

**Please!**  
Turn **OFF** cell phones  
and paging devices



# Designing, Deploying, and Using ArcGIS Server Map Caches

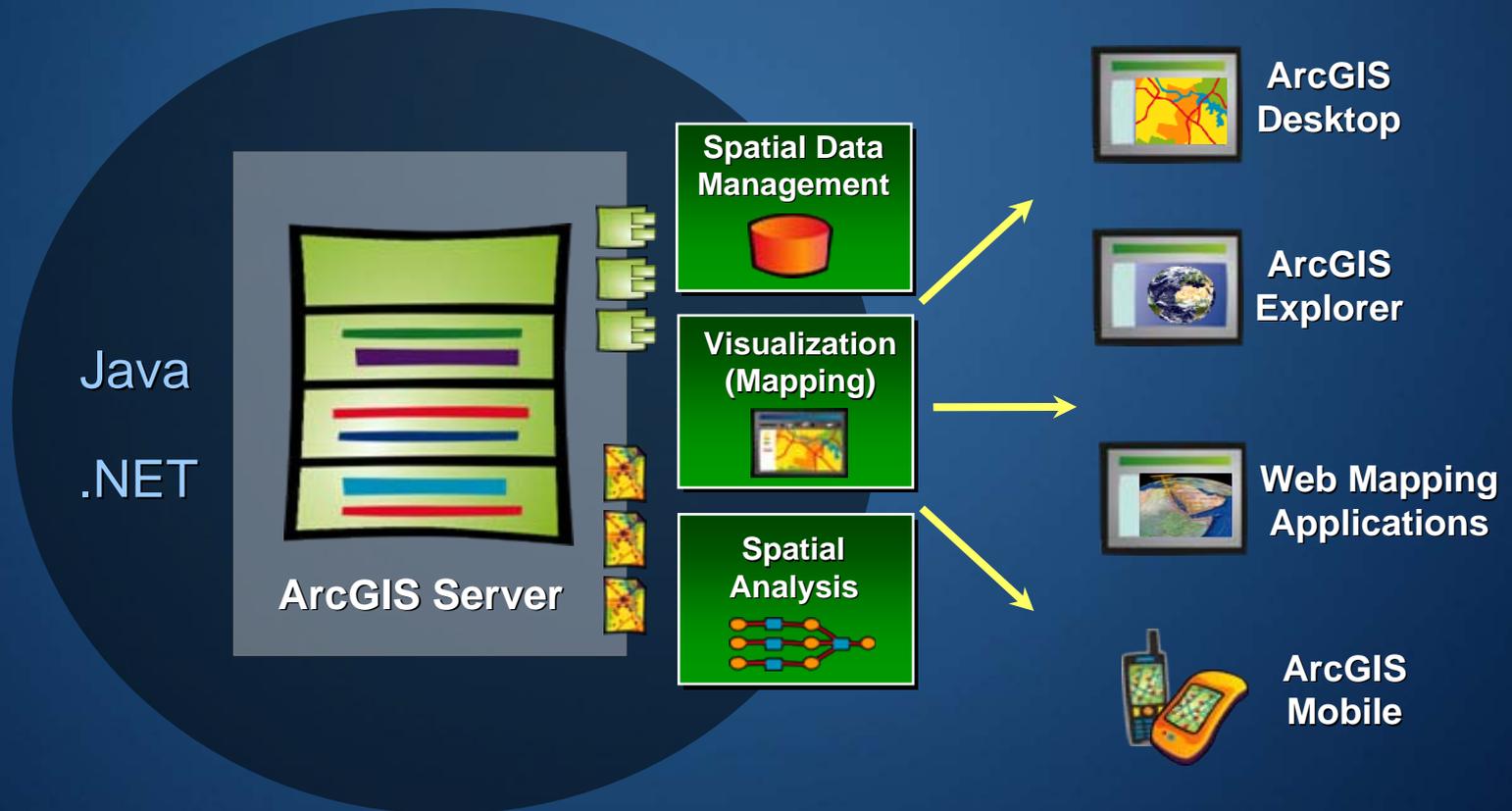
*Jeremy Bartley*  
*[jbartley@esri.com](mailto:jbartley@esri.com)*

