Understanding GPS Accuracy
Mastering Android SDK Geolocation

Andy Gup
www.andygup.net
@agup

Goals/Objectives

Go beyond the native SDK documentation

Use Cases, Use Cases, Use Cases

Coding Patterns

Android GPS Test Tool

Save you hours, days or weeks of ramp-up

Assumptions

Familiar with Android SDK

Worked with Android projects

Understand Java

Who am I?

Andy Gup
Developer Evangelist
www.andygup.net
github.com/andygup
agup@esri.com
@agup

esri

This is not about...

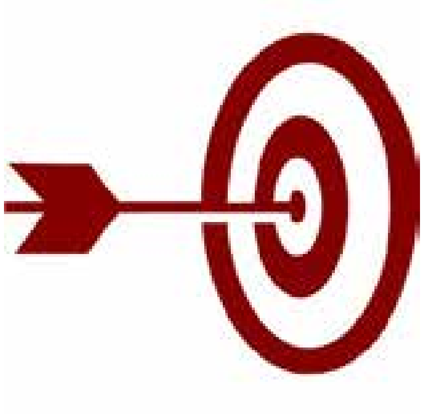
New Google Play Services SDK

Fused Location Provider
Activity Recognition
Geofencing APIs

≠

http://developer.android.com/google/play-services/location.html

Accuracy



Depends on many things:

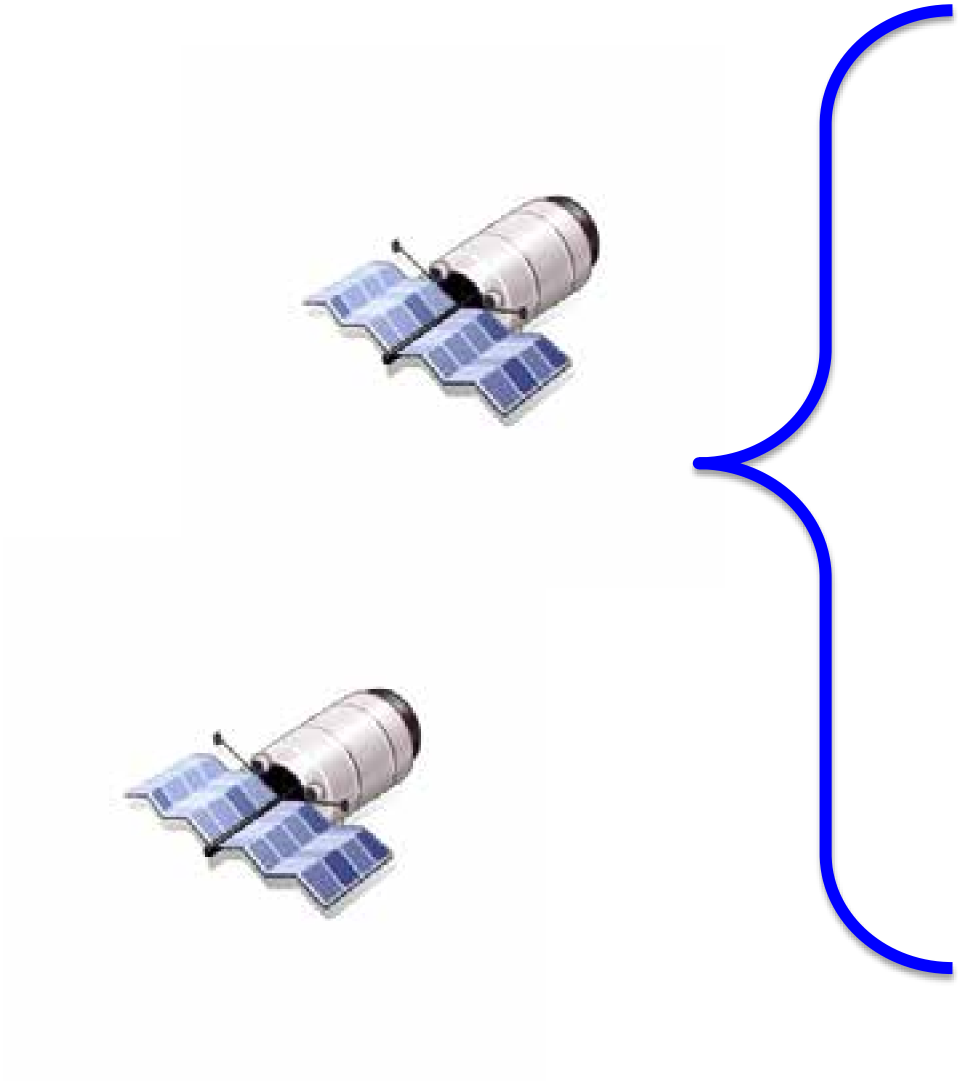
- Device type
- Which providers enabled
- Internet connectivity
- Cellular provider

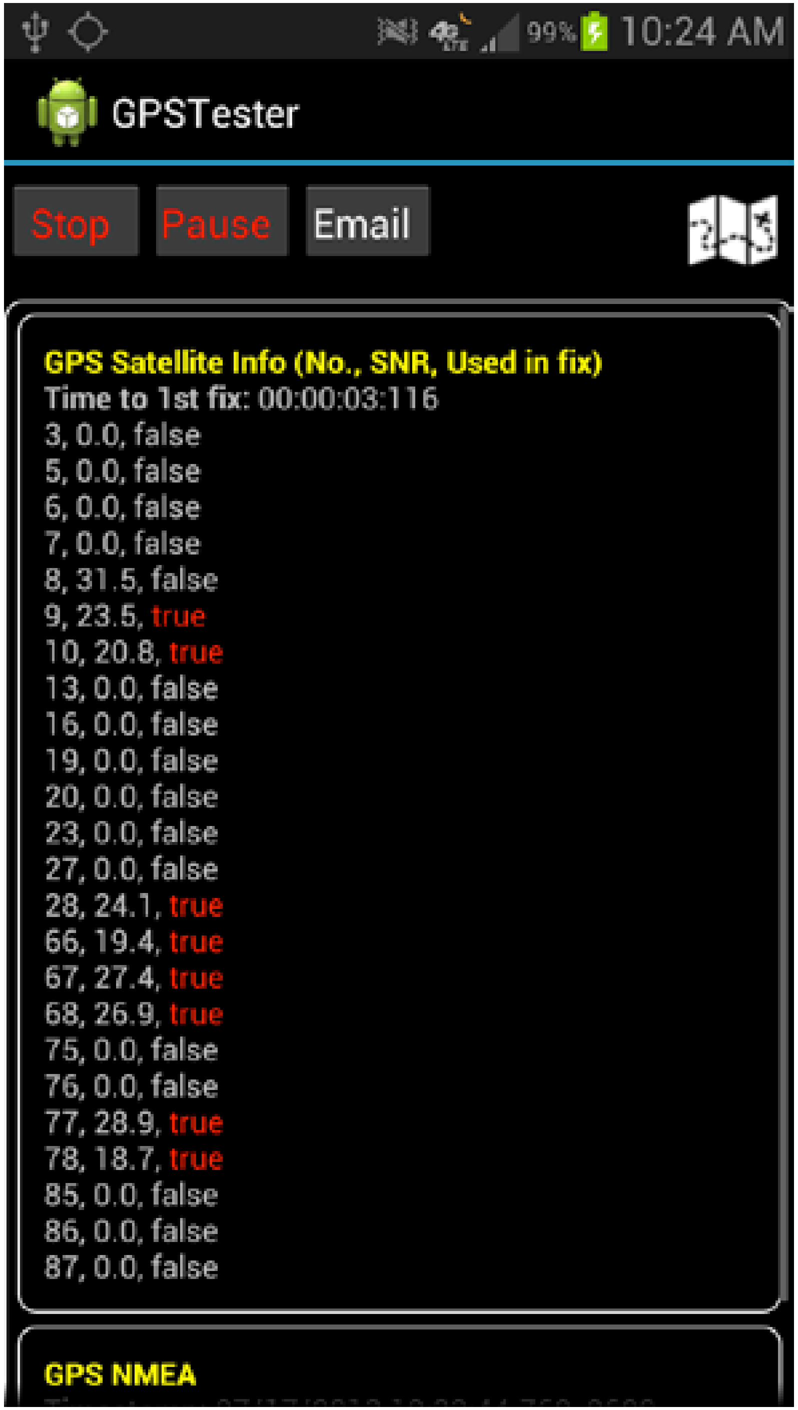
Accuracy

Depends on many things:

- In a parked car
- In a moving car
- In a building
- In a parking garage

Accuracy



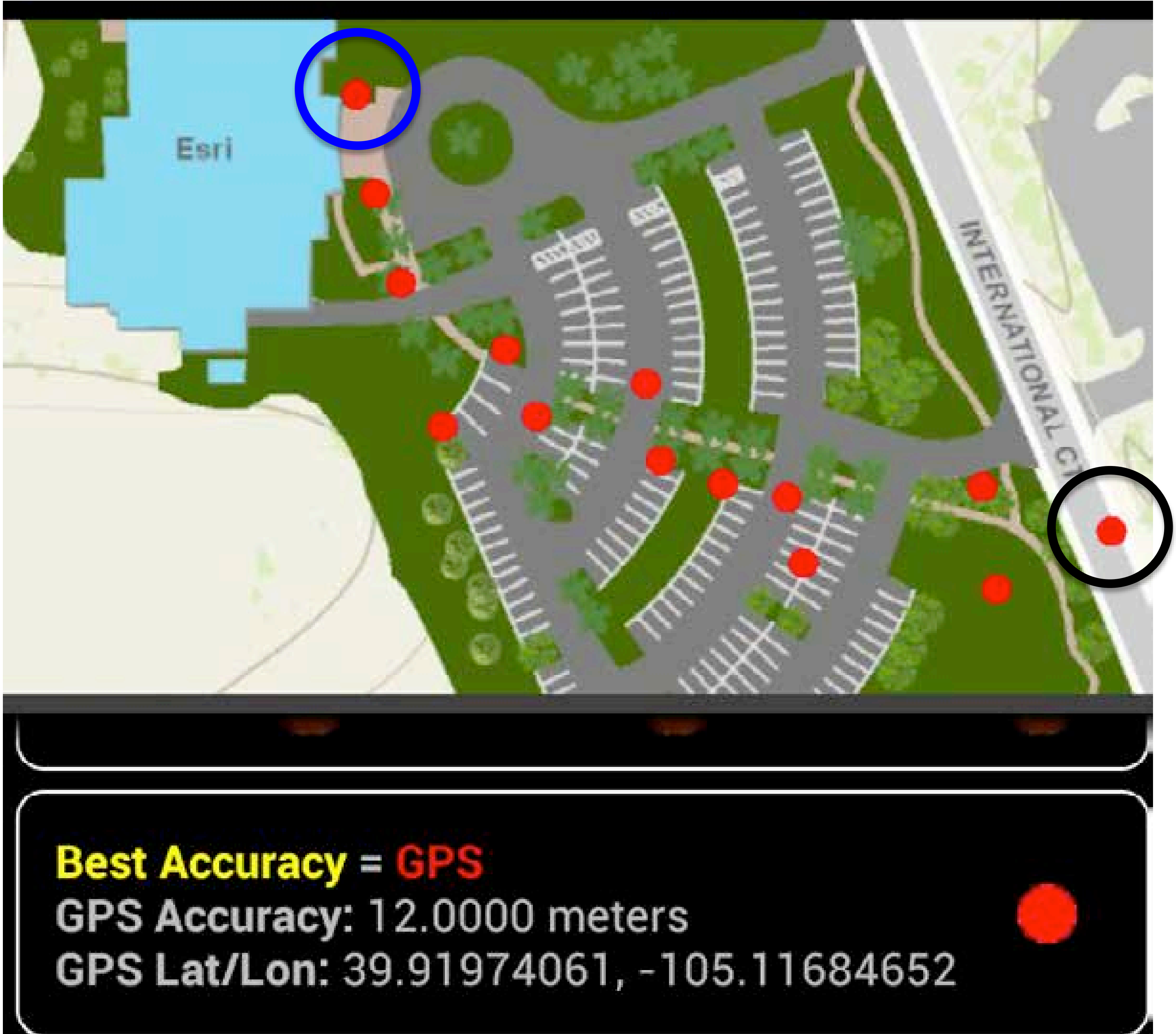


GPS Satellite Info (No., SNR, Used in fix)

