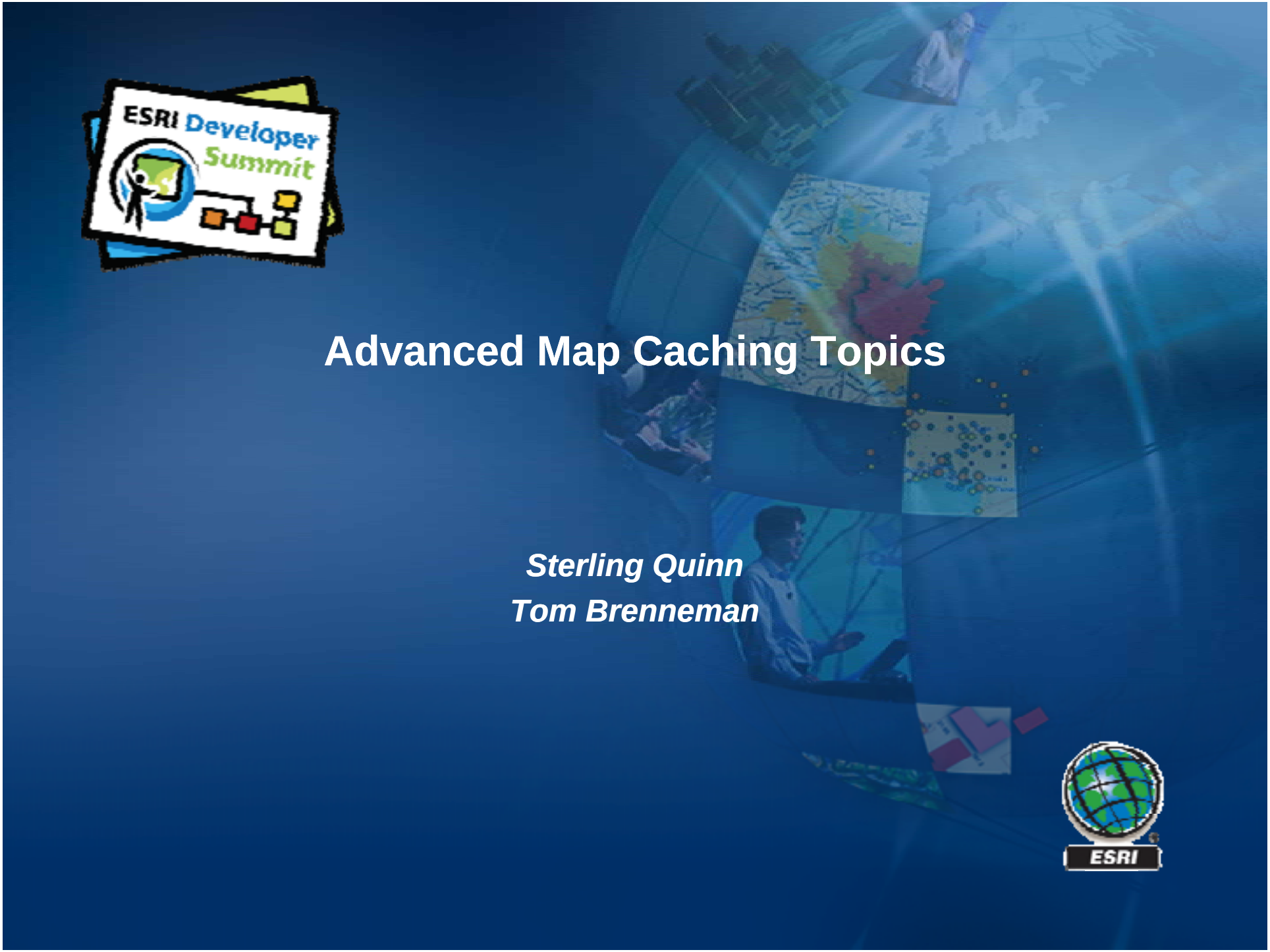


Advanced Map Caching Topics

Sterling Quinn
Tom Brenneman

Schedule

- Today we will cover
  - Why and what to cache
  - Key components of a map cache
  - Authoring considerations for cached map services
  - Caching strategies
  - System architecture

- We will answer questions at the end of the session

Please complete the session survey!

Why make the effort to cache?

- Speed
- Scalability
- Appearance
- Orange County Property Appraiser

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**

Key components of a map cache

- Tiling scheme
  - Scales
  - Tile size
- Image format
- Antialiasing

Set the tiling scheme

- Choose from well-known tiling schemes of Web map services
  - ArcGIS Online
  - Google Maps & Virtual Earth
- Create your own
  - Do this only if other options will not work
- Navigation limited to the tiling scheme scales
- Tiles cannot be projected

Using a well known tiling scheme

- **ArcGIS Online**
  - Map must use WGS 1984 coordinate system
- **Google Maps and Virtual Earth**
  - Map must use WGS 1984 Web Mercator coordinate system
  - May need to apply appropriate transformation to get data to align
    - WGS_1984_Major_Auxiliary_Sphere_To_WGS_1984
    - See KB article 34749

Choosing the scales for a tiling scheme

- **Build just the scales you need**
  - **Determine closest scale (Raster resolution)**
  - **Divide scale by 2 for each subsequent scale**
  - **Adjust smallest scale to full extent**

Sample 10 level cache

Level	Scale	Tiles	% of total
1	1:16,000,000	1	0.000%
2	1:8,000,000	4	0.001%
3	1:4,000,000	16	0.005%
4	1:2,000,000	64	0.018%
5	1:1,000,000	256	0.073%
6	1:500,000	1,024	0.293%
7	1:250,000	4,096	1.172%
8	1:125,000	16,384	4.688%
9	1:62,500	65,536	18.750%
10	1:31,250	262,144	75.000%