*Sterling Quinn*  
*[squinn@esri.com](mailto:squinn@esri.com)*

*ESRI ArcGIS Server Development*

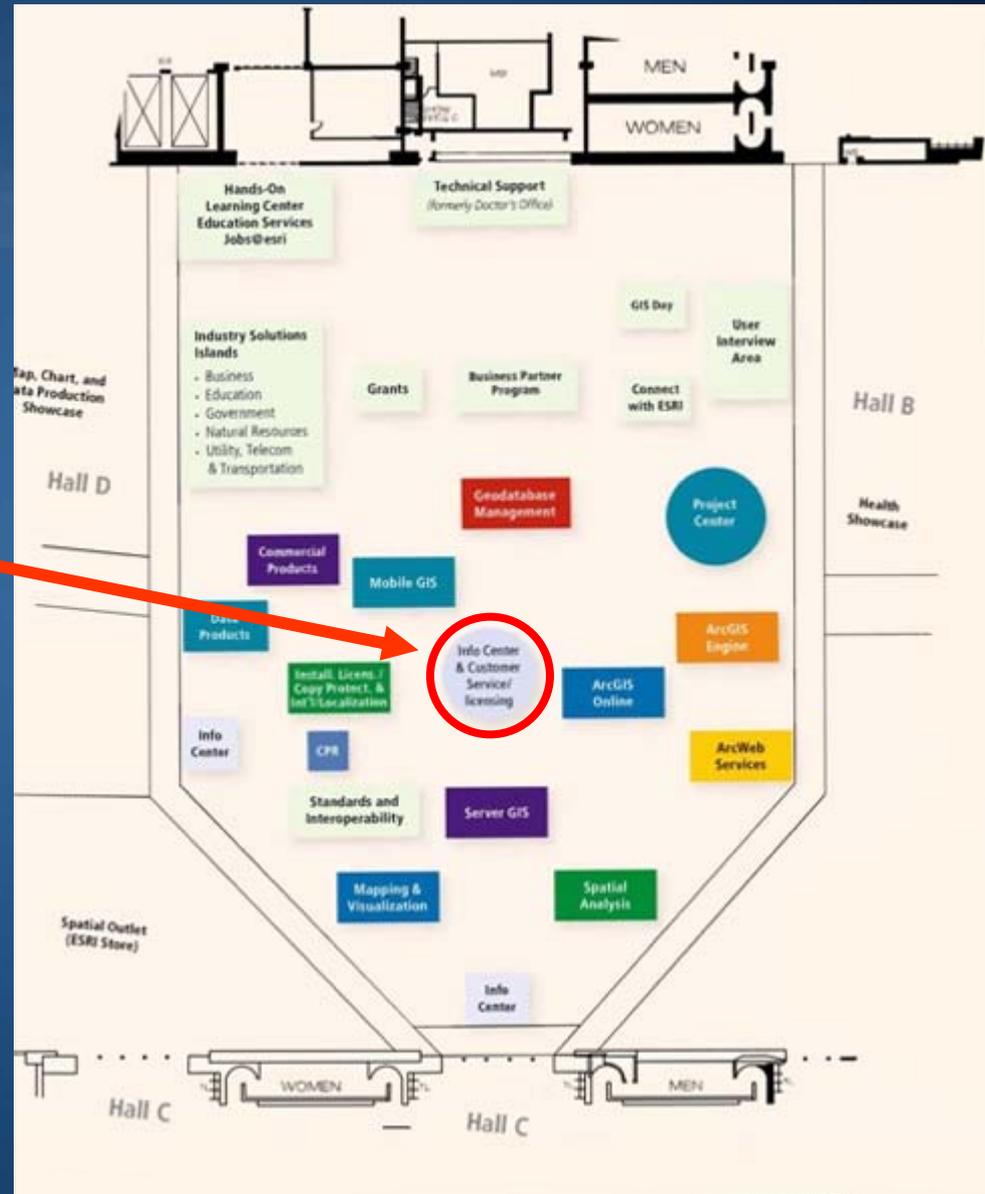
# ArcGIS Server 9.2

- Complete & Integrated server-based GIS
- Out-of-the-box applications and services
- Tremendous developer opportunities



# ArcGIS Server 9.2: Licensing and Migration Questions and Issues

Information Center  
& Customer  
Service/Licensing  
(Yellow Banner)



# What to expect

- This presentation covers basic and advanced topics
- Topics covered:
  - Map caching fundamentals
  - Designing and authoring a map cache
  - Deconstructing the tiling scheme
  - Generating the map cache
  - Using map caches in different clients
    - ArcGIS Desktop
    - ArcGIS Server Web mapping applications
  - Globe caches
  - What's ahead for caching
  - Questions

# Prerequisites

- It's helpful, but not required, that you're already familiar with:
  - ArcGIS Desktop (ArcMap and ArcCatalog)
  - ArcGIS Server deployment
  - Building Web applications with ArcGIS Server

# Map caching fundamentals

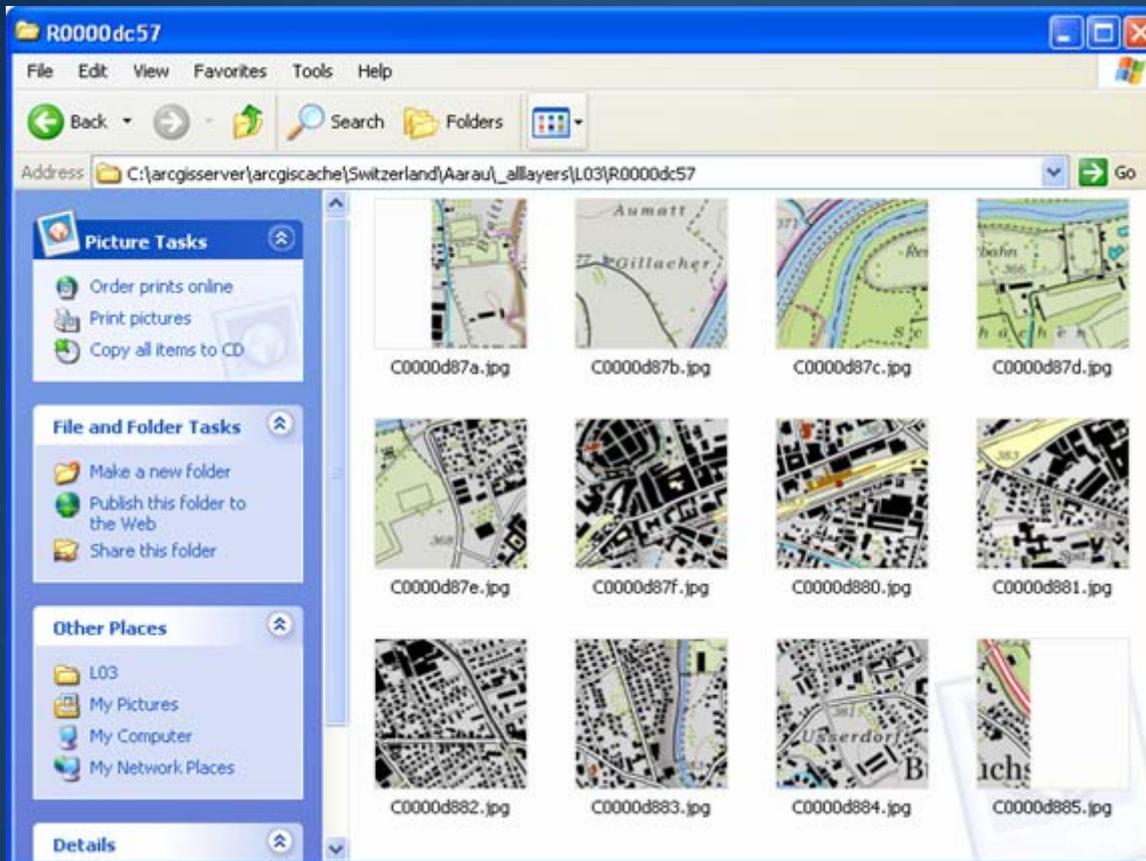
# What is a map cache?

- A map cache is a set of map images that have been pre-rendered for rapid display.



- You create map caches at pre-determined scale levels.

# The cached images are stored on disk



Example of a cached map service

# Why cache a map?

- Which is faster?
  - Let the server draw the map OR
  - Get the map image from a cache
- By caching, you only have to render the map once: When you create the cache.
- Caching is an *investment*.

