



Esri International Developer Summit

Palm Springs, CA

Getting Started with ArcGIS Runtime SDK for Java SE

Elise Acheson, Vijay Gandhi, and Eric Bader

Demo Source code:

<https://github.com/Esri/arcgis-runtime-samples-java/tree/master/dev-summit-2014>

Video Recording:

<http://video.esri.com>

Outline

- **Intro to ArcGIS Runtime SDKs**
- **Get started: download and install the SDK**
- **Tour of the functionality of the API**
- **Basics of building a map application**
- **Online workflow: services, ArcGIS Online, web maps**
- **Offline workflows: local data, create and update**
- **Deployment and licensing**

ArcGIS Runtime

Runtime built using C++

EXPLOITS THE CAPABILITIES OF THE DEVICE

Functionality exposed to developers via an API
native to the platform

INTUITIVE TO LEARN

Common functionality set and conceptual model

EASES MULTI PLATFORM DEVELOPMENT

Device Platforms



PHONE



TABLET



LAPTOP



DESKTOP



EMBEDDED

Java SE

ArcGIS Runtime SDK for Java

- Build native apps for Windows and Linux
 - Windows 8 / 7 / **Vista, XP***
 - Ubuntu, RedHat
 - 32 and 64 bit Windows, **32*** and 64 bit Linux
- Java SE API, Swing-based
- Eclipse plugin (Juno, Kepler)
- Integrates with the ArcGIS Platform
- Developed alongside Runtime SDK for Android

* **Dropping Vista and XP support at 10.2.2**

* **32-bit Linux support new at 10.2 release**



ArcGIS Runtime SDK for Java

- **Get it:** free download on developers.arcgis.com
- **What you get:**
 - Set of jars to code against
 - Open-source toolkit (mainly UI components)
 - Eclipse plugin, includes map application template
 - Runtime tools: deploy / debug
 - Documentation: Guide, API reference
 - Tons of samples
- **Get help:** Guide, API Reference, Forum
- **Give feedback:** website pages, sessions, Forum

DEMO

The SDK

Elise Acheson

Home

Guide

API Reference

Sample Code

Forum

Map options

↓ DOWNLOAD SAMPLE VIEWER



This application shows how to create a `JMap` using a `MapOptions` instance, giving you the option (base layer), latitude and longitude around which to center the map, and zoom level for the map. The `MapOptions` class is then used to switch the type of basemap in the map on-the-fly. In addition, simple marker graphics can be added directly to the `JMap` using the `addMarkerGraphic` methods. Popups are enabled by default on the map. To disable these popups, use `setMarkerGraphicPopupsEnabled(boolean)`, passing in `false` to disable. For a given address or location, static methods on the `Locator` class exist which either take or return input as a `String`. In this application, the `Locator.findAddress` static method is used to locate (geocode) the search string entered in the text field. The top result is shown on the map using a marker graphic.

What you can do

- Mapping
- Searching (query, find, identify, address finding)
- Editing
- Geometry operations
- GPS
- Network Analysis (route finding, drive times, closest facility)
- Spatial Analysis (Geoprocessing)
- Advanced Symbology



Online and offline

DEMO

Functionality Tour

Vijay Gandhi

Time slider



Shows how to display a time-aware layer from a map package (.mpk) as a local dynamic layer.

Local tiled layer



Tiled map service layer



Loads an ArcGIS Server tiled map service from its URL.

OpenStreetMap layer



OpenStreetMap custom layer



Shows how to display a custom tiled layer adhering to the OpenStreetMap tile naming conventions using the OpenStreetMapLayer.

'No Data' tiles



Download tile cache



Shows how to download a tile cache from an online service which supports the 'exportTiles' operation.

