

# Flex and Map Caching

## A Combination for Faster Web Maps



**City of Phoenix**

**Gady Pitaru**

*IT Analyst/Programmer II*

[gady.pitaru@phoenix.gov](mailto:gady.pitaru@phoenix.gov)

July 13, 2010



# Background

- Changes in user expectations

Google maps

bing

YAHOO! LOCAL

- Goal: Replace ArcIMS and desktop GIS applications with fast web mapping apps
- Staff
  - Analyst/Programmer II
  - Analyst/Programmer I
  - Senior GIS Tech & 2 GIS Techs

# Introduction

- Street Transportation's **EZ Map**
  - ArcGIS Server & Adobe Flex
  - Replaces primary ArcIMS viewer application
  
- Stages of Development
  - User Needs Assessment
  - Authoring & Caching Map Services
  - Application Development with Flex
  - Deployment



# Caching Map Services

- Slowest performing and least-frequently updated Map Services
- Tiling scheme – 11 scales
  - Optimal symbology & labeling
  - Design at each scale
- 10-14 hours for basemaps
  - Scheduled on weekends via Python script
  - Caching by updated quarter-sections (daily)



# Planning for Caching

## ■ Tiling Scheme

- Predefined scales where map is viewed
  - Design map at each scale
- Bottom scale uses 75% of storage
- If using Google/Bing Maps...
  - Use their tiling scheme
  - Use web mercator projection
- Experiment with what works best
  - Tested on sample area

1:500,000  
1:128,000  
1:64,000  
1:32,000  
1:16,000  
1:8,000  
1:4,000  
1:2,000  
1:1,500  
1:1,000  
1:500



# Caching Aerial Photography

- One layer in MXD
  - Raster source stored in SDE geodatabase
  - Displayed when zoomed in beyond 1:8,200
- Cached as PNG24 image files
  - Better quality, but larger file size than JPG
  - No difference in quality & smaller than PNG32
  - PNG24 transparency issues in IE6



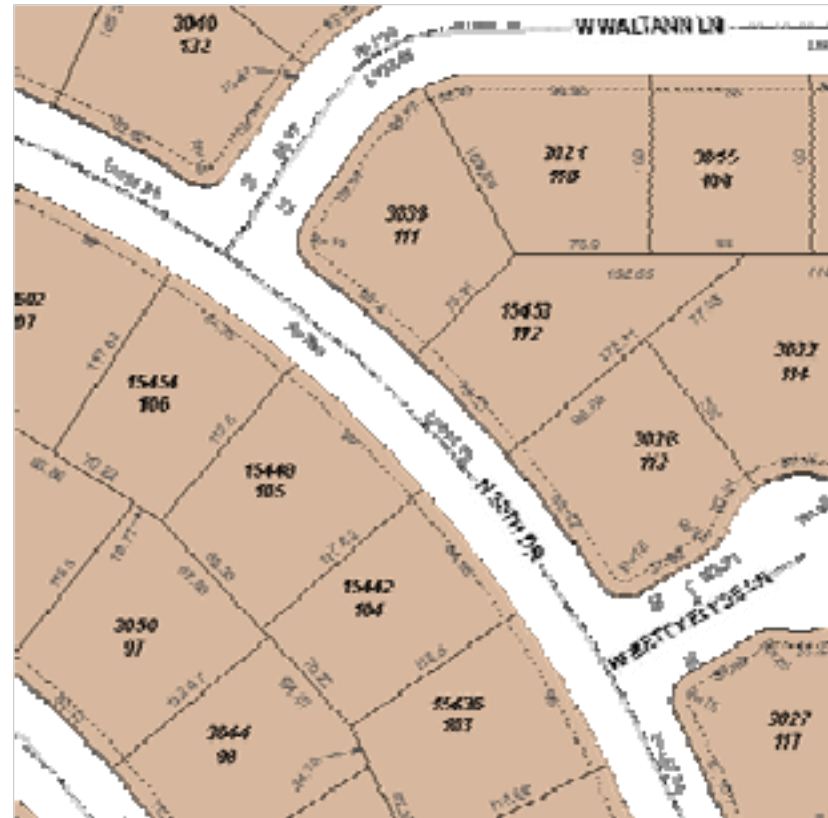
# Basemaps

Designed for use with Aerials



Displayed at all scales.  
City Limits filled above 1:8,000.

Designed for "Engineering" view



Displayed at 1:8,000 and below.

