Oklahoma SCAUG 2011

Best Practices for Designing Effective Map Services

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

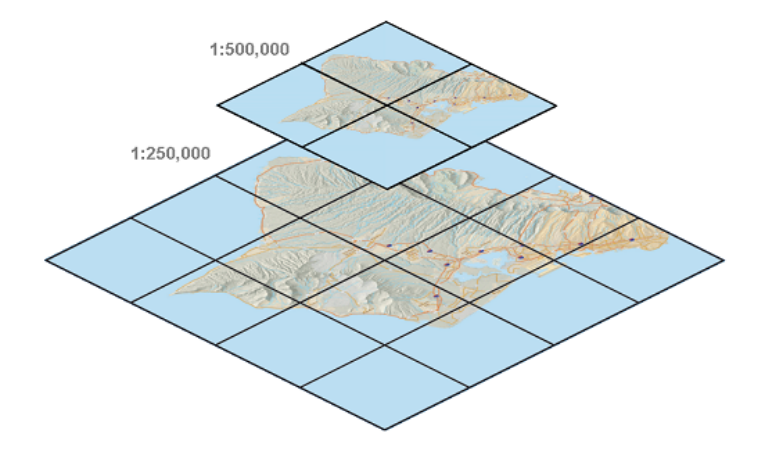
Displayed on top of base map

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

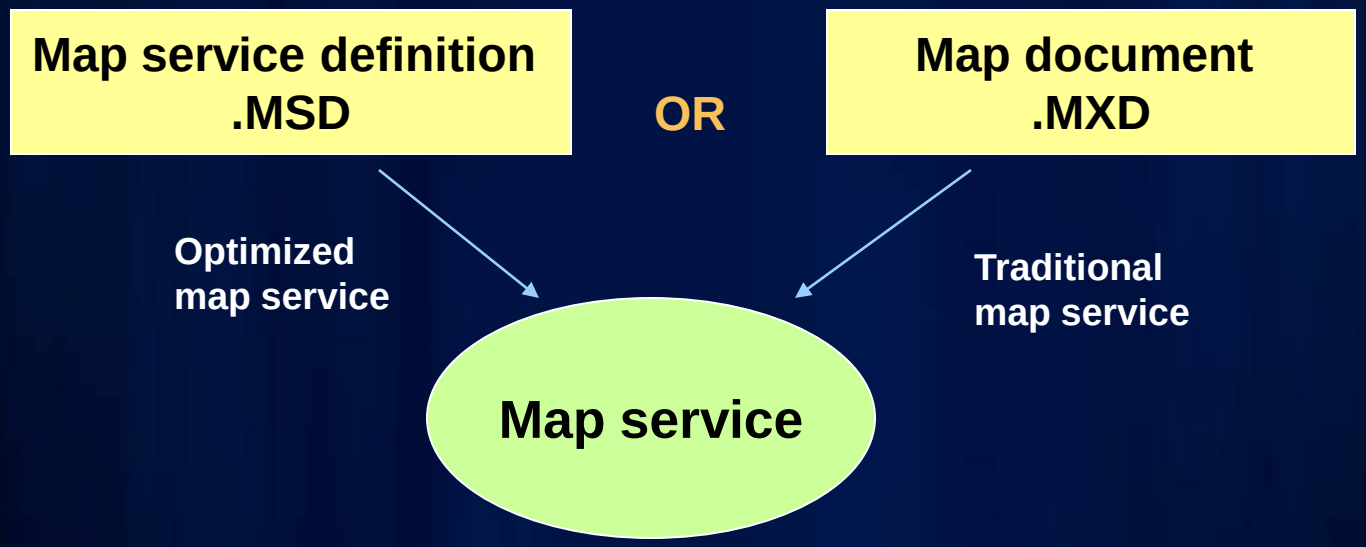
Dynamically drawn map services

- 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



Optimized map services

