Bringing Your Data to Life in the ArcGIS API for JavaScript: Vector Tiles

Craig Williams and Jeremy Bartley
Overview

- Why vector tiles?
- Vector tiles in ArcGIS
- ArcGIS vector tile basemaps
- Consuming and styling vector tiles
- Overview of creating vector tiles
- Authoring a map for vector tiles
- Vector tiles in the JavaScript APIs
- Common questions
Web and mobile mapping over the last 10+ years

- Typically vector content (points, lines, polygons)
- Displayed on top of basemaps
- Since ~2005, basemaps have usually been raster tiles

- Dynamic updates of the map consist of two things:
  - Updating overlay content as drawn in client
  - Changing the basemap to a new tile set

- Paradigm is changing
Why vector tiles?
Graphics processing units

• **GPUs have changed the landscape**
  - On your devices (OpenGL ES2+)
  - In your browser (WebGL+)
  - On your desktop (DirectX, OpenGL)
  - Even in virtualized systems (vGPU)

• **Vector data can remain vector, draw at native resolution**

• **Raster data still best served as raster in most circumstances**
Advantages of vector tiles

- **Display quality**
  - Best possible resolution for Retina displays
  - Small efficient format
- **Dynamic labeling**
  - Clearer, more readable text
  - On the fly labeling for heads up display
- **Map Styling**
  - Streets, Topo, Canvas from one tileset
  - Day and Night mode
  - Restyling
Vector tiles in ArcGIS

• Tiles produced in ArcGIS Pro
  - Use the Mapbox vector tile spec
    - Which uses Google protocol buffers
    - Styling converted to Mapbox gl style spec

• More aggressive overzoom
  - Builds on generalization work done in past ArcGIS releases
  - Support for traditional tiling also exists
Vector tiles in ArcGIS (con’t)

• Vector tiles can be produced in any coordinate system
  - All Esri clients support drawing tiles from any coordinate system

• Recent work
  - More support for data driven workflows
    - ArcGIS Pro 2.1 and JavaScript (4.6/3.23) and Runtime 100.2 releases
  - Support for vector tiles in the print service (10.6)

• Future
  - Creation of tiles from feature services
  - Popups (Identify)
  - Circle layer
Vector tile format

- Vector tiles are stored using protocol buffers
  - Compact binary format for transferring data
  - Data is organized into layers of geometry with key/value pairs of attributes

- A style file defines
  - The layer order
  - Filter for each symbol layer
  - Symbol information for each symbol layer
  - Pro maps are converted to this model
    - Is a downgrade in some cases
Tile creation process: Esri basemaps

• Entire world
  - ~ 10hrs on a desktop machine
  - Tiles ~ 25 GB
  - Multiple styles can use the same tiles

• Compared to raster for the entire world
  - ~ many weeks on a server cluster per map style
  - Tiles ~ 20 TB
Vector tile basemaps

- Available on ArcGIS.com since November 2015
- Street (with and w/o relief), Topo, Night, Navigation, Dark Canvas, Light Canvas, Hybrid
Case study:

ARCGIS BASEMAPS
ArcGIS vector tiles - consumption

- **Tile consumption**
  - ArcGIS JavaScript 3.15+ and 4.x APIs
    - 3.18+ use a new implementation
    - Need a WebGL capable browser
  - ArcGIS Runtime 100.0+
    - OpenGL ES2 and DirectX (depends on platform)
  - ArcGIS Pro 1.3+
    - Shares ArcGIS Runtime implementation
Using and styling vector tiles
Using vector tiles in your applications

- **Multiple ways to use vector tiles:**
  - A) Use Esri provided vector tiles / styles
  - B) Style Esri vector tiles for your own use
    - Change colors
    - Drop layers
    - Match the needs of your application
  - C) Create your own vector tiles from your own data
Styling vector tiles

• Simple Style Copy
  - Save tile layer to your Portal or Online account

• Hand editing JSON
  - Update map item

• Two additional sample Vector Styling Apps simplify this:
  - Vector Style JSON Editor - GitHub
  - Vector Basemap Style Editor - GitHub
Demo
STYLING VECTOR TILES
Using vector tiles in your applications

• Multiple ways to use vector tiles:
  
  - A) Use Esri provided vector tiles / styles
  
  - B) Style Esri vector tiles for your own use
    - Change colors
    - Drop layers
    - Match the needs of your application
  
  - C) Create your own vector tiles from your own data
Creating vector tiles
Creating vector tiles

• Tile creation in ArcGIS Pro 1.2+
• Publish tile layers in ArcGIS Online and ArcGIS Server/Portal 10.4+

• At ArcGIS Pro 1.4+
  - Share as Web Layer experience supports vector tiles
  - Create, upload, and publish in one step
  - Previously, this was just available via Geoprocessing tools