Final level is ~75% of the total

Tile size

- Pixel dimensions of each image
- 256x256 and 512x512 are defacto standards
  - ArcGIS Online uses 512 X 512
  - Google Maps and Virtual Earth use 256 X 256
- Larger dimensions are faster to build, but tiles take longer to download

Choosing an image format

- Image format affects
  - Tile storage space requirements
  - Web application performance (speed and supported browsers)
  - Tile image quality
  - Tile transparency
- Choose carefully and build a test cache

Basemaps: Image format guidelines

- **JPEG**
  - Over 256 colors
  - Small file size for many colors, no transparency

JPEG 75 (magnified)

JPEG 90 (magnified)

- **PNG 8**
  - Under 256 colors
  - Very small images, crisp lines, antialiasing

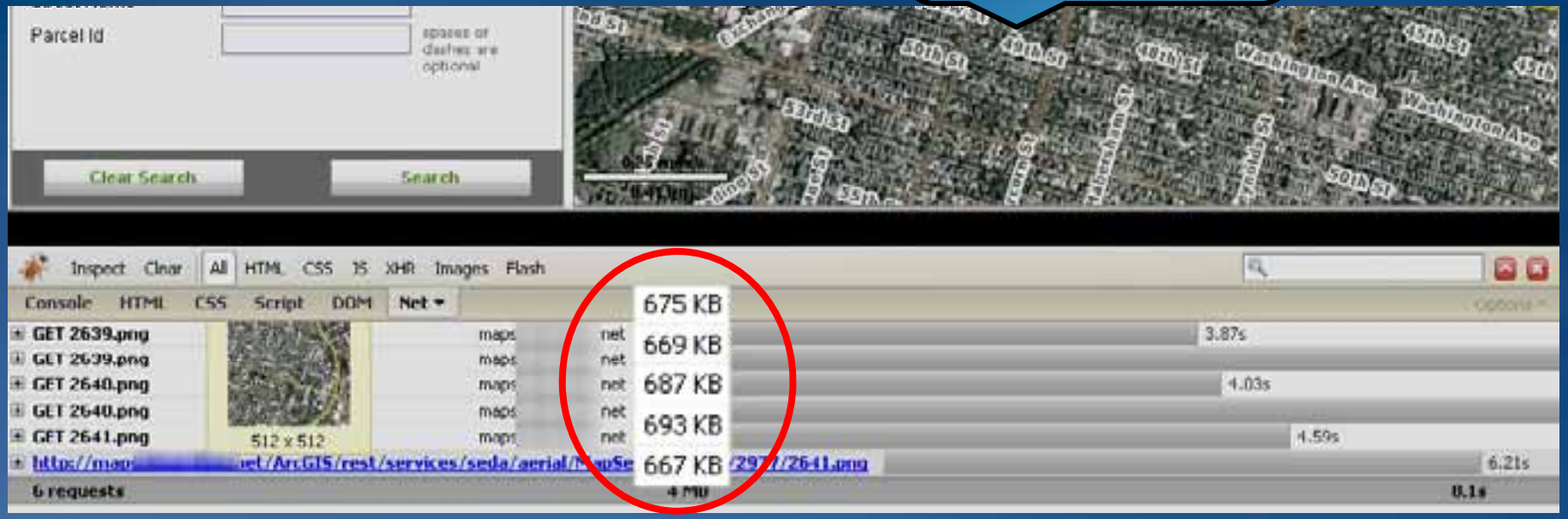
Operational layers: Image format guidelines

- PNG 8
  - Under 256 colors
  - Imagery, hillshades, gradient fills, highway symbols, and antialiasing can push your map over 256 colors
  - Free tool – Irfanview- can help with this
- PNG 32
  - Over 256 colors
  - Maps with antialiasing
- PNG 24
  - Poor support in IE 6

- Solar Boston

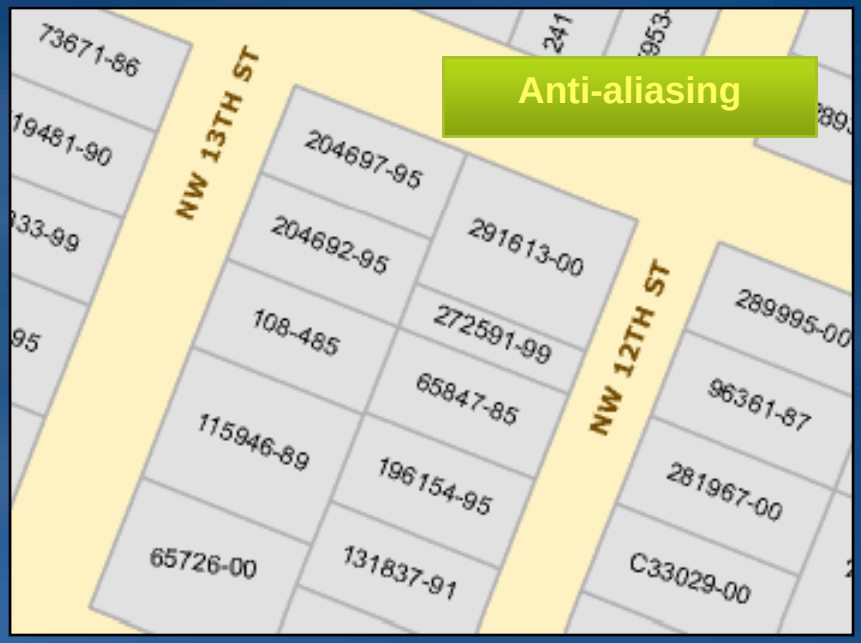
Example: Tiles are too large

Aerial photo and vector blend using PNG 32



When should I use antialiasing?

