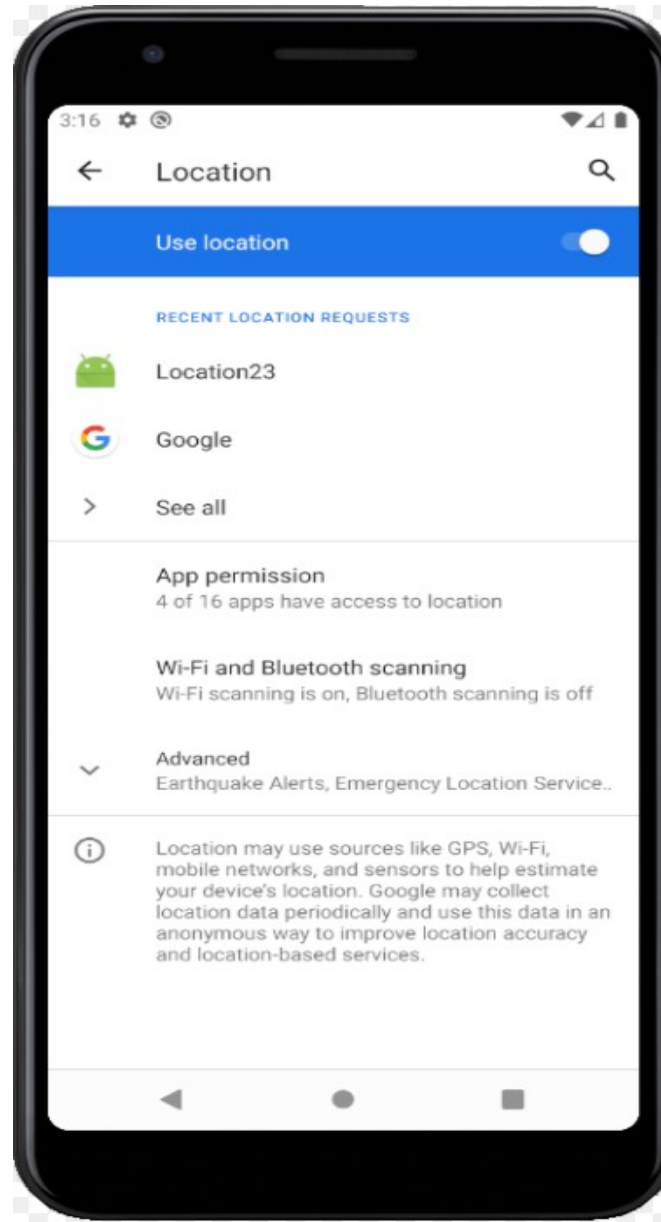


Location Based Services (LBS) and Geocoding

Location Based Services (LBS)

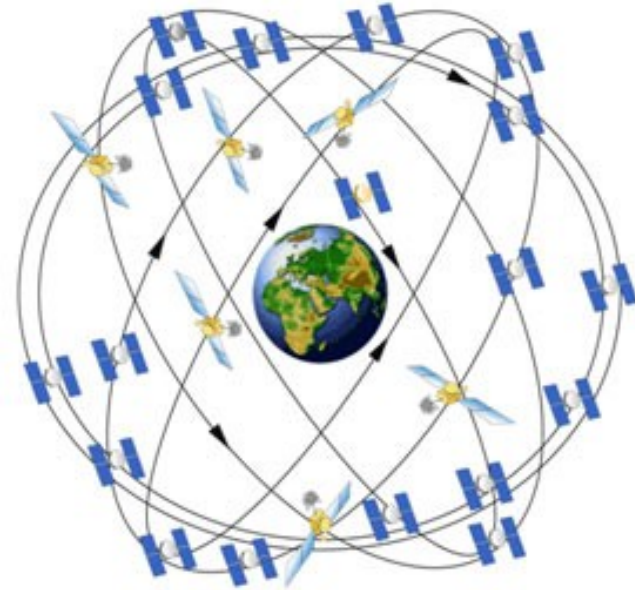
- **Location and Google maps are separate!!**
- Location for Android devices include:
 - GPS
 - cell-ID (cell tower)
 - Wi-Fi networks
- Geocoding – converting from latitude and longitude to street address