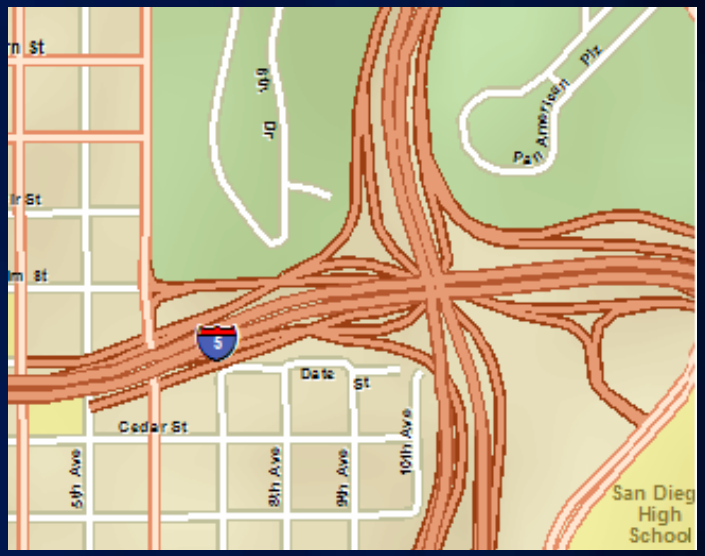
- Obtained through Map Services Publishing toolbar in ArcMap
- Supports the most common layer and symbol types
- Faster dynamic drawing than ArcIMS

Demo

Publishing an optimized map service

Antialiasing with optimized map services

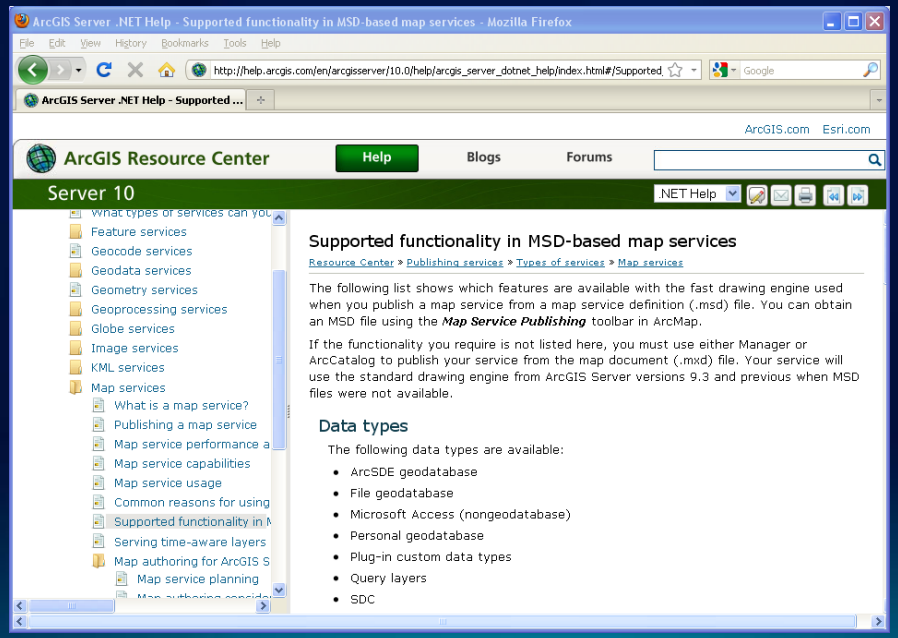
- Improves visual quality
- Slight performance cost
  - Use Preview button to see effect on performance



