.
8. Callout box margins are not ported across into vector tiles.
9. Remember that feature label weights are specific to Maplex and not supported in vector tiles. Use label priority and label buffer to limit conflict.
10. Note that vector tiles are in the RGB color space as per their specification. So be advised that any other color models such as CMYK will convert to RGB on cooking which might affect their appearance. Colors can be stored as JSON strings in a variety of permitted formats.
11. Multiple fonts in a single label class will cook to vector tiles but is not currently supported when read back in ArcGIS Pro.
12. The text direction setting in ArcGIS Pro is not compatible with vector tiles, and mixed script characters may be handled differently in ArcGIS Pro versus Map Viewer.
13. Any layer-based transparency is transferred to the features when cooked into vector tiles, as this behavior is beneficial in more cases than not.

About the Author

Chris Wesson is a Principal Product Engineer at Esri on ArcGIS Pro, where he shapes multiscale mapping capabilities and advanced cartographic workflows. With more than 20 years in the GIS and mapping industry, he brings a sustained passion for turning complex location data into clear, insightful, and impactful visual narratives that scale from analysis to communication.

Pertinent to this paper, Chris has worked with many tiled mapping solutions both raster and vector over his career as well as on projects with map tile vendors and European-wide collaborations.

His academic background and broader intellectual interests span both applied social and physical sciences. Chris holds a Master of Social Science in Accounting & Management Science and a Bachelor of Science in Oceanography with Physics from the University of Southampton, UK—an interdisciplinary foundation that informs his systems-level perspective on geographic and data visualization.

esri | THE SCIENCE OF WHERE®

Esri, the global market leader in geographic information system (GIS) software, location intelligence, and mapping, helps customers unlock the full potential of data to improve operational and business results.

Founded in 1969 in Redlands, California, USA, Esri software is deployed in more than 350,000 organizations globally and in over 200,000 institutions in the Americas, Asia and the Pacific, Europe, Africa, and the Middle East. Esri has partners and local distributors in over 100 countries on six continents, including Fortune 500 companies, government agencies, nonprofits, and universities. With its pioneering commitment to geospatial information technology, Esri engineers the most innovative solutions for digital transformation, the Internet of Things (IoT), and advanced analytics.

Visit us at esri.com.

Contact Esri

380 New York Street
Redlands, California 92373-8100 USA

1 800 447 9778
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations

For more information, visit **esri.com/URL**.