LBS System Settings



GPS

- 31 core satellites
- medium earth orbit, 20k km above the earth
- 6 orbital planes with 4 satellites each
- generally 4 satellites in line of sight at any spot on the earth

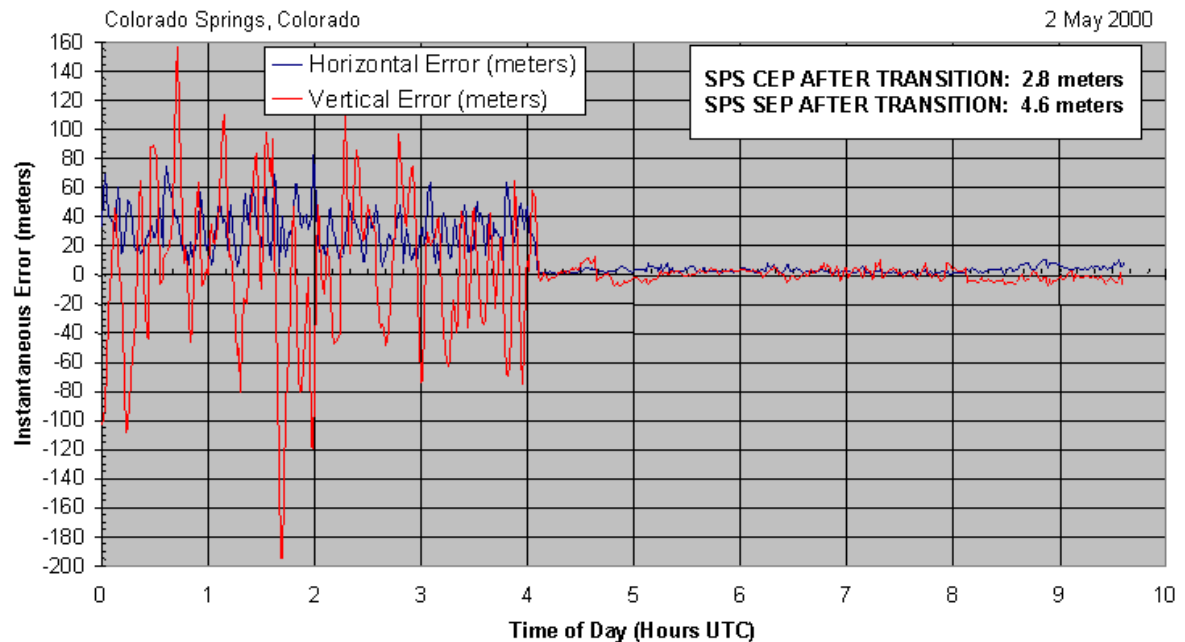


GPS Accuracy

- Selective Availability: intentional degradation of signals for civilian use
 - ended in 2000



SA Transition -- 2 May 2000



GPS

- Accurate within 20ft
- only works OUTDOORS
- quickly consumes battery power
- Slow to acquire satellites or re-acquire if lost

