



Advanced tips & tricks for building powerful iOS/Mac apps

Divesh Goyal
Mark Dostal



Overview

- **Tips and tricks any developer can use to make great apps**
- **All code written in Swift**
- **Uses the ArcGIS Runtime for iOS/OS X SDK version 10.2.4**
- **Demo app written for iOS, but can be used on OS X as well**
- **Demo app's GitHub repository here:**
 - <https://github.com/Esri/tips-and-tricks-ios>

Why Swift?

- **The “...new language for the future of Apple software development” ***
- **Fast**
- **Modern**
- **Safe**
- **Interactive**



Tips and Tricks...

Geometry

Configure Location Display

Custom Map Data

Map Options

Networking

Miscellaneous



Geometry

1. Coordinate Conversion

44 56 25.847N 093 17 55.127W

4Q 612341 2356781

34.0547 N, 117.1825 W

- Show coordinates in a user-friendly notation, instead of as x, y
- Easily retrieve a coordinate string from an `AGSPoint` or create an `AGSPoint` from an existing coordinate string
 - `degreesMinutesSecondsStringWithNumDigits`
 - `init!(fromDegreesMinutesSecondsString degreesMinutesSeconds,...)`
- 8 different coordinate notations supported in `AGSPoint_CoordinateConversion`
 - MGRS, DMS, DD, DDM, GARS, UTM, GEOREF, USNG

2. Geodesic Operations

- For when you remember the Earth is round... true distance or flight paths between locations
- Allows for geometry manipulation through geodetic calculations
- Part of the **AGSGeometryEngine** class
 - **geodesicDistanceBetweenPoint1**
 - **geodesicDensifyGeometry**
- Methods to create, modify, and query geometries

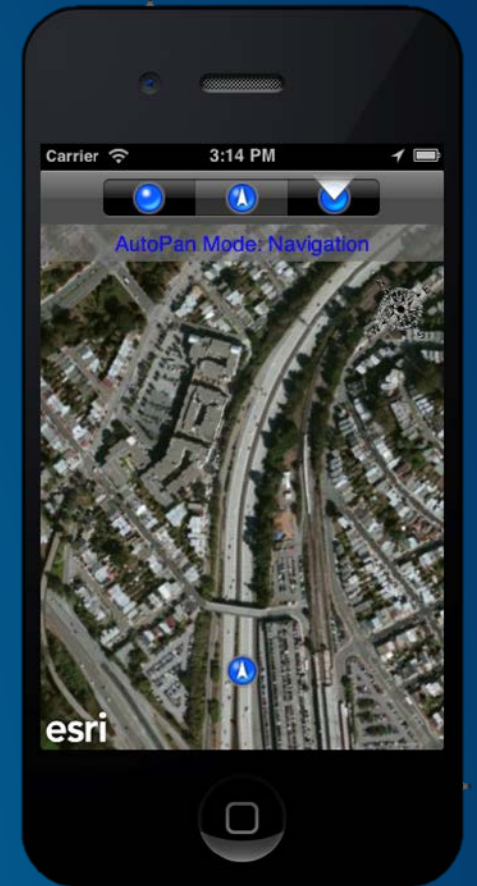


Configuring Location Display



3. Pluggable location datasources

- Use on-board GPS (default)
 - `AGSCLLocationManagerLocationDisplayDataSource`
- Simulate location updates for demo or testing
 - Using GPX Track – `AGSGPXLocationDisplayDataSource`
 - Using polyline geometry – `AGSSimulatedLocationDisplayDataSource`
- Plug in a custom datasource
 - High accuracy external GPS receivers
 - Adopt `<AGSLocationDisplayDataSource>` protocol



4. Customize location symbology

- Match application theme or style
- Indicate changing conditions
 - Moving in/out of geofences
- AGSLocationDisplay properties
 - defaultSymbol
 - headingSymbol
 - courseSymbol
 - accuracySymbol, pingSymbol