# Basemaps

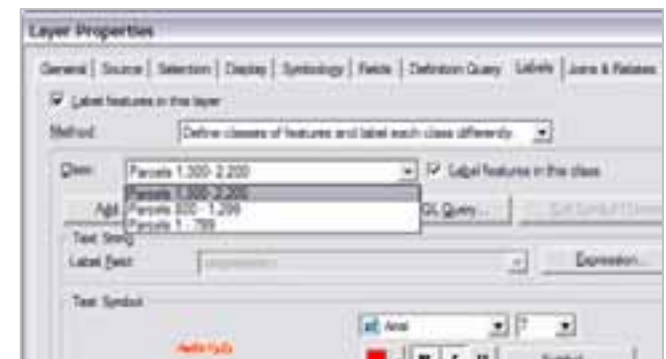
## ■ Layers

- Not grouped by scale
- Visible scale ranges
  - Match tiling scheme scales + 200



## ■ Labels

- Classes for each scale range
  - Visible scale range
  - Sizes/symbology
- Used Maplex engine
  - Cannot utilize MSD files (until 9.4)





# Basemaps

## ■ Anti-aliasing

- Background color of Data Frame must be considered
- Used average RGB pixel values of Aerials



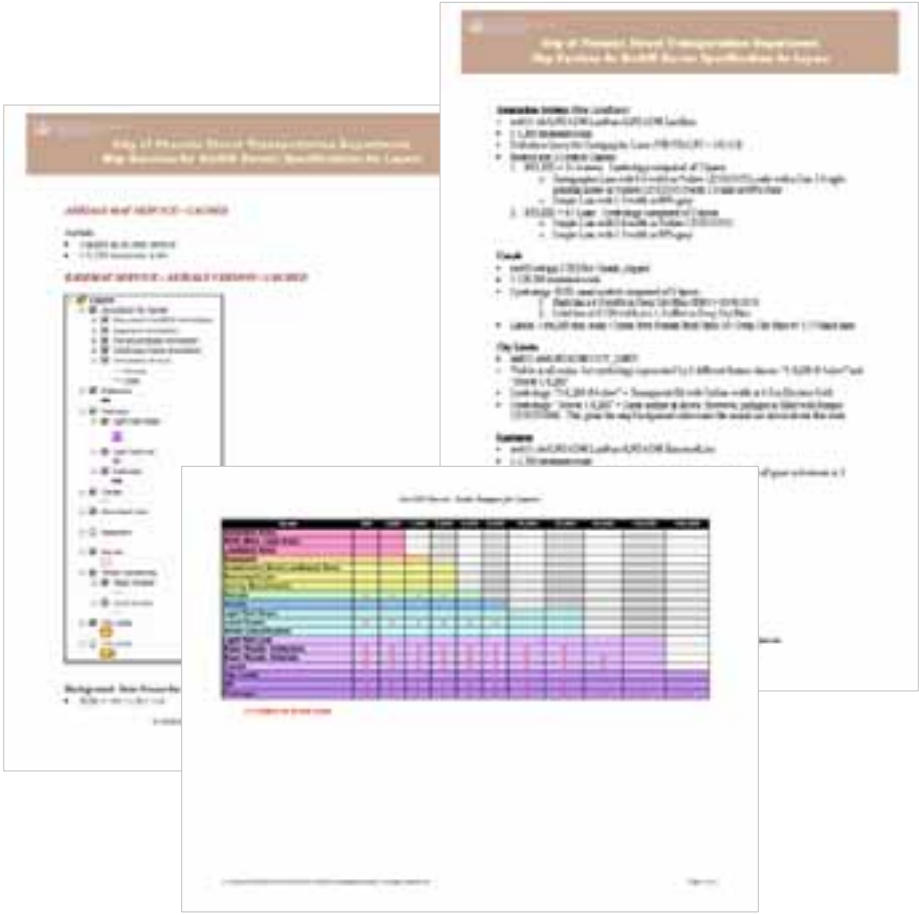


# Stormwater Management

- Storm Drain System & Floodplain data
  - Good candidates for caching
    - Storm Drain data is static
    - Annual reload of Floodplain data
    - Complex geometries

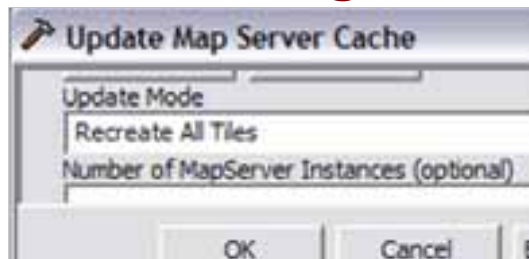
# Documentation

- Documented all layers and properties for each Map Service
  - Data sources
  - Visible scale ranges
  - Feature symbology
  - Label styles
  - Definition queries