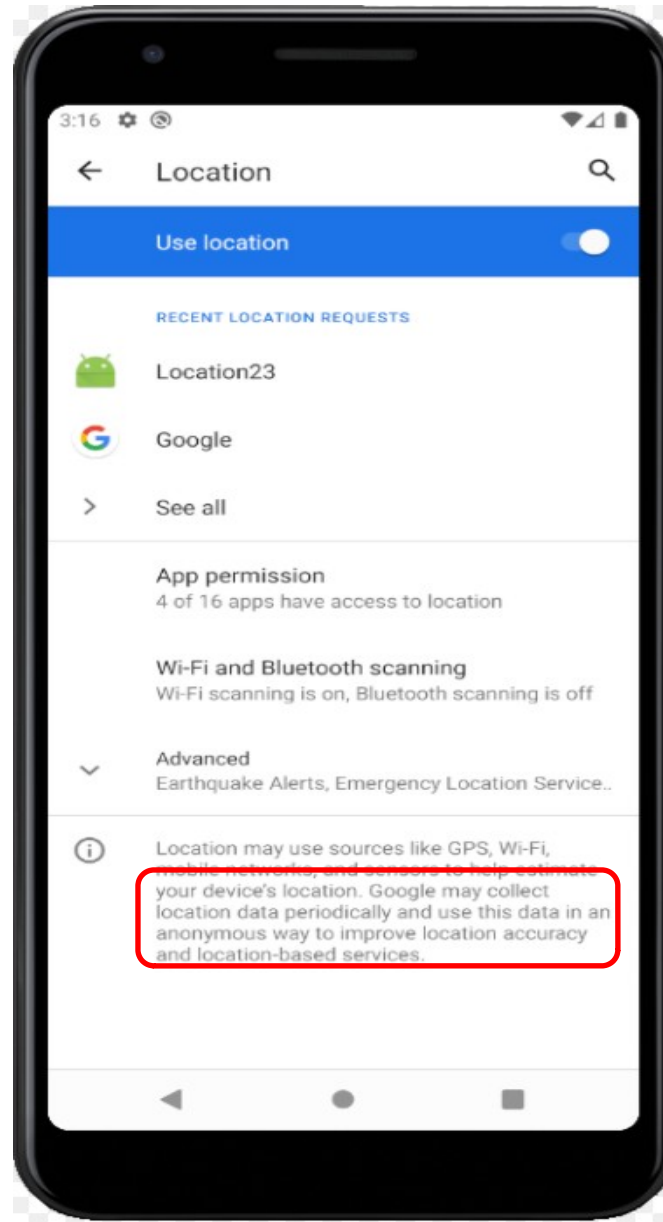


How does Wi-Fi Fix Location?

- Used to use StreetView cars
- Now, use devices themselves to map locations to wi-fi spots
- Apple and Microsoft do the same thing



Google Location Services

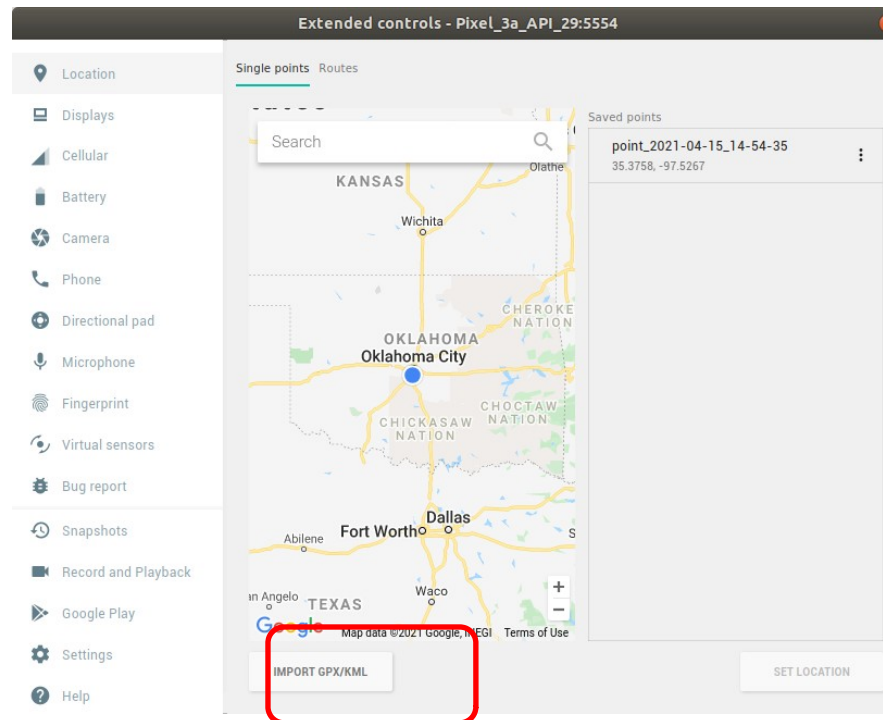


WiFi or Network

- Accurate within 1 mile
- Works inside and out
- Low power consumption
- Fast location acquisition