Time to 1st fix: 00:00:03:116

```
3, 0.0, false
5, 0.0, false
6, 0.0, false
7, 0.0, false
8, 31.5, false
9, 23.5, true
10, 20.8, true
13, 0.0, false
16, 0.0, false
19, 0.0, false
20, 0.0, false
23, 0.0, false
27, 0.0, false
28, 24.1, true
66, 19.4, true
67, 27.4, true
68, 26.9, true
75, 0.0, false
76, 0.0, false
77, 28.9, true
78, 18.7, true
85, 0.0, false
86, 0.0, false
87, 0.0, false
```

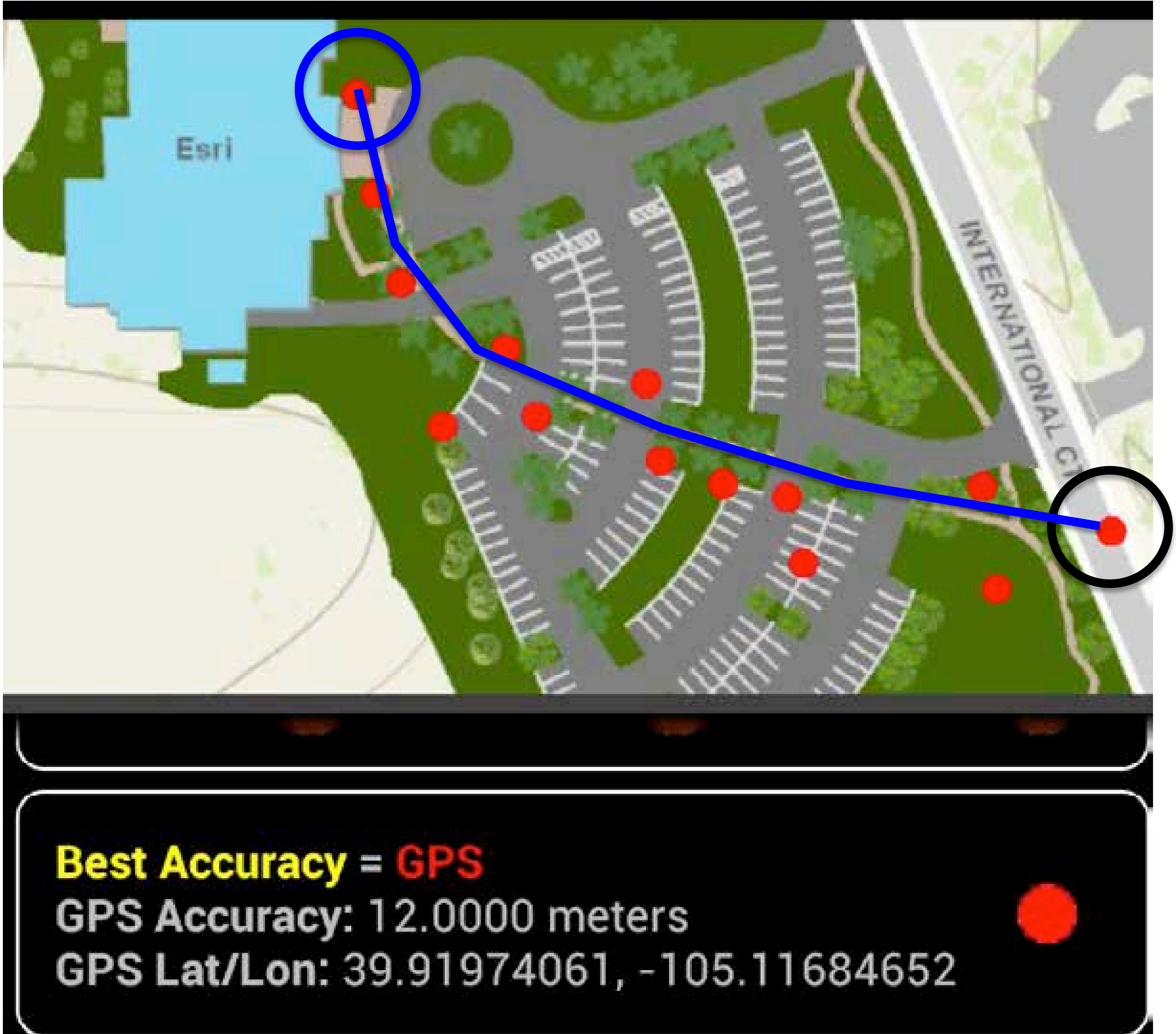
GPS NMEA



Best Accuracy = **GPS**

GPS Accuracy: 12.0000 meters

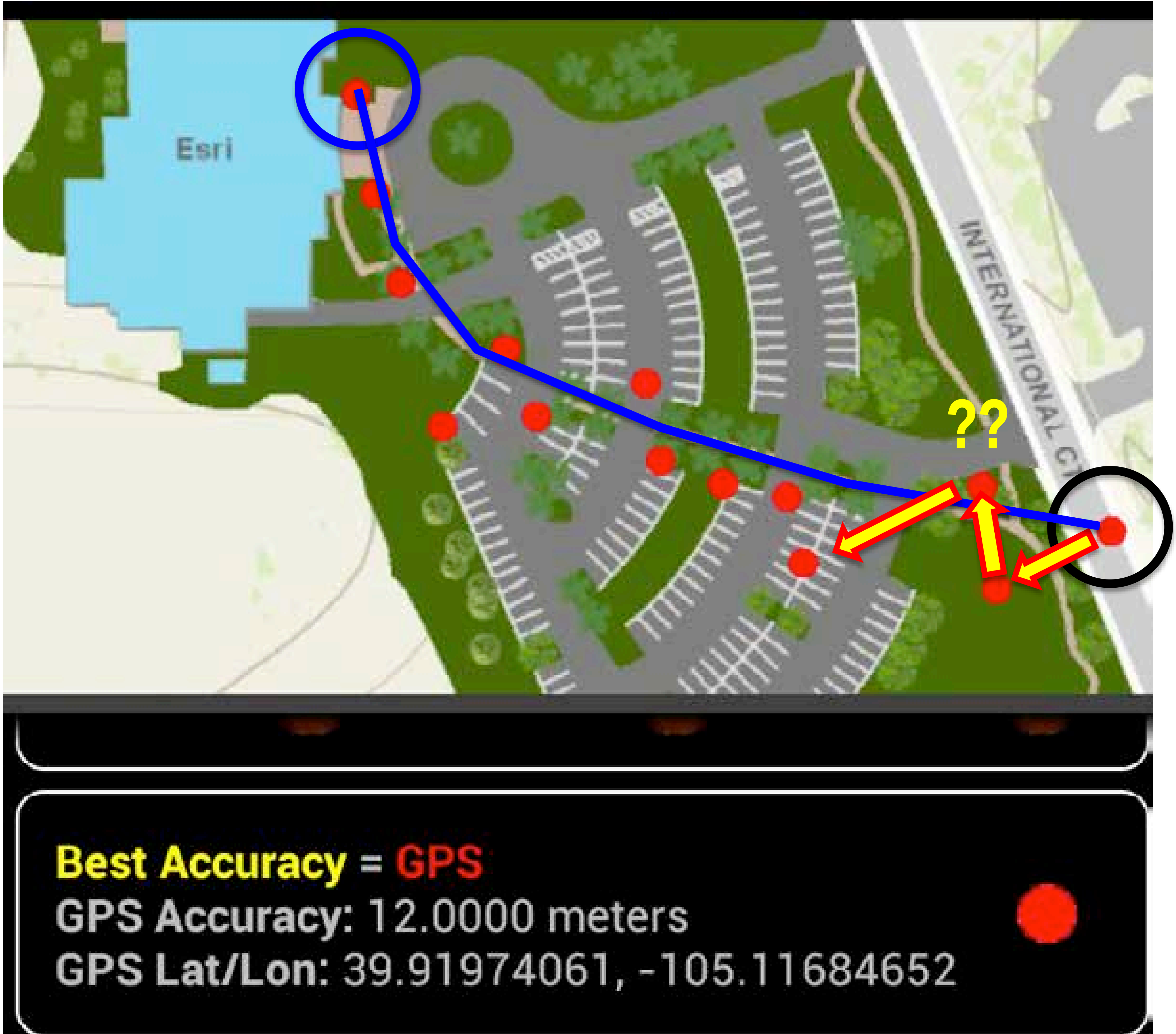
GPS Lat/Lon: 39.91974061, -105.11684652



Best Accuracy = **GPS**

GPS Accuracy: 12.0000 meters

GPS Lat/Lon: 39.91974061, -105.11684652



Best Accuracy = GPS

GPS Accuracy: 12.0000 meters

GPS Lat/Lon: 39.91974061, -105.11684652

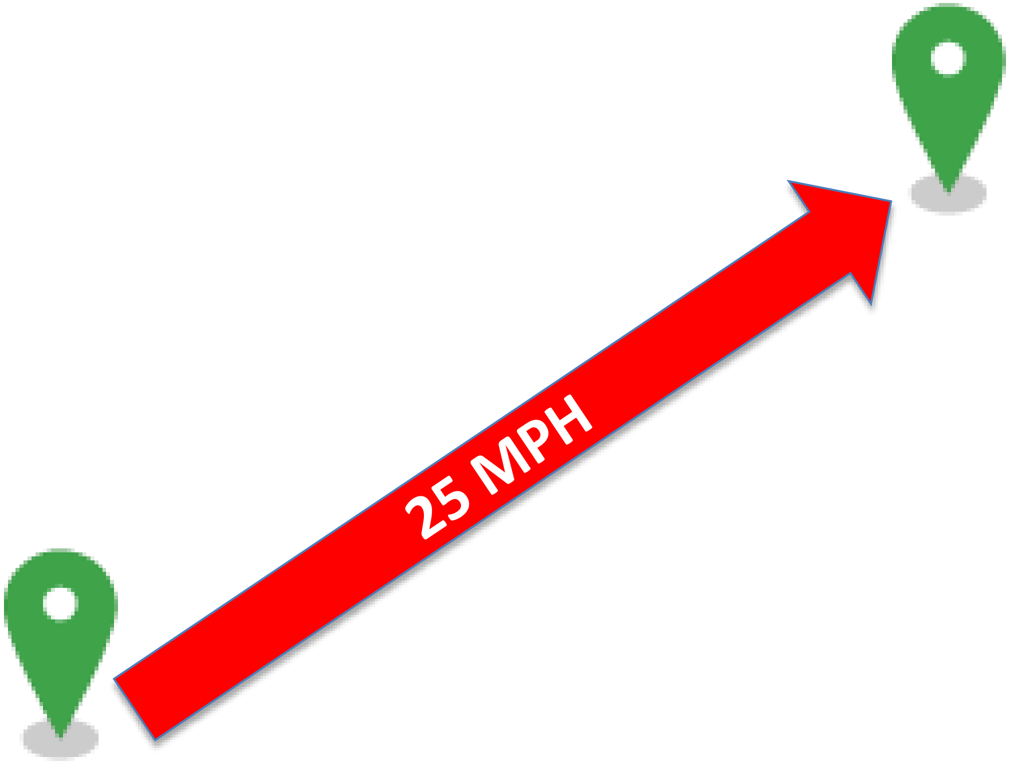
Define accuracy

Time

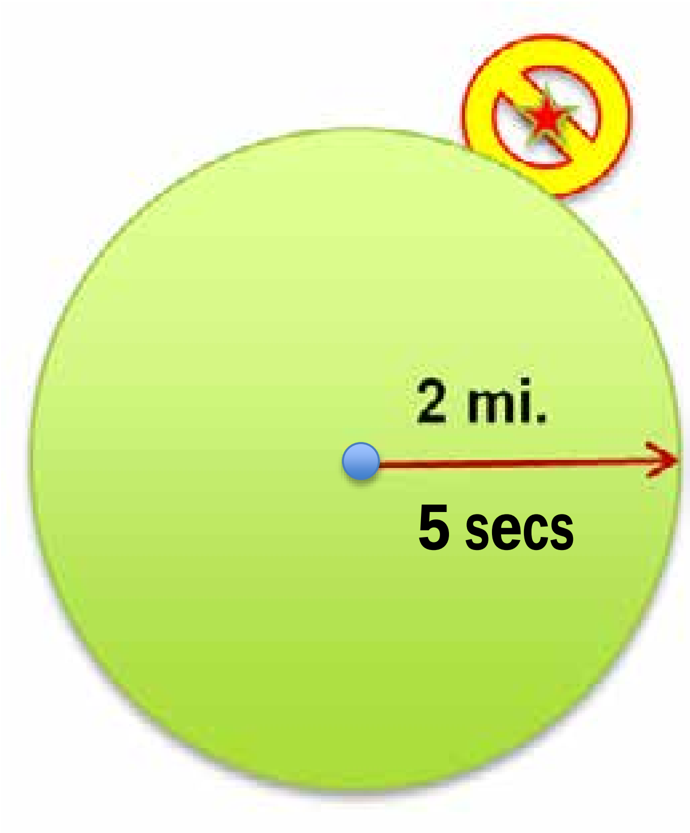
Distance

Speed

Heading



Reject bad results



2 mi.

5 secs

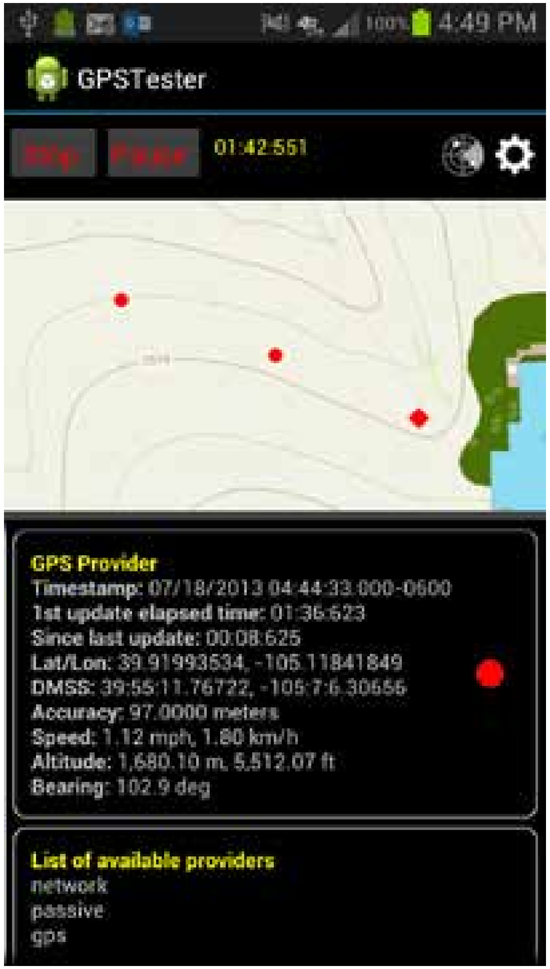
East (270 deg)

AndroidManifest.xml

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

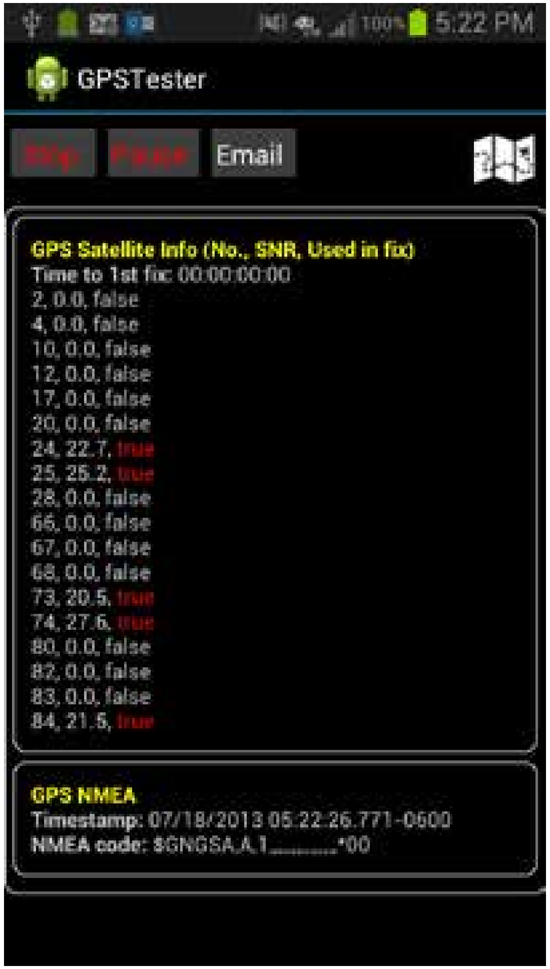
Android GPS Test Tool

https://github.com/Esri/android-gps-test-tool



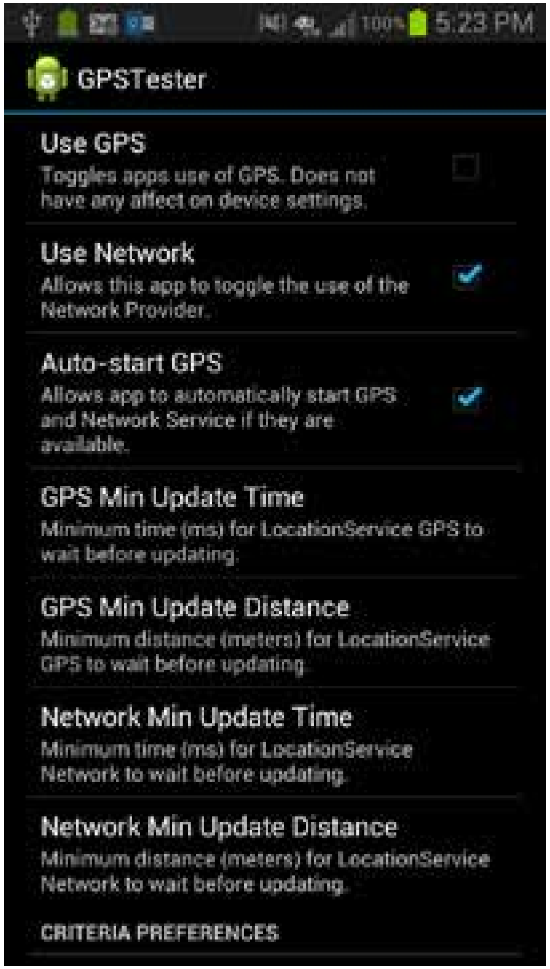