Tiled image service layer



Map and layers

- **JMap : Swing component (JComponent)**
- **Spatial Reference (SR)**
 - First layer in map (basemap) sets SR
 - Subsequent layers reprojected to map SR
- **Extent**
 - Envelope in SR coordinates
 - Set initial extent to area of interest
- **Layers**
 - Collection of layer classes with different behaviours
 - All inherit from `Layer` class
 - Order in map is order in which they are added
 - Add a layer to map's layer list:

```
jMap.getLayers().add(Layer);
```

Build a map



- **Live Data**
 - Graphics layers
- **Operational Data**
 - Dynamic layers / Feature layers
- **Basemap**
 - Tiled layers
- **Map**

DEMO

Build a map

Vijay Gandhi



Work with graphics

- API classes: `GraphicsLayer`, `Graphic`
- A graphics layer contains graphics (you guessed it!)
- `Graphic` class is immutable: so don't hold references to `Graphic` objects
- Update / move / remove graphics using methods on `GraphicsLayer`
- Work with graphics via the layer using their unique ID

```
id = addGraphic(Graphic)
graphic = getGraphic(id)
...
updateGraphic(id, Graphic)
updateGraphic(id, Symbol)
updateGraphic(id, Geometry)
...
removeGraphic(id)
setGraphicVisible(id, visible)
select(id)
...
```

Interact with the map

- **Extend MapOverlay class, implement the methods you need:**
 - onMouseClicked, onMouseMoved, onMouseDragged, ...
 - override onPaint to draw onto map

- **Use the toolkit overlays!**
 - Including overlays for:
 - editing
 - popups
 - scale bar & navigator
 - hit tests (responding to graphics being clicked)

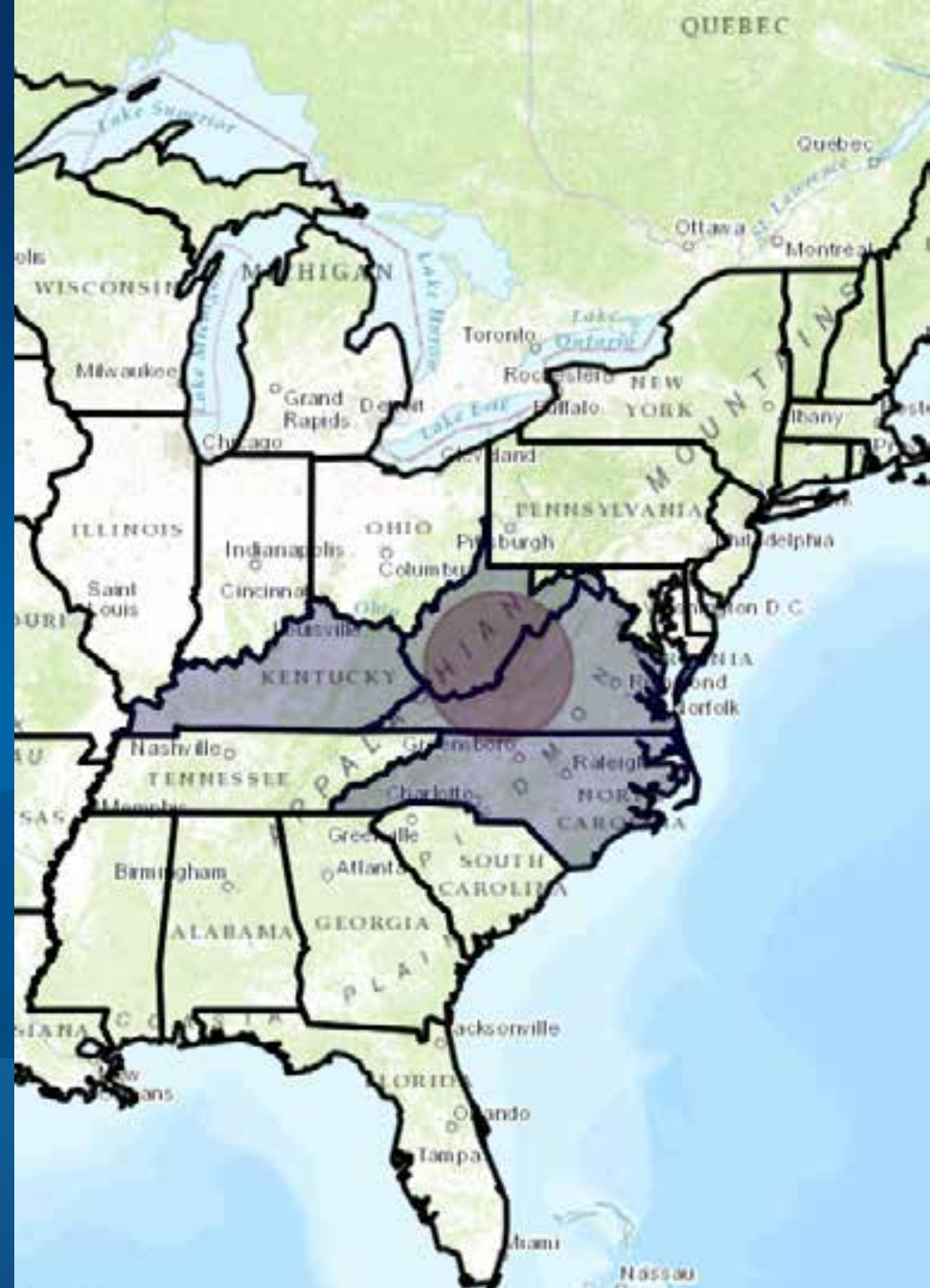
- **Add an overlay to the JMap:**

```
jMap.addMapOverlay(...);
```