- High quality line and label appearance on vector maps
- Web standard (Google, VE, AGOL)



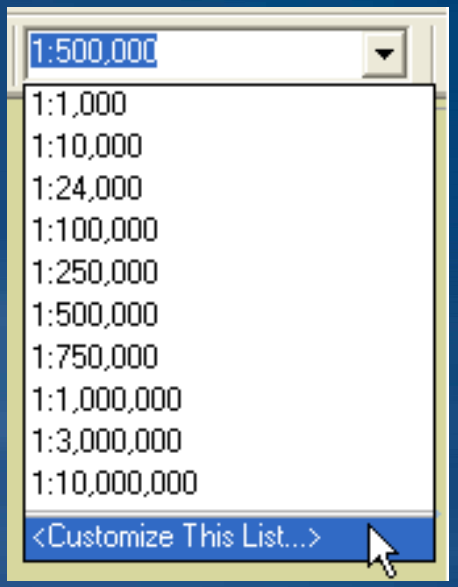
- Optimized map services provide more antialiasing options
- PNG 32 is better for antialiasing but larger file sizes

Authoring considerations for cached map services

- Scales
- Map text

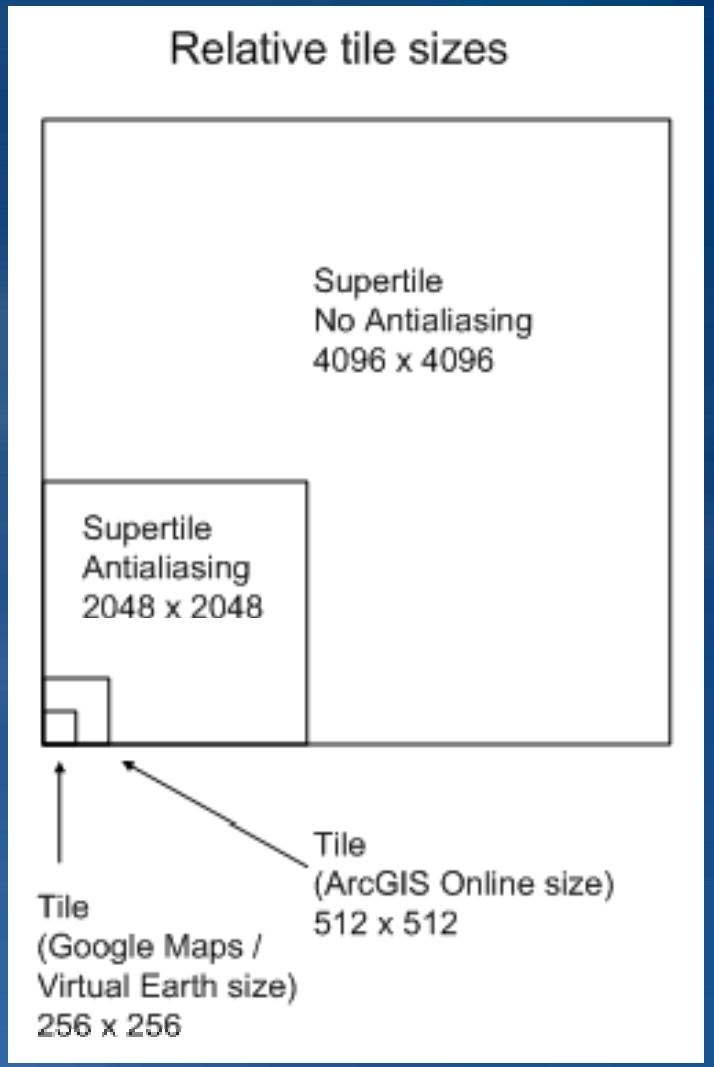
Design for your cache scales in ArcMap

- If map will overlay other services, match the projection
- Choose a set of scale levels and design at those
  - 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



Super tiles and labeling

- individual tiles are cut from large area (supertile)
  - 4096 x 4096
  - 2048 x 2048 if using antialiasing
- Supertile necessary to
  - Reduce duplicate labeling
  - Reduce requests to map service when caching
- Labeling rules can repeat across super tile boundaries
  - Use the Maplex labeling engine
  - Annotation



Caching strategies

- Test cache
- Strategic cache creation
- On-demand caching
- Very large cache creation
- Strategic updates

Creating a test cache

1. Select test area with varying geography
2. Create a simple feature class covering the test area
3. Create tiles at all scale levels based on this feature class