# Industry Standard

- Google Maps
  - Microsoft Virtual Earth
  - Yahoo! Maps
  - ArcGIS Online
  - others...
- 
- You want **performance, scalability, and high quality!**

# Keep Cartographic Quality and Map Performance with ArcGIS Server cached map services

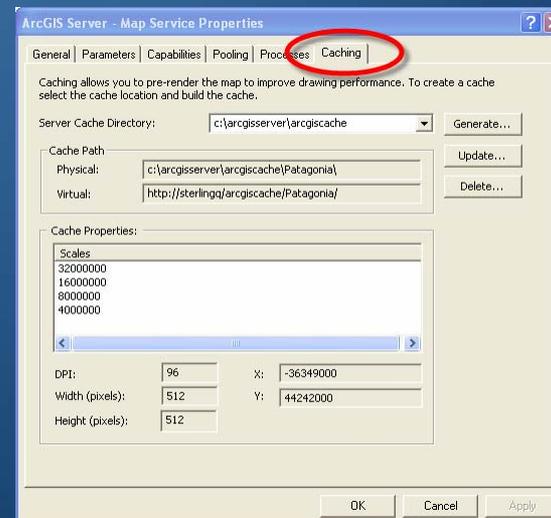
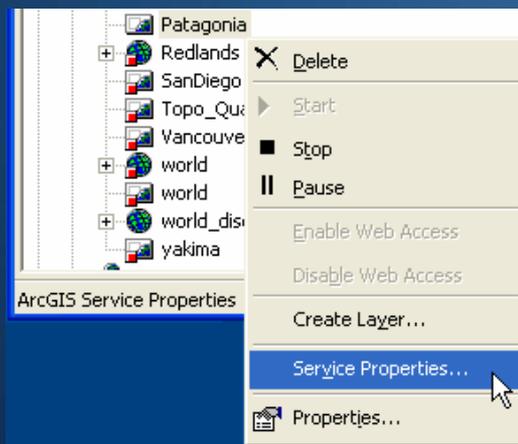


- Shaded Relief
- Transparent Layers
- Maplex Labeling

- Low-res relief
- Solid colors
- Annotation

# How do I create a map cache?

- You can create the cache in ArcCatalog
- You must have an existing map service running
- Use the Caching tab of the Service Properties

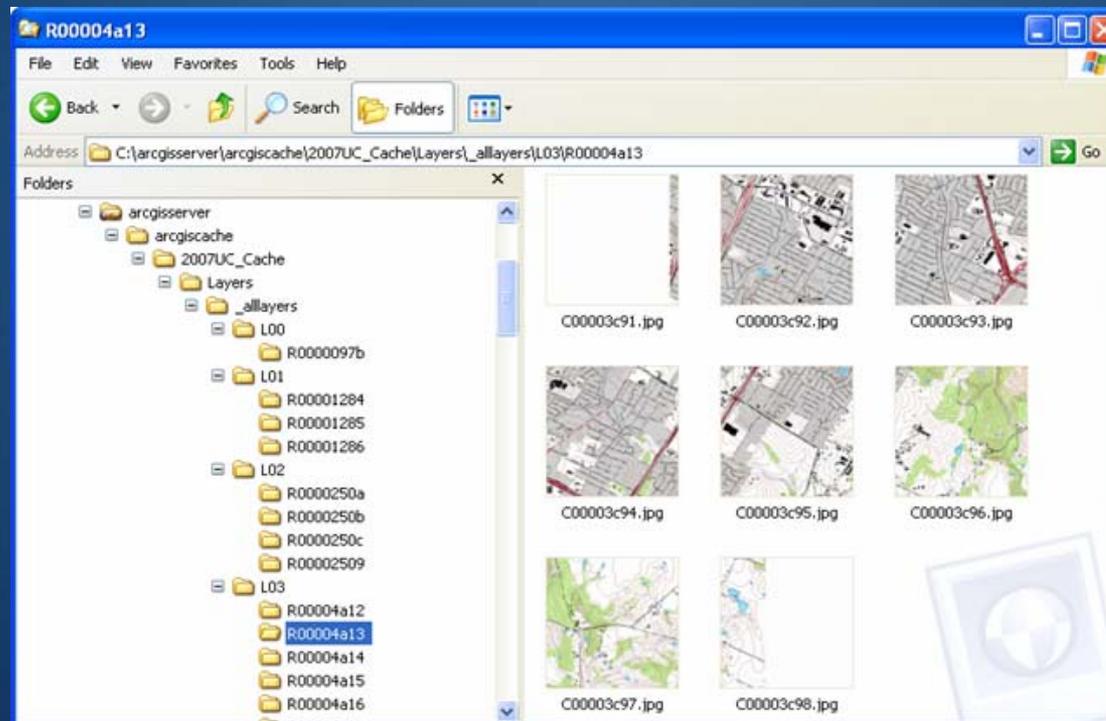


# Demo

- Creating a map cache

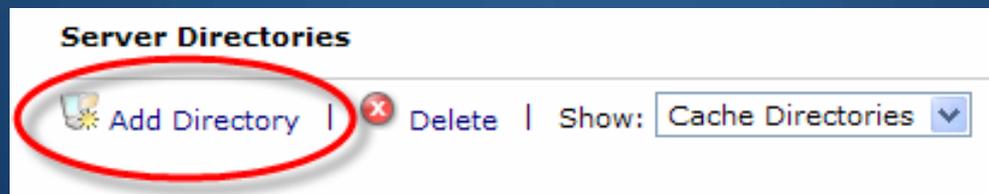
# What happens during caching?

- The server draws the map at all of the scale levels you specified.
- Cached tiles are stored in a folder hierarchy in your server cache directory



# Where is my server cache directory?

- If you installed all of the components of ArcGIS Server on one machine, you get a server cache directory by default.
- This default server cache directory is  
C:\arcgisserver\arcgiscache
- If you installed the components of ArcGIS Server over multiple machines, you must create a server cache directory in ArcCatalog or Manager



# What types of maps should I cache?

- Base maps



- Maps that don't change frequently



- Maps you won't be editing



# What type of cache should I create?

- Fused cache
  - Includes all layers in map in one “fused” image
  - Good performance
  - Can’t toggle layers on and off
  
- Multilayer cache
  - Can choose groups of layers to be cached separately
  - Performance decreases with number of layer groups
  - Can toggle layers on and off

# What type of cache should I create? (continued)

- **Fused** is recommended for most applications
- If you need to turn layers on and off, consider overlaying two fused map services instead of creating a multilayer cache
- ArcGIS Server 9.2 Service Pack 2 contains enhanced support for overlaying caches (More about this later)
- Multilayer caches work best in ArcMap

# Cache functionality matrix

	ArcMap	ArcGlobe	ArcGIS Explorer	Web Mapping Application
Fused 2D cached map service	*****	***	***	*****
Multi-layer 2D cached map service	*****	*	*	**

# Can I still access the underlying data?

- The GIS server still has access to the data for operations such as:
  - Query
  - Identify
  - Selection

# Demo

- Let's look at the cache that we created

# Designing and authoring a map cache

# Before you start

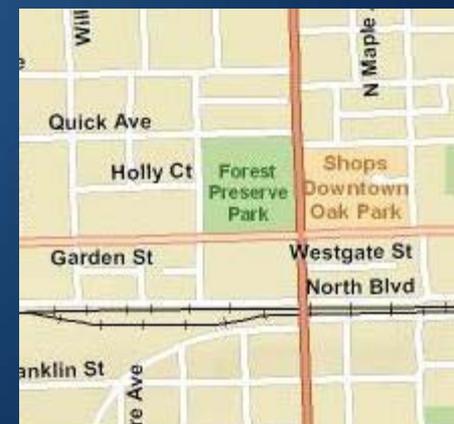
- What tiling scheme will you use?
- What resources will you need to build the cache?
- **Tip:** Use one tiling scheme for all caches in your organization
  - Cache the same map area at the same set of scale levels for all caches.
  - Ensures your fused map caches can be efficiently combined within your end client (ArcGIS Desktop, Web Mapping Application)

