

Esri Developer Summit

March 26–29, 2012 | Palm Springs, California

esri.com/events/devsummit



Building Applications with ArcGIS Runtime SDK for iOS — Part II

Eric Ito and Scott Sirowy

A decorative graphic at the bottom of the slide. It features a curved orange border at the top. Below the border is a semi-transparent map of a landscape with green fields and blue water. Overlaid on the map is white text representing code snippets from the ArcGIS Runtime SDK for iOS. The code includes symbols like 'esri.symbol.SimpleLineSymbol', 'new dojo.Color([0,0,0,0.5])', and 'feature.setSymbol(polySymbolGreen)'.

```
esri.symbol.SimpleLineSymbol
new dojo.Color([0,0,0,0.5])
polySymbolGreen
feature.setSymbol(polySymbolGreen)
} else if(f == 1) {
var polySymbolGreen =
polySymbolGreen.setOutline
symbol.SimpleLineSymbol(esri.symbol
Color([0,0,0,0.5]), 1));
polySymbolGreen.setSymbol(polySymbolGreen)
} else if(f == 2) {
polyBlue = new esri.symbol.SimpleLineSymbol
setOutline(new
esri.symbol.SimpleLineSymbol
new dojo.Color([0,0,0,0.5]), 1));
```

Esri Developer Summit

March 26–29, 2012 | Palm Springs, California
esri.com/events/devsummit



Building Applications with ArcGIS Runtime SDK for iOS — Part II



Eric Ito and Scott Sirowy

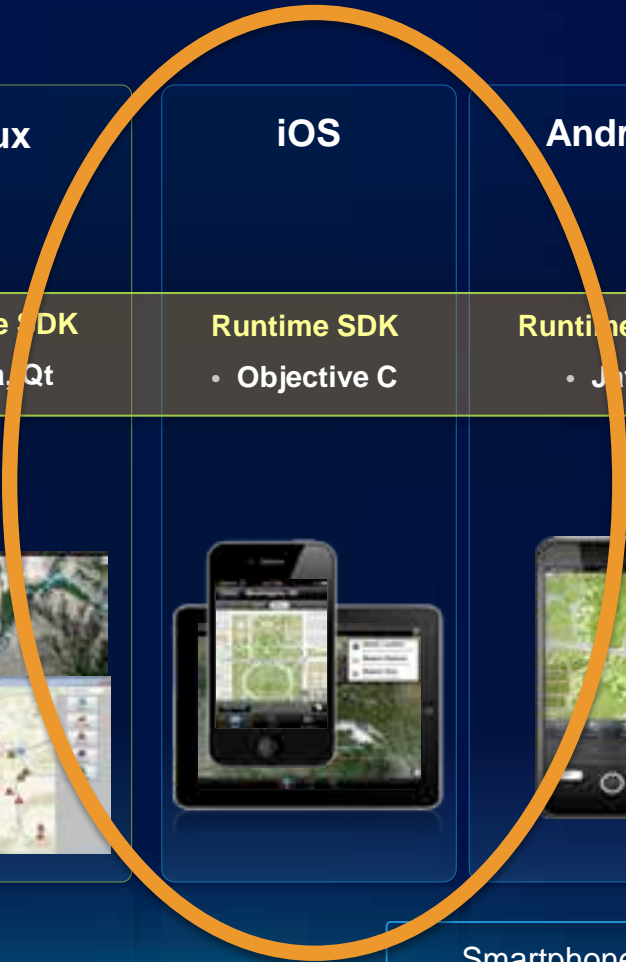


2011 Dev Summit
Dodgeball Champions



ArcGIS Runtime SDKs

Windows Mobile	Windows	Linux	iOS	Android	Windows Phone
Runtime SDK <ul style="list-style-type: none">.NET CF	Runtime SDK <ul style="list-style-type: none">WPF, Java	Runtime SDK <ul style="list-style-type: none">Java, Qt	Runtime SDK <ul style="list-style-type: none">Objective C	Runtime SDK <ul style="list-style-type: none">Java	Runtime SDK <ul style="list-style-type: none">Silverlight
					



