Developing Qt Apps with the Runtime SDK

Thomas Dunn and Michael Tims
Agenda

Getting Started
Creating the Map
Geocoding and Routing
Geoprocessing
Message Processing
Work Offline
The Next Release
Native Apps
A Native App

- Works connected and disconnected from the internet
- Exploits device capabilities
- Works with sensors connected to the device
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
Target Platforms

- OS X
- Desktop
- Desktop Client
- .NET
- JavaSE
- Mobile
- iOS
- Android
- Windows Phone
- Windows Mobile
- Embedded
- Desktop
Runtime is part of the ArcGIS enterprise system

- ArcGIS Online & ArcGIS for Portal
  - Maps, services, content and Organization branding

- ArcGIS for Server
  - Services

- ArcGIS for Desktop
  - Packages
Qt Enables Cross-Platform Development

- **Easy**
  - Cross-platform libraries
  - High-level abstractions

- **Portable**
  - Write once, run anywhere
  - Builds as native C++

- **Open**
  - Pre-built platforms
  - Source code
Extensive cross-platform development library

Qt components:
- Graphics
- System
- Networking
- Signals and Slots
- Containers
- Text
- UNICODE Translation (I18N)
- XML
- QML
- Consistent style
- Threading
- Memory management
- SQL Sockets
Getting Started
A brief word about **System Requirements** at 10.2.3

- **Operating Systems (development and deployment)**
  - Windows 7, 8 and 8.1, Server 2008 R2, 2012 and 2012 R2
  - Red Hat 6.2
  - Ubuntu 12.04 LTS
  - 32 or 64-bits

- **Qt SDK from** [http://qt-project.org](http://qt-project.org)
  - Version 5.1.1 for Windows or Linux

- **Qt Creator IDE**
  - Version 2.7 or higher
Creating a map
Map Layers

Layers are added to the map in order, bottom-to-top

Operational layers: dynamic features
Facilities, buildings, zones, networks

Basemap layer: spatial context
Imagery, topography

Graphics layers: live / temporary data:
Vehicles, people, events
Spatial Reference

- Decide on the spatial reference to be used by your map
  - e.g. Web Mercator Auxiliary Sphere, WGS-84, UTM …

- Use a tiled service or tile package for your base map that uses that spatial reference
  - Tiled basemaps cannot be re-projected

- Spatial reference of map determined by first layer added
Provisioning Content

• **Online Services:** access these via a URL
  - Tiled services, dynamic map services, feature services, Geoprocessing services, Geocoding services

• **Local Content:** author packages and deploy them with your app
  - Tile Packages as a high performance base map layer
  - Map Packages for querying or editing
  - Geoprocessing packages built from ArcGIS models or Python scripts
  - Geocoding locator packages or locator files
Creating the Map
Geocoding

Santa Claus
North Pole

Where is this address?
(geocoding)

Van Ittersumstraat 14,
9621 CV Groingen,
The Netherlands
(reverse geocoding)

Reverse
Geocoding
Task-Based Framework

1. Construct a task object
2. Provide parameters
3. Connect slots for completion signals
4. Execute task
5. Grab results in slot when task completes

Runtime tasks:
- Locator (geocoding)
- Routing
- Geoprocessing
- Geodatabase sync
- Find
- Query
- Identify features
- Find closest facility
- Calculate service area
- Tile cache generation
Geocoding and Routing
Geoprocessing

• Geoprocessing provides advanced GIS analysis in Runtime
  - See Qt Sample Application for some samples
  - [https://developers.arcgis.com/qt/guide/supported-geoprocessing-tools.htm](https://developers.arcgis.com/qt/guide/supported-geoprocessing-tools.htm)

• Use one tool, or build a Model or Python script to chain tools

• Display results on map or use in subsequent processing

• Connected
  - ArcGIS Server and ArcGIS Online geoprocessing services

• Disconnected
  - Local geoprocessing service from geoprocessing packages (GPKs)
Message Processor