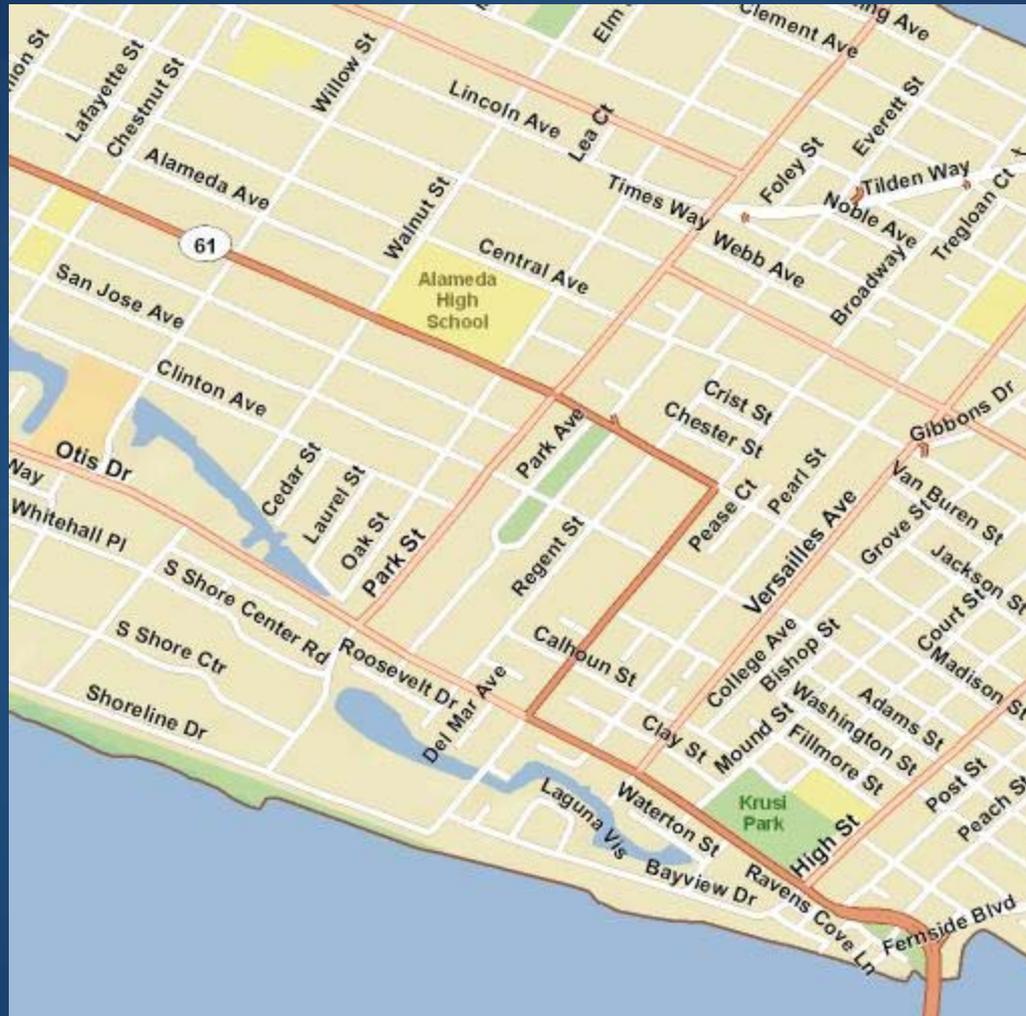
# For one organization...

- One tiling schema
- Identify a set of basemaps
  - Try not to duplicate basemaps across multiple tiling schemes
- Group layers into logical thematic maps
  - Aerial Imagery
  - Hydrography (lakes, streams, ponds, rivers, etc.)
  - Transportation (local roads, main thoroughfares, highways, etc.)
  - Landbase (parcels, building footprints)
  - Elevation
- Best performance will be achieved by blending multiple fused cached map services

# Example...

- You are the GIS Analyst at a county government. Your boss would like you to build a cached map service of your county. She would like you to build this map so that it can be viewable from 1:1,000,000 down to 1:10,000.
- Easy right?



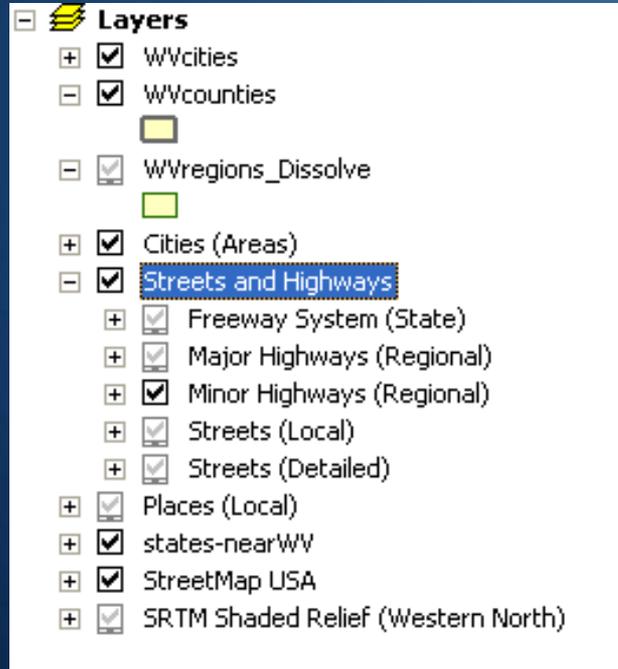


**-122.25, 37.75**

# Cached map design tips

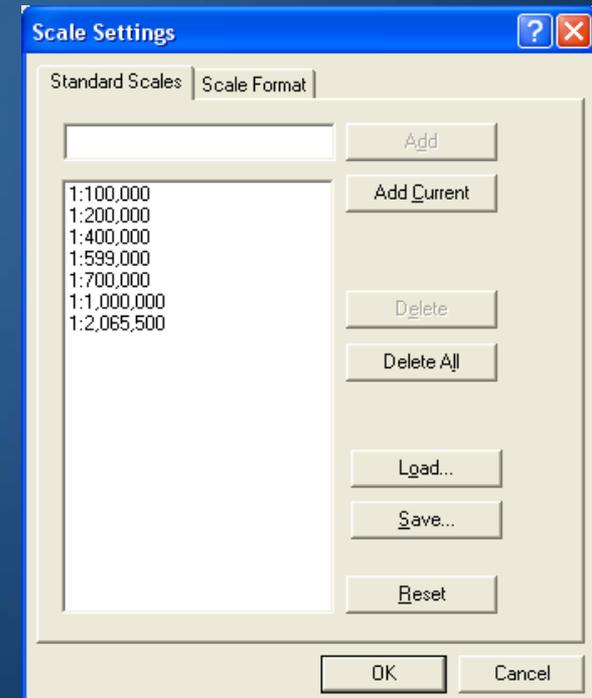
## Layer Visibility

- Each Layer has its own scale properties
- Each layer is visible within an upper and lower scale threshold
- Layers are only visible at appropriate scale
- Together the layers work as one map