- Only Position
- ◒ Position & Heading
- Ⓜ Position & Course

Custom Map Data

5. Custom Dynamic Layer

- We may not understand all your data formats or all your rendering options
- A custom `AGSDynamicLayer` class can display custom data, such as a heat map based on USGS earthquake data
- Class derives from `AGSDynamicLayer` and overrides only a handful of methods
 - `fullEnvelope`
 - `initialEnvelope`
 - `spatialReference`
 - `requestImage(...)`



6. Web Tiled Layer

- Add and display tiles from any provider on the web
- Simply create `AGSWebTiledLayer` using 3rd Party URL scheme and add it to a map
 - `init!(templateURL: String!, ...)`



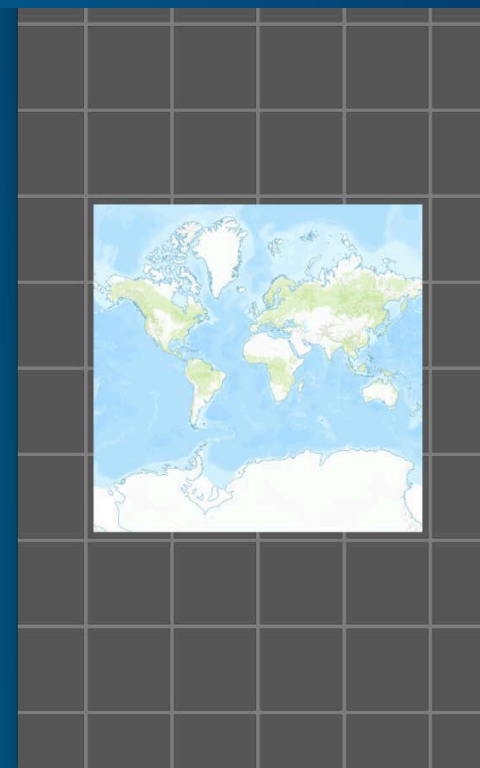
Map Display Options

7. Customize Map Background

- Match basemap
- Match application color theme

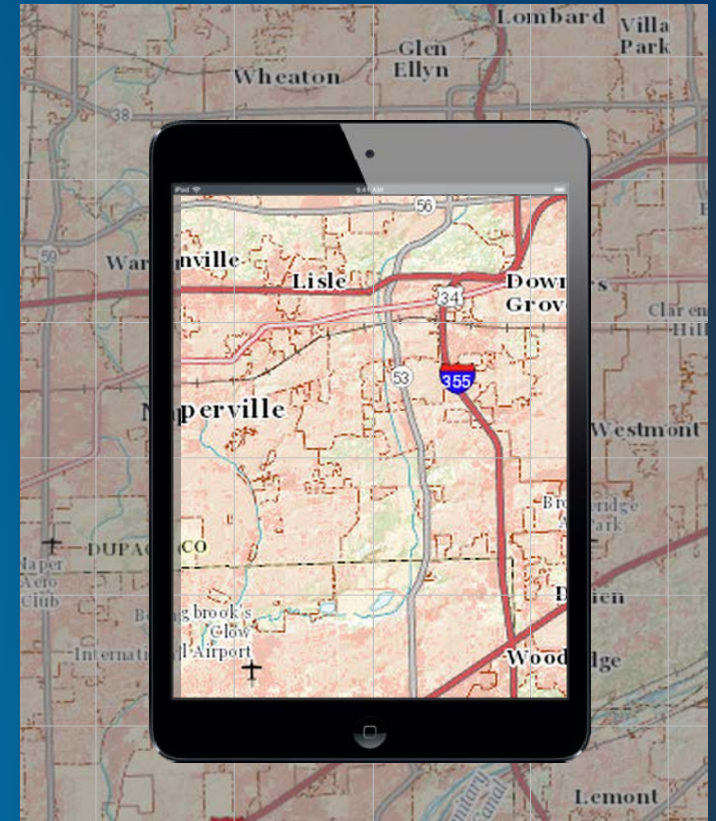
- **AGSMapView**

- backgroundColor
- gridLineColor
- gridLineWidth
- gridSize



8. Prefetch neighboring tiles

- To prevent seeing the background when user pans
- **AGSTiledLayer**
 - **bufferFactor**
 - Value between 1 & 2;