# Caching Methods at 9.2

- Recreate All Tiles using Geoprocessing tool



- Windows Scheduler using Python script

```
# Load required toolboxes...
gp.AddToolbox("c:/Program Files/ArcGIS/ServerTools/Toolboxes/Server Tools.tbx")
message("Loaded toolboxes...")

# Local variables...
Output_Base = ""
Output_Map_Server = ""
message("Declared locals...")
message("")

# Process Manage Map Server Cache Tiles...
message("Cleaning Map Server Cache Tiles started for EsriMapServer...")
message("")
gp.ManageMapServerCacheTiles_server("arcgis", "EsriMapServer", "Layers", "None")
message(gp.GetMessages())
message("Map Server Cache Tiles cleaned for EsriMapServer.")
message("")

# Print out time
message("Time " + TIME.strftime("%H %M"))

# Save/Close log file
logfile.close()

#####

# Get messages on error
message(gp.GetMessages())
message("")
message("Time " + TIME.strftime("%H %M"))
logfile.close()
```

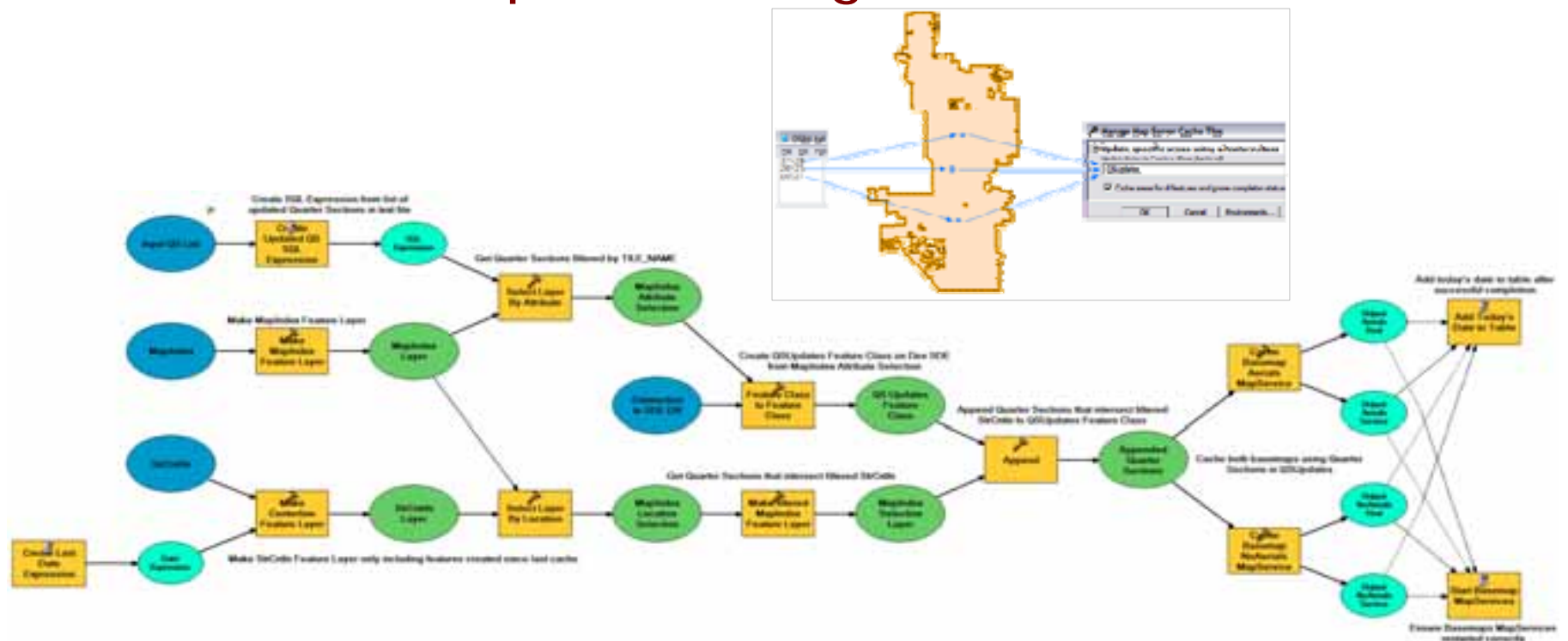


# Caching Methods at 9.2

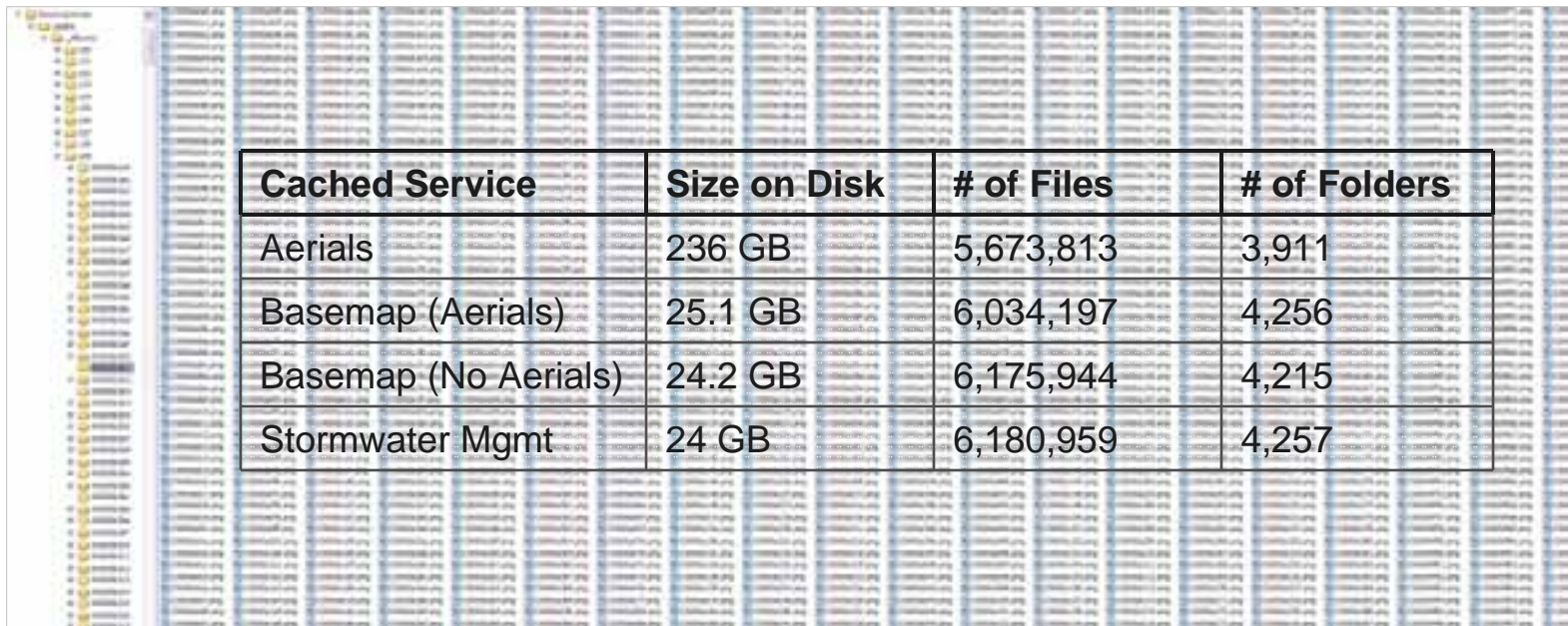


# Caching by Quarter Section

- Update cache from feature extents at 9.3
  - Recreate cache tiles only in areas with landbase updates using QS as reference



# Storage Requirements



Cached Service	Size on Disk	# of Files	# of Folders
Aerials	236 GB	5,673,813	3,911
Basemap (Aerials)	25.1 GB	6,034,197	4,256
Basemap (No Aerials)	24.2 GB	6,175,944	4,215
Stormwater Mgmt	24 GB	6,180,959	4,257



# Application Development

- Finding the right tool for the job
  - .NET Web ADF
    - Out-of-the-box template
    - Too robust for our use
    - Customization cumbersome
    - ASP.NET, VB/C#.NET, HTML, JavaScript, CSS