## Map Scales (target scales)

- You can define a list of scale thresholds for the entire map
- Use these to test the map at specific scales as if you were to create cache at these scales



# Remember...

- Cached map services must be authored at EACH scale that is to be cached.
- You can ensure that your map is well-designed by applying scale-dependencies to labels and features.
- These scale-dependent changes help the map convey more information in an organized manner.
- The scale dependencies you set for your map layers and the scale levels you choose for your tiling scheme should be carefully coordinated.

# Background Color

- The background color is used to define the transparent part of the image
- Explicitly define the background color
  - If the background color is not defined then the transparent color will be set to 253,253,253
- Use a color not used in the symbology

# Anti-aliasing



- Smooths the edges of labels and lines by blending them with the background.
- The resulting screen display quality can be better than standard rendering in ArcMap.
- Tiles are rendered at finer resolution by down sampling (takes twice as long to cache an area when using anti-aliasing)
- If using anti-aliasing with a cached overlay service, set the background color close to the average color of your basemap
  - <http://serverx.esri.com/antialiasingexamples/>

# Deconstructing the tiling scheme

# The cache configuration file (conf.xml)

- Cache structure is defined by the parameters you enter into the cache generation tool
- This information is stored in a file called conf.xml
- Every cache has a conf.xml
- [http://jbartley/agoDev/arcgis\\_online\\_conf.xml](http://jbartley/agoDev/arcgis_online_conf.xml)

# Two ways to work with the tiling scheme programmatically

- Use ArcObjects or SOAP to access the tile configuration information
  - ITiledMapServer Interface
    - GetTileCacheInfo
    - GetTileImageInfo
- Directly read and parse a cache configuration file (conf.xml).
  - Useful when programming in Python

# Spatial Reference

- All map caches must have a defined spatial reference
- Derived automatically from the data frame being cached

# Tiling Origin

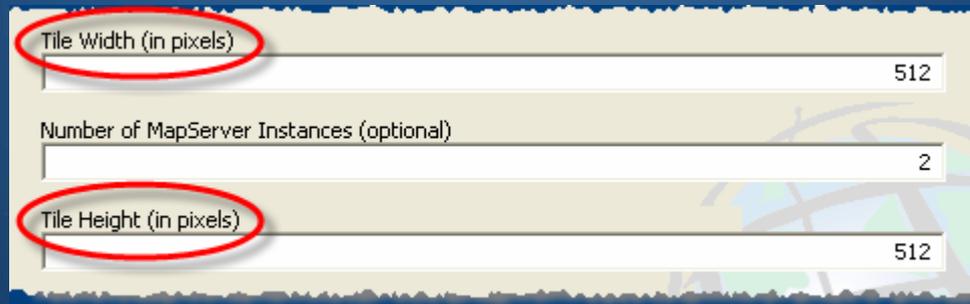
- Specified in map units
- Can be located inside or outside of your data frame extent
  - Tiles will not be created outside the tile origin.

Tiling origin in map units (optional)

X Coordinate	Y Coordinate
<input type="text" value="-118489100"/>	<input type="text" value="142221800"/>

# Tile image properties

- Tile Width
  - Number of pixel columns in an individual tile
  - Appears as TileCols in conf.xml
- Tile Height
  - Number of pixel rows in an individual tile
  - Appears as TileRows in conf.xml



The image shows a screenshot of a configuration form with a light beige background and a dark blue border. The form contains three input fields. The first field is labeled 'Tile Width (in pixels)' and has the value '512' entered. The second field is labeled 'Number of MapServer Instances (optional)' and has the value '2' entered. The third field is labeled 'Tile Height (in pixels)' and has the value '512' entered. The labels 'Tile Width (in pixels)' and 'Tile Height (in pixels)' are circled in red.

Tile Width (in pixels)	512
Number of MapServer Instances (optional)	2
Tile Height (in pixels)	512

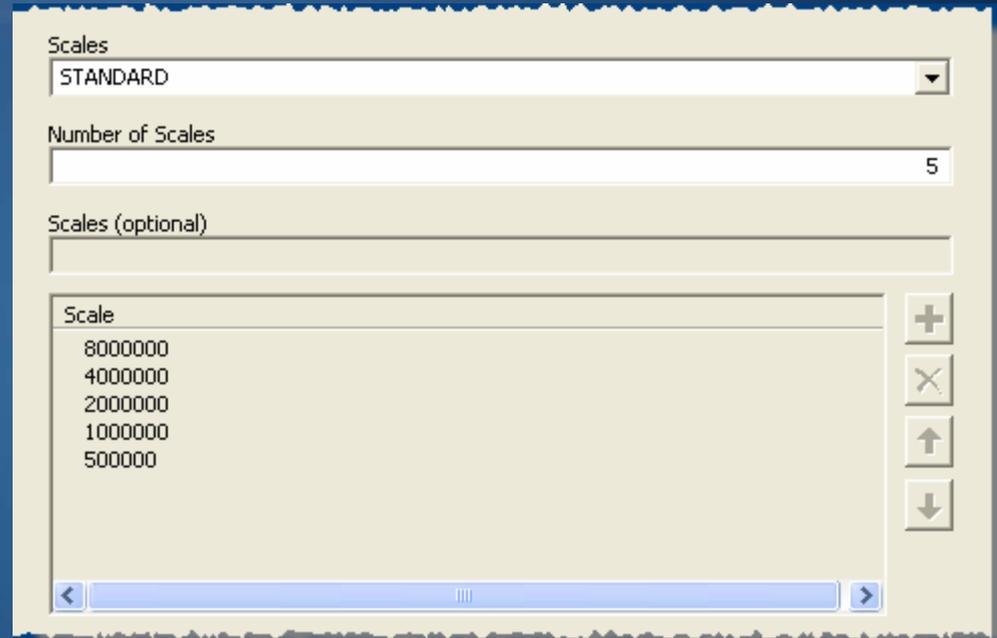
# DPI (Dots per inch)

- Used to compute the resolution of an individual pixel for the cache tile that the server generates.
- If printing map caches you may find it useful to increase the DPI beyond the default (96) to match the output device's DPI setting. This will increase the number of the files in the cache.



