ArcGIS Runtime SDK for .NET: Building Apps

Rich Zwaap
Thad Tilton
Agenda
ArcGIS Runtime SDK for .NET: Building Apps

• ArcGIS Runtime SDK Overview
  - Architecture
  - Functionality

• Getting started with ArcGIS Runtime SDK for .NET
  - SDK tools and resources
  - Build your first mapping app

• What is Universal Windows Platform (UWP)?
• What is Xamarin?
• Developing cross platform apps
ArcGIS Runtime SDK 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

<table>
<thead>
<tr>
<th>Android</th>
<th>Linux</th>
<th>iOS</th>
<th>OS X</th>
<th>Win</th>
<th>WinRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86</td>
<td>x64</td>
<td></td>
<td></td>
<td></td>
<td>ARM</td>
</tr>
</tbody>
</table>

C++ ‘Runtime Core’
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
SDK Highlights

- **High-performance 2D and 3D mapping**
- Perform geometric operations locally
- Task-based asynchronous pattern
- Work offline with local data
  - Read mobile map packages (.mmpk) created with ArcGIS Pro
  - Take web maps offline
  - Feature service editing and sync
  - Geocode and Routing
- Work with device sensors
- **Integration with Portal and ArcGIS Online**
  - Load, edit, and save web maps
- MVVM friendly
New with 100.1

- Offline Map Task
- Related tables
- Client-side labeling
- Enhanced network analysis capability
  - Service areas
  - Closest facilities
- Heatmap renderer
- Support for StreetMap Premium map packages

- Additional layer types
  - Image Service
  - Dynamic sublayers from an ArcGIS Map Service
  - OpenStreetMap
  - Bing
- Scene view camera controllers
  - Follow graphics
  - Orbit a location
ArcGIS Runtime 101: Display maps and layers

• GeoView control: MapView and SceneView
  - UI container for a single Map (2D) or Scene (3D)
  - Manages a collection of GraphicsOverlays
  - Events for user interaction
  - Facilitates MVVM design

• Map / Scene
  - Container for a collection of layers

• Layer
  - Display base maps or geographic features
  - Various types
Demo:
- Follow a graphic
- Create a simple app

Thad Tilton
ArcGIS Runtime SDK for .NET

- *Software Development Kit* is a set of tools for developers
  - **Visual Studio templates**: For all supported platforms
  - **API NuGet packages**: For each platform
  - **Documentation**: Developers Guide, API reference, Samples
  - **Samples viewer**: Source code in GitHub repository
  - **Toolkit**: Open source GitHub repository
  - **GeoNet**: Discussion, blogs
Where to start?

- Developers site
  https://developers.arcgis.com/
  Sign up for free developer account
  Download APIs
  Credits for dev and testing

- GitHub repos
  https://github.com/Esri/
  Toolkit
  Samples
  Demos
  Example Apps

- GeoNet
  https://geonet.esri.com/
  Blogs, discussions, and more
### System requirements

<table>
<thead>
<tr>
<th></th>
<th>Windows 7 SP1</th>
<th>Windows 8.1</th>
<th>Windows 10</th>
<th>macOS 10.12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDE</strong></td>
<td>Visual Studio</td>
<td>Visual Studio</td>
<td>Visual Studio</td>
<td>Visual Studio for Mac</td>
</tr>
<tr>
<td><strong>WPF</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>UWP</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Android</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>iOS</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Forms</strong></td>
<td>No</td>
<td>Android, iOS</td>
<td>Android, iOS, UWP</td>
<td>Android, iOS</td>
</tr>
</tbody>
</table>

- All iOS development requires a Mac computer to act as a build host
- Xamarin development requires VS 2015 Update 3, VS 2017, or VS 2017 for Mac

• All iOS development requires a Mac computer to act as a build host
• Xamarin development requires VS 2015 Update 3, VS 2017, or VS 2017 for Mac
Runtime Example Apps
More than just sample code

• Real world apps based on use cases collected from users
• Complete working apps and getting started data
• Open sourced on GitHub (Apache 2.0 license)
• Supporting documentation (code, data creation, app workflows, customization)
Runtime Example Apps
More than just sample code

• Current apps:
  - .NET SDK:
    - Indoor Routing (iOS)
    - Offline Mapbook (Windows)
  - Android SDK:
    - Maps App, Nearby, Ecological Marine Unit, Offline Mapbook

• Future apps:
  - .NET SDK:
    - Maps App (iOS, Android, and Windows)
    - Indoor Routing (Android and Windows)
  - Android SDK: Tree Survey
  - iOS SDK: Maps App
Runtime Example Apps
More than just sample code

https://developers.arcgis.com/example-apps/

Example Apps Demonstrating the ArcGIS Platform

Learn about the ArcGIS Platform from these complete example applications.

Indoor Routing Xamarin
Find your way around indoor spaces with this iOS app built with the ArcGIS Runtime SDK.

Ecological Marine Unit Android
Explore our ocean ecosystems with Ecological Marine Units, or EMUs using the ArcGIS Runtime SDK.