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

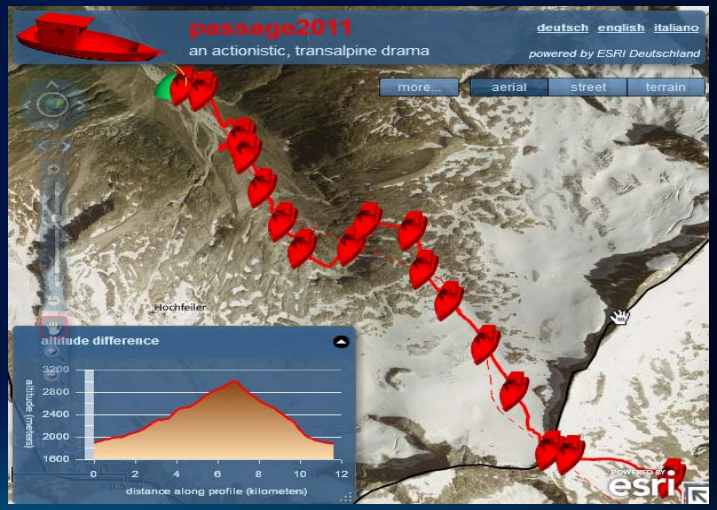
Prepare

1 Error | 12 Warnings | 6 Messages | Search:

Severity	Status	Code	Description	Name	Type	Data Frame
High	Unresolved	00005	Layer type is not supported	BufferPoints	Layer	BaseMap
High	Unresolved	10001	Layer's data source has a different projection [NAD_198…	Hillshade	Layer	BaseMap
High	Unresolved	10002	Layer's data source doesn't have a spatial index	Place labels	Layer	BaseMap
Medium	Unresolved	10009	Enabling the option to convert layer transparency to colo…	Place labels	Layer	BaseMap
Medium	Unresolved	20007	Layer contains a multilayer line symbol whose symbol widt…	Major Highways (Regional)	Layer	BaseMap

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

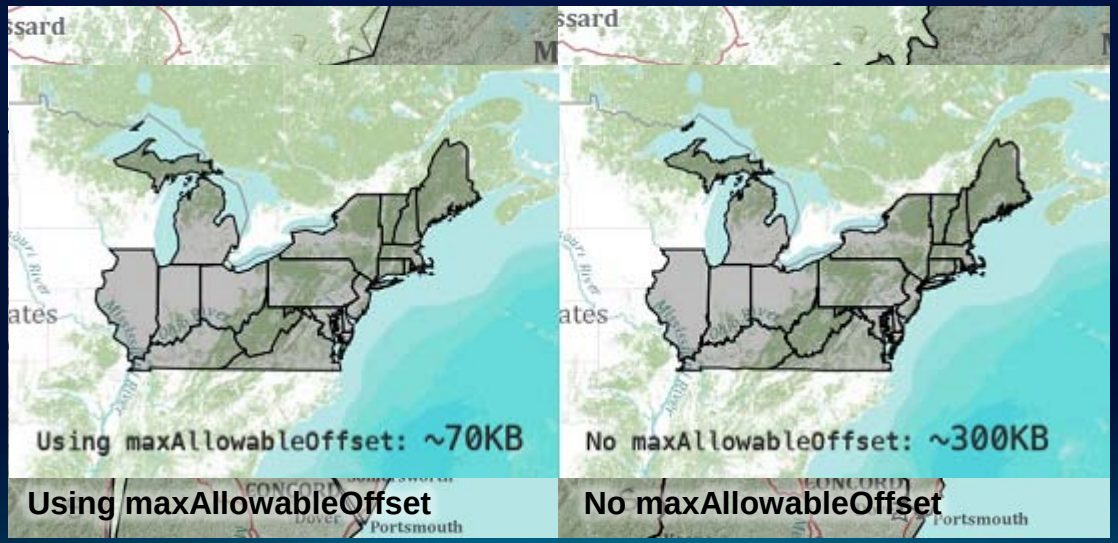
ArcGIS Server Blog Posts

- 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?

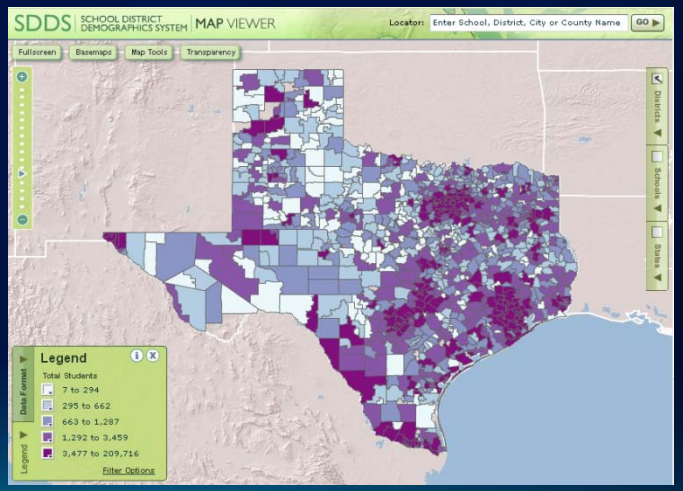
maxAllowableOffset:

- 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

esri