Best Practices for Designing Effective Map Services

Ty Fitzpatrick

Tanu Hoque
What’s in this session

- Map service planning and design

- Ways to serve your maps
  - Dynamic map service
  - Cached tiles
  - Client-side graphics

- DynamicLayers capability (new at 10.1)

- Performance tips for map services

Please! Turn OFF cell phones and paging devices
Organize data into logical groupings

**Basemaps**
- Geographic frame of reference
- Contain static vector and raster data
- Reusable in multiple applications

**Operational Layers**
- Show a focused item of interest
- Support functionality of the application
- Displayed on top of base map
Community templates
Ways to serve your maps
Three options for displaying map services

• As cached tiles

• As a dynamically drawn image

• As client-side graphics
Demo

Publishing Dynamic Map Services
Dynamically drawn map services

- Server retrieves data, draws an image, sends image to client
- Slower than caching, but may be satisfactory using dynamic map service
Data that’s OK to draw dynamically

- Real-time data
- Frequently-changing data with large scope
- Internal maps accessed by smaller groups of people
Demo

Publishing Cached Map Services
Cached tiles

- Pre-draw map tiles and serve them to clients
- Best performance and scalability
- Standard for online maps (ArcGIS Online, Google, Bing, Yahoo, etc)
- Requires you to create and maintain cache
What should you cache?

- Base maps
- Operational layers that satisfy one of the following:
  - High volumes of traffic
  - Don’t change often
  - Cover small scales only
Cache image formats

- MIXED for most basemaps
- PNG for overlay networks (boundaries, roads)
- PNG 8 for classified rasters < 256 colors
Authoring a basemap from scratch

- Only a good option if there are no templates

- Design map for cache scales
  - Add your tiling scheme scales to the ArcMap dropdown list

- Group layers by scale level
  - Only have to set the scale range at the group layer level
  - Copy layers between groups
Antialiasing with map services

- Improves visual quality

- Slight performance cost
  - Use Preview button to see effect on performance
Publishing your map service
Dynamic Layers
Map service dilemmas

• cannot change without republishing
• end users can’t modify how or which order a layer draws

streets basemap
light gray canvas basemap
ocean basemap
national geographic basemap
topographic basemap

e.g. the same roads are drawn differently depending on purpose
Map service dilemmas

- One layer with many many many… attributes!!!

  e.g. layer containing census, health indicator data
Map service dilemmas

- many many... FeatureClasses
  Ê ? map services
## Functionality matrix

<table>
<thead>
<tr>
<th></th>
<th>Cached map service</th>
<th>Dynamic map service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toggle layer visibility</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Toggle label visibility</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Reorder layers</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Modify layer's renderer</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Define new labels</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Add new layer dynamically</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Create joins dynamically</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
Dynamic layers: the concept

• New behavior with the map service that allows for per-request changes to the map
  - Server side change
  - Stateless

• Optional capability of map services
Demo
Dynamic Layers – Thematic Mapping
Dynamic layers: use cases

- **Simple updates to the map service**
  - Remove layers or reorder layers

- **Thematic mapping**
  - Updates to renderers
  - Change data sources – including joins

- **Adding content to the map service**
  - Add data from registered workspaces
    - Including query layers
Dynamic layers:
- Server Side Utility

- Generate renderer operation
  - New map server API for constructing renderer classes
  - Supports
    - class breaks
    - unique value
  - Popular classification types from ArcGIS Desktop
    - natural breaks
    - equal interval
    - standard deviation etc.
# Functionality matrix

<table>
<thead>
<tr>
<th></th>
<th>Cached map service</th>
<th>Dynamic map service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dynamicLayers‡</strong></td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>Toggle layer visibility</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle label visibility</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorder layers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify layer's renderer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define new labels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add new layer dynamically</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create joins dynamically</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic layers: where can I learn more?

