Best Practices for Designing Effective Map Services
Sterling Quinn
Tanu Hoque
What’s in this session

- Map service planning and design

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

- Authoring a good Web map

- Performance tips for map services
How Web maps have changed!
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
Case study: Google Maps
Google Maps base maps

**“Map”**
- Highways
- Streets
- Ferries
- Railroads
- Transit centers
- Cities
- Parks
- Military reservations
- Municipal boundaries
- Lakes
- Rivers
- Golf courses
- Hospitals
- Shopping centers
- Airports
- Colleges
- Cemeteries
- Amusement parks

**“Terrain”**
- Shaded relief
- Vegetation
- Highways
- Streets
- Cities
- Parks
- Military reservations
- Municipal boundaries
- Lakes
- Rivers
- Golf courses
- Hospitals
- Shopping centers
- Airports
- Colleges
- Cemeteries
- Amusement parks

**“Satellite”**
- Imagery
- Source information
Google Maps operational layers

- Street overlay for imagery
- Traffic
- Photos
- Videos
- Wikipedia
- StreetView coverage
- Web cams
- Bicycle routes
Some ArcGIS Server examples

- Orange County Property Appraiser Map
- City of St George
- North Vancouver Projects
- Solar Boston
- City of Greeley Property Information Map
Ways to serve your maps
Three options for displaying map services

- As cached tiles
- As a dynamically drawn image
- As client-side graphics
Cached tiles

- Pre-draw map tiles and serve them to clients
- Best performance and scalability
- Standard for online maps (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
  - High quality (~90) for vectors
  - Lower quality (55 – 75) for imagery

- PNG for overlay networks (boundaries, roads)

- PNG 8 for classified rasters < 256 colors
Related Session

- Designing and using cached map services
  - Wednesday, 8:30 AM    Room 6C
  - Thursday, 3:15 PM    Room 8

- Advanced map caching topics
  - Wednesday, 1:30 PM    Room 2
Dynamically drawn map services

- Server retrieves data, draws an image, sends image to client

- Slower than caching, but may be satisfactory using optimized map service
Data that’s OK to draw dynamically

- Real-time data
- Frequently-changing data with large scope
- Internal maps accessed by just a few people
Two types of files can support a map service

- Map service definition .MSD
- Map document .MXD

OR

- Optimized map service
- Traditional map service
Optimized map services

- Obtained through Map Services Publishing toolbar in ArcMap
- Supports the most common layer and symbol types
- Faster dynamic drawing than ArcIMS
Demo

Publishing an optimized map service
Antialiasing with optimized map services

• Improves visual quality

• Slight performance cost
  - Use Preview button to see effect on performance
What's available through optimized services?

- Most data and layer types
  - You'll get an Error in analyzer if not supported

- New at 10.0: Maplex and cartographic representations
  - Recommended for caching only
If you have to use an MXD-based service…

- Move whatever layers you can into a separate optimized map service

- Use ESRI_Optimized style for drawing

- Still use the Analyze button to catch performance warnings
Client-side graphics

- “Data on demand” pattern treats map service as a feature server
  - Queries from map services
  - Feature services
- Server sends geometries and attributes to client
- Features drawn in browser
What should you draw with client-side graphics?

- 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
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
maxAllowa... what?

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
Graphics performance considerations

- **Generalize geometries**
  - Do not generalize geometries in Editing scenario
- Be careful not to request too many features
  - Scale dependencies with Feature Services
- Beware of server limits on number of features that can be returned
  - Default 1000
- Beware “1=1” firewall filters
Where can I learn more about these techniques?

• Implementation differs depending on the web API being used

• See the Web API Sessions (Javascript, Flex, Silverlight) in the agenda.

• Online examples at the ArcGIS Resource Center
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
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
Authoring a good web map
Start with a template

• Fully functional apps, maps and data
  - Esri: Resources.arcgis.com
    - User Communities
  - Community: www.arcgis.com > Gallery
    - Look for “Configurable”
  - Download and point at your own data
Demo

Building a basemap from a template
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
Authoring for feature service: 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

ArcMap

WebMap
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
Sneak peek: server side thematic mapping at 10.1

- New at ArcGIS Server 10.1
- Allows to modify renderer without downloading geometries on the client side
  - Specially helps with too many features
- Maps are drawn at the server side
  - Only image is returned to the client
- Helper function available on the server side to compute class breaks based on different classification methods.
Demo

Server side thematic mapping at 10.1
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

- 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:** resources.esri.com
Questions

Please fill out the survey for this session
www.esri.com/sessionevals