GPS Provider
Timestamp: 07/18/2013 04:44:33.000-0600
1st update elapsed time: 01:36:623
Since last update: 00:08:625
Lat/Lon: 39.91993534, -105.11841849
DMSS: 39:55:11.76722, -105:7:6.30666
Accuracy: 97.0000 meters
Speed: 1.12 mph, 1.80 km/h
Altitude: 1,680.10 m, 5,512.07 ft
Bearing: 102.9 deg

List of available providers
network
passive
gps



GPS Satellite Info (No., SNR, Used in fix)
Time to 1st fix: 00:00:00:00
2, 0.0, false
4, 0.0, false
10, 0.0, false
12, 0.0, false
17, 0.0, false
20, 0.0, false
24, 22.7, true
25, 25.2, true
28, 0.0, false
66, 0.0, false
67, 0.0, false
68, 0.0, false
73, 20.5, true
74, 27.6, true
80, 0.0, false
82, 0.0, false
83, 0.0, false
84, 21.5, true

GPS NMEA
Timestamp: 07/18/2013 05:22:26.771-0600
NMEA code: $GNGSA,A,1,,,,,,,,,,,,,,*00



Use GPS ☐
Toggles apps use of GPS. Does not have any affect on device settings.

Use Network ☑
Allows this app to toggle the use of the Network Provider.

Auto-start GPS ☑
Allows app to automatically start GPS and Network Service if they are available.

GPS Min Update Time
Minimum time (ms) for LocationService GPS to wait before updating.

GPS Min Update Distance
Minimum distance (meters) for LocationService GPS to wait before updating.