DEMO

Interact with the map

Vijay Gandhi



Online and offline workflows



Online

- **ArcGIS for Server services**
 - **ArcGIS Online (web maps)**
 - **Portal for ArcGIS**
-
- **Basemaps: map services**
 - **Dynamic layers: map services**
 - **Feature layers: feature services**
 - **Geocoding: geocode services**
 - **Route finding: network analyst service**
 - **Analysis: geoprocessing services**



Offline

- **ArcGIS for Desktop: prepare data**
 - **Download data from online services**
 - **Local Server (services)**
-
- **Basemaps: local tile cache**
 - **Dynamic layers: local map services (mpk)**
 - **Feature layers: local geodatabase**
 - **Geocoding: local geocoding**
 - **Route finding: local routing**
 - **Analysis: geoprocessing services (gpk)**

Online workflows

- **ArcGIS Services**
 - REST API
 - Create via ArcGIS for Desktop, ArcGIS for Server
 - **Map services**
 - tiled layers, dynamic layers
 - **Feature services**
 - feature layers, editing, search tasks (query)
 - **Image services**
 - image service layers
 - **Geocode services**
 - geocode task
 - **Network Analysis services**
 - route task, closest facility task, service area task
- **Other online data sources: WMS, OpenStreetMap, Bing basemaps, KML**

WebMap and Portal

- Open via web map ID, Portal, user credentials if secure
 - get ID from URL
- Retrieve web map via Portal API
 - query for web map items on a Portal
- Create a `WebMap` instance then load into `JMap`:

```
WebMap webmap = new WebMap("webmap_id");  
jMap.loadWebMap(webmap);
```

- `JMap` loads all the web map's layers
 - JSON of web map passed to client API, displays the layers according to order, rendering info, popup info, etc.



