Practical Applications of JSR 179

Ryan Wick — Sr. Engineer

Zane Lyon — Manager

Sprint Nextel—Advanced Wireless Solutions

http://developer.sprint.com

TS-1515

Goal of This Talk

Learn how to build portable location aware mobile solutions using Location API for J2ME™ (JSR 179) and accommodate implementation irregularities

Agenda

Review JSR 179 Goals, Design, and Best Practices

Implementation Variances and Portability Challenges

Case Study: Tracking Application

Summary

Agenda

Review JSR 179 Goals, Design, and Best Practices

Implementation Variances and Portability Challenges

Case Study: Tracking Application

Summary

JSR 179 Overview

Location on Cellular Networks

- Location capabilities on handsets started to appear in 2002
  - Primary push was to meet FCC E-911 requirements
- Carriers deploy different technologies to meet E-911 mandate
- Not all carriers designed a solution with building location services in mind

JSR 179 Overview

Background

- Devices and networks implement location determination using very different underlying technologies and approaches
  - GPS/assisted GPS
  - Location of cell site/sector
  - AFLT
  - Short-range (Bluetooth™, WLAN)
  - Hybrid approaches
- Goal of the standard is to abstract any underlying detail of implementation and publish generalized configuration parameters

JSR 179 Overview

javax.microedition.location.LocationProvider

- Starting point for location request
- Specify Criteria to get proper provider

```
getInstance(Criteria criteria)
```

- One-shot fix based on criteria

```
getLocation(int timeout)
```

- Set up a LocationListener to get updates

```
setLocationListener(LocationListener l, int interval,
  int timeout, int maxAge)
```

JSR 179 Overview

javax.microedition.location.Criteria

- Specifies the best suited LocationProvider for your needs
- Knobs to turn
  - Cost, speed and course, horizontal/vert accuracy, power consumption, response time, alt, address
- Criteria class is key to influencing how the device will acquire location
  - Interpretation of settings by implementation not dictated by specification or standards

JSR 179 Overview

javax.microedition.location.Criteria

```
Criteria cr = new Criteria();
cr.setCostAllowed(true); //default value
cr.setSpeedAndCourseRequired(true);
cr.setHorizontalAccuracy(500);
cr.setAltitudeRequired(true);
lp = LocationProvider.getInstance(cr);
```

JSR 179 Overview

javax.microedition.location.LocationListener

- Receives updates from a LocationProvider

```
setLocationListener(listener,interval,timeout,maxAge);
```

  - Interval—how often you want fix updates, in seconds
  - Timeout—how late update can be from interval, in seconds
  - maxAge—acceptable age for cached info to be returned
- Best effort is made to return in specified interval but should not be considered a timer
- Updates should be considered a trigger, processing should be handled in separate thread

JSR 179 Overview

javax.microedition.location.LocationListener

```
class NavigationHelm implements LocationListener{
  void locationUpdated(LocationProvider p, Location l){
    // call another thread to process
    handler.handleUpdate(l);
  }
  void providerStateChanged(LocationProvider p, int
   state){
    // not common for state to change
  }
}
```

JSR 179 Overview

javax.microedition.location.Location

- Response for both positive and negative attempts

```
Location.isValid()
```

- Location source (CELLID, SATELLITE, TOA)

```
Location.getLocationMethod()
```

- Retrieve coordinates, speed, course, and time
- Getting formatted data (NMEA, LIF) and detailed error description

```
Location.getExtraInfo(String mimetype)
getExtraInfo("application/X-jsr179-location-
  nmea")
getExtraInfo("text/plain")
```

JSR 179 Overview

Tradeoffs and Considerations

- The choices you make in your criteria and application design may affect
  - Battery life
  - Latency in the application
  - Adequacy of precision
  - Whether or not you get a usable location at all (e.g., GPS indoors)
  - Cost

Agenda

Review JSR 179 Goals, Design, and Best Practices

Implementation Variances and Portability Challenges

Case Study: Tracking Application

Summary

Quick Sidebar

What Did Sprint Nextel Do?