# Scales (Levels of Detail)

- Each cache has a set of user-defined scale levels
- Choose scale levels carefully!
  - They determine cache creation time and storage size on disk
  - Web application users will be limited to these scales
- Scales appear as LODInfos in conf.xml (LOD = “Levels of detail”)



# How big will my cache be?

- Lets take a look at an ArcGIS Online Service:
- <http://jbartley/TileCacheInfo>
- Code available from EDN:  
<http://arcscripts.esri.com/details.asp?dbid=15045>

# Cache Tile Format



- PNG8 — a lossless, 8-bit color, image format that uses an indexed color palette
  - Transparency is stored in the color index palette, excellent browser support
- PNG24 — a lossless, three-channel image format that supports large color variations (16 million colors) and has limited support for transparency.
  - Transparency value is stored in the image header. Versions of Internet Explorer less than version 7 do not support this type of transparency.

# Cache Tile Format

- PNG32 — a lossless, four-channel image format that supports large color variations (16 million colors) and transparency.
- JPEG — a lossy, three-channel image format that supports large color variations (16 million colors) but does not support transparency.

# Cache tile format best practices

- Use JPEG for raster-based base maps such as imagery
- Use JPEG or PNG for vector-based base maps such as street maps
- Use PNG8 for overlay services that need to be supported in all browsers

# Generating the map cache

# Generating the map cache

- Can be time consuming
- **Tip:** Create the cache for a small area before building the whole thing.
  - Examine appearance of symbology, labels, etc.
  - Test the performance of the cache in your preferred client application
  - Make necessary adjustments and build again

# Two types of caching scenarios

1. Small scope: Cities, Counties, etc.
2. Large scope: The continental United States down to 1:10,000
  - The geographic extent and the levels of detail (scale levels) of the area of your caching job directly influence the amount time you will need to devote to the caching process

# Caching jobs are defined by...

- Cache tiling scheme
- Full extent of the map service (derived from the data frame full extent in the source MXD)
- For example:
  - Tiling origin is -180, 90
  - Map service extent is the bounding box of Colorado
  - Tiles will only be generated within and around the bounding box of Colorado
    - The tile rows and columns will be referenced from the tile origin (-180, 90)

# Demo

- Setting a custom Full Extent in ArcMap

# Factors that influence cache creation time

- Geographic extent
- Number and choice of scale levels
- Complexity of the map
  - If it takes a long time to draw in ArcMap, it will take a long time to cache.
- Anti-aliasing
  - Generally twice as long to generate
- Data source type
  - Enterprise geodatabase vs. local copies of data
- Server resources
  - For example, SOC machines and available service instances
- Network bandwidth between SOC machine and cache directory

# Factors that influence cache creation time

- Tile size also affects cache creation time
  - Larger size produces fewer tiles
    - Less disk space (block size)
    - Faster creation
    - Easier to manage
  - Smaller size
    - Allows partial update of the display
  - Takes approximately 5X as long and takes up 1Gb more of space when creating a cache at 128x128 tile size versus 512x512 tile size with the same data (Hawaii)
  - In most cases, it's best to keep the default of 512x512

# Impact of scale selection: StreetMap USA

- 48 states
- Cached on 6 dual-CPU servers

Scale	Files	Creation Time
1:500K	4K	2 min
...	...	...
1:64K	0.3M	2 hours
1:32K	1.1M	4.5 hours
1:16K	4.7M	37 hours

Total Size on Disk: 57 GB

# Tips for large scope caching jobs

- When caching very large geographic areas break up caching job to distinct areas.
  - Use UpdateMapServerCache at specific user defined extents in a script environment (Service Pack 2)
- Areas that don't need to be cached should be built using custom extents
  - Alaska, Hawaii, Continental US, but not all scale levels of the Pacific Ocean
  - You can cache the same map service using different full extents as long as your cache tiling schema doesn't change

# Generating the map cache – 9.2 tools

Note: You must manually add Server Tools to ArcToolbox

- **GenerateMapServerTilingScheme**
  - Generates a tiling scheme that can be used to create caches for multiple services
- **GenerateMapServerCache**
  - Generates the cache for a map service
  - Works with either a pre-defined or a newly defined tiling scheme
  - Creates the cache and populates it
- **UpdateMapServerCache**
  - Updates the cache for a map service within a specified extent
  - Creates only missing and empty tiles OR
  - Recreates all tiles
- **DeleteMapServerCache**
  - Deletes the cache for a map service

# Difference between caching tools

- **GenerateMapServerCache**
  - Used to generate a map cache from a map service
  - Generates map tiles for all scales defined in the cache schema
  - Partially checks to see if tiles have been created previously
  - Caches everything within map service full extent at all defined levels of detail
  
- **UpdateMapServerCache**
  - Used to update individual scales and extents
  - Updates can be for all tiles or for missing tiles in given extent
    - When updating missing tiles, performs a check on cache to find missing tiles and tiles of 0k size

# Using map caches in client applications

# Demo

- Using a cached map in ArcMap

# Using map caches in ArcMap

- Gives you a high level of control of how you view the cache
- Continuous zoom
  - Associates current view with a level of detail in the map cache and then resamples up to match the current view
  - Generally uses the next lowest scale compared to the current scale
  - Image quality can vary depending on how far away you are from a cached scale
- Can easily blend with other services and data
- Reprojection supported
  - ArcMap reprojects an existing tile instead of generating a dynamic map. Some performance lost.

# ArcMap creates a local cache

- Can get out of sync with server side cache
- Cache administrator must notify clients to clear their caches when an update is available
- Clear cache in layer Properties
- Located at %temp%\esrimapcache\

# Demo

- Using a cached map in a Web mapping application

# ArcGIS Server Web mapping applications

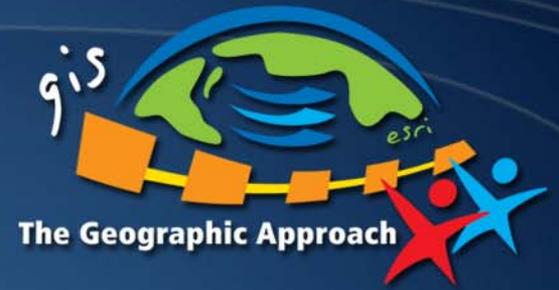
- 9.2 Service Pack 2 *highly recommended*
  - Allows for easier overlay of caches
- To get the performance benefit of the cache:
  - Navigation is limited to the cached scales
  - Cached tiles cannot be projected on the fly
- Differences between Java and .NET implementations
  - Check the ArcGIS Server Help online (<http://webhelp.esri.com>) to see specific instructions for your platform

# Using map caches in 9.2 Service Pack 2

- Criteria to overlay multiple cached services in Java:
  - Projection must be the same
  - Scales that are common to both layers should match
- Criteria to overlay multiple cached services in .NET:
  - Projection must be the same
  - Scales that are common to both layers should match
  - Tile origin and tile size should be identical between services
- Loading a cache configuration file can ensure that scale levels and tile sizes match

# Using map caches in 9.2 Service Pack 2

- Internet Explorer 6 with transparent PNG images.
  - IE 6 and earlier does not honor transparency in PNG24
  - Best practice is to use PNG8 or PNG32
- In .NET Web mapping applications with IE 6
  - If you try to use PNG24 for an overlay cache, the Web application will generate the image dynamically, thereby slowing performance.
- In Java Web mapping applications with IE 6
  - If you try to use PNG24 for an overlay cache, you will not be able to see what's below it.