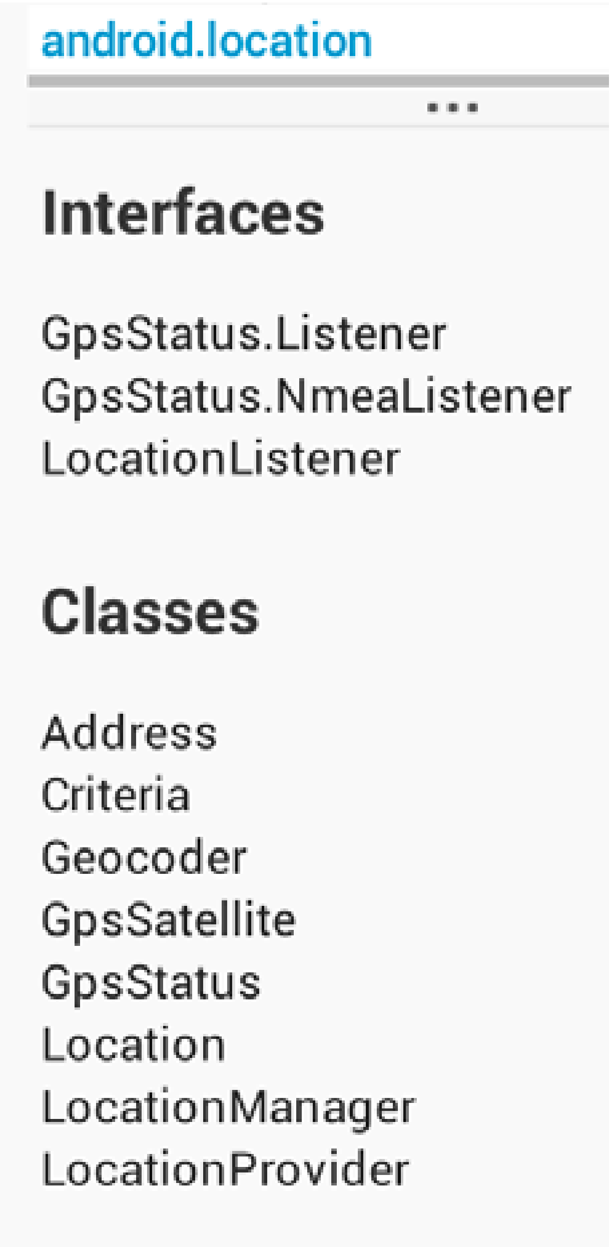
Network Min Update Time
Minimum time (ms) for LocationService Network to wait before updating.

Network Min Update Distance
Minimum distance (meters) for LocationService Network to wait before updating.

CRITERIA PREFERENCES

package

android.location



android.location

Interfaces

GpsStatus.Listener
GpsStatus.NmeaListener
LocationListener

Classes

Address
Criteria
Geocoder
GpsSatellite
GpsStatus
Location
LocationManager
LocationProvider

android.location.LocationManager

Provides access to system location services.

android.location.LocationProvider

Provides geographic location

Has a set of criteria

GPS_PROVIDER

NETWORK_PROVIDER

PASSIVE_PROVIDER

Requesting Updates

`android.location.LocationManager`

`requestLocationUpdates()`

`requestSingleUpdate()`

Requesting Updates

```
requestLocationUpdates(
    provider,
    minTime, /* ms */
    minDistance, /* meters */
    listener
);
```

Requesting Updates (GPS)

```
if(gpsProviderEnabled == true){
  _locationManager.requestLocationUpdates(
     LocationManager.GPS_PROVIDER,
     10000 /* minTime */,
     10 /* minDistance */,
     _locationListenerGPSProvider
  );
}
```

Six types of location data

- Real-time
  - GPS
  - Network
- Cached
  - GPS
  - Network
- Passive
- NMEA

Location data: Real-time GPS

Latitude - (decimal degrees)

Longitude - (decimal degrees)

Time - (UTC since Jan 1, 1970)

Bearing - (degrees)

Speed - (meters/second)

Altitude - (meters >= sea level)

Accuracy - (meters)

Location data: Real-time GPS

```
Location[
    mProvider=gps,
    mTime=1374081473000,
    mLatitude=39.91824753,
    mLongitude=-105.10395774,
    mHasAltitude=true, mAltitude=1065.0,
    mHasSpeed=true, mSpeed=0.0,
    mHasBearing=false, mBearing=0.0,
    mHasAccuracy=true, mAccuracy=72.0,
    mExtras=Bundle
[mParcelledData.dataSize=44]]
```

Location data: Real-time Network

Latitude - (decimal degrees)

Longitude - (decimal degrees)

Time - (UTC since Jan 1, 1970)

Accuracy - (meters)

Location data: Real-time Network

```
Location[
    mProvider=network,
    mTime=1373643571572,
    mLatitude=33.6985455,
    mLongitude=-117.9914674,
    mHasAltitude=false, mAltitude=0.0,
    mHasSpeed=false, mSpeed=0.0,
    mHasBearing=false, mBearing=0.0,
    mHasAccuracy=true, mAccuracy=55.161,
    mExtras=Bundle
[mParcelledData.dataSize=212]]
```

Location: Cached GPS

Latitude - (decimal degrees)

Longitude - (decimal degrees)

Time - (UTC since Jan 1, 1970)

Bearing - (degrees)

Speed - (meters/second)

Altitude - (meters >= sea level)

Accuracy - (meters)

Location data: Cached Network

Latitude - (decimal degrees)

Longitude - (decimal degrees)

Time - (UTC since Jan 1, 1970)

Accuracy - (meters)

Location data: Passive

Latitude - (decimal degrees)

Longitude - (decimal degrees)

Time - (UTC since Jan 1, 1970)

Bearing - (degrees) ???

Speed - (meters/second) ???

Altitude - (meters >= sea level) ??

Accuracy - (meters) ???

Location data: NMEA

```
addNmeaListener(GpsStatus.NmeaListener);
```

```
$GNGSA,A,2,67,68,78,,,,,,,,,,2.4,2.2,0.8*23
```

7 Technical use cases

Cold start

Warm start

Minimized

Passive

Snapshot

Intermittent

Continuous

Technical Use Case: Cold start

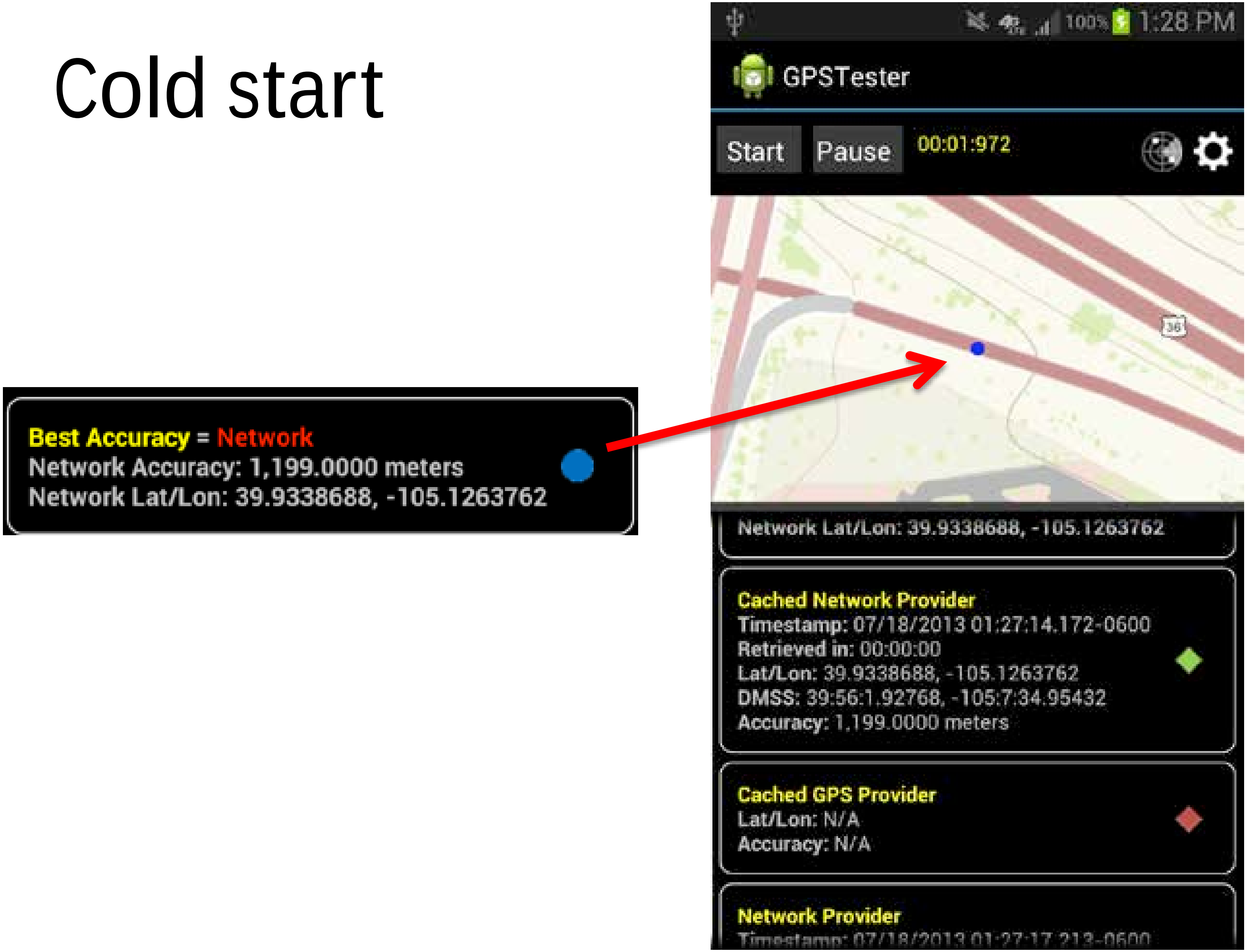
Phone rebooted

Updated phone OS

Potential for no cached values

Potential for large inaccuracy

Cold start



Best Accuracy = Network
Network Accuracy: 1,199.0000 meters
Network Lat/Lon: 39.9338688, -105.1263762

Network Lat/Lon: 39.9338688, -105.1263762

Cached Network Provider
Timestamp: 07/18/2013 01:27:14.172-0600
Retrieved in: 00:00:00
Lat/Lon: 39.9338688, -105.1263762
DMSS: 39:56:1.92768, -105:7:34.95432
Accuracy: 1,199.0000 meters

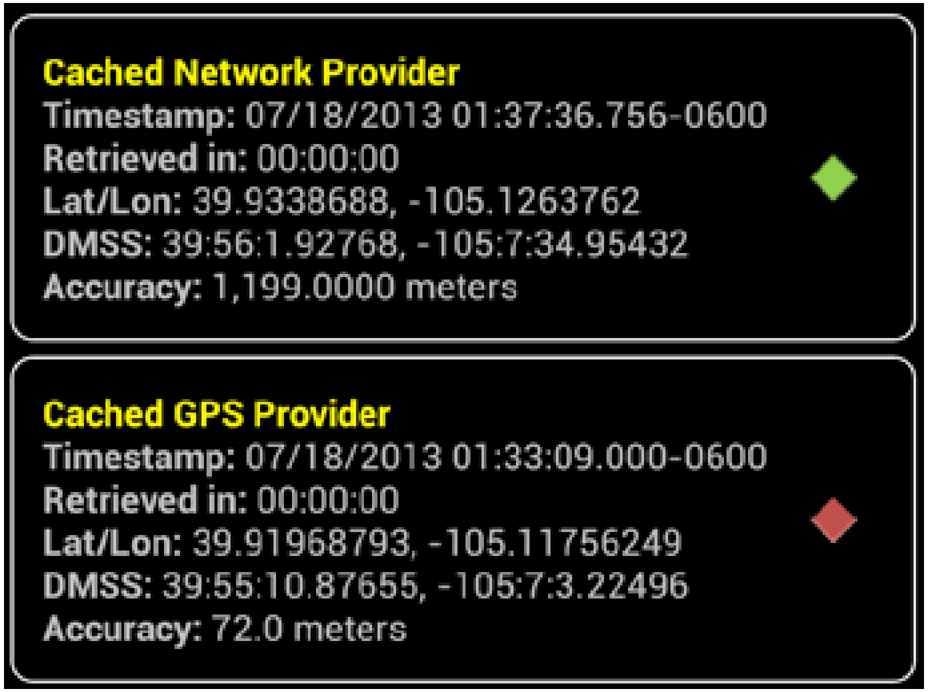
Cached GPS Provider
Lat/Lon: N/A
Accuracy: N/A