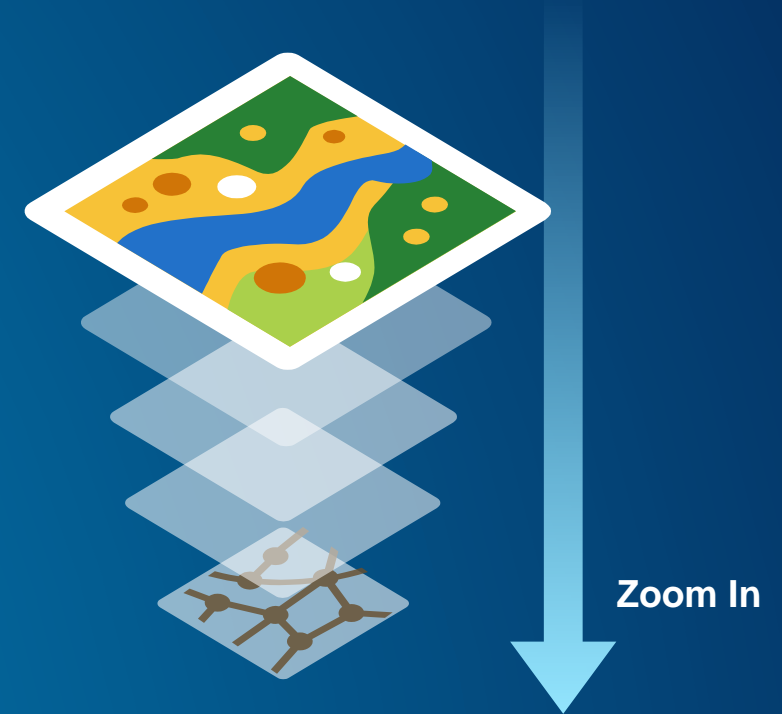


9. Constrain map bounds

- **Restrict the map to certain geographic bounds**
 - To focus on a city/county even when basemap covers entire world
- **AGSMapView**
 - `maxEnvelope`
 - Defaults to basemap layer's full envelope

10. Expand map scale range

- Allow zooming-in beyond the basemap
 - To see full resolution of operational data
- AGSMapView
 - `maxScale`
 - Defaults to basemap layer's scale range



11. Generate map snapshot

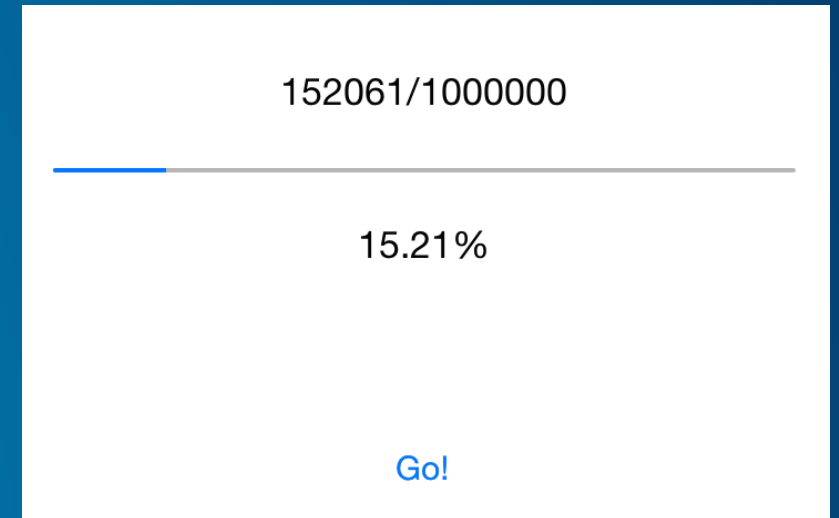
- Save to photos, post to social network, print, ...
- Use as placeholder for animation, image effects, ...