Note the following from your test cache

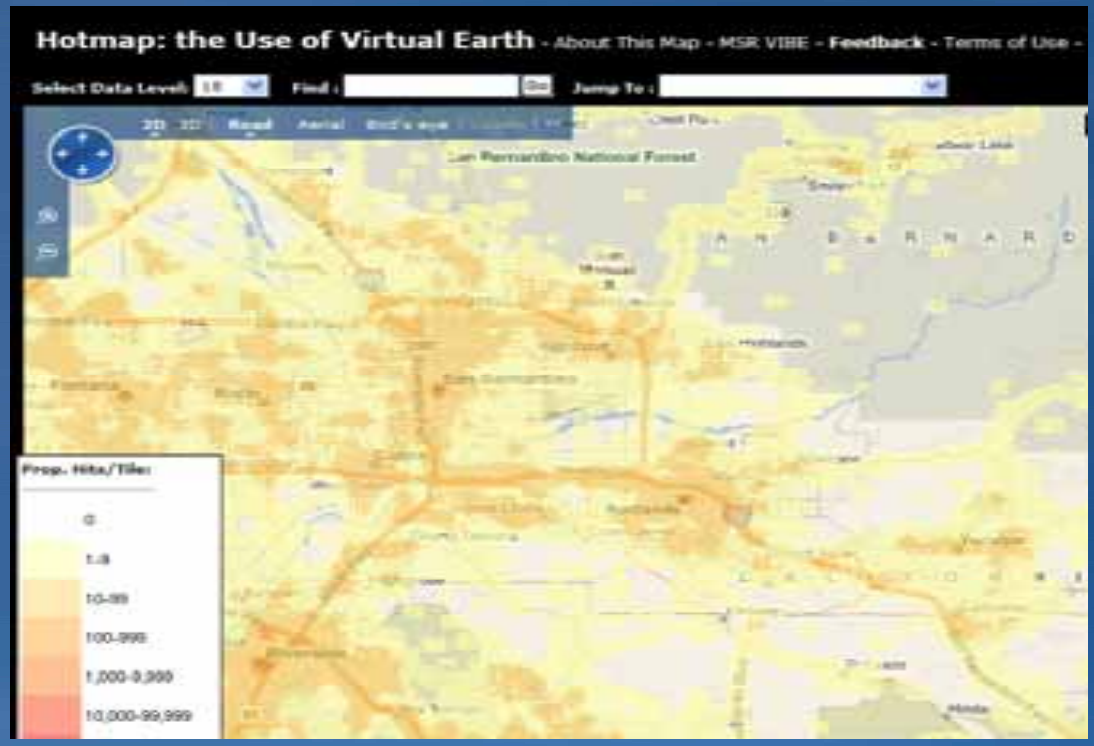
- Appearance of tiles
- Performance of tiles in client
- Cache creation time
- Cache size on disk

Strategic cache creation

Strategic cache creation

- Do you really need to create all tiles at all scales?
  - At street level scales some tiles might never be used
- Manage Map Server Cache Tiles tool
  - Update specific areas using a feature class

http://hotmap.msresearch.us



How do you know which tiles will be popular?

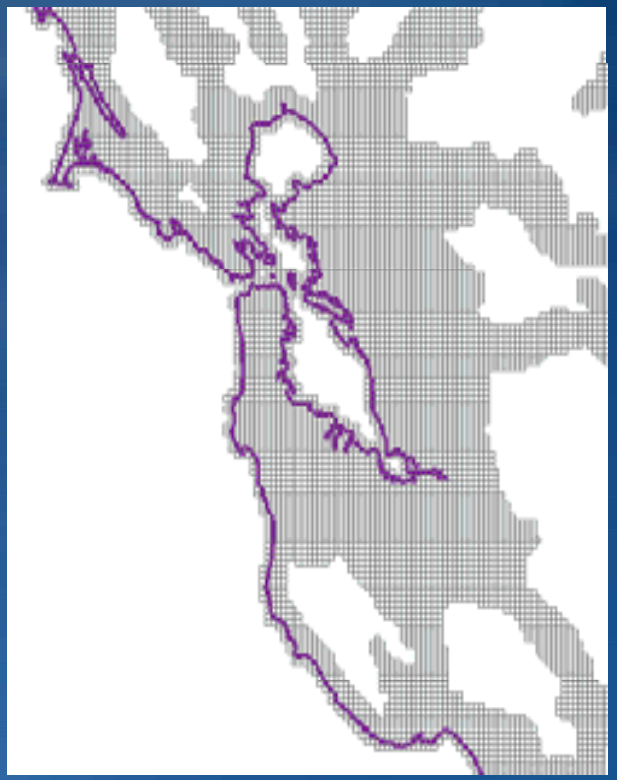
- Study usage patterns of your maps
- Examine known example of MS Hotmap
  - Urban areas
  - Roads
  - Coastlines
  - Points of interest

Popular areas can be modeled

MS Hotmap

Modeled “hot” areas: Urban, roads, coasts, POIs

Tiles created using model output

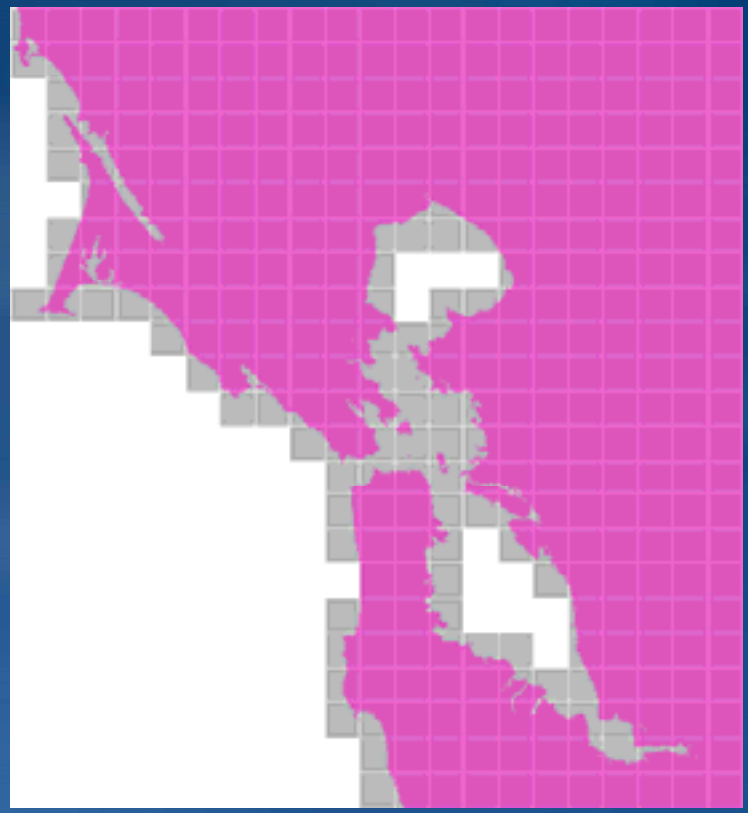


Example model output

- Covers about 25% of California
- Contains about 96% of California's population
- At 1:9000 (neighborhood / street level) creates 56% of California's area in tiles
- Saves 1.7 GB when caching a street map down to 1:9000