• Accessory tools:
  - Share Package updated to support vector tiles
  - Extract Package updated to support vector tiles
ArcGIS vector tiles service

http://<catalog-url>/<folder>/<serviceName>/VectorTileServer

|--root.json
|--tilemap/
|--tile
|--resources
  |--fonts/
  |--styles/
  |--sprites/
  |--info/

Service example
Authoring vector tiles
Authoring maps

• Only feature layers with simple, unique value, graduated, unclassed or class breaks symbology supported

• Maps should be re-authored for vector tiles
  - Limit number of layers
  - Limit duplication of content

• Several improvements have been made in ArcGIS Pro to assist with this
  - Scale dependent capabilities added to symbology
  - Alternate symbols added to symbology
  - Scale based sizing added to symbology
  - Display filtering based on attributes
  - Improvement to scale logic
Scale dependent symbology

- Each symbol class can be assigned a scale range
  - Unique value
  - Class breaks

- Allows a multiscale map to be authored without duplicating content

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Scale Range (min to max)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7; 18; 12; 20; 22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9; 14; 15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16; 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11; 17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Image of a software interface showing scale range adjustments for different symbols]
Alternate symbols for symbology

- Symbol classes can switch symbols at scales
  - Unique value

- Allows you to change the appearance of a symbol without duplicating layer
Scale based symbol sizing

• Each symbol can have scale based sizing configured
  - Single symbol
  - Unique value
  - Class breaks

• Allows for small changes to symbol size across scales
Display filters

- Filter features via attributes other than what you’re symbolizing on
- Allows for scale dependent feature filtering
  - Lighten payload in your tiles
Scale logic changes

• **ArcMap and ArcGIS Pro 1.1**
  - Layers will draw AT and BETWEEN minimum and maximum scales

• **ArcGIS Pro 1.2**, layers don’t draw at max scale by default
  - Check “Draw up to and including the maximum scale in scale ranges” to revert to old behavior
    - This is checked for old Pro projects or imported ArcMap maps
Demo

AUTHORING AND CREATING VECTOR TILES IN ARCGIS PRO
Vector tile in APIs
Vector tiles in the JavaScript API

- JavaScript API samples in SDK
- Developers can use Tile Layers published to Portal in custom apps
Vector tiles in the JavaScript API

• Available in both 3.x and 4.x
• Works with all published Vector Tiles
• Tiles drawn in WebGL
• IE 11+
Common questions
Q: Why would I need to create raster tiles anymore?

A: Consider consuming clients and map requirements before committing to vector tiles. At this time, it’s not an answer for everything. Vector tiles will never be a solution for most raster datasets.
Q: Can my data be extracted from vector tiles?

A: Think of vector tiles as generalized graphic derivations of your data. In many cases features are cut at tile boundaries, overlapped at tile boundaries, or are dissolved for optimal draw. Only a minimum number of attributes needed for feature draw are stored. It’s not raw data.
Q: Can I show popups for vector tiles?

A: Not at this time, we have this on the roadmap for our client implementations of vector tiles (JavaScript API, ArcGIS Runtime, ArcGIS Pro)
Q: Can I create vector tiles for any map projection?

A: Only Web Mercator (Auxiliary Sphere) supported for the initial releases (ArcGIS Pro 1.2 and 1.3). From ArcGIS Pro 1.4+ vector tiles can be created in any projection and used with the JavaScript API version 3.18+, ArcGIS Runtime 100+, and ArcGIS Pro 1.4+
Q: Can I project vector tiles on the fly?

A: ArcGIS Pro supports this with a much improved implementation at version 1.4. We do not expect to add this to other clients.
Q: Should I re-author my maps for vector tiles?

A: Yes, start by reading the help topic titled Author a map for vector tile creation
Q: Will ArcMap support vector tiles?

A: It is unlikely that ArcMap will ever support viewing vector tiles. Creation of vector tiles will not be implemented in ArcMap.
Q: Can I do server side vector tile creation?

A: Not at this time, we have this on our roadmap for Portal / ArcGIS Online including support for incremental updates.
Q: Vector tiles don’t draw correctly in my browser, is this a bug?

A: You’re likely experiencing a problem with WebGL support in your browser. Ensure you have the latest driver for your video card from the driver manufacturer. Do not rely on Windows Update on Windows machines for video drivers.
Q: Can I use any font for vector tiles?

A: From a technical standpoint any TrueType or OpenType font can be processed into the vector tile font format. However, font licenses vary widely and you should ensure you’re licensed for such use.
Q: Can Esri clients view non-Esri tiles conforming to the spec?

A: This is our goal. There are examples of this in the JavaScript API.

Example: Mapillary
Q: Does the print service support vector tiles?

A: Vector tiles can be printed in ArcGIS Pro. The Print Service in ArcGIS Enterprise 10.6 does support vector tiles. The 4.6 JavaScript API will fully leverage this service. The 3.24 release of the JavaScript API will fully leverage this service when released.

If the JavaScript API detects an older print service, it sends a raster rendering of vector tiles through to the print service.