- **UIView**
 - `drawViewHierarchyInRect:afterScreenUpdates:`

Networking

12. Request Operation - Download file

- Download any file for use locally on the device.
- Image, data, text, etc.
- **AGSRequestOperation**
 - `init!(URL url: NSURL!)`



13. Request Operation - Invoke Web Service

- More dynamic access to data than a single file
- Example: using a web service to find weather data for a single map location
 - <http://api.openweathermap.org/data/2.5/weather?lat=33.815995&lon=-116.518781>

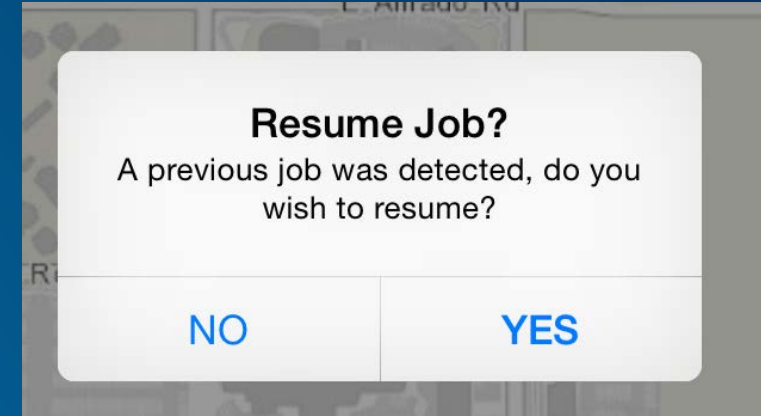
- AGSJSONRequestOperation

```
{  
  "main": {  
    "temp": 299.52,  
    "pressure": 1012,  
    "humidity": 9,  
  },  
  "name": "Palm Springs",  
}
```



14. Resume Jobs

- Network operations like generating a TPK can take a long time. If the user leaves the app or the app gets killed before the job finishes, you can resume the existing job instead of starting a new one
- The `resumeID` for the started job is stored in `NSUserDefaults` in case the app is killed and then retrieved upon app restart
- `AGSEExportTileCacheTask`
 - `exportTileCacheWithParameters`
 - `exportTileCacheWithResumeID`



15. Background Fetch

- For time-intensive tasks, allow the app to be notified if a download completes while in the background
- Code is similar to the code in 14. Resume Jobs but we've added a 'BackgroundHelper' class to handle background notifications



TipsAndTricks_DevSummit2015 now
Tile cache downloaded.

Miscellaneous

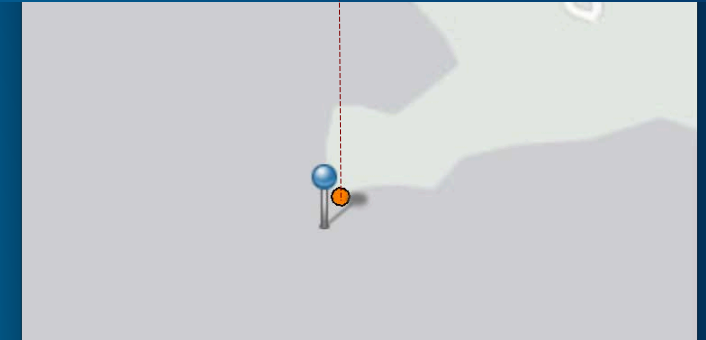
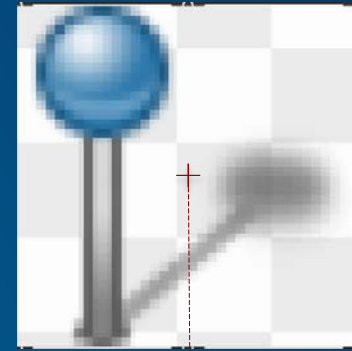
16. Choose graphics layer rendering mode