Rugged and Embedded Devices

Smartphones and Tablets

Inspections App



iOS Part I - Recap

- Using AGSMapView in your application
 - Adding a tile layer basemap
 - Add your own feature layers



iOS Part I - Recap

- **Using AGSMapView in your application**
 - Adding a tile layer basemap
 - Add your own feature layers
- **Adding callouts, custom graphics, etc**

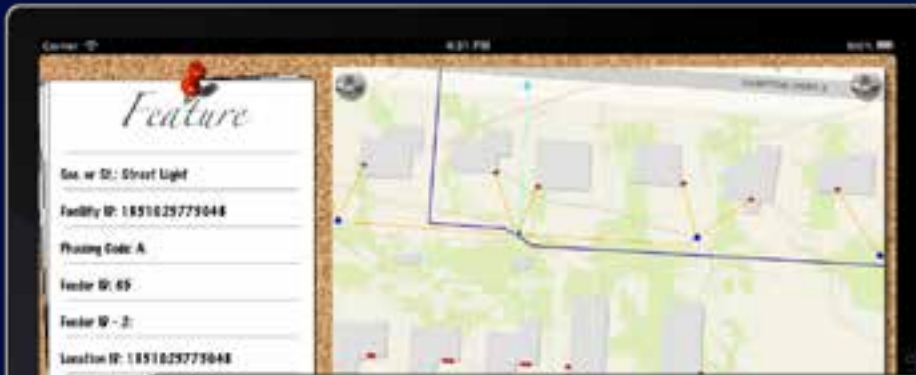


iOS Part I - Recap

- Using AGSMapView in your application
 - Adding a tile layer basemap
 - Add your own feature layers
- Adding callouts, custom graphics, etc
- Incorporating routing into your application
 - Directions
 - GPS Modes



iOS Part II - Where we are headed



- Portal/Geowarehouse APIs (new release)
- Web Map APIs
- Viewing and editing information

iOS Part II - Where we are headed



- Support for Portal REST API (just released)
- Web Map APIs
- Viewing and editing information
- Taking your application offline
 - Provisioning offline base maps