- See the topic “About dynamic layers” in the ArcGIS Server help
- REST API – See the “Dynamic Layer / Table” resource help
Dynamic layers: ArcGIS Server blog posts

- Mapping Health Indicators using Dynamic Layers in ArcGIS Server 10.1
- Uploading client data to perform thematic mapping on the Server
Client-side Graphics
Client-side Graphics: The Concept

- Server sends geometries and attributes to client
- Features drawn in browser
- Client side layer types:
  - GraphicsLayer
  - FeatureLayer
- Source Service types:
  - Map services
  - Feature services
- Query mode:
  - Snapshot
  - On demand
  - Selection
Client-side graphics: what should you draw?

- Interactive operational layers for mashups
- Query or geoprocessing results
- Web editing: Feature Services
- Layers that need to be thematically symbolized on the fly
  - National Center for Education Statistics
Client-side graphics: considerations

- Symbology

  - Rendered in the client
  
  - Symbols can be Simple (Marker, Line, Fill) or Picture (Marker, Fill)
    - Complex symbols are converted to picture (PNG)
      - Most point symbols reproduced well – test
    - Avoid gradient fills
Client-side graphics: considerations

- Be careful not to request too many features
  - Scale dependencies with Feature Services
- Generalize geometries - *maxAllowableOffset*
  - Do not generalize geometries in Editing scenario
- Beware of server limits on number of features that can be returned
  - Default 1000
Client-side graphics: considerations
- maxAllowableOffset

- A way of reducing the number of points in a curve
- Suggestion: a feature’s geometry should not display more than one vertex per pixel

http://storymaps.esri.com/globalfootprint/
Dynamic layers vs Client-side graphics

• When do I use dynamic layers instead of feature layers on the client for thematic mapping?
  - large number of features
  - complex geometries that cannot be generalized
  - when it provides a performance advantage

• Each approach has tradeoffs
  - e.g. Client side features scale better and provide more interactive behavior

• GenerateRenderer operation can be used with both approaches
Client-side graphics: where can I learn more?

- See the Web API Sessions (Javascript, Flex, Silverlight) in the agenda.
- Online examples at the ArcGIS Resource Center
Client-side graphics: ArcGIS Server blog posts

- Determining Limits for Map Graphics
- High Performance Web Map with Large Dataset as FeatureLayer
- Out of Box Vector Tiling using FeatureLayer
- FeatureLayer can Generalize Geometries on the fly
Performance tips for map services
Pre-compute when possible

- Cache
- Annotation
- Projection
  - Tip: You can re-project geodatabase features during replication
- Spatial indexes
  - Keep up to date
  - Correct size relative to map extent
- Scale dependent layer visibility
Data access tips

- ArcSDE geodatabase tips
  - Tune ArcSDE
  - Use direct connect
- Avoid UNC paths for file-based data
- Cached query or tool results
  - Example: Solar Boston
- Avoid downloading all attributes unless you have to
- Attribute indexes
  - Use for joins and common queries
Maps for Mobile Devices
Authoring mobile maps
Specific cartography for mobility

- Design for purpose
  - Remove unnecessary layers of information
  - Set scale dependency (walk, drive, etc)
  - Render editable layers to define feature types

- Design for the environment
  - Establish contrast, choose meaningful symbology
Authoring mobile maps (continued)
Specific cartography for mobility

• **Design for device form factor**
  - Set scale dependency based on device resolution
  - Set symbol width based upon device resolution

• **Architect and Deploy Map Data**
  - Build Compressed Base Map Datasets
  - Build Operational Mobile Caches
Review

• Organize map services in logical groups
  - Base maps
  - Operational layers

• Use a high-performance blend of display techniques
  - Cached tiles
  - Dynamically drawn services
  - Client-side graphics
  - Dynamic layers

• Follow performance tips, pre-computing when possible
Additional resources

• ESRI Showcase
  - Meet ESRI Development staff

• Other sessions
  - Advanced Map Caching Topics
  - Javascript, Flex, Silverlight web API sessions
  - Many other ArcGIS Server sessions

• Resource centers: http://resources.esri.com
Thank you…

- Please fill out the session survey:

  **First Offering ID:** 1121
  **Second Offering ID:** 1239

  Online – [www.esri.com/ucsessionsurveys](http://www.esri.com/ucsessionsurveys)
  Paper – pick up and put in drop box