Styling Vector Basemaps

WESLEY JONES, ANDY SKINNER
What is a Basemap?

A map that is designed to provide a *foundation* for one or more online maps.
What are Vector Tile Basemaps?

- Basemaps are cached and delivered as vector tiles (PBF format)
- Rendered client-side
- Based on resource files delivered with vector tiles
  - Fonts
  - Info (font list)
  - Sprites (patterns & symbols)
  - Styles (root.json)
Esri Vector Tile Basemaps

• The maps and data are prepared in ArcGIS Pro
• Can be viewed in ArcGIS Pro, but not ArcMap
• They display in most current web browsers
Esri Vector Tile Basemaps

... are available in some desktop and mobile apps, and we are increasing the number of those all the time.

Web
- ArcGIS Online
- JavaScript API Map Viewer
- Scene Viewer
- Web App Viewer
- Story Maps

Runtime/Quartz Mobile Applications
- Explorer
- Collector
Esri Vector Tile Basemaps

- We provide a core set of pre-prepared map styles
- All built from one vector tile service
- Some are like existing Esri basemaps
- Others are new.
Customizing our Basemaps

ANDY SKINNER
The Reference Document

... is available as a pdf via the ‘Details’ page for your source basemap.

Any of the content listed here can be added to your map.

Supporting Information for Esri Vector Basemaps (Production v1.3)

This document is updated to account for the changes applied to our production vs. vector file maps release. There are four sections of information. Refer to this *vector file maps* release for more information on the latest Esri vector basemaps and the ability to customize them for your own web maps and apps.

Esri Vector Basemaps Feature Names and Label Subtypes

This provides a list of the feature IDs, subtypes and label information found in the root .JSON file. Knowing what the feature names are from this list is beneficial when editing the .JSON code to style and customize your vector basemap.

Esri Vector Basemaps Min/Max Zoom Levels | Symbols & Labels

The list below shows the current full list of feature names ("ID") and their corresponding min and max zoom levels. You can control when features appear on the map by adjusting the "minzoom" and "maxzoom" values in the root .JSON file.

Esri Vector Basemaps Disputed Boundaries and IDs

See the link for an example of how to change the .JSON code and apply the .DISPUTED code in the filter for boundary lines.

Esri Vector Basemaps Resources/Fonts

See the link for an example of how to change the .JSON code and apply different fonts to your maps, following the CSS file to achieve the desired style.
Preparing the basemap

- Choose your basemap
- Is the sprite symbolization going to work with your final result?
Preparing the basemap

- Make a copy of the basemap layer, and save to your account
Preparing the basemap

- Make a copy of the basemap layer, and save to your account
- Download the style, and open it in an editor
Preparing the basemap

- Keep the layer open in a webmap
- Edit the json
- Update it on the ‘Details’ page
- Refresh the map
Editing Apps

http://esri.github.io/arcgis-vectortile-style-editor/#
Simple json editor
Editing Apps

http://esri.github.io/arcgis-vectortile-style-editor/#

Make changes to the relevant code, apply them, and if they look OK, save them
GUI-based editor
Editing Apps

https://maps.esri.com/jg/VectorBasemapStyleEditor/index.html

Style controls can be accessed through the list on the right
Changing content

Deleting features:

```
1  
2  "layout": {},
3  "paint": {
4       "fill-color": "#1f1f1f",
5       "source": "test",
6   },
7   "id": "background"
8 },
9 },
10 "layout": {},
11 "paint": {
12       "fill-color": "#ededed",
13       "source": "test",
14   },
15   "id": "land"
16 },
17 "layout": {
18   "visibility": "hidden"
19 },
20 "paint": {
21       "fill-color": "#ededed",
22       "source": "test",
23   },
24   "id": "land"
25 },
26 "layout": {
27   "visibility": "hidden"
28 },
29 "paint": {
30       "fill-color": "#ededed",
31       "source": "test",
32   },
33   "id": "land"
34 }
```

Adding features:

- Copy/paste a block of code into the json from one of our other basemaps …
- or find the relevant layer in our reference document and construct a new block of code around it
Changing color
Code-based app

- Find the “paint” section in the relevant code
- Adjust the ‘hex’ value, and save
Changing color
Code-based app

- Find the “paint” section in the relevant code
- Adjust the ‘hex’ value, and save
Changing color
GUI-based app

- Click on the relevant color swatch

- Input as HEX, RGB or HSV
Changing stroke value
Code-based app

- Find the “line width’ section in the relevant code

- Adjust the relevant stroke value, or values
Changing stroke value

GUI-based app

- Click on the layer name to reveal the json code

- Adjust values
Changing fonts
Both apps

• Find the “line width’ section in the relevant code

• Adjust the relevant stroke value, or values
Available Fonts for use with Esri Vector Basemaps

Arial

Avenir Next

Belleza

Ubuntu

Ubuntu Condensed

Ubuntu Mono

Redressed

Oswald

Cabin Sketch

Palatino Linotype

Special Elite

Syncopate

Josefin Slab

Playfair Display

Playfair Display SC
Styling in Pro

Vector Tile Packages
Vector Tile Package Customizations

- Vector Tile Package
  - Fonts
  - Sprites
  - Your own data
  - Pro / json

- Children’s Map example

- Demo
  - Stylize
  - Examine Resources
  - Push to Online
Extras
Things we didn’t cover but are still really interesting

- Languages
- Disputed Boundaries
- Rotation
- Projections
- Raster components
Summing Up

ANDY SKINNER
Power!

Use it Wisely …
Power!

Use it Wisely …
Power!

Use it Wisely …
Links

ArcGIS.com ‘Vector Basemaps’
All of our core set of Vector Tile Basemaps

ArcGIS.com ‘Sample Vector Tile Layers’
Our experiments with using some of the techniques discussed here

JSON Map Editor
The simple vector tile editor with JSON code alongside a preview map

‘GUI’-style Map Editor
The vector tile editor that itemizes by layer

JSON Lint
A useful (and free) online tool for reformatting JSON files and validating the final result