DEMO

WebMap and Portal

Elise Acheson



Offline workflows

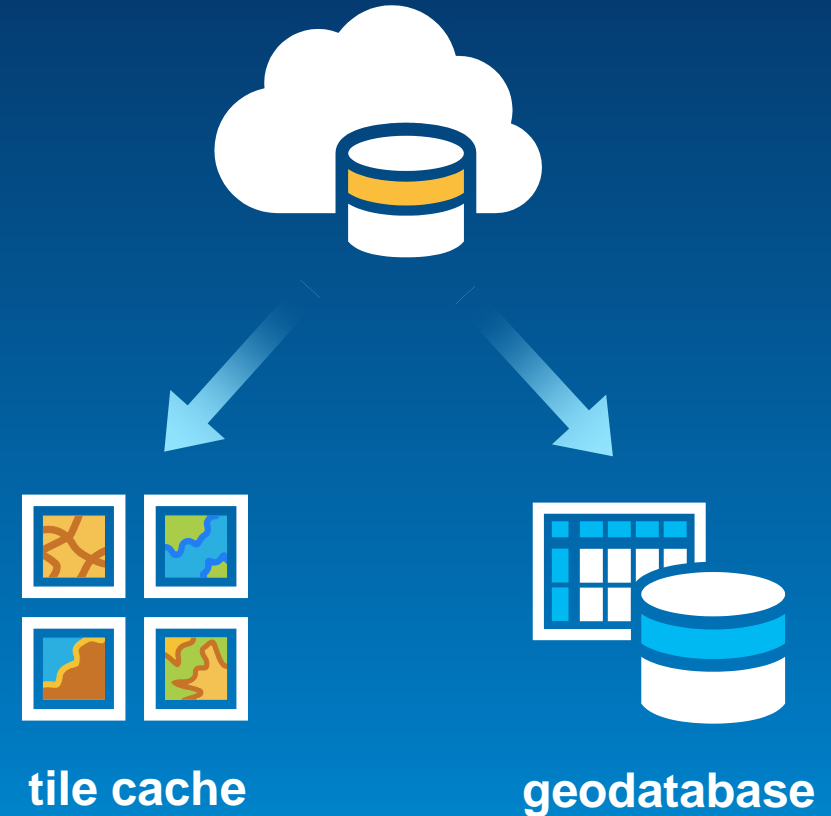
- **Pre-10.2: Local Server**
- **10.2 Beta / 10.2.2 Final: local data access**
 - **Used across Runtime SDKs: mobile as well as desktop platforms**
 - **performant, lightweight**
- **Create data from services: tile caches, geodatabases**
- **Offline routing**
- **Offline geocoding**

- **Geometry operations done locally via API**
- **Local Server for offline geoprocessing**
- **Local Server for offline dynamic map services**

Offline

- **Services pattern: create from ArcGIS services**
- **Desktop pattern: create in ArcGIS for Desktop**

- **Create local tile caches (basemaps)**
- **Create local geodatabases**
 - geodatabase for storing feature data locally
 - edit offline
 - query offline
 - sync edits back with service