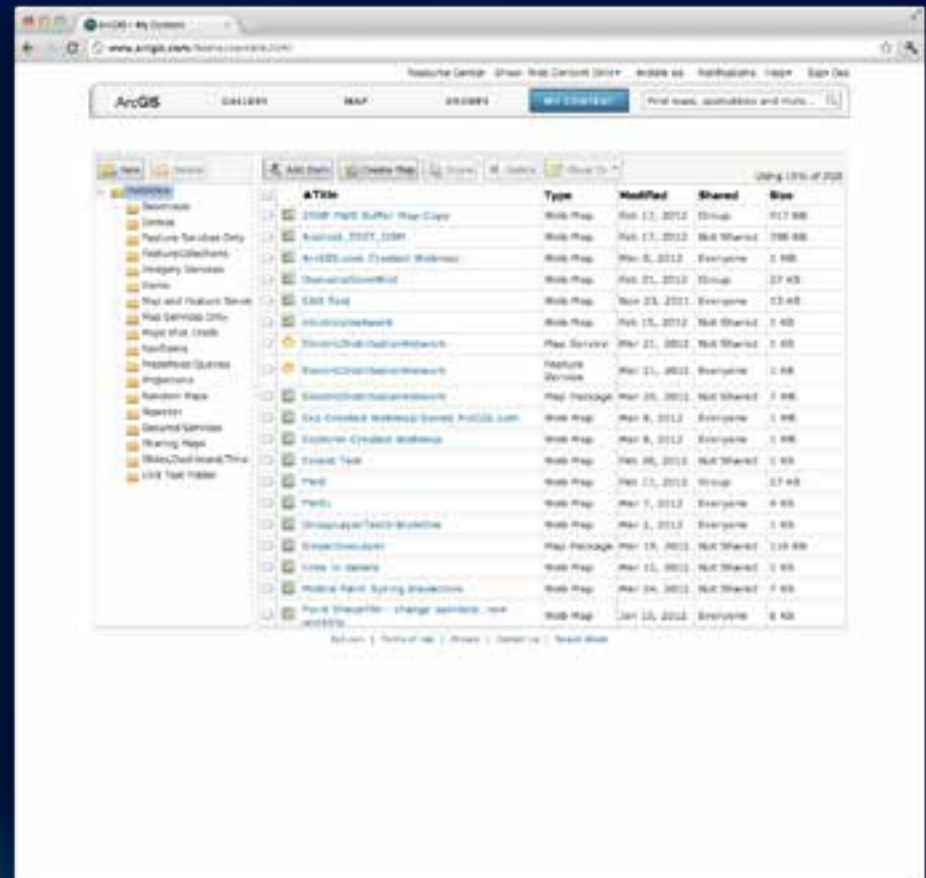
iOS Part II - Where we are headed



- Support for Portal REST API (just released)
- Web Map APIs
- Viewing and editing information
- Taking your application offline
 - Provisioning offline base maps
- Creating a stand-out mapping application for iOS

Support for Portal REST API

- **New support in iOS Runtime SDK 2.2**
- **Access to an online portal**
 - ArcGIS Online
 - Custom Portal
- **Access users, groups, web maps, and other related content.**



Portal REST API – New iOS Classes

```
@interface AGSPortal : NSObject
```

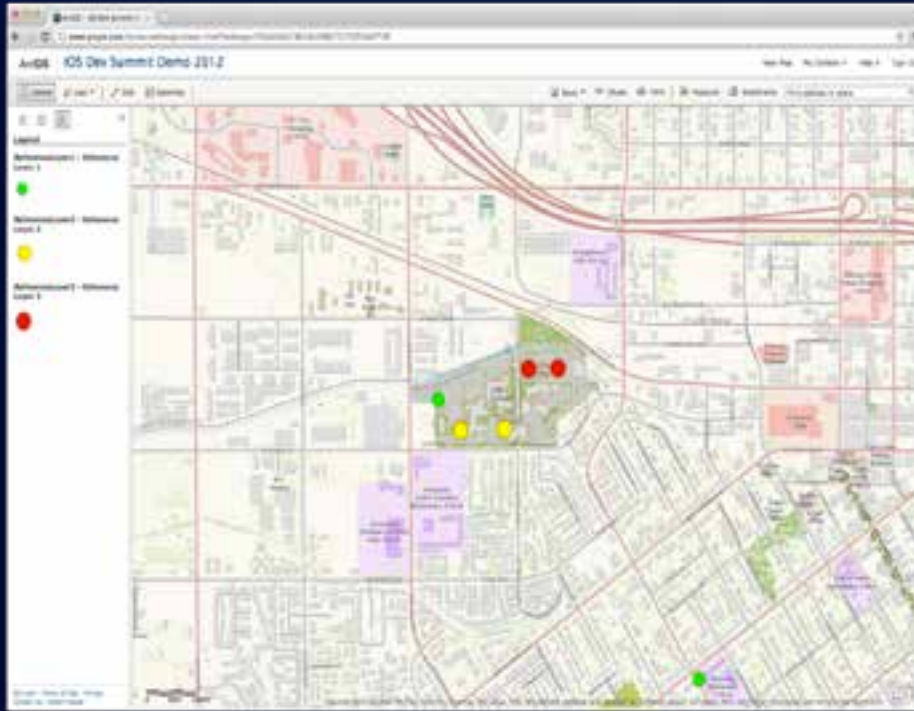
```
@interface AGSPortalItem: NSObject
```

```
@interface AGSPortalGroup: NSObject
```

```
@interface AGSPortalUser: NSObject
```