- **Static Mode (Default)**
 - Better suited for displaying lot of (1000's) graphics
 - U/X
 - Map remains responsive even under load
 - But, graphics appear fuzzy temporarily when zooming in
 - CPU based

- **Dynamic Mode**
 - Better suited for displaying a few (100's), rapidly changing graphics
 - U/X
 - Graphics remain crisp when zooming in/out
 - But, map becomes sluggish under load
 - GPU based

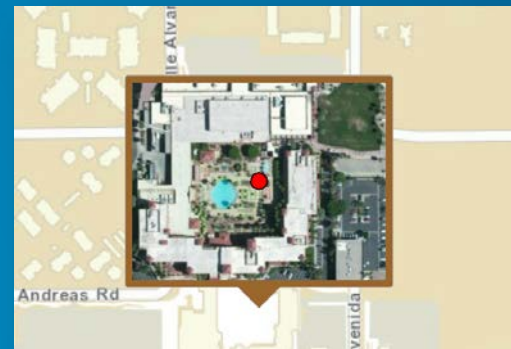
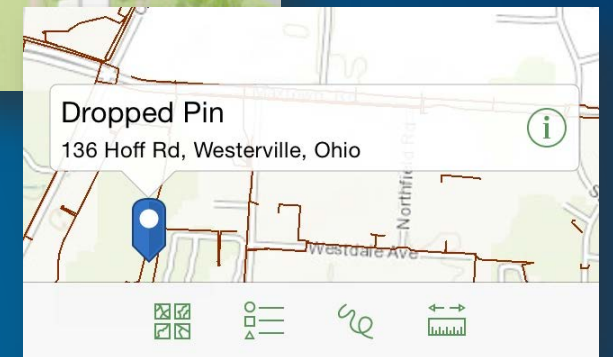
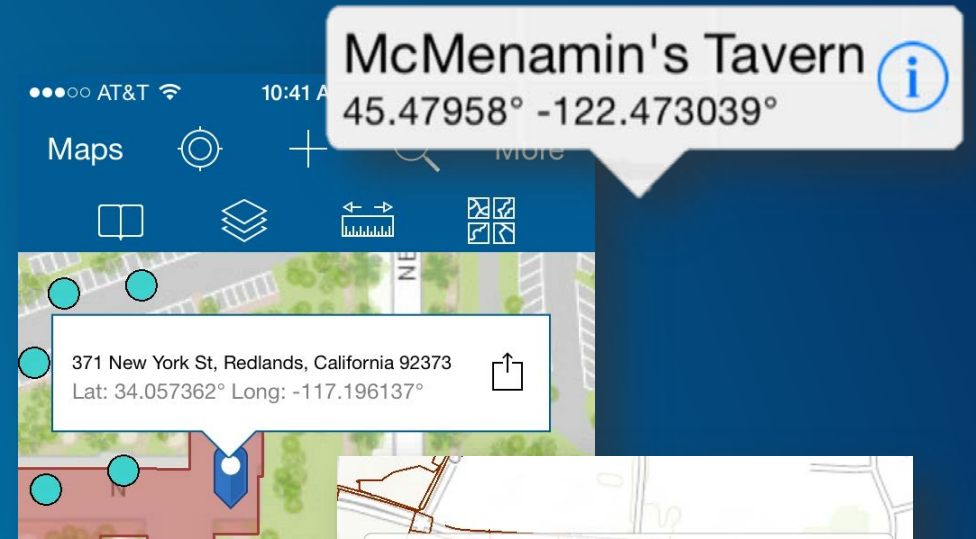
17. Fine-Tune symbol alignment