DEMO

Offline analysis

Elise Acheson

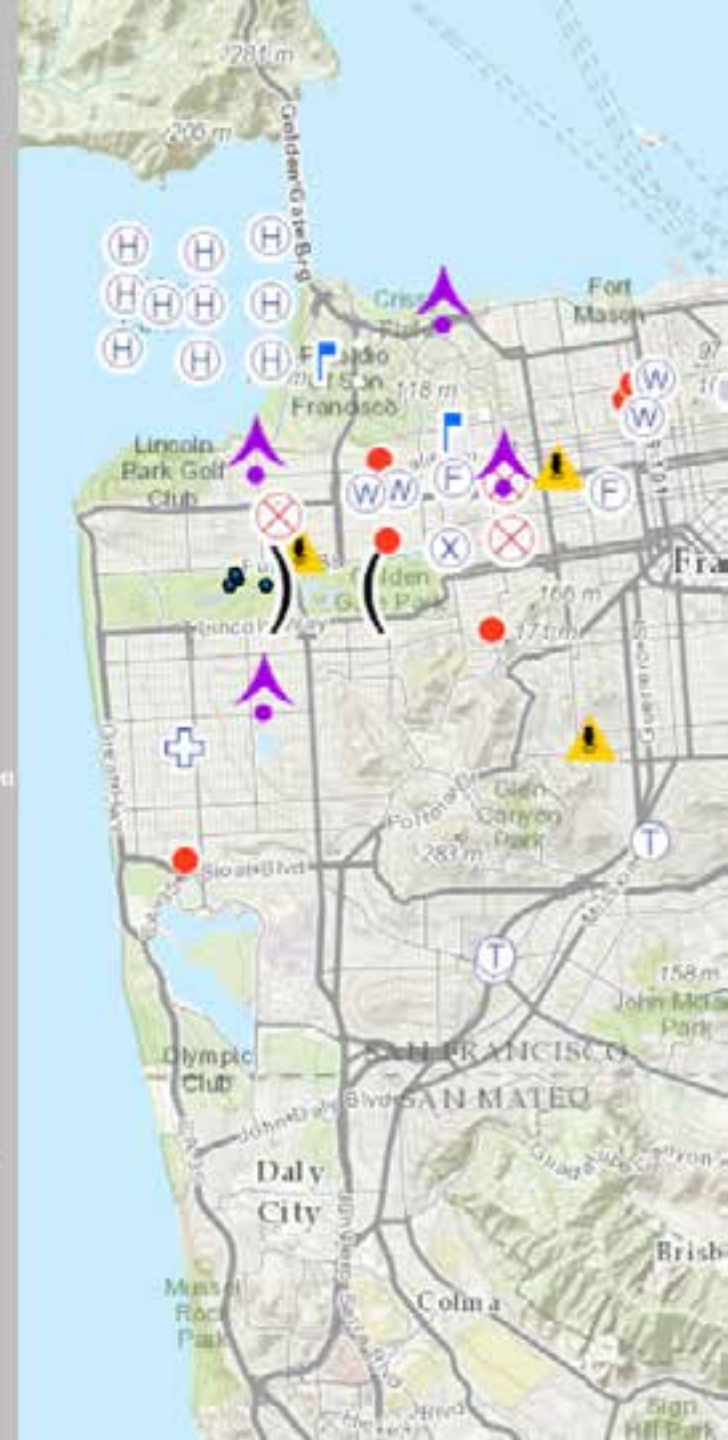


DEMO

Offline editing

Vijay Gandhi

- Division Break
- Aerial Hazard
- Camp
- Drop Point
- Fire Origin
- Fire Station
- First Aid Station
- Safety Zone
- Spot Fire
- Water Source
- Wind Speed Direction
- Hellbase
- Hot Spot
- Lookout
- Med/Vac Site
- Mobile Weather Unit



Runtime Licensing

Development and Deployment Workflow



1. Download and Install



2. Develop and Test



3. Deploy and Distribute

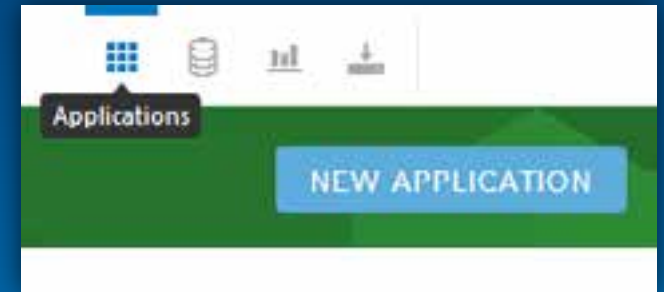
License levels and functionality

License Level	Available functionality
Developer (development and testing only)	All functionality including Local Server extensions (watermarks and debug messages will be generated, nag screens with Local Server)
Basic	All functionality, except: <ul style="list-style-type: none">• Local locators (geocoding)• Local routing• Local geodatabase editing• Local geodatabase sync operations with an upload• Local Server
Standard	All functionality Local Server extension licenses are additional and require the Standard license.