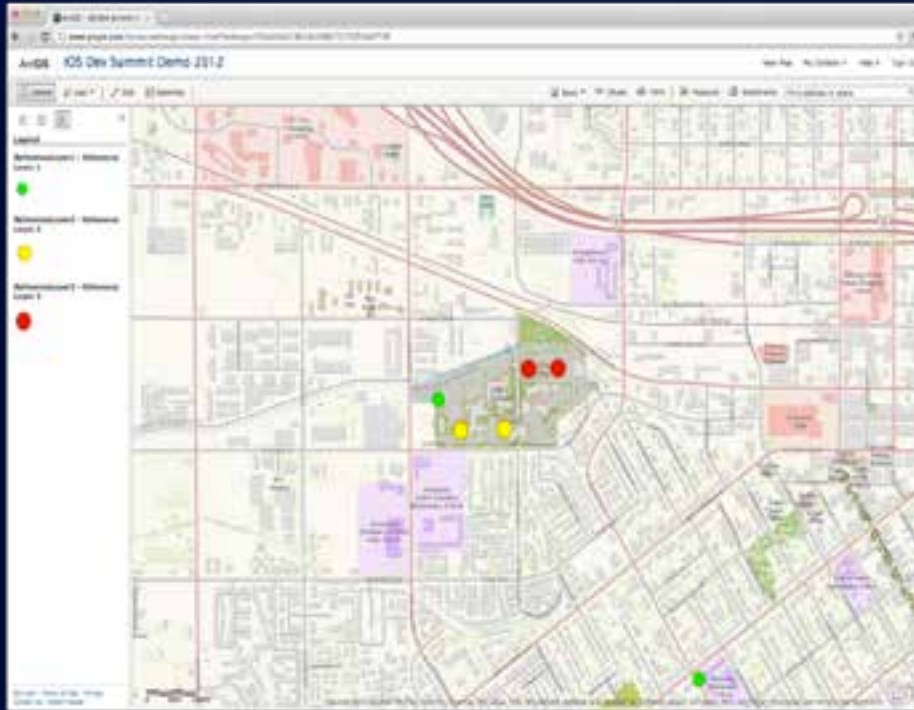
Several other supporting classes too...

AGSWebMap (since 2.0 release)



- +(AGSWebMap*)webMapWithItemId:(NSString*)itemId credential:(AGSCredential*)cred;
- +(AGSWebMap*)webMapWithURL:(NSURL *)url credential:(AGSCredential*)cred;

AGSWebMap – Supports new Portal REST API

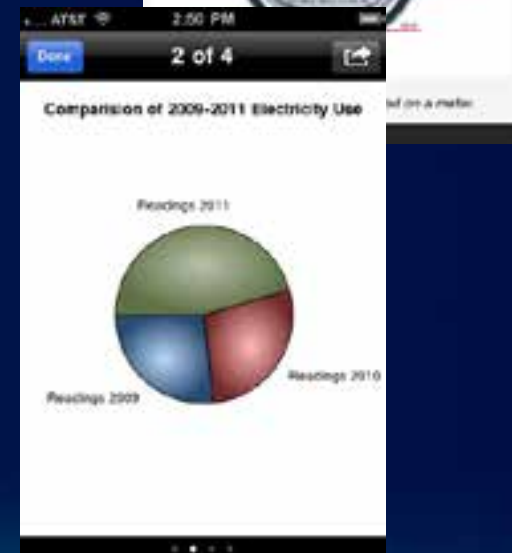
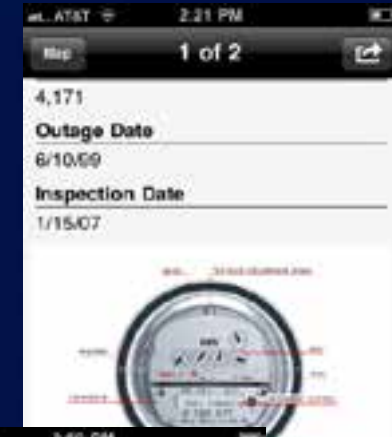
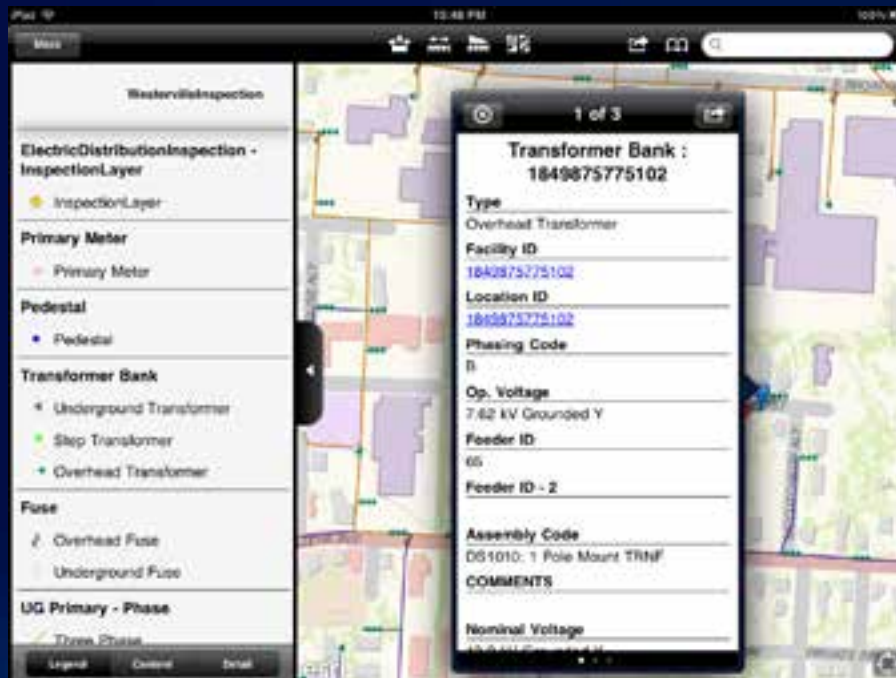


