Developing Windows Desktop Apps with ArcGIS Runtime SDK for Microsoft .NET Framework

Thad Tilton - @T_hadde_us
Apurva Goyal
Session overview
What will we cover?

- What is the ArcGIS Runtime SDK for .NET?
  - ArcGIS Runtime Core
  - ArcGIS Runtime APIs and SDKs

- Get started
  - Installation and system requirements
  - Documentation, tutorials, and samples

- Display a map
  - MapView and Map controls
  - Types of layers
Session overview (continued)

What will we cover?

• Work with tasks
  - Query
  - Geocode and Routing

• Work offline
  - Store features locally (offline geodatabase)
  - Make edits while disconnected
  - Sync changes to the server

• Licensing
  - Development and testing
  - Basic
  - Standard
ArcGIS Runtime Overview

Thad Tilton
ArcGIS Runtime overview

• Runtime Core (C++)
  - Small footprint, high performance
  - Core functionality: Display, geometry, data access, …
  - Compiled for multiple platforms and architectures
ArcGIS Runtime overview

- Access core functionality via a native API for each platform:
  - Application Programming Interface
  - .NET, Android, Java, etc …
  - No need to be concerned with details of Core
ArcGIS Runtime SDK

- **Software Development Kit**: tools for developers
  - Conceptual doc, API reference, samples, and the developer community
  - GitHub: Samples, Toolkit, Portal Viewer app
ArcGIS Runtime SDK Highlights

• Support for many platforms
• Runs natively on devices
• Built for performance

• Use data from ArcGIS for Portal, ArcGIS for Server, and ArcGIS Online
• Work offline with local basemaps and data
  - Editing and sync, Geocode, Routing
• Perform advanced geometric operations locally
• Task-based asynchronous pattern
• MVVM friendly
• Simplified licensing model
Get started

Thad Tilton
System Requirements for Windows Desktop development

- **Operating system**
  - Windows 7
  - Windows 8
  - Windows 8.1

- **.Net framework**
  - 4.5
  - 4.5.1

- **IDE**
  - Visual Studio 2013 (all editions)
  - Visual Studio Express 2013 for Windows Desktop
  - Visual Studio 2012 with Update 3 (all editions)
  - Visual Studio Express 2012 for Windows Desktop with Update 3

For Windows Store
- Windows 8.1
- Visual Studio 2013 (or Express)

For Windows Phone
- Windows 8.1 (64 bit only)
- Windows Phone OS 8.1
- Visual Studio 2013 with Update 2 (or Express)
Where to start?

• Developers site
  - https://developers.arcgis.com/net
  - Check system requirements

• Download and install the public beta
  - http://betacommunity.esri.com

• Code resources
  - GitHub repos - https://github.com/Esri
    - Toolkit
    - Samples

• Provide feedback
  - Beta community (use forums, log issues)
Demo: Developer Resources

Thad Tilton
Display a map

Thad Tilton
Mapping classes

- **MapView (control)**
  - UI container for a single Map
  - Facilitates MVVM design

- **Map**
  - Container for a collection of layers

- **Layer**
  - Display geographic features
  - Various types
Define a map with XAML

<!-- Add XML Namespace declaration for ArcGIS Runtime -->
xmlns:esri="http://schemas.esri.com/arcgis/runtime/2013"

<!-- Use namespace prefix to access ArcGIS Runtime objects -->

<esri:MapView x:Name="MyMapView">
  <esri:Map x:Name="MyMap">
    <esri:ArcGISTiledMapServiceLayer
      ServiceUri="http://services.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer"
      ID="Basemap"/>
    <esri:GraphicsLayer ID="Graphics"/>
  </esri:Map>
</esri:MapView>

Note: Syntax differs in Store/Phone apps

Use x:Name to identify MapView or Map

Use ID to identify Layers
Create a map with code

```javascript
var theMap = new Esri.ArcGISRuntime.Controls.Map();

var uri = new Uri("http://services.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer");
var basemap = new Esri.ArcGISRuntime.Layers.ArcGISTiledMapServiceLayer(uri);
basemap.ID = "Basemap";

graphicsLayer.ID = "Graphics";

theMap.Layers.Add(basemap);
theMap.Layers.Add(graphicsLayer);

this.MyMapView.Map = theMap;
```