Network Provider
Timestamp: 07/18/2013 01:27:17.213-0600

Technical Use Case: Warm start

Cached locations available

GPS has been run recently



Cached Network Provider
Timestamp: 07/18/2013 01:37:36.756-0600
Retrieved in: 00:00:00
Lat/Lon: 39.9338688, -105.1263762
DMSS: 39:56:1.92768, -105:7:34.95432
Accuracy: 1,199.0000 meters

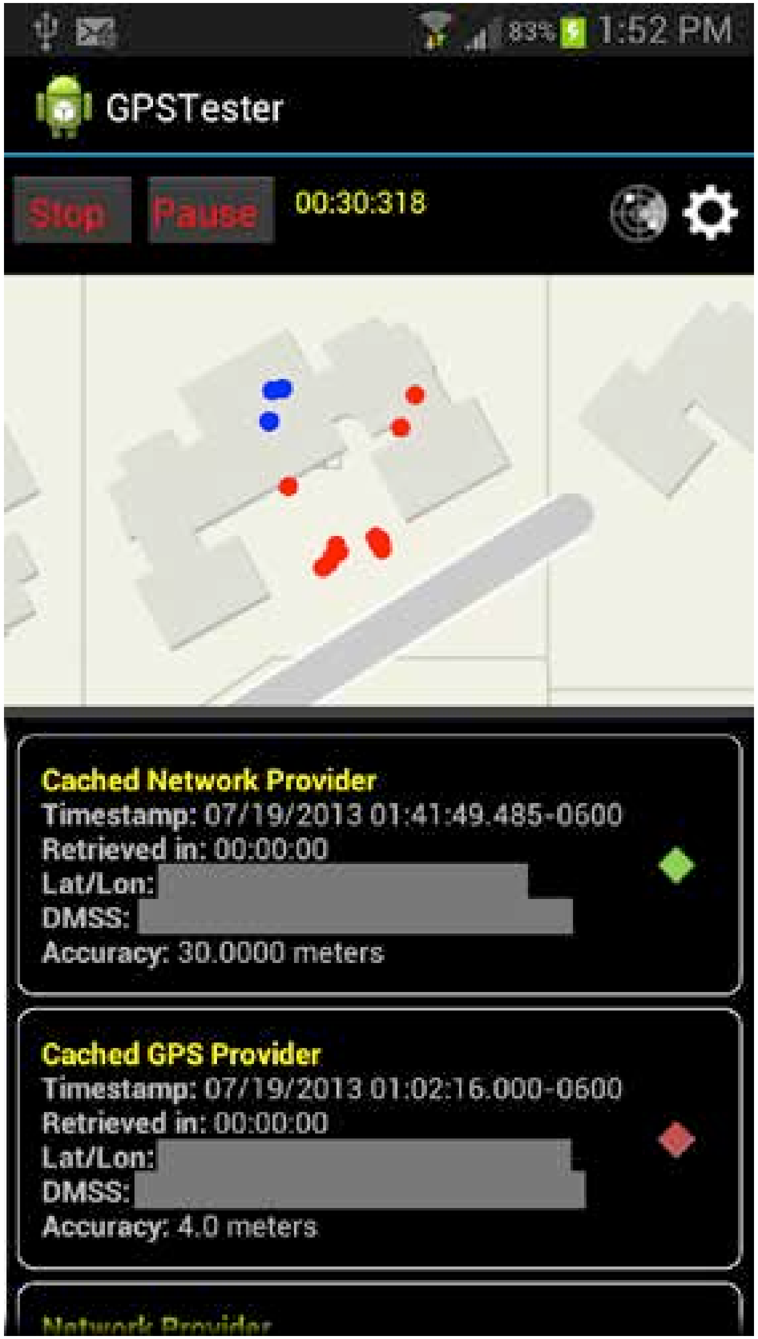
Cached GPS Provider
Timestamp: 07/18/2013 01:33:09.000-0600
Retrieved in: 00:00:00
Lat/Lon: 39.91968793, -105.11756249
DMSS: 39:55:10.87655, -105:7:3.22496
Accuracy: 72.0 meters

Warm start

Compare

Cached network & GPS

Check timestamps!!



Cached Network Provider
Timestamp: 07/19/2013 01:41:49.485-0600
Retrieved in: 00:00:00
Lat/Lon: [redacted]
DMSS: [redacted]
Accuracy: 30.0000 meters

Cached GPS Provider
Timestamp: 07/19/2013 01:02:16.000-0600
Retrieved in: 00:00:00
Lat/Lon: [redacted]
DMSS: [redacted]
Accuracy: 4.0 meters

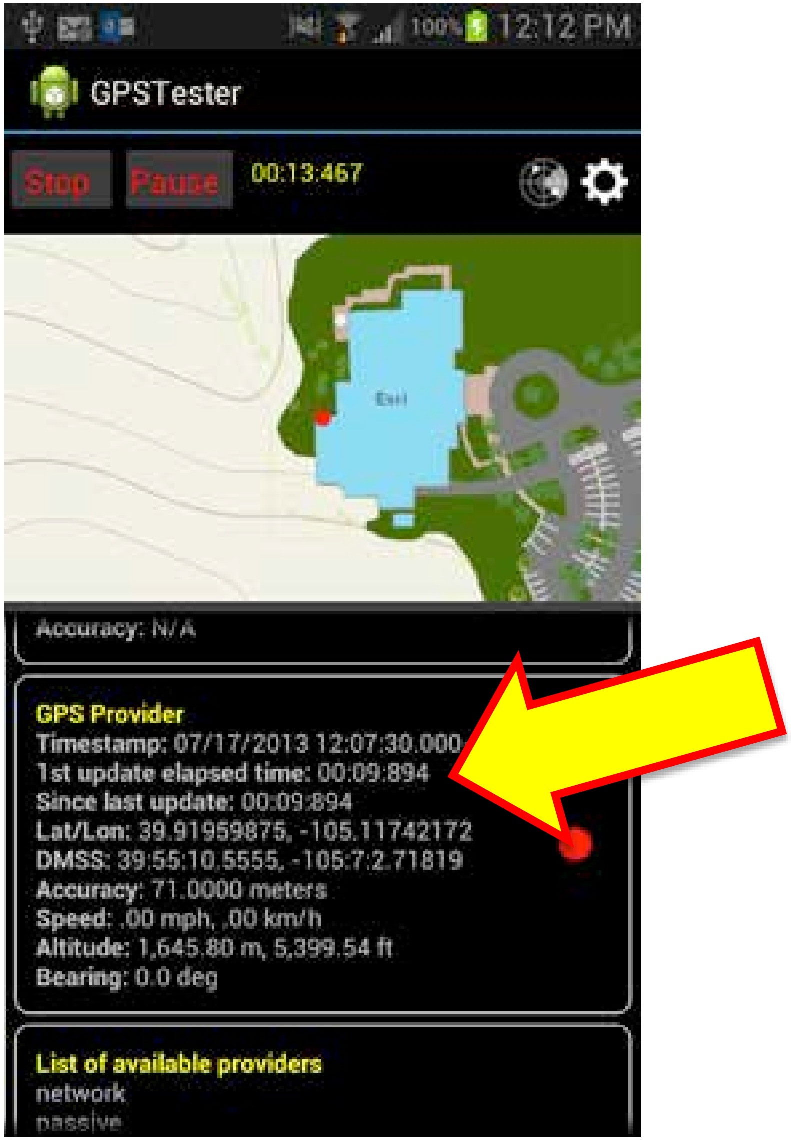
Network Provider

Warm start

Example:

~10 seconds

Accuracy 71 meters



Accuracy: N/A

GPS Provider
Timestamp: 07/17/2013 12:07:30.000
1st update elapsed time: 00:09:894
Since last update: 00:09:894
Lat/Lon: 39.91959875, -105.11742172
DMSS: 39:55:10.5555, -105:7:2.71819
Accuracy: 71.0000 meters
Speed: .00 mph, .00 km/h
Altitude: 1,645.80 m, 5,399.54 ft
Bearing: 0.0 deg

List of available providers
network
passive

Technical Use Case: Minimized

App in background

GPS can be run

Can kill battery quickly

Technical Use Case: Passive

Dependent on some other app

Only receives if other app requests location

No guarantees!

```
<receiver
    android:name=".receivers.PassiveLocationChangedReceiver"/>
```

Technical Use Case: Snapshot

One-time location

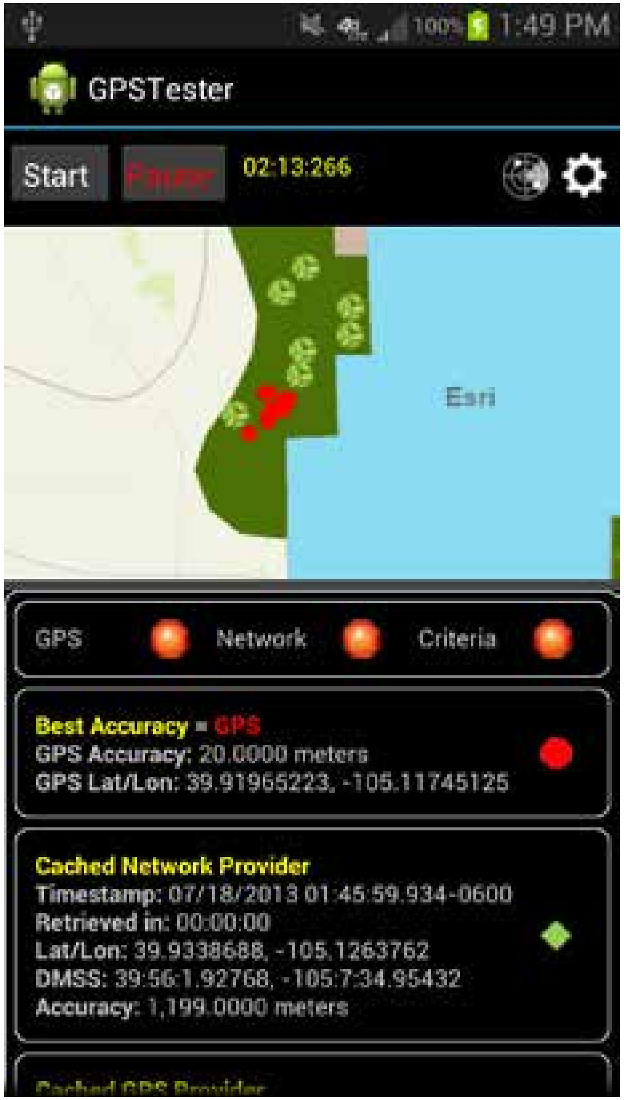
>= minimum accuracy

Example:

Standing indoors

~2 mins

Accuracy 20 meters



Best Accuracy = GPS
GPS Accuracy: 20.0000 meters
GPS Lat/Lon: 39.91965223, -105.11745125

Cached Network Provider
Timestamp: 07/18/2013 01:45:59.934-0600
Retrieved in: 00:00:00
Lat/Lon: 39.9338688, -105.1263762
DMSS: 39:56:1.92768, -105:7:34.95432
Accuracy: 1,199.0000 meters

Cached GPS Provider

Technical Use Case: Snapshot

One-time location

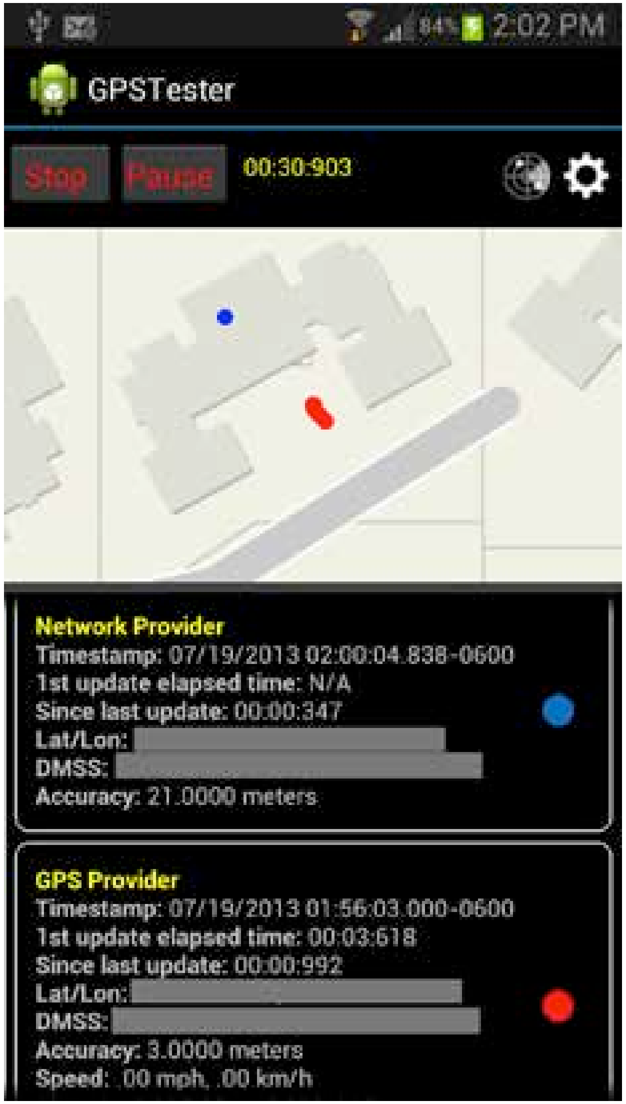
>= minimum accuracy

Example:

Next to building

~30 seconds

Accuracy 3 meters



Network Provider
Timestamp: 07/19/2013 02:00:04.838-0600
1st update elapsed time: N/A
Since last update: 00:00:347
Lat/Lon: [redacted]
DMSS: [redacted]
Accuracy: 21.0000 meters

GPS Provider
Timestamp: 07/19/2013 01:56:03.000-0600
1st update elapsed time: 00:03:618
Since last update: 00:00:992
Lat/Lon: [redacted]
DMSS: [redacted]
Accuracy: 3.0000 meters
Speed: .00 mph, .00 km/h

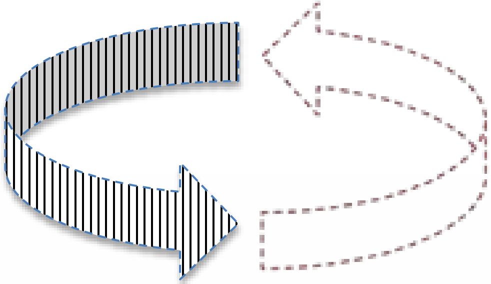
Technical Use Case: Intermittent

No movement detected for period of time

Adjusts minTime & minDistance

Temporarily sleep GPS

Keeps GPS ‘warm’



Technical Use Case: Continuous

GPS stays on

Highest level of accuracy

Most battery usage

End-User scenarios - Continuous

Walking

Running

Hiking

Biking

Driving

Biking

Example Technical Requirements

```
Start
    minTime = 0
    minDistance = 0
if
    accuracy <= 20 meters
    speed > 5 mph && speed < 45 mph
    time > 2 minutes
    power source = battery
then
    reset location updates
    minTime = 5000
    minDistance = 10
```

End-User scenarios - Snapshot

Find coffee

Find nearby places

Get start location for driving directions

Find nearby coffee shops

Example Technical Requirements

```
    Start
        minTime = 0
        minDistance = 0
    if
        accuracy <= 1000 meters
    then
        shutoff location
```

Questions?

Andy Gup
Developer Evangelist
www.andygup.net
github.com/andygup
agup@esri.com
[@agup](https://twitter.com/agup)

Bonus Slides

Step 1: AndroidManifest.xml

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

Step 2a: retrieve LocationManager

```
LocationManager _locationManager =
Context.getSystemService(Context.LOCATION_SERVICE);
```

Step 2b: verify providers

```
final Boolean gpsProviderEnabled =
   _locationManager.isProviderEnabled(
      LocationManager.GPS_PROVIDER);

final Boolean networkProviderEnabled =
   _locationManager.isProviderEnabled(
      LocationManager.NETWORK_PROVIDER);

final Boolean passiveProviderEnabled =
   _locationManager.isProviderEnabled(
      LocationManager.PASSIVE_PROVIDER);
```

Step 3: Last Known Location

```
Location _lastKnownLocationNetworkProvider =
    _locationManager.getLastKnownLocation(
        LocationManager.NETWORK_PROVIDER);

Location _lastKnownLocationGPSProvider =
    _locationManager.getLastKnownLocation(
        LocationManager.GPS_PROVIDER);
```

```
if(_lastKnowLocationNetworkProvider != null){
    final double cachedNetworkLatitude =
        _lastKnownLocationNetworkProvider.getLatitude();

    final double cachedNetworkLongitude =
        _lastKnownLocationNetworkProvider.getLongitude();
}
```

Step 4a: set GPS Listener

```
_gpsLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
    }

    @Override
    public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub
    }
}
```

Step 4b: set Network Listener

```
_networkLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
    }

    @Override
    public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub
    }
}
```

NOTE: multiple listeners

You will not be able to shut off a provider if it has multiple listeners.

Step 5: Requesting Updates

```
android.location.LocationManager

    requestLocationUpdates()

    requestSingleUpdate()
```

Requesting Updates

```
requestLocationUpdates(
    provider,
    minTime, /* ms */
    minDistance, /* meters */
    listener
);
```

Requesting Updates (GPS)

```
if(gpsProviderEnabled == true){
  _locationManager.requestLocationUpdates(
     LocationManager.GPS_PROVIDER,
     10000 /* minTime */,
     10 /* minDistance */,
     _locationListenerGPSProvider
  );
}
```

Requesting Updates (Network)

```
if(networkProviderEnabled == true){
  _locationManager.requestLocationUpdates(
     LocationManager.NETWORK_PROVIDER,
     10000 /* minTime */,
     10 /* minDistance */,
     _locationListenerNetworkProvider
  );
}
```

Requesting Updates (Passive)

```
if(passiveProviderEnabled == true){
   _locationManager.requestLocationUpdates(
     LocationManager.PASSIVE_PROVIDER,
     10000 /* minTime */,
     10 /* minDistance */,
     _locationListenerPassiveProvider
   );
}
```

Requesting Updates

minTime

milliseconds >= 0

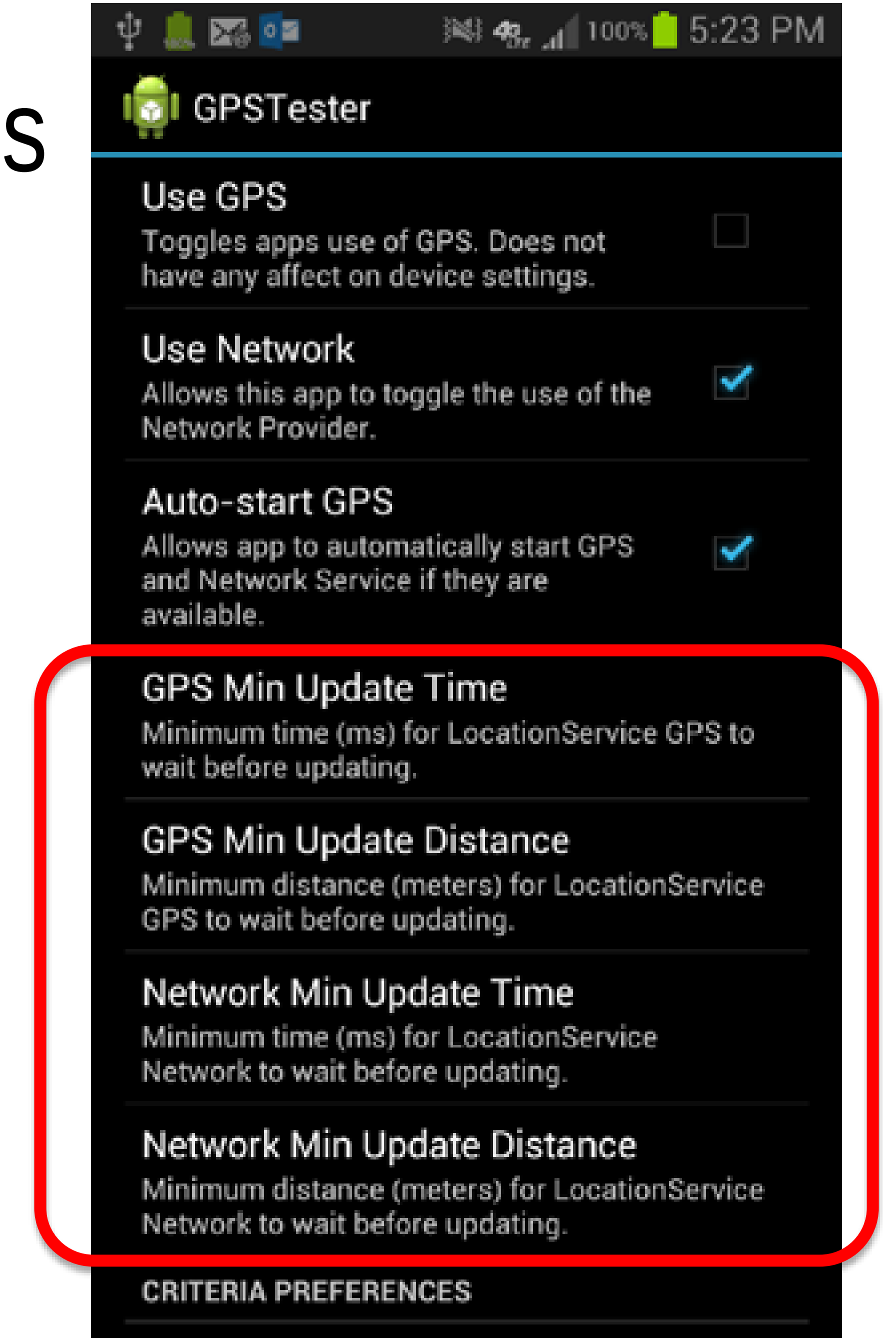
```
if(minTime > 0 && minDistance)
```

minDistance

meters >=0

Less battery efficient than minTime

Requesting Updates



100% 5:23 PM

GPSTester

Use GPS ☐
Toggles apps use of GPS. Does not have any affect on device settings.