Emulator Sidebar

- Use extended Emulator Location controls
- Import GPX location files



LBS How to use

- Add appropriate permission to `AndroidManifest.xml`
- Get instance of `LocationManager` using `getSystemService` method using `LOCATION_SERVICE`
- Choose location provider Implement a `LocationListener` class
- Call `requestLocationUpdates` method with chosen provider so `LocationListener` can start receiving location information

Manifest Permissions

- FINE = GPS

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

- COURSE = WiFi (network) or cell

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

LBS- Main Classes

- LocationManager
 - Obtain current location
 - Track movement
 - Set proximity alerts for areas
 - Find available Location Providers
- Location Provider
 - Various location-finding technologies

Location Manager

- Gatekeeper
 - Selects particular location providers
 - Lists available location providers
 - Registers for location updates
- Can be simple to use

```
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

Location Providers

- Entities that encapsulate the functionality of a particular location provider
- Distinguished directly by
 - LocationManager.GPS_PROVIDER
 - LocationManager.NETWORK_PROVIDER
 - LocationManager.PASSIVE_PROVIDER

```
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

Location Providers

Select by Criteria

- Can ask for a location provider indirectly by criteria;
- Suppose low power consumption is the most important criteria.

```
Criteria criteria = new Criteria();  
criteria.setPowerRequirement(Criteria.POWER_LOW);  
provider = locationManager.getBestProvider(criteria, false);  
Location location = locationManager.getLastKnownLocation(provider);
```

- Use Criteria to filter on accuracy of returned location, accuracy of bearing, speed and altitude etc.

Location Based Services Strategy

1. Get LocationManager object
2. Register and listen for Location Updates
3. Unregister when app no longer visible

Update Current Location

LocationListener

- Listens for location changes

Public Methods	
abstract void	<code>onLocationChanged (Location location)</code> Called when the location has changed.
abstract void	<code>onProviderDisabled (String provider)</code> Called when the provider is disabled by the user.
abstract void	<code>onProviderEnabled (String provider)</code> Called when the provider is enabled by the user.
abstract void	<code>onStatusChanged (String provider, int status, Bundle extras)</code> Called when the provider status changes.

Update Current Location

- Register the `LocationListener` to receive location updates
- `locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 15000, 10, locationListener);`
 - provider: name of provider to register with
 - minTime: the minimum time interval for notifications, in milliseconds. only a hint to conserve power, and actual time between location updates may be greater or lesser than this value.
 - minDistance: min distance interval for notifications in meters
 - the listener itself

STOP Updating Current Location

- Stop getting updates when you are finished
- `locationManager.removeUpdates(my
LocationListener);`

Location Listener

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE)

// Define a listener that responds to location updates
LocationListener locationManagerListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationManagerListener);
```

LocationManager - Useful Methods

- `addProximityAlert(double latitude, double longitude, float radius, long expiration, PendingIntent intent)`
 - Sets a proximity alert for the location given by the position (latitude, longitude) and the given radius.
- `List<String> getAllProviders()`
 - Returns a list of the names of all known location providers.
- `Location getLastKnownLocation(String provider)`
 - Returns a Location indicating the data from the last known location fix obtained from the given provider.

Best Practices

- Battery Life verses Accuracy
- Startup time
- Update Rate

Location Based Services

GeoCoding

- Translate between street addresses and latitude/longitude coordinates
- Uses a webservice
 - Many geocoding services available
- Forward Geocoding
 - Finds Latitude/longitude of an address
- Reverse Geocoding
 - Finds street address given latitude/longitude

Location Based Services

Geocoding

1. Uses a webservice so do the call in a separate thread
2. Pass it a location object
3. It will return an address

References

- <https://developer.android.com/training/location/index.html>
- There are a lot of tutorials here