XAML

```xml
<esri:MapView x:Name="MyMapView">
</esri:MapView>
```
Create a map with code (MVVM)

```csharp
class MapViewModel
{
    public MapViewModel()
    {
        var map = new Map();
        var uri = new Uri("http://services.arcgisonline.com/...");
        var basemap = new ArcGISTiledMapServiceLayer(uri);
        basemap.ID = "Basemap";

        map.Layers.Add(basemap);

        ParcelMap = map;
    }

    public Map ParcelMap { get; private set; }
}

<esri:MapView Map="{Binding ParcelMap}">
</esri:MapView>
```
Several types of layers

• Tiled layers
  - ArcGISTiledMapServiceLayer
  - ArcGISLocalTiledLayer
  - BingLayer
  - WebTiledLayer
  - WmtsLayer

• Dynamic layers
  - ArcGISDynamicMapServiceLayer
  - DynamicMapServiceLayer
  - FeatureLayer
  - GraphicsLayer
Tiled layers

• Base map to provide context
  - Data that don’t change often
• Cached for performance

• Find on ArcGIS Online

<esri:ArcGISTiledMapServiceLayer
  ServiceUri="http://services.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer"
  ID="Basemap" />
Dynamic layers

- Used to display changing information
- Online or local data source
- Draw on top of base map layer(s)

<esri:FeatureLayer ID="MyFeatureLayer">
  <esri:FeatureLayer.FeatureTable>
    <esri:GeodatabaseFeatureServiceTable
      ServiceUri="http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/Fire/Sheep/FeatureServer/2" />
  </esri:FeatureLayer.FeatureTable>
</esri:FeatureLayer>
Demo: Display a map

Apurva Goyal
Work with tasks

Thad Tilton
Tasks are used to perform specialized pieces of work

- **QueryTask**: use attribute or spatial criteria to find features
- **IdentifyTask**: get information about features at a location
- **FindTask**: find a text value across several layers and fields
- **LocatorTask**: geocode an address
- **RouteTask**: find the best path between points on a network
- **PrintTask**: output data to a specific device

- Many others …
Tasks …

• Have corresponding Parameter and Result objects
  - FindTask – FindParameters, FindResult
  - IdentifyTask – IdentifyParameter, IdentifyResult
  - GenerateRendererTask – GenerateRendererParameter, GenerateRendererResult
  - QueryTask – Query, QueryResult

• Execute asynchronously
  - Keep UI responsive
  - Can be cancelled

• May have local and online versions
  - LocalLocatorTask / OnlineLocatorTask
Consistent task workflow

1. Create Task object, point it to an appropriate resource (online or local)
   
   ```csharp
   var myTask = new SomeTask(new Uri("http://…"));
   ```

2. Define task parameters
   
   ```csharp
   var taskParams = new SomeTaskParameter();
   ```

3. Execute the task asynchronously
   
   ```csharp
   var taskResult = await myTask.ExecuteAsync(taskParams);
   ```

4. Process the result
   
   ```csharp
   ```
Work offline

Thad Tilton
Work with data while disconnected from the internet

• Download online data sources for local access
  - Tiled service as image cache
  - Feature service as geodatabase
  - Can also make functionality available offline (Geocode, Route)

• Consistent user experience
• Consistent developer experience

• Synchronize data sources when connected
  - Push edits made to local data
  - Pull newest version of online data
Going Offline – Tiled Map Services

- Use `ExportTileCacheTask` to take tiled map services offline
  - `GenerateTileCacheAsync` – creates a tile package (.tpk) or compact cache
  - `DownloadTileCacheAsync` – downloads a .tpk or compact cache
  - `GenerateTileCacheAndDownloadAsync` – does both in one call

```javascript
var task = new ExportTileCacheTask(new Uri(onlineTiledLayer.ServiceUri));
var downloadResult = await task.GenerateTileCacheAndDownloadAsync(generateOptions, downloadOptions, TimeSpan.FromSeconds(3), CancellationToken.None, onGenerateUpdate, onDownloadUpdate);
```
Going Offline – Feature Services