- Align contents of the picture with geographic location
 - By default, the picture and callout are centered on the location
 - But you may want to offset the picture and the callout leader line to better align with the location
- AGSMarkerSymbol
 - `offset`
 - `leaderPoint`



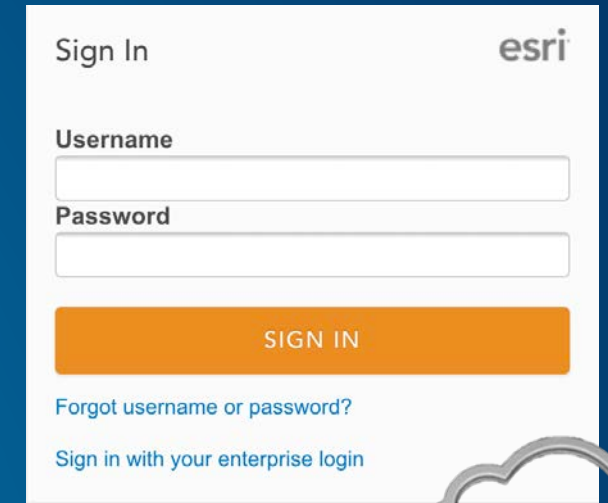
18. Customize the callout

- Match application's style
- Provide additional information/actions
- Default layout
 - Title text, Detail text, Accessory Button
 - Change colors, shadow, outline, button style...
- Custom layout
 - Embed any UIView
- AGSCallout
 - color, borderWidth,...
 - customView



19. Saving credentials to keychain

- **User doesn't have to sign-in everytime app is restarted**
 - Persist credential to keychain when user first signs in
 - Read credential from keychain when app is restarted
 - Clear keychain when user signs out
- **AGSKeychainItemWrapper**
 - Persist/Read AGSCredential objects
 - `setKeychainObject`
 - `keychainObject`



Sign In esri

Username

Password

SIGN IN

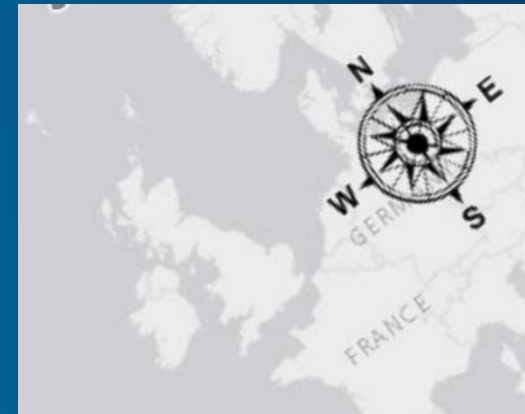
[Forgot username or password?](#)

[Sign in with your enterprise login](#)



20. KVO on properties

- Key-value-observing (KVO) provides automatic change notification
 - Use mapview's `rotationAngle` property to display a North Arrow



Rate This Session

www.esri.com/RateMyDevSummitSession



Related Sessions

Session Name	Time	Location
ArcGIS Runtime SDKs: Building Offline Apps, Part I	Thursday 9:00 – 10:00am	Primrose A
ArcGIS Runtime SDK: Building Offline Apps, Part II	Thursday 10:30– 11:30am	Primrose A
ArcGIS Runtime SDK for iOS and OS X: How We Build the Drone Apps	Thursday 1:00 – 2:00pm	Primrose A
ArcGIS Runtime SDKs: Offline Routing and Geocoding	Thursday 4:00-5:00pm	Smoketree A-E
Collector for ArcGIS: How we did it	Thursday 5:30 – 6:30pm	San Jacinto
The Road Ahead: ArcGIS Runtime	Friday 8:30 – 9:30am	Primrose A
Everything (or Anything) You Wanted to Know about the ArcGIS Runtime SDKs but Were Afraid to Ask	10:00 – 11:00am	Primrose A
Sneak peek into the Next Generation iOS and OS X API	1:00pm – 2:00pm	Primrose C/D



Q & A...





Understanding our world.