All supertiles intersecting your feature class are created

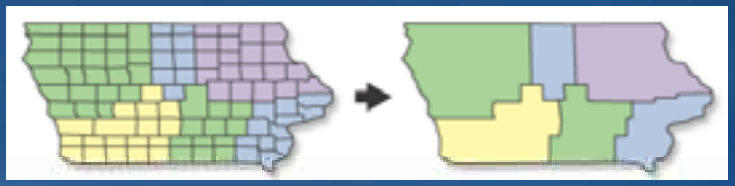
- Extra tiles created on periphery of your area of interest
- Complex feature class = many tiles on the periphery
- Cache by feature class most effective at large (zoomed in) scales



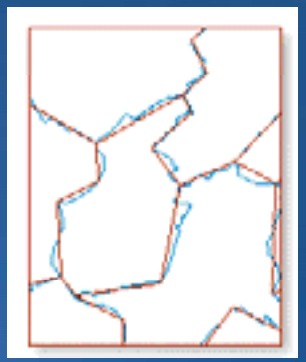
More tips for caching by feature class

- Feature class coordinate system should match map you're caching

- Avoid numerous small features
  - Aggregate Polygons tool
  - Dissolve tool



- Avoid excessive vertices
  - Simplify Polygons tool



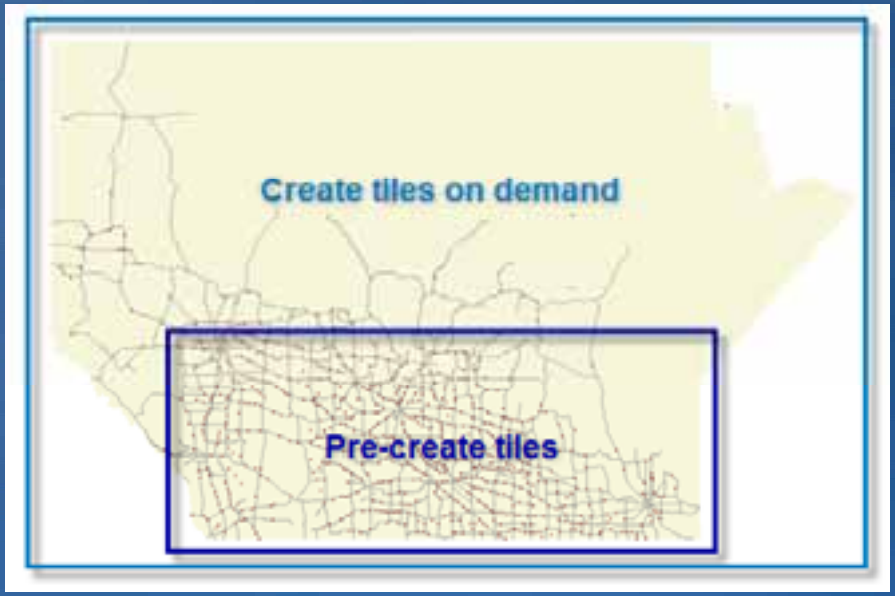

On-demand caching

On-demand caching

1. User navigates to uncached area
2. Supertile is created and added to the cache

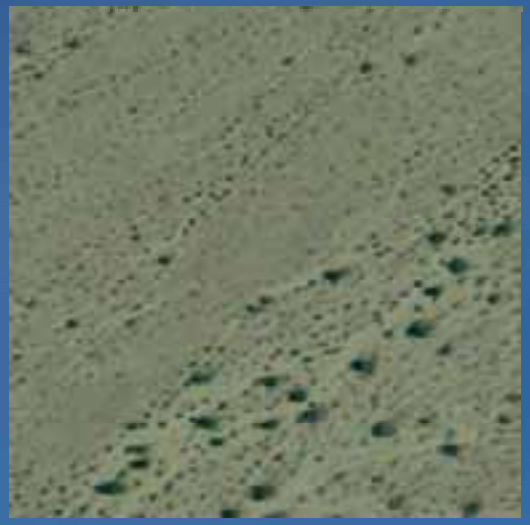
On-demand caching and pre-created tiles

- **Pre-create tiles for areas that you anticipate will be most popular**
  - **Use on-demand for everything else**



The ideal tiles to cache on demand

- Few simple features
  - Barren homogenous area
- Rarely accessed
- Draw relatively fast
- Large scale




Creating a very large cache

Creating a very large cache

- Use *Update specific areas using a feature class* option
  - Use status field to track creation
  - Divide study area into manageable chucks
    - i.e. one days worth of cache creation
- Keep your CPU below 100%
  - Use N+1 instances where N = the number of sockets on server
  - Adjust number of instances based on test cache

Update strategies

Should this data be cached

- Some data shouldn’t be cached
  - Real time data
  - Dynamic rendering
- Alternatives
  - Optimized map services
  - Graphics

What affects cache update strategies?