- **Use GeodatabaseSyncTask to take feature services offline**
  - GenerateGeodatabaseAsync – creates a geodatabase
  - Handle onComplete callback to save results when task completed
    - Check for exceptions

```javascript
GeodatabaseSyncTask gdbTask = new GeodatabaseSyncTask(new Uri(featureServiceUrl));
var result = await gdbTask.GenerateGeodatabaseAsync(
gdbParameters,
onGenerateCompleted,       // Callback to handle operation completion
TimeSpan.FromSeconds(3),             // Interval to check status
onGenerateProgress,                   // Callback to handle operation status updates
CancellationToken.None);            // Token to handle cancellation
```
Generate geodatabase parameters

- **GenerateGeodatabaseParameters** object
  - Layers to include in the output
  - Geometry (polygon) for filtering features
  - Output spatial reference
  - Synchronization model: per layer or per geodatabase

```csharp
var layerIDs = new List<int>{0,1,2};
var extent = MyMapView.Extent;

var gdbParameters = new GenerateGeodatabaseParameters(layerIDs, extent);
gdbParameters.SyncModel = SyncModel.PerLayer;
gdbParameters.OutSpatialReference = MyMapView.SpatialReference;
```
Download generated geodatabase

- Make request to GeodatabaseStatusInfo.ResultUri

```csharp
private async void onGenerateCompleted(GeodatabaseStatusInfo statusInfo, Exception ex)
{
    // if unsuccessful, return
    if (ex != null) { return; }

    // read the generated geodatabase from the server
    var client = new ArcGISHttpClient();
    var gdbStream = client.GetOrPostAsync(statusInfo.ResultUri, null);

    // write geodatabase to local location
    await Task.Factory.StartNew(async () =>
    {
        using (var stream = System.IO.File.Create(_geodatabasePath))
        {
            await gdbStream.Result.Content.CopyToAsync(stream);
        }
    });
}
```
Demo: Take data offline

Apurva Goyal
Offline Editing: Editing Geometry Locally

• Use GeometryEngine static class

```javascript
var reshapedGeometry = GeometryEngine.Reshape(geometryToBeReShaped, reshaperLine);
```

• Use Editor control

```javascript
var editGeometry = await MyMapView.Editor.EditGeometryAsync(geometryToBeEdited);
var newGeometry = await MyMapView.Editor.RequestShapeAsync(DrawShape.Polygon);
```
GeometryEngine methods

- Area
- Buffer
- Clip
- Contains
- Crosses
- Cut
- DistanceFromGeometry
- Extend
- Generalize
- GeodesicArea
- GeodesicBuffer
- GeodesicLength
- Intersection
- Intersects
- NearestCoordinateInGeometry
- NearestVertexInGeometry
- Overlaps
- Project
- Relate
- Simplify
- Touches
- Union
- And others …
Demo: Edit features

Apurva Goyal
Offline Editing: Synchronize changes

• Check if local data has changes

```javascript
var gdbFeatureSvcTable = parcelsFeatureLayer.FeatureTable as GeodatabaseFeatureTable;
var adds = gdbFeatureSvcTable.AddedFeaturesCount;
var updates = gdbFeatureSvcTable.UpdatedFeaturesCount;
var deletes = gdbFeatureSvcTable.DeletedFeaturesCount;

if (adds > 0 || updates > 0 || deletes > 0)
    DoSync();
```

• Create sync task and parameters

```javascript
var syncTask = new GeodatabaseSyncTask( new Uri("http://...") );
var syncParams = new SyncGeodatabaseParameters();
sp.SyncDirection = SyncDirection.Bidirectional;
```
Offline Editing: Synchronize changes