Use Network ☑
Allows this app to toggle the use of the Network Provider.

Auto-start GPS ☑
Allows app to automatically start GPS and Network Service if they are available.

GPS Min Update Time
Minimum time (ms) for LocationService GPS to wait before updating.

GPS Min Update Distance
Minimum distance (meters) for LocationService GPS to wait before updating.

Network Min Update Time
Minimum time (ms) for LocationService Network to wait before updating.

Network Min Update Distance
Minimum distance (meters) for LocationService Network to wait before updating.

CRITERIA PREFERENCES

Step 6: process location updates

```
@Override
public void onLocationChanged(Location location) {
   double lat = location.getLatitude();
   double long = location.getLongitude();
   double altitude = location.getAltitude();
   float accuracy = location.getAccuracy();
   float bearing = location.getBearing();
   float speed = location.getSpeed();
}
```

Best Provider via Criteria

```
Criteria criteria = new Criteria();
criteria.setAccuracy(accuracy);
criteria.setCostAllowed(cost);
criteria.setPowerRequirement(power);
...
...

final String bestProviderName =
_locationManager.getBestProvider(criteria, true);
```

Best Provider by comparison

```
if(_networkAccuracy > _gpsAccuracy &&
   _gpsTime > _networkTime &&
   _gpsTimeDiff > _MIN_UPDATE_TIME){
   //Use network Location data
}

if(_gpsAccuracy > _networkAccuracy &&
   _networkTime > _gpsTime &&
   _networkTimeDiff > _MIN_UPDATE_TIME){
   //use gps Location data
}
```

Step 7: shutdown updates

```
if(_locationManager != null &&
   _locationListenerNetworkProvider != null &&
   _locationListenerGPSProvider != null){

      _locationManager.removeUpdates(
         _locationListenerNetworkProvider);

      _locationManager.removeUpdates(
         _locationListenerGPSProvider);

      _locationListenerNetworkProvider = null;
      _locationListenerGPSProvider = null;
}
```

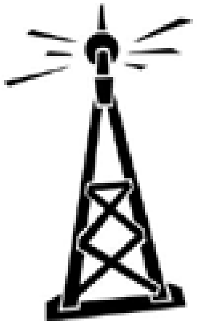
Consuming real-time locations

Time to 1st result

- Elapsed time
- Accuracy value

Compare GPS vs Network

!==



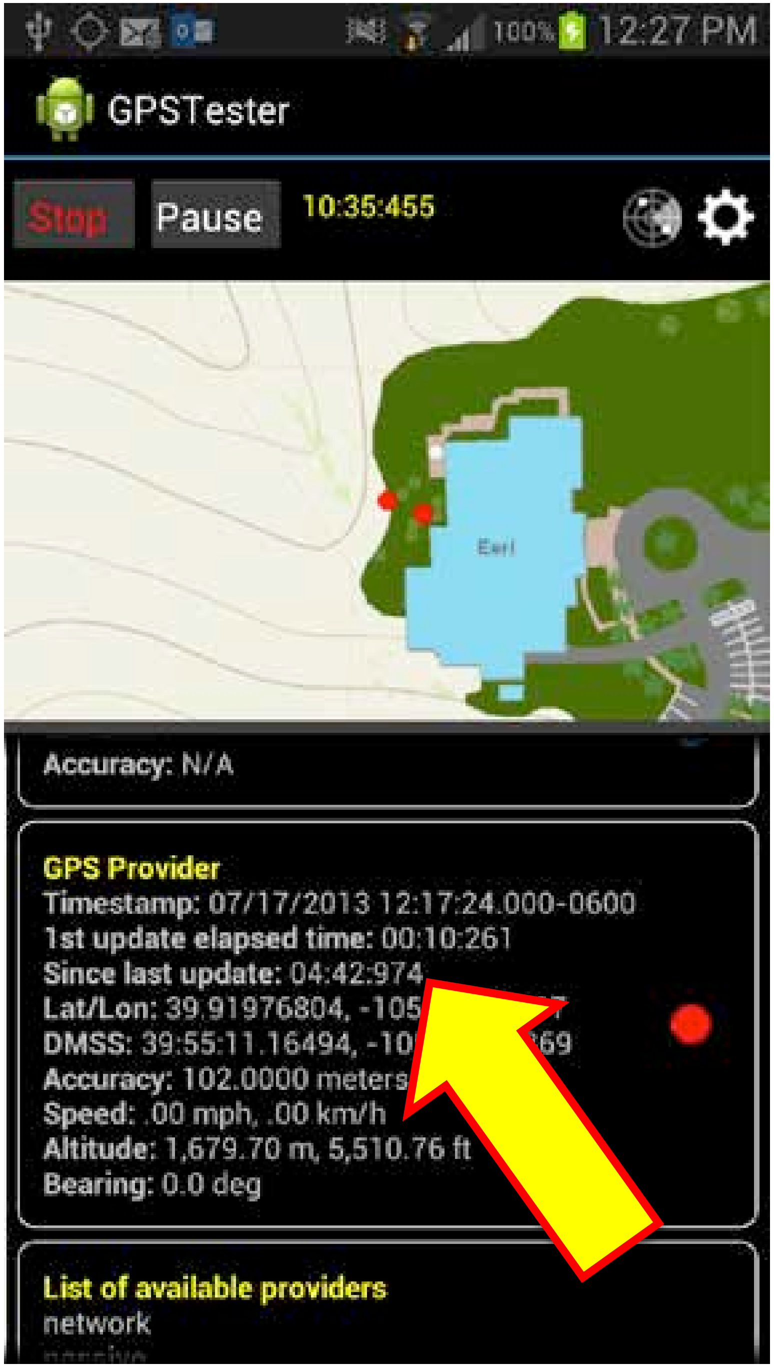
Consuming real-time locations

Subsequent results

- Elapsed time
- Accuracy value
- Bearing
- Distance traveled

Streaming updates

Time between updates



Accuracy: N/A

GPS Provider
Timestamp: 07/17/2013 12:17:24.000-0600
1st update elapsed time: 00:10:261
Since last update: 04:42:974
Lat/Lon: 39.91976804, -105
DMSS: 39:55:11.16494, -10
Accuracy: 102.0000 meters
Speed: .00 mph, .00 km/h
Altitude: 1,679.70 m, 5,510.76 ft
Bearing: 0.0 deg

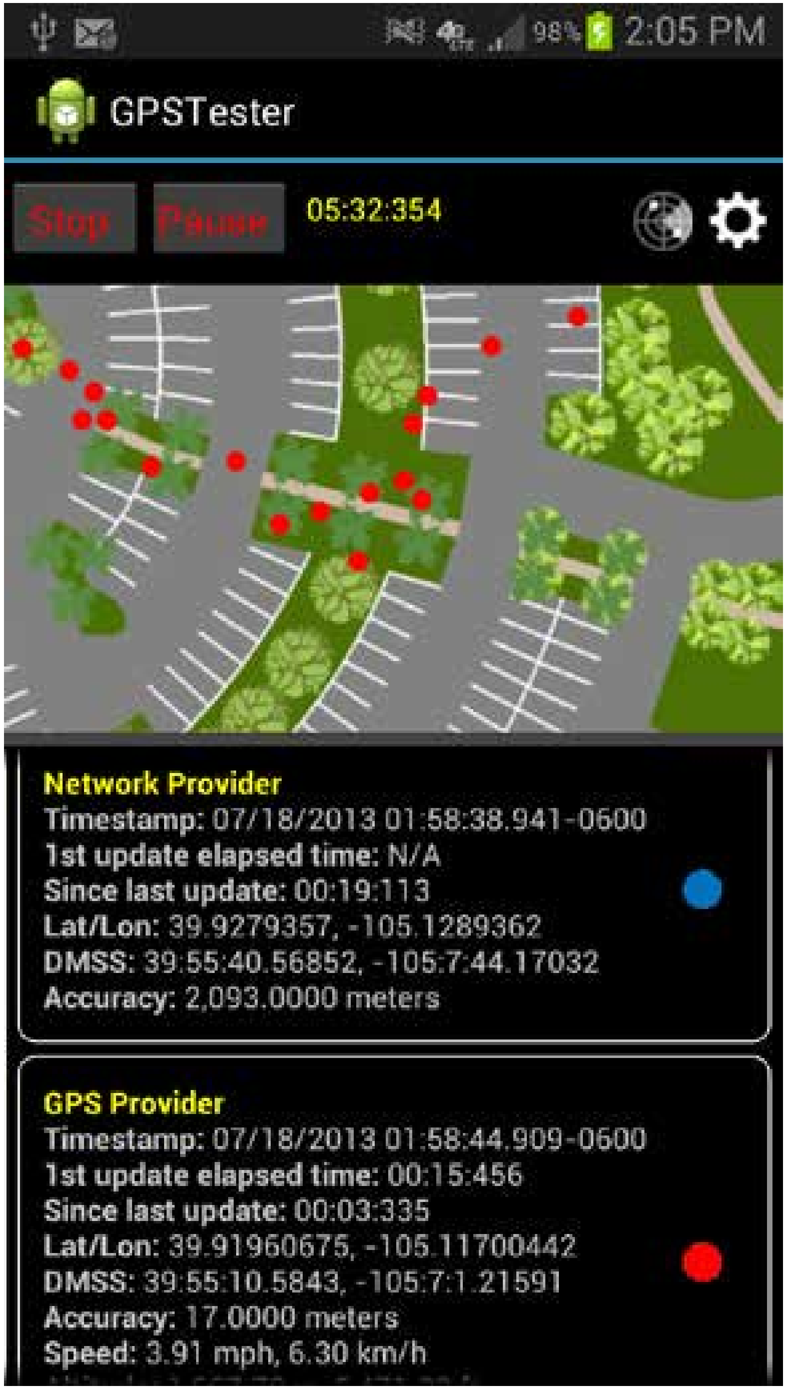
List of available providers
network
passive

Streaming updates

Network vs GPS

Example:

2093m vs 17m



Network Provider
Timestamp: 07/18/2013 01:58:38.941-0600
1st update elapsed time: N/A
Since last update: 00:19:113
Lat/Lon: 39.9279357, -105.1289362
DMSS: 39:55:40.56852, -105:7:44.17032
Accuracy: 2,093.0000 meters

