VECTOR TILES
From Creation to Implementation
Introduction

- Need for consistent basemap across all agency applications
- Need to represent our base layers in basemap
  - e.g. Districts, Planning Boundaries, etc.
  - Roadway network
- Assets along roadway (e.g. pavement condition, projects, etc.) must properly align with our road network
Previous Process - Raster Basemap Tiles

- Huge MXD onerous to manage
- Group layer for each zoom level
- Complicated definition queries and label queries
- Annotation for each layer for each zoom level
Previous Process - Raster Basemap Tiles

Issues:

– Multiple zoom levels increase likelihood of inconsistency between each level
  • Duplication of effort for each
    – Schema changes
    – Easy to miss something
    – Symbology or labeling
  – Sometimes you would not see the results until after creating tiles
Previous Process - Raster Basemap Tiles

- In order to avoid labels being cutoff or appearing to be duplicated by bordering tiles

  • Had to create tile boundary polygons
  
  • Most labeling had to be converted to annotation
    
    – Any update requires a change to the annotation at each zoom level
  
  » This made the case for our switch to Vector Tiles
Enormous overhead in creating Raster Tiles:

– Small mistakes were sometimes not corrected due to enormous time to produce tiles

– PCs not powerful enough to handle processing

– Processing in AGO is expensive (~1,000 credits)

– Ultimately used Amazon machine to process tiles

Missing Interstate Highway Shields at this major intersection in Houston. Only occurs at this zoom level. Not worth reproducing tiles to fix.
Vector Tiles

Compared with Raster Tiles:

- 10 minutes to create tiles for state of Texas all the way down through zoom level 22!

- Map used to create tiles is simpler:
  - Single layer for each feature class
  - No need to duplicate definition queries for each zoom level
  - Labeling is dynamic and is only set once rather than for each zoom level
  - Control what draws at each level using GUI sliders for each symbol class

- If you find a mistake you can quickly re-run the tiles after making the correction
Vector Tile Usage & Configuration

- Can re-symbolize existing tiles
  - Edit JSON
- Implemented with single line of code change in applications
- Will use in all web map applications as standard basemap

```
// Raster basemap service
// tiled = new Tiled("http://tiles.arcgis.com/tiles/KTcxITD9dsQw4r7Z/arcgis/rest/services/Statewide_Planning_Map/MapServer");
// map.addLayer(tiled);

// Vector basemap service

// TxDOT Vector Tile Layer
TxDOTVectorTileLayer = new VectorTileLayer("http://tiles.arcgis.com/tiles/KTcxITD9dsQw4r7Z/arcgis/rest/services/TxDOT_Vector_Tile_Basemap/VectorTileServer");
map.addLayer(TxDOTVectorTileLayer);
```
Vector Tile Usage & Configuration

My Content

Folders

From the computer
From the web

<table>
<thead>
<tr>
<th>From the web</th>
<th>Type</th>
<th>Modified</th>
<th>Shared</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Web Mapping Application</td>
<td>Jan 9, 2017</td>
<td>Not Shared</td>
</tr>
</tbody>
</table>

Item from the web

Reference an item on the Web.

Type:
- ArcGIS Server web service
- KML
- WMS (OGC)
- WFS (OGC)
- WMTS (OGC)
- Document

URL: http://tiles.arcgis.com/tiles/KTcxiTD9dsQw4r7Z/arcgis/rest/services

Title: TxDOT_Vector_Tile_Basemap

Tags: Add tag(s)

ADD ITEM CANCEL
Vector Tile Usage & Configuration

Style
Vector Tiles

Downsides and possible improvements:

– Less control in ArcPro for point symbology such as highway shield as available in ArcMap
  • Unable to adjust placement of text inside shield

– Still minor tile boundary overlap issues, however considerably improved

– Cannot overwrite existing tiles in AGO. Must delete and reproduce in order to maintain URL, however time to produce new tile set makes this a negligible/manageable interruption