- Use `GeodatabaseSyncTask.SyncGeodatabaseAsync`
  - Push updates from the client and download changes from the service
  - Only changes (deltas) are downloaded/uploaded

```javascript
var gdbTask = new GeodatabaseSyncTask(new Uri(featureServiceUrl));
var result = await gdbTask.SyncGeodatabaseAsync(
    syncParameters,
    featureLayer.FeatureTable.Geodatabase, // Geodatabase to sync
    onSyncCompleted,
    onUploadCompleted,
    TimeSpan.FromSeconds(3),
    onStatusUpdate,
    CancellationToken.None); // Operation parameters
```
Demo: Synchronize data

Apurva Goyal
Runtime licensing

Thad Tilton
# License levels and functionality

<table>
<thead>
<tr>
<th>License Level</th>
<th>Available functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer (development and testing only)</td>
<td>All functionality (watermarks and debug messages will be generated, nag screens with local server *)</td>
</tr>
</tbody>
</table>
| Basic                                | Connected - all functionality  
Offline - map viewing only                                                                    |
| Standard                             | Connected and offline - all functionality, includes:  
• Local locators (geocoding)  
• Local routing  
• Local geodatabase editing  
• Local geodatabase sync operations  
• Local server * |

* For those SDKs that support it
How to license your app at the basic level

- [http://developers.arcgis.com](http://developers.arcgis.com)

- Under Application section, create a New Application (or select existing)

- Click on Runtime SDK Licensing

- Copy the Client ID and use it to programmatically set your ClientID
How to license your app at the standard level

• You have 2 options:

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

For more info speak to sales or product management
Summary

• Use ArcGIS Runtime SDK for .NET to develop for …
  - Windows Desktop, Windows Store apps, and Windows Phone

• Key functionality
  - Map visualization and query
  - Analysis, geocoding, routing, etc.
  - Offline capabilities
  - Editing and synchronization

• Resources
  - developers.arcgis.com/net – documentation, tutorials, API reference, forum
  - github.com/esri – code repositories for API toolkits, samples, full applications
## ArcGIS Runtime SDK sessions Wednesday

<table>
<thead>
<tr>
<th>Session Name</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcGIS Runtime SDK for Qt: Tips and Tricks</td>
<td>9:30am – 10:00am</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td><strong>Building .NET Apps with ArcGIS Runtime SDK: Tips and Tricks</strong></td>
<td>11:30am – 12:00pm</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>Offline Routing and Geocoding in ArcGIS Runtime SDK</td>
<td>3:00pm – 3:30pm</td>
<td>General Theater 2 (demo theatre)</td>
</tr>
<tr>
<td>Developing Windows Desktop Apps with ArcGIS Runtime SDK for .NET</td>
<td>8:30am – 9:45am</td>
<td>Room 09</td>
</tr>
</tbody>
</table>
## ArcGIS Runtime SDK sessions Thursday

<table>
<thead>
<tr>
<th>Session Name</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create your own Android App Tools Using ArcGIS Runtime SDKs</td>
<td>9:30am – 10:00am</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>Dive Deep into the Performance of the ArcGIS Runtime SDKs Core Display Architecture</td>
<td>10:30am – 11:00am</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>10 Things you Didn’t Know You Can Do with ArcGIS Runtime SDK for iOS</td>
<td>11:30am – 12:00pm</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>Animating Thousands of Graphics and Features with ArcGIS Runtime SDK for Java SE</td>
<td>12:30pm – 1:00pm</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>Developing Mobile Apps with ArcGIS Runtime SDK for .NET</td>
<td>10:15am – 11:30am</td>
<td>Room 05 A</td>
</tr>
<tr>
<td>ArcGIS Runtime SDKs: The Road Ahead</td>
<td>1:30pm – 2:45pm</td>
<td>Room 07 A/B</td>
</tr>
</tbody>
</table>
• Questions?

Вопросы?

¿ Preguntas?

Spørgsmål?

প্রশ্ন?

有问题吗？

Mga Katanungan?

질문?

الأسئلة؟
Thank you...

- Please fill out the session survey:

**Offering ID: 1347**

Online – [www.esri.com/ucsessionsurveys](http://www.esri.com/ucsessionsurveys)

Paper – fill out and put in drop box