- Nextel (Sprint's iDEN network)
  - A-GPS
    - Handset receives assist data or Ephemeris data to quicken the TTF (time-to-first fix)
    - Handset listens for GPS satellites and calculates locally the position
- SprintPCS (Sprint's CDMA network)
  - GPS/Hybrid
    - Handset listens for GPS satellites
    - Handset forwards results to PDE (Position Determining Equipment)
    - PDE uses satellite data and potentially network data to calculate location of handset

Portability Challenges

- Platform specific configuration items not addressed in JSR 179 API
- Interpretation of criteria mapping
- Support for LocationListener callbacks
- Support for multiple providers
- Proximity listener implementation differences
- Carrier/device code signing requirements

Portability Challenges

Settings Not Covered in JSR 179

- No generic facility in JSR 179 for setting platform specific configurable items
- Examples:
  - IP and Port settings for PDE; requires importing a carrier specific utility class to setup the Location environment
  - Bluetooth and WLAN settings
  - Privacy settings
- Could be handled by platform or application manager, outside of application

Portability Challenges

Inconsistent Criteria Mappings

- Inconsistent interpretations of criteria mappings across implementations
- Examples:
  - PreferredPowerConsumption—High/Med/Low—Used by some implementations to determine GPS vs. Cell
  - AltitudeRequired—Some implementations won't return until altitude is calculated; others use this simply to determine whether or not to include the info in the Location returned
  - CostAllowed—Different significance on different implementations
  - Platforms give different precedence to various criteria

Portability Challenges

Supporting LocationListener

- Not all LocationProviders allow you to register a LocationListener
- Differing behaviour if a position is not available at the time of the callback

Portability Challenges

Using Multiple Providers at Once

- Platforms may or may not allow a single app to register multiple providers at the same time
  - Use Case:
    - Using LocationListener with a “GPS” provider
    - If call to listener indicates a failure use a second provider to give a less accurate location, such as “cell-site location”
- Platforms may or may not support running multiple applications accessing location services simultaneously
  - Use case:
    - Running a “breadcrumbing” tracking application while using a navigation app to find a route

Portability Challenges

Proximity Listener Implementation Differences

- Differences in update timing
  - On a time schedule
  - Different for each type of provider
  - Dynamic depending on movement
  - Dynamic depending on projected potential to cross proximity
- May or may not be able to update by calling

```
Location.getLocation(int timeout)
```

Portability Challenges

Code Signing Requirements

- Not specific to JSR 179, but critical to porting your application across devices and networks
- Different signing processes and certificates required
- May require device to be “unlocked” just to test
- May require carrier verification of application before going to market or releasing to production

Portability Challenges

Device-Specific Quirks

- Not specifically a JSR 179 compliance issue, but another potential barrier to porting your applications
- Platform work-arounds that should be built into API implementation
- For lack of a euphemism … “bugs”

Agenda

Review JSR 179 Goals, Design, and Best Practices

Implementation Variances and Portability Challenges

Case Study: Tracking Application

Summary

Case Study—Tracking Application

Simple Requirements

- When user clocks in to application position is acquired every five minutes and stored locally on the device
- Every 30 minutes coordinates are transmitted back to server application

Case Study—Tracking Application

Implementation Details

- Attempt to acquire high precision position
- Switch to lower precision if high precision fails to return
- Implement a LocationListener to receive the results from the high precision provider
- Attempt to write portable solution

Case Study—Tracking Application

Coding Demo Setup

- Motorola iDEN sample
- Qualcomm-based VM on CDMA
- Blackberry sample

DEMO

Build JSR 179 Applications in the New Sprint SDK from Sun

Agenda

Review JSR 179 Goals, Design, and Best Practices

Implementation Variances and Portability Challenges

Case Study: Tracking application

Summary

Summary

- Compliance doesn't guarantee portability
- Know your implementation and position determination mechanism
- Test, test, test

For More Information

- http://developer.sprint.com
- http://www.jcp.org/en/jsr/detail?id=179
- http://www.jcp.org/en/jsr/detail?id=293

Q&A

Ryan Wick
Zane Lyon

Practical Applications of JSR 179

Ryan Wick — Sr. Engineer

Zane Lyon — Manager

Sprint Nextel—Advanced Wireless Solutions

http://developer.sprint.com

TS-1515