- Size of cache
  - Rebuild entire cache
- Scales containing updates
  - Rebuild specific scales
- Location of updates
  - Rebuild specific areas

- Cache updates can be scripted with geoprocessing

Targeting cache updates to edited areas only

- **Versioned Geodatabase data**
  - **Custom Geoprocessing tool: Show Edits Since Reconcile**
- **Data that is not versioned**
  - **Select by location equal features, then invert selection**
  - **Custom solution**
- **Note: This technique can capture geometry changes, but not symbology changes**

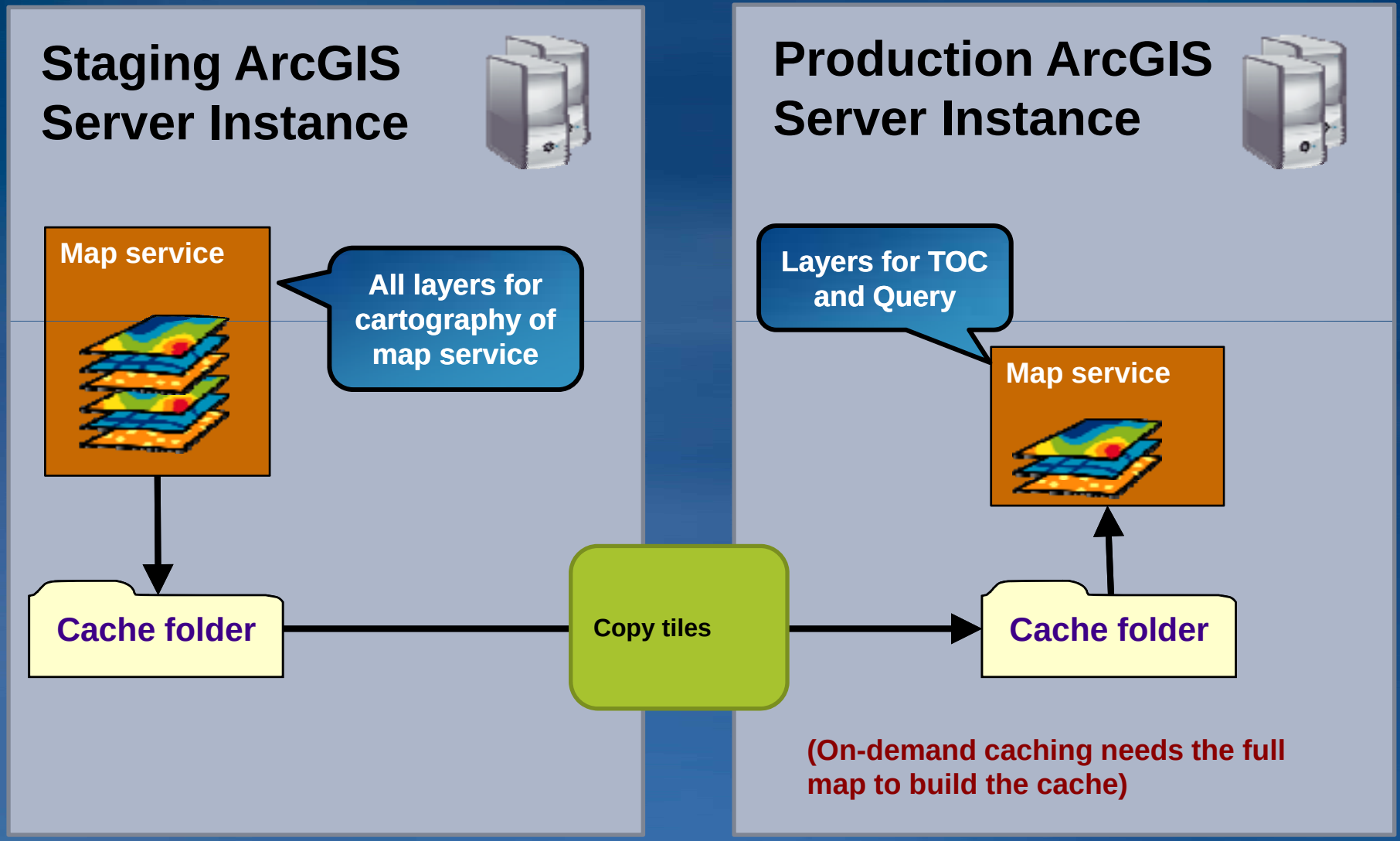
When updating cache based on edited areas…

- Full update at small scales, targeted update at large scales
- Generalize features that are numerous and close-together
  - Aggregate Polygons tool
- Dissolve edited areas into one multipart feature
  - Prevents numerous service restarts and tile duplication