  - JavaScript API
    - Performed well
    - Scalable
    - Spend too much time on web design
    - JavaScript, HTML, CSS





# Adobe Flex

## ■ Flex API

- Fastest performing
- Very scalable
- Minimal time spent on web design
  - Flex apps are inherently beautiful
- Ease of development
- ActionScript, MXML
- Flash Player (9+) required on client
- Adobe FlexBuilder IDE or 3<sup>rd</sup>-party



# User Interface Design

- Keep it simple
  - Docked toolbar
  - Categorized layers list
- Exploit Flexy look & feel
  - Drag-able & animated windows
  - Clean, sleek, smooth experience
- Intuitive & familiar
  - Keep user expectations in mind
    - Google/Bing Maps, etc.



# GIS & Mapping Functionality

- Toggling layer visibility
  
- Only provide the necessary tools
  - Zoom to intersection, address, etc.
  - Identify features
  - Google Street View
  - Measure lines & areas
  - Add graphics & text
  - Export map to various formats
  - Bookmarks
  - Report Issues



# Deployment

- User Committee testing & feedback
  - Within the month prior to going live
- Hyperlink on Department's intranet webpage
- Replace hyperlinks in web reports
- Department-wide training
  - 2 sessions/week at various advertised locations



# Results

- 220 unique visitors in first 3 weeks
  - About 44% of Department
- Averaging almost 90 visitors/day
  - More than 3x ArcIMS app visitors/day
- Positive user feedback
- Replaced ArcGIS Desktop installations

# If It Ain't Broke...

- Redesigned for Aviation Department
- "Bypass" ArcGIS Server for database calls





# Thank You

Questions?

[gady.pitaru@phoenix.gov](mailto:gady.pitaru@phoenix.gov)