# Globe Caches

# What is a Globe Service?

- Ability to publish your 3D GIS data to the Web
- Supports visualization, animation, and query
- Selection of two caching models:
  - Just in time (on-demand)
  - Pre-cached

# Who can author Globe services?

- Must have ArcGIS Server 9.2
- Globe Services are available in both ArcGIS Server Standard and Advanced editions
- ArcGIS Desktop with the 3D Analyst Extension is required to author globe documents

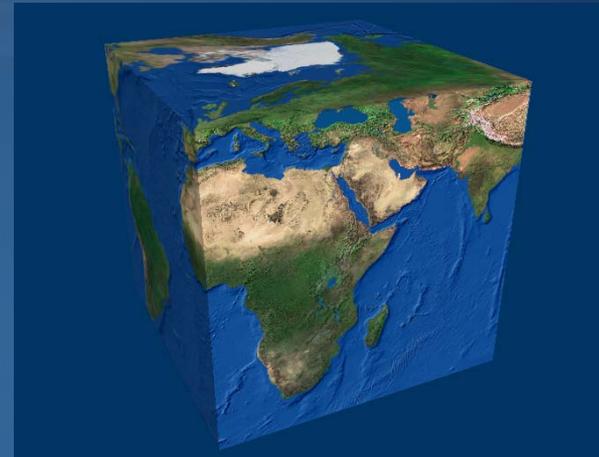
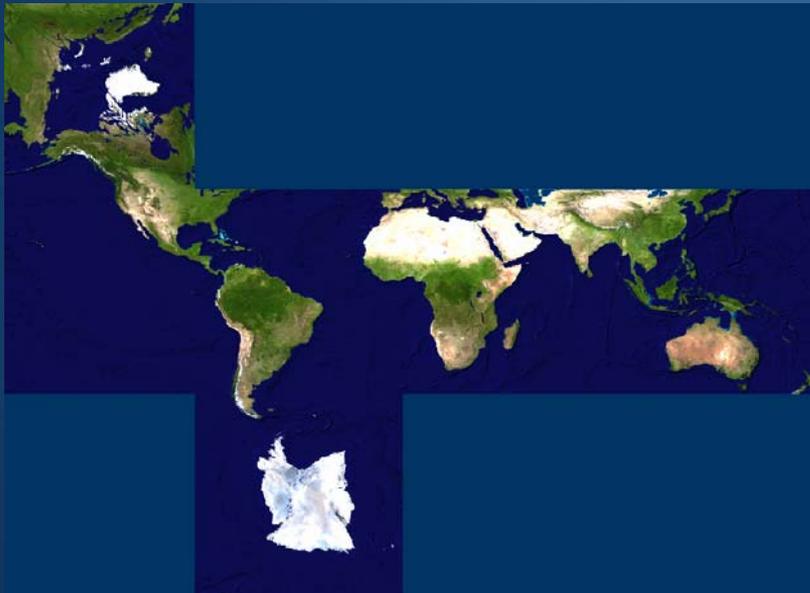
# How to build 3D caches-- Generate Data Cache

- Use ArcCatalog (Generate Globe Server Cache tool) to cache these data types:
  - Rasters as Imagery
  - Rasters as Elevation
  - Vector data as rasterized 2D feature layer
  - Web based data such as Map Service layers, IMS, WMS...
- Use ArcGlobe to cache the following data types:
  - Vector Data as non-rasterized 2D feature layer
  - 3D feature layers (multipatches and extruded feature layers)
  - Point and Line feature layers symbolized by 3d symbols
  - Graphics Layers

# Globe display management

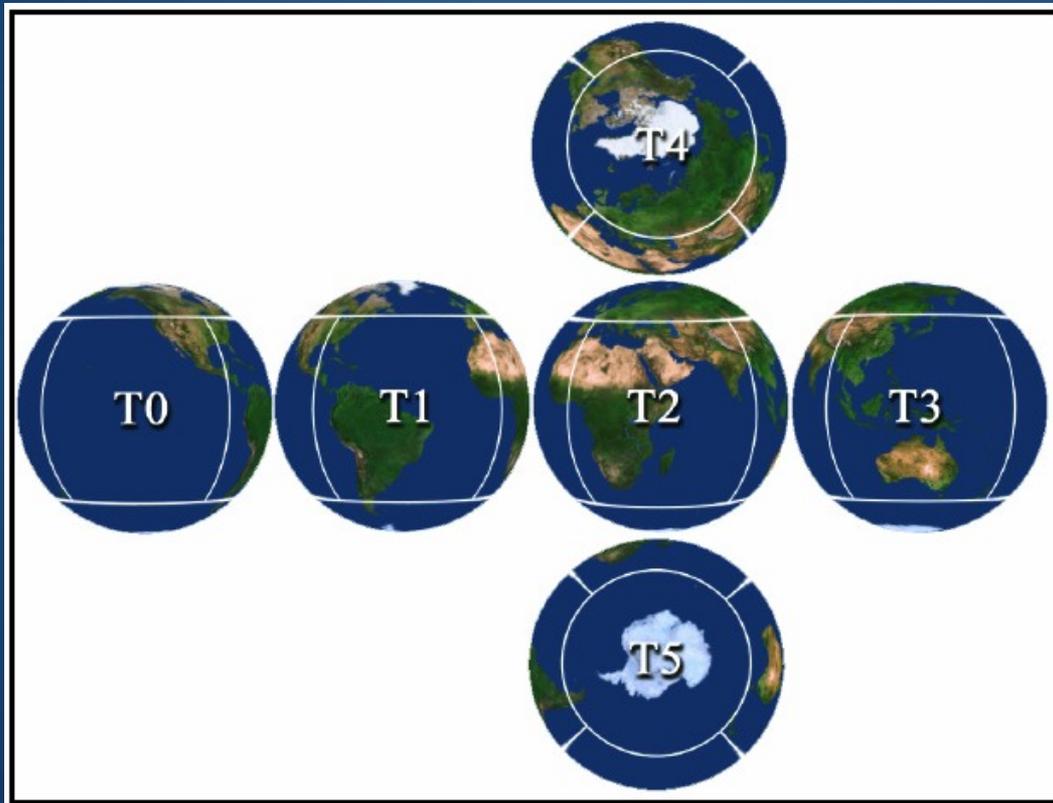
---

- ArcGlobe internally uses the Cube Projection to organize data display



# The cache structure

- This results in a cache structure of six faces which is different from a map cache structure.