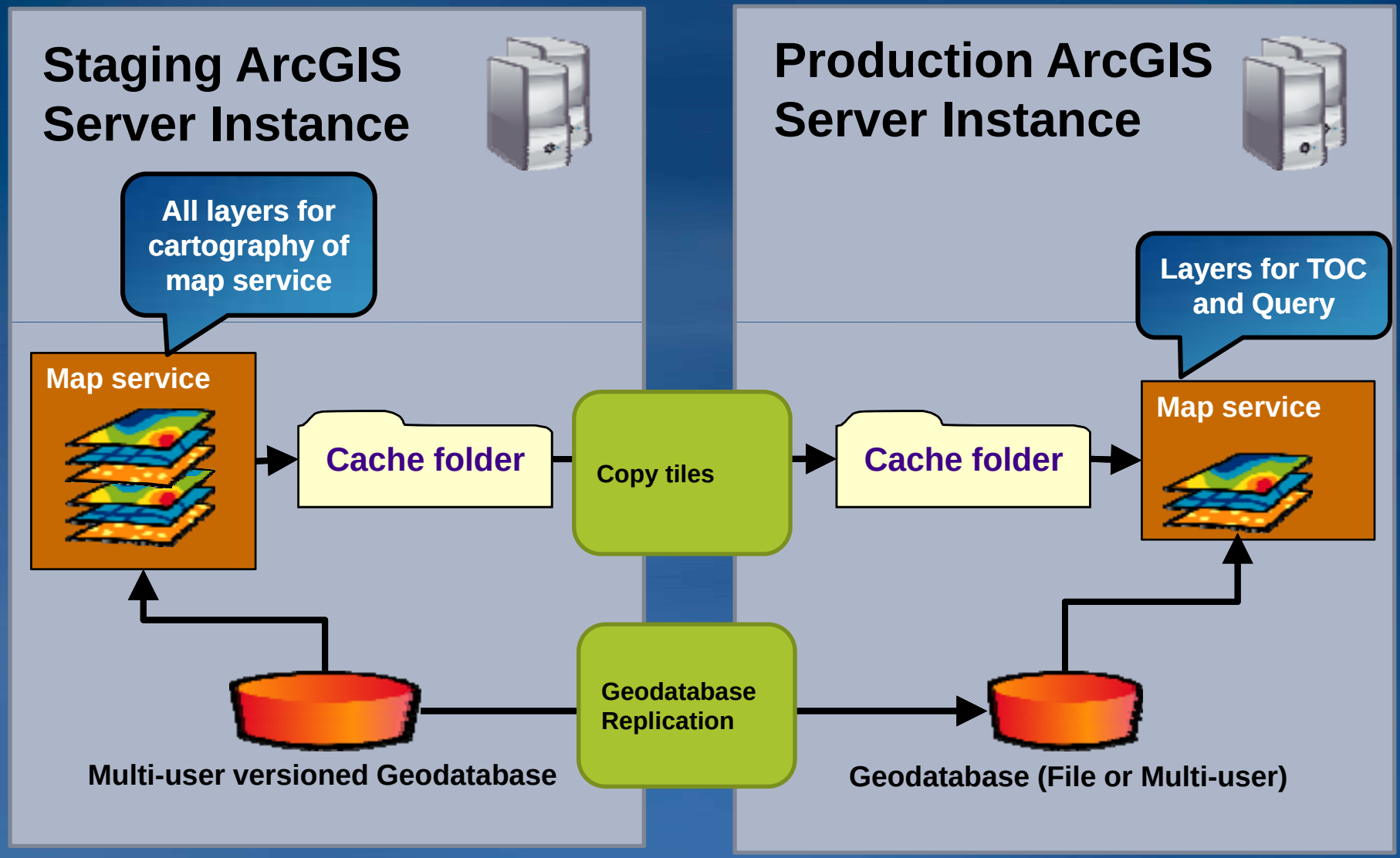
System architecture

- Staging server
- Cache storage
- Cache distribution
- HTTP connections

Update a cache using a staging server



Update a cache and geodatabase using a staging server



Copying caches

- Windows Copy/Paste inefficient for large caches
- Use XCOPY command
  - /D switch will only copy new tiles
- Third party utilities (SecureCopy) may be helpful for large caches

Reducing “Size on disk”

- Default minimum cluster size on Windows is 4k
- Lowering to 1k (or less) can reduce “Size on disk” if you have many simple tiles
- Requires you to store cache on dedicated partition or disk

Cache distribution

- HTTP 1.1 spec constricts browsers to two simultaneous downloads
- Improve cache retrieval performance by using multiple hosts
  - Can be one server with multiple DNS entries (cache1.mydomain.com, cache2.mydomain.com)
- Geographically distribute ArcGIS Server instances

Using multiple domains

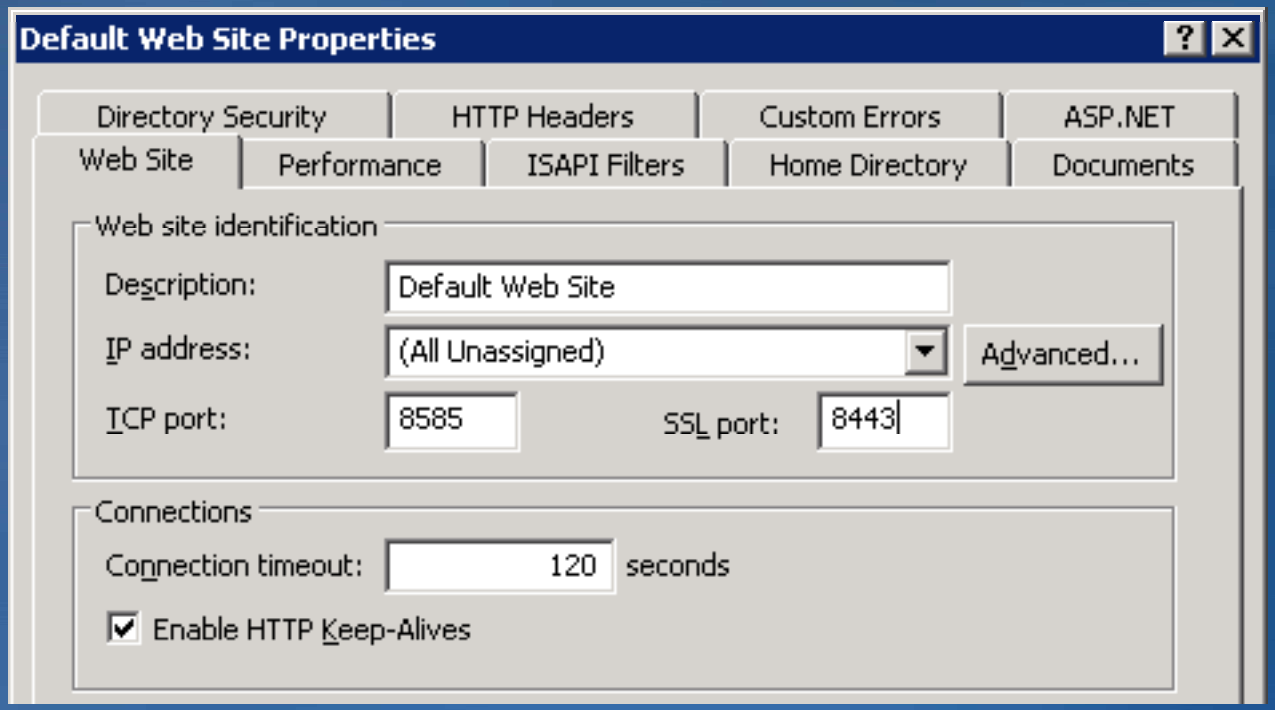
- With multiple services
  - Use a different domain for each services
- With one service
  - API's support multiple web services endpoints for a single layer

```
var layer = new esri.layers.ArcGISTiledMapServiceLayer(
      "http://www.mydomain.com/ArcGIS/rest/services/myservice/MapServer",
      { tileServers: [
      " http://cache1.mydomain.com/ArcGIS/rest/services/myservice/MapServer ",
      " http://cache2.mydomain.com/ArcGIS/rest/services/myservice/MapServer "]
      });
```