- +(AGSWebMap*)webMapWithItemId:(NSString*)itemId credential:(AGSCredential*)cred;
- +(AGSWebMap*)webMapWithURL:(NSURL *)url credential:(AGSCredential*)cred;
- **+(AGSWebMap*)webMapWithPortalItem:(AGSPortalItem*)item;**
- **+(AGSWebMap*)webMapWithItemId:(NSString*)itemId portal:(AGSPortal*)portal;**

Viewing/Editing Data with the iOS SDK

Using Feature layers & Popups

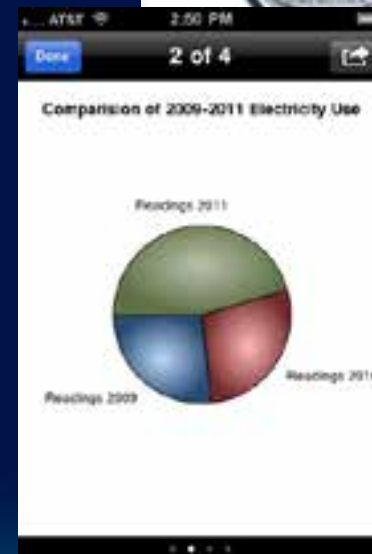
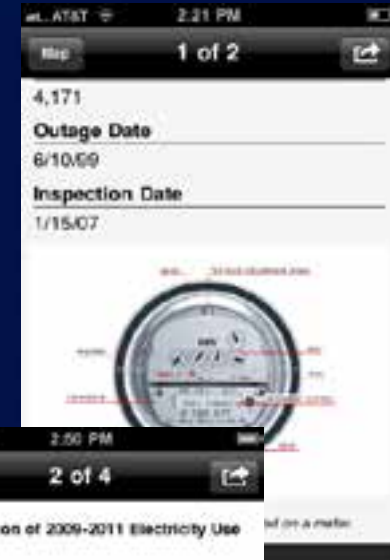
- Popups provide UI to
 - Display and edit attributes
 - Manage attachments
 - View charts, media