- Streamline display of command and control symbols
  - Message communicates a symbol’s type, location and status
  - Message Processor displays the symbol on a special graphics layer

- Works with standard symbol dictionaries
  - MIL-STD-2525C
  - App6B
Using the Message Processor

• Create MessageGroupLayer object, add to map
• Get reference to MessageProcessor object that was created within the MessageGroupLayer
• For each message received from outside source
  1. Construct message object from message content
  2. Pass message object to the MessageProcessor
  3. MessageProcessor interprets the message and adds / updates / deletes symbol in MessageGroupLayer
Work Offline

- Download and view basemaps from Esri or your own authoritative basemaps
- Edit data from feature services (hosted in the cloud or on premises) and sync changes back
- Query and analyze your operational data locally
- Search for places (geocode and reverse geocode) using your organization’s spatial data assets
- Find optimal routes using your organization’s network dataset
Work Offline
Runtime Licensing

Development and Deployment Workflow

1. Download and Install
2. Develop and Test
3. Deploy and Distribute
## 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>
<tr>
<td>Basic</td>
<td>Connected - all functionality</td>
</tr>
<tr>
<td></td>
<td>Offline - map viewing only</td>
</tr>
<tr>
<td>Standard</td>
<td>Connected and offline - all functionality, includes:</td>
</tr>
<tr>
<td></td>
<td>• Local locators (geocoding)</td>
</tr>
<tr>
<td></td>
<td>• Local routing</td>
</tr>
<tr>
<td></td>
<td>• Local geodatabase editing</td>
</tr>
<tr>
<td></td>
<td>• Local geodatabase sync operations</td>
</tr>
<tr>
<td></td>
<td>• Local server *</td>
</tr>
</tbody>
</table>

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

- 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 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

1. 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
Licensing and Deployment

- Qt
- OS X
- Windows
- Store
- Desktop Client
- JavaSE
- .NET
- Mobile
- iOS
- Android
- Mobile
- Windows
- Phone
- Desktop
- Embedded
- .NET
- Desktop
- Client
- JavaSE
- .NET
- Desktop
- Client
What’s next for Runtime?
Next Release

- QML (Qt Quick) API
- Direct read of raster datasets*
- Direct read of vector data (KML, Shapefiles)
- 3D*
- Toolkits
- Common conceptual model
- More offline capabilities
  - Feature service table
  - Versioned data support

*Some SDKs will release items before others
QML supported platforms

- Windows
- Linux
- OS X
- iOS
- Android
Qt
## ArcGIS Runtime SDK sessions Tuesday

<table>
<thead>
<tr>
<th>Session Name</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating Your WPF Apps to ArcGIS Runtime SDK for .NET</td>
<td>12:30pm – 1:00pm</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>ArcGIS Runtime Offline APIs</td>
<td>3:30pm – 4:00pm</td>
<td>Developer Island (demo theatre)</td>
</tr>
<tr>
<td>Developing Apps with ArcGIS Runtime SDK for Android</td>
<td>8:30am – 9:45am</td>
<td>Room 15 B</td>
</tr>
<tr>
<td>Developing iOS and Mac Apps with ArcGIS Runtime SDK</td>
<td>10:15am – 11:30am</td>
<td>Room 31 C</td>
</tr>
<tr>
<td>Developing Java Apps with ArcGIS Runtime SDK</td>
<td>3:15pm – 4:30pm</td>
<td>Room 31 C</td>
</tr>
<tr>
<td>Session Name</td>
<td>Time</td>
<td>Location</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------------------------</td>
</tr>
<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>Building .NET Apps with ArcGIS Runtime SDK: Tips and Tricks</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>
Thank you...

- Please fill out the session survey:

  Offering ID: 1218

  Online – www.esri.com/ucsessionssurveys
  Paper – pick up and put in drop box