- Use with small cache tiles
- Can reduce browser caching and result in more HTTP connections

HTTP connections

- **Turn On HTTP KeepAlive for best performance**
  - **Multiple image requests don't need to open a new connection**
  - **Biggest benefit with one host (~2-3 times faster tile requests)**
- **Apache: KeepAlive directive**
- **IIS**



Summary

- Today we covered
  - Why and what to cache
  - Key components of a map cache
  - Authoring considerations for cached map services
  - Caching strategies
  - System architecture

Still have questions?

Additional Resources

Questions, answers and information…

- ***Tech Talk***
  - ***Outside this room right now!***
- ***Other sessions***
  - ***Best Practices for Designing Effective Map Services***
- **Meet the team**
  - **6:00 – 7:00 pm during the party in Oasis 2**
- ***ESRI Resource Centers***
  - **PPTs, code and video**

resources.esri.com

- ***Social Networking***

www.twitter.com/ESRIDevSummit

tinyurl.com/ESRIDevSummitFB

Want to Learn More?

ESRI Training and Education Resources

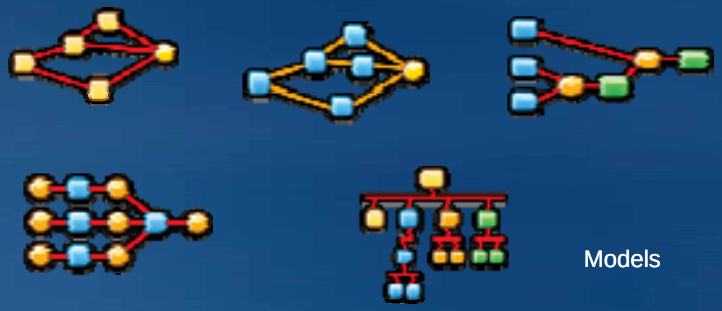
- **Instructor-Led Training**
  - Introduction to ArcGIS Server
  - ArcGIS Server: Web Administration Using the Microsoft .NET Framework
- **Free Web Training Seminars**
  - Authoring and Publishing Optimized Map Services
  - Implementing and Optimizing ArcGIS Server Map Caches

http://www.esri.com/training

Questions


ArcGIS Desktop Authors
ArcGIS Servers
GIS Users
Geodatabase
Web Users
Web GIS
Raster Files
Desktop
Desktop
Desktop
Mobile
Mobile
Mobile
Mobile
Mashups
Explorer
ArcGIS Server
ArcGIS Online
Map
Map
Map
Web Map
Mashups
Browser
Browser
Web Blog
Web Blog
Web Map
Web Map
Open Standards
Web Map

Designing & Planning



Models

Layers

Professional Services

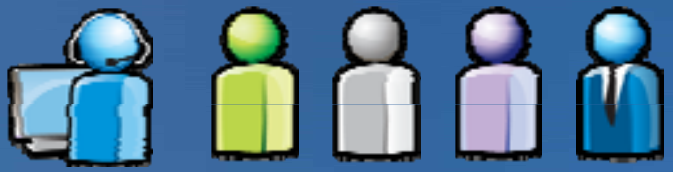
Professional Services

Situational Awareness

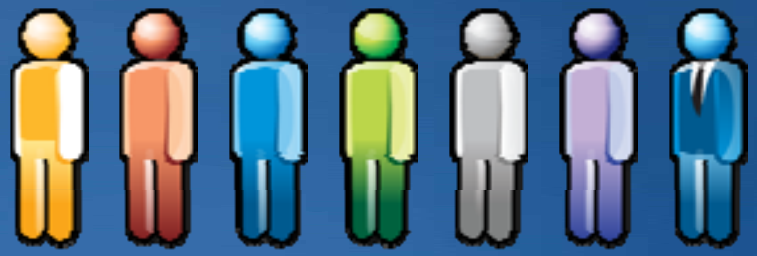
Education

Education

Tech Support



Tech Support



Business Partner


Files
Documentation
Internet
Data Appliance
CD/DVD
GIS
GIS
ArcGIS
ArcGIS
Database
ArcGIS
ArcGIS
ESRI
Satellite
Satellite

Body Content Master (24 point white)

Subtitle (16 point yellow italic)

- **Bulleted text (24 point with drop shadow)**
  - **Sub-bullet**

Body text (24 point with drop shadow)

Code (Min 16 Point with background window)

```
public class BufferTask {
  double bufferDistance = 40;
  String selectLayer;
  . . .
}
```

Closing Statement… (16 point yellow italic)