Viewing/Editing Data with the iOS SDK

Using Feature layers & Popups

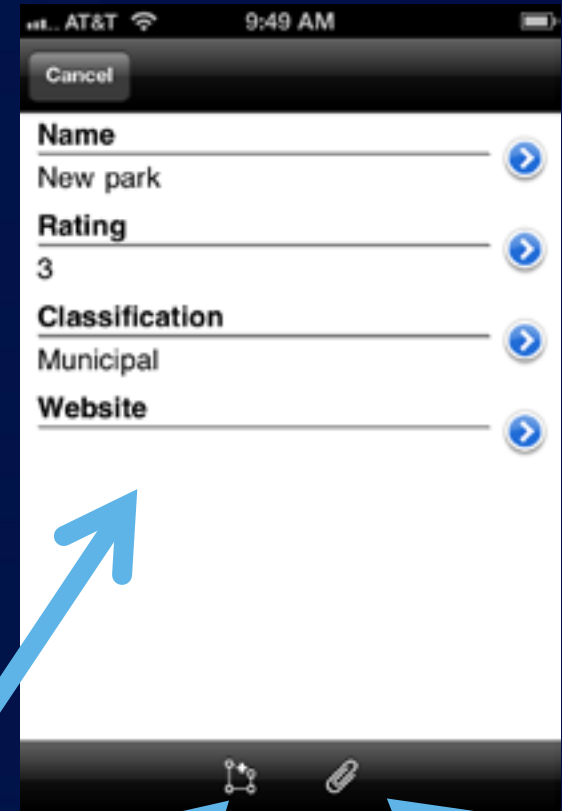
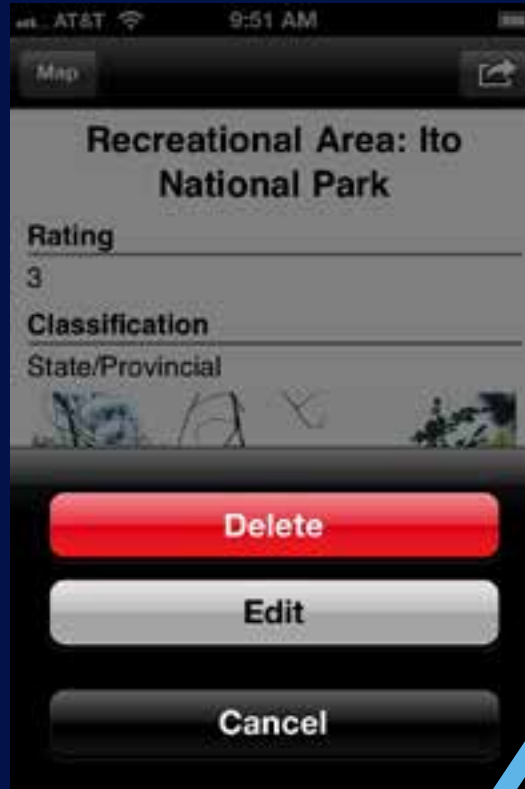
- **Popups provide UI to**
 - Display and edit attributes
 - Manage attachments
 - View charts, media
- **Popups configured through web maps**
 - Attributes to display & edit
 - User friendly aliases and hints
 - Formatting for numbers, dates



Editing/Collecting Data

Using Popups

- **Edit feature**
 - **Attributes**
 - **Geometry**
 - **Attachments**



Popups iOS Classes (since 2.0)

```
@interface AGSPopup : NSObject
```

```
@interface AGSPopupInfo : NSObject
```

```
@interface AGSPopupsContainerViewController : UIViewController
```

```
NSArray *popups = [NSArray arrayWithObject: myPopup];  
AGSPopupsContainerViewController *vc =  
    [[AGSPopupsContainerViewController alloc] initWithPopups: popups];  
[self presentViewController: vc animated: YES];
```

Going Offline

- **Provisioning device with a basemap cache**
 - **Supported Caches**
 - **Application Configuration**
 - **Using Itunes**
- **Taking your data offline**



Creating a great (GIS/mapping) application for iOS



Creating a great (GIS/mapping) application for iOS



- “Great iOS Apps Embrace the Platform and Human Interface Design Principles”
- “A Great User Experience Is Rooted in Your Attention to Detail”

~Apple's Mobile Human Interface Guidelines



esri