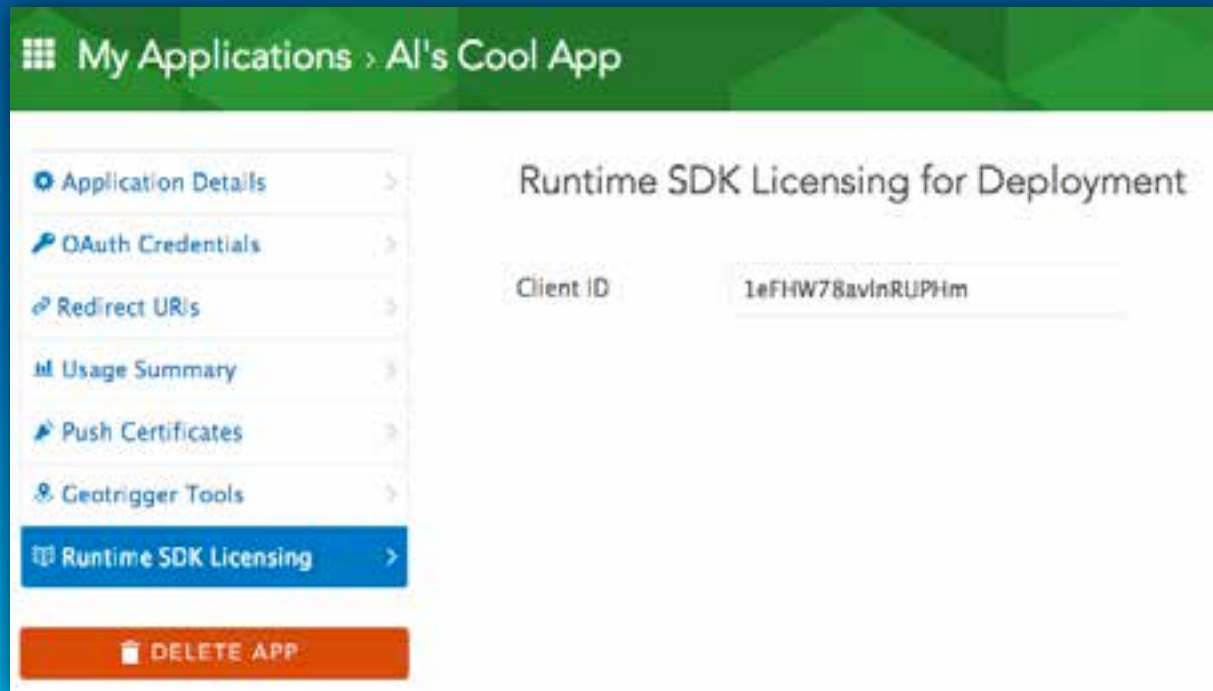
License your app at Basic level

1. Go to developers.arcgis.com and log in (or create a developer account)
2. Create a New Application (or select existing)
3. Click on Runtime SDK Licensing
4. Copy the Client ID and set it in your app

2.



3.



4.

```
// set the client ID  
ArcGISRuntime.setClientID("myClientID");
```

License your app at Standard level

2 ways:

1. Use an **organization account** (ArcGIS Online or Portal for ArcGIS)

- Requires users of your app to log in with their account

2. Use a **license string** obtained from Customer Service or your international distributor

- License burnt into the app
- Extensions can also be added with this option (e.g. Local Server geoprocessing)

**** You must use workflow 2 if you want to license any extensions ****

Questions?

Next sessions:

- Wednesday, 1:00pm - 1:30pm, Demo Theatre 1 – Oasis 1

Animating Thousands of Graphics and Features Using ArcGIS Runtime SDK for Java

- Wednesday, 4:00pm - 5:00pm, Smoketree F

Building Java Apps using ArcGIS Runtime SDK

- Thursday 8:30am - 9:30am, Primrose A

Road Ahead for ArcGIS Runtime SDKs

- Thursday 10:00am – 11:00am, Primrose A

Everything (or Anything) You Wanted to Know about ArcGIS Runtime SDKs



Understanding our world.