Maps App Android
Your organisation’s mapping app built with the ArcGIS Runtime SDK.
Demo: SDK tools and resources

Thad Tilton
Universal Windows Platform (UWP)

Rich Zwaap
What is the Universal Windows Platform (UWP)?
One app to rule them all (or more importantly one ArcGIS Runtime SDK to rule them all!)

- Only one Windows 10 operating system (AKA “One Core”)
- Several “Device Families”
Demo: UWP

Rich Zwaap
What is Xamarin?

Rich Zwaap
What is Xamarin?

• Xamarin
  Based on the Mono runtime*
  Compiles into a native Android or iOS app
  Exposes all Android and iOS APIs

• Xamarin is not a cross-platform SDK. It’s a cross-platform language (C#)
  Most of .NET’s core libraries are shareable code
  UI code is very platform specific
  Device code not shareable (Bluetooth, GPS, sensors etc)

• Abstraction-libraries exist that simplifies this
  Xamarin.Forms: Cross-platform UI framework which supports XAML
  Lots of nuget-libraries
The ArcGIS Runtime for Xamarin

Why code-sharing works

• One Common API surface
  Same API on Windows Desktop, UWP, iOS, and Android
  Same underlying code, same functionality
  Most code becomes shareable cross-platform

• Streamlined Development
  Changes inherently apply to all platforms
  All platforms remain in sync

• Tooling in Visual Studio
  Shared projects
Xamarin options
Two primary approaches

• Xamarin Forms: lots of shared code, less control
  - Use XAML to define the UI
    Rendered appropriately for each platform
    ‘Lowest common denominator’ UI elements
  - Basic cross-platform functionality

• Xamarin Native: less shared code, more control
  - Customize UIs with platform-specific elements and designers
  - More platform-specific control
  - Native behavior for user interactions
Which one to pick?

Xamarin.Forms

- Apps that require little platform-specific functionality
- Apps where code sharing is more important than custom UI
- Time until delivery

Xamarin.iOS / Xamarin.Android

- Apps that uses many platform-specific APIs
- Apps where custom UX is more important than code sharing
- Apps that require specialized interaction
Demo: Xamarin apps

Rich Zwaap
Cross platform apps

Rich Zwaap
Cross-platform
The good, the bad, and the ugly

• **Good**
  - Makes your app available to more users
  - Enforces good design patterns

• **Bad**
  - User experience and quality of your app may vary
  - Requires more testing

• **Ugly**
  - Creating platform-specific UIs
  - Handling platform idiosyncrasies (file locations, sensors, security, bugs, etc)
Options for creating cross-platform apps

- HTML5 and JavaScript: Sencha, PhoneGap, Appcelerator Titanium
- C# Development: Xamarin, Alpha Anywhere, Unity 3D

- Cross-platform ArcGIS Runtime SDKs
  - Java, .NET
  - Qt (C++ / QML)
  - Xamarin
Why Xamarin is a good option

- Fully native iOS and Android apps
- Exposes all functionality of the iOS and Android APIs
- Ability to share the majority of an app’s code (60-100%)
- Performance: code is compiled to native binary, not interpreted
- Immediate updates to support iOS and Android releases
- Support for 3rd-party .NET libraries

- Visual Studio and C#!
Organizing your Xamarin code

- Individual project for each platform
  - UI and app code (‘Views’)
- One project for shared code (core)
  - Portable Class Library
  - Shared project

Note: If using Xamarin Forms, UI (.xaml) can be shared
Demo: Cross platform apps

Rich Zwaap
ArcGIS Runtime SDKs: The Road Ahead
Wednesday 1:30 pm - 2:45 pm, Ballroom 6B

• Geopackge layers (vector and raster)
• WMS
• KML
• Time aware layers (2D and 3D)
• Geotransformations
• Enhanced 3D analysis
Esri Developer Summit Europe
October 24-26 Berlin Congress Center Germany

- Pre-Summit Hands-On Training
  - Introduction to the ArcGIS Python API
  - Introduction to the ArcGIS Pro SDK for .NET
  - Introduction to the ArcGIS API for JavaScript
  - ArcGIS Developer Workshop for Esri Distributors and Esri Partners

- Technical Sessions
  - Over 60 sessions on ArcGIS Runtime, Enterprise, Online, Python, JavaScript, Desktop, and Pro

- User Presentations

- Speedgeeking

- Exhibit and Networking Opportunities for Partners
Thank you!

Questions?

Visit us and our team in the ArcGIS Runtime SDK area at the Expo!

Other sessions of interest …

• **ArcGIS Runtime: Building Offline Applications**, **Wednesday 8:30**, Ballroom 5B
• **Building 3D Apps**, **Wednesday 9:30 & Thursday 11:30**, Demo Theater 11
• **ArcGIS Runtime: Building Cross-Platform Apps**, **Wednesday 10:15**, Ballroom 5B
• **ArcGIS Runtime: The Road Ahead**, **Wednesday 1:30**, Ballroom 6B
• **Migrating your Apps from ArcGIS Engine**, **Wednesday 3:30**, Demo Theater 11
• **Maximizing Performance**, **Thursday 10:30**, Demo Theater 11
• **Upgrading Common Workflows from 10.2.x to 100.x with ArcGIS Runtime SDK for .NET**, **Thursday 10:30**, Demo Theater 9