# Cache structure facts

- All tiles have same size (images 512; elevation 256)
- Simple indexing: (face, level, row, column)
- Has a fixed custom 'spatial reference' (Cube projection)
- Can be consumed by ArcGlobe and ArcGIS Explorer

# Cache functionality matrix

	ArcMap	ArcGlobe	ArcGIS Explorer	Web ADF
Fused 2D cached map service	*****	****	***	*****
Multi-layer 2D cached map service	*****	*	*	**
Globe cache	X	*****	*****	X

# Using existing map caches to create globe caches

- Allows you to create efficient Globe Caches (in creation and delivery)
- Superior Symbology/Labeling
  - Unless you need 3D labels (billboard display)
- See the ArcGIS Server Development Blog for detailed instructions
  - <http://blogs.esri.com/Dev/blogs/arcgisserver/archive/2007/05/30/Creating-a-3D-globe-cache-from-a-2D-map-cache.aspx>

# What's ahead for caching

- Easy workflow to define a cache
  - ArcGIS Manager and ArcGIS Desktop
  
- Support for on-demand map caching
  - Build your cache dynamically

# Cache on demand

## How We Watch the City: Popularity and Online Maps

Danyel Fisher  
Microsoft Research  
1 Microsoft Way, Redmond, WA  
danyelf@microsoft.com

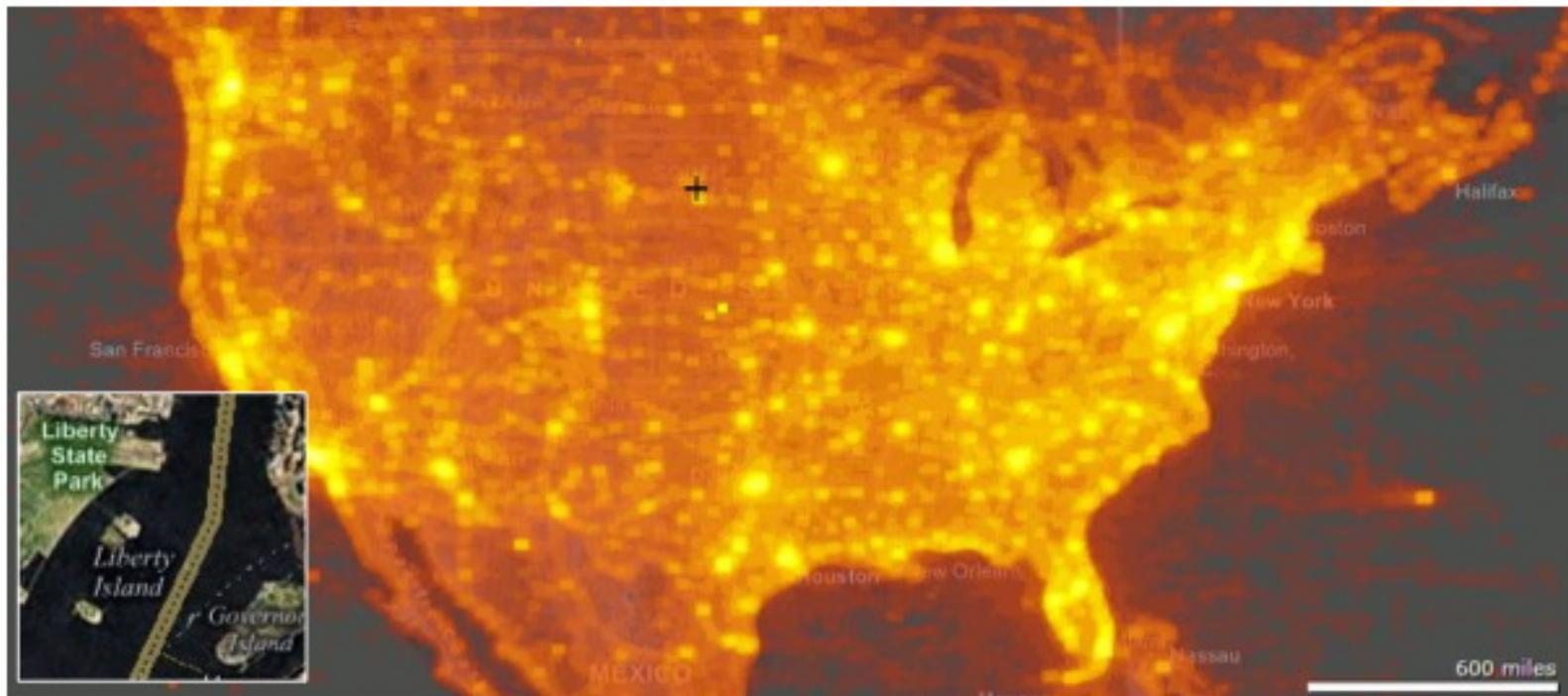


Figure 1: A heat map of popularity of tiles over the US at level 12 (approximately 40 meters per pixel scale). The brightest points have on the order of half a million hits, while the dimmest visible points show closer to a thousand hits. Note that usage patterns at this scale seem to follow population. Inset: mapping imagery at the level (but not size) represented by one pixel.

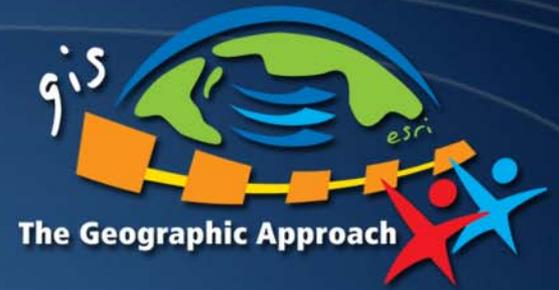
# What's ahead for caching

- Options to build the map cache for an area of interest
  - Pre cache only highly populated areas
  - Cache the rest of the area on the fly

# Summary

- Map caching fundamentals
- Designing and authoring a map cache
- Deconstructing the tiling scheme
- Generating the map cache
- Using map caches in client applications
  - ArcGIS Desktop
  - ArcGIS Server Web ADF
- Globe caches
- What's ahead for caching

# Questions?



# Map and Globe Caches

*Thanks!*

*Jeremy Bartley*  
*[jbartley@esri.com](mailto:jbartley@esri.com)*

*Sterling Quinn*  
*[squinn@esri.com](mailto:squinn@esri.com)*

*ESRI ArcGIS Server Development*