GPS Provider
Timestamp: 07/18/2013 01:58:44.909-0600
1st update elapsed time: 00:15:456
Since last update: 00:03:335
Lat/Lon: 39.91960675, -105.11700442
DMSS: 39:55:10.5843, -105:7:1.21591
Accuracy: 17.0000 meters
Speed: 3.91 mph, 6.30 km/h

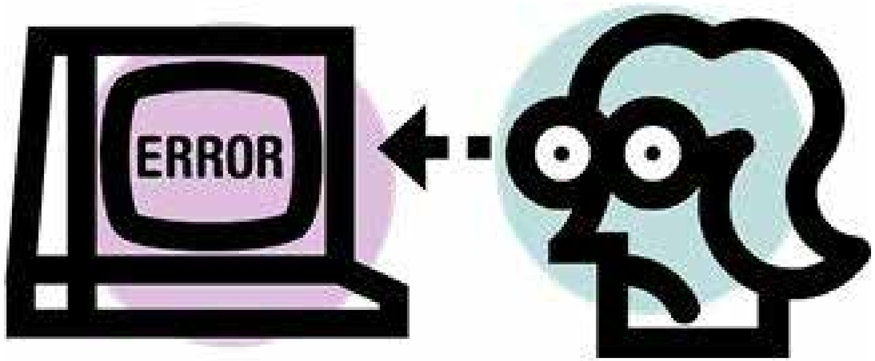
Consuming real-time locations

Handle bad locations

Recovering from errors

Lost connection:

- GPS
- Network



Detect power on app startup

```
@Override
public void onCreate(Bundle savedInstanceState) {

    Intent intent = registerReceiver(null, new
        IntentFilter(Intent.ACTION_BATTERY_CHANGED));

    boolean isBatteryOn =
        intent.getIntExtra(
            BatteryManager.EXTRA_PLUGGED, -1) > 0

}
```

Detect power state change

BroadcastReciever

```
<action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
<action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
```

Detect low battery

```
<action android:name="android.intent.action.ACTION_BATTERY_LOW"/>
<action
android:name="android.intent.action.ACTION_BATTERY_OKAY"/>
```

```
public class PowerStateChangedReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent) {
        boolean batteryLow =
            intent.getAction().equals(Intent.ACTION_BATTERY_LOW);
        if(batteryLow == true){
            //Do something
        }
    }
}
```

Detecting provider changes

```
@Override
public void onStatusChanged(String provider, in
   status,  Bundle extras) {
   // TODO Auto-generated method stub
}

@Override
public void onProviderEnabled(String provider) {
   // TODO Auto-generated method stub
}

@Override
public void onProviderDisabled(String provider) {
   // TODO Auto-generated method stub
}
```

onStatusChanged Event

OUT_OF_SERVICE

TEMPORARILY_UNAVAILABLE

AVAILABLE

onProviderDisabled Event

Let your user know

Store the timestamp for comparison

Attempt to reestablish connection/reset

onProviderEnabled Event

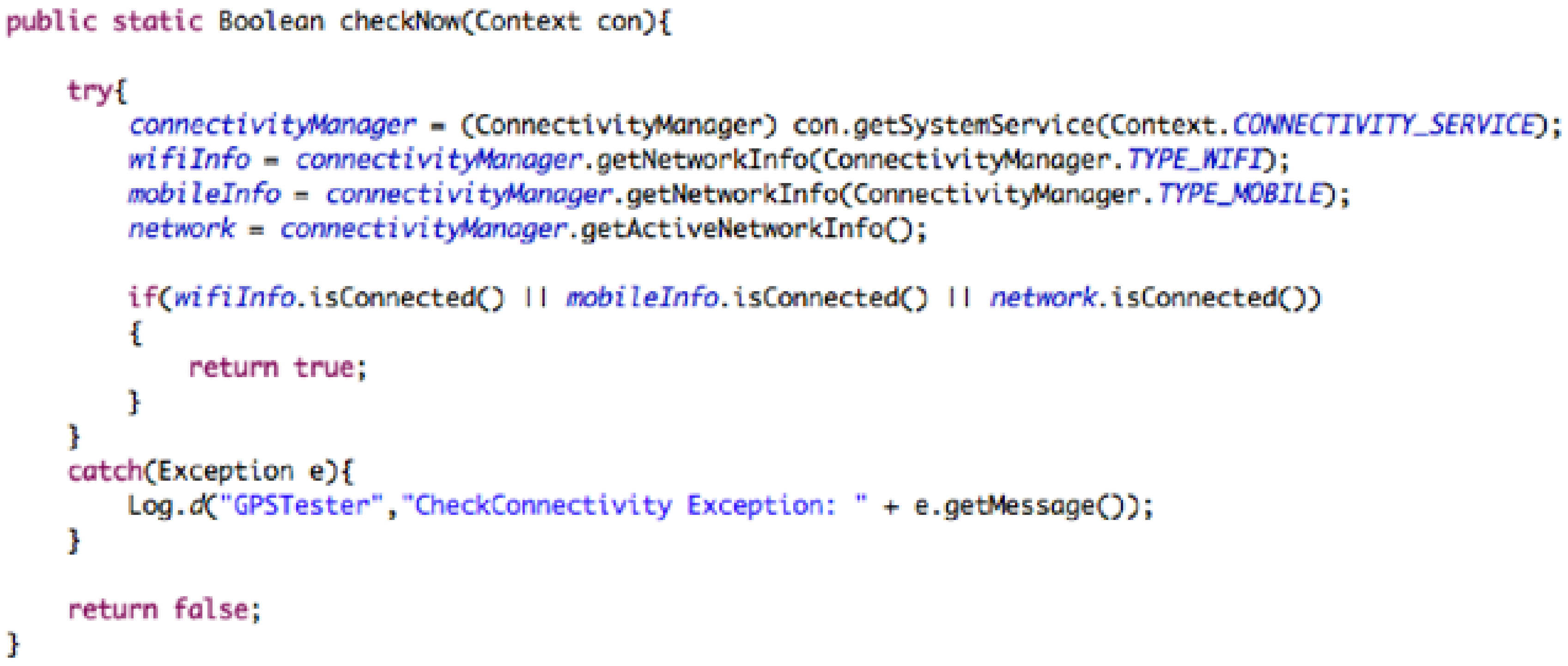
Verify accuracy, timestamp

Let user know data was interrupted

Activity starting and stopping

```
@Override
protected void onPause() {
    stopLocation();
    super.onPause();
}

@Override
protected void onStop(){
    super.onStop();
    stopLocation();
}

@Override
protected void onResume() {
    super.onResume();
    startLocation();
}
```

Detect connectivity change (static)



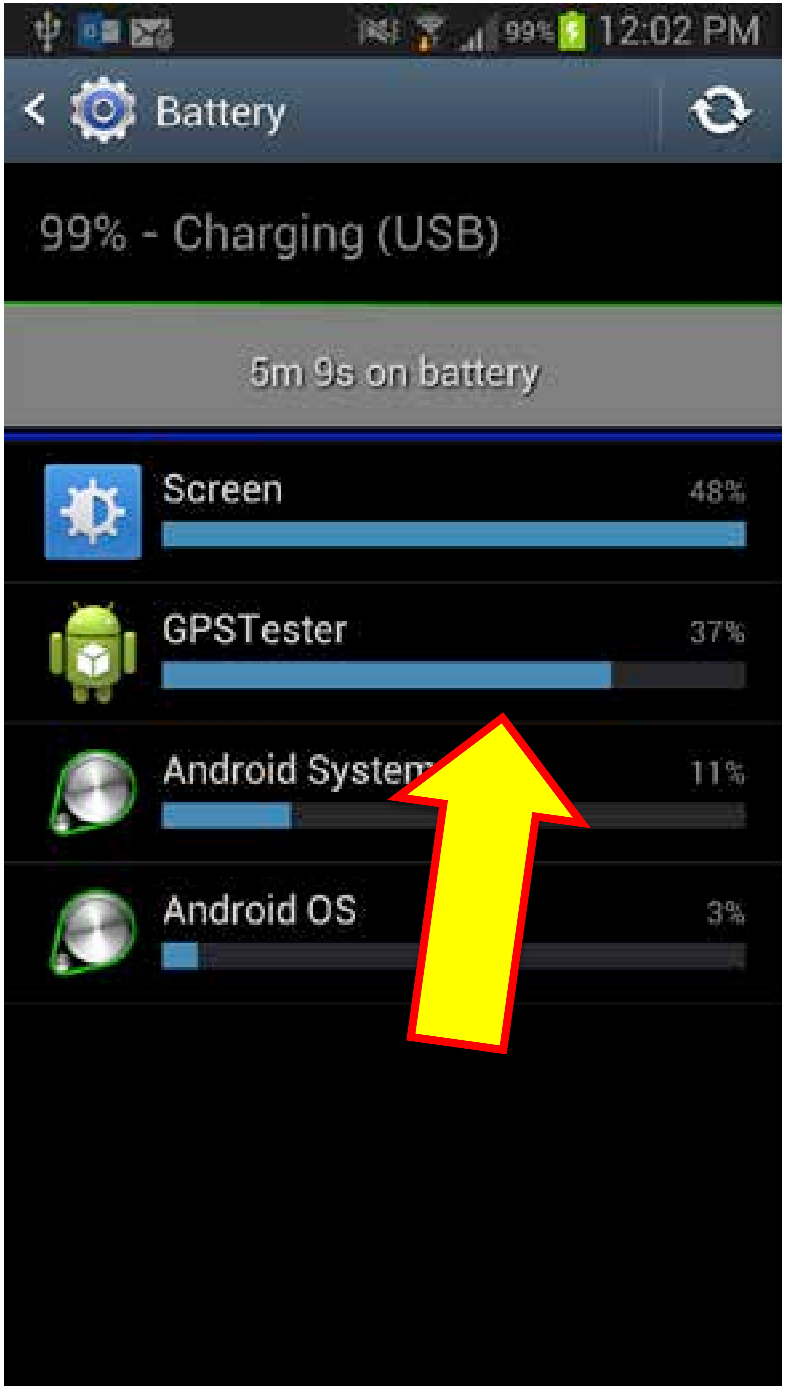
```java
public static Boolean checkNow(Context con){

    try{
        connectivityManager = (ConnectivityManager) con.getSystemService(Context.CONNECTIVITY_SERVICE);
        wifiInfo = connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
        mobileInfo = connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
        network = connectivityManager.getActiveNetworkInfo();

        if(wifiInfo.isConnected() || mobileInfo.isConnected() || network.isConnected())
        {
            return true;
        }
    }
    catch(Exception e){
        Log.d("GPSTester","CheckConnectivity Exception: " + e.getMessage());
    }

    return false;
}
```

Helper methods: Distance Traveled

`android.location.Location`

```
   distanceBetween()
   distanceTo()
   bearingTo()
```

Battery life



12:02 PM

Battery

99% - Charging (USB)

5m 9s on battery

Screen 48%

GPSTester 37%

Android System 11%

Android OS 3%

Battery life

minTime > 0

Shutoff location when minimized.

Shutoff location at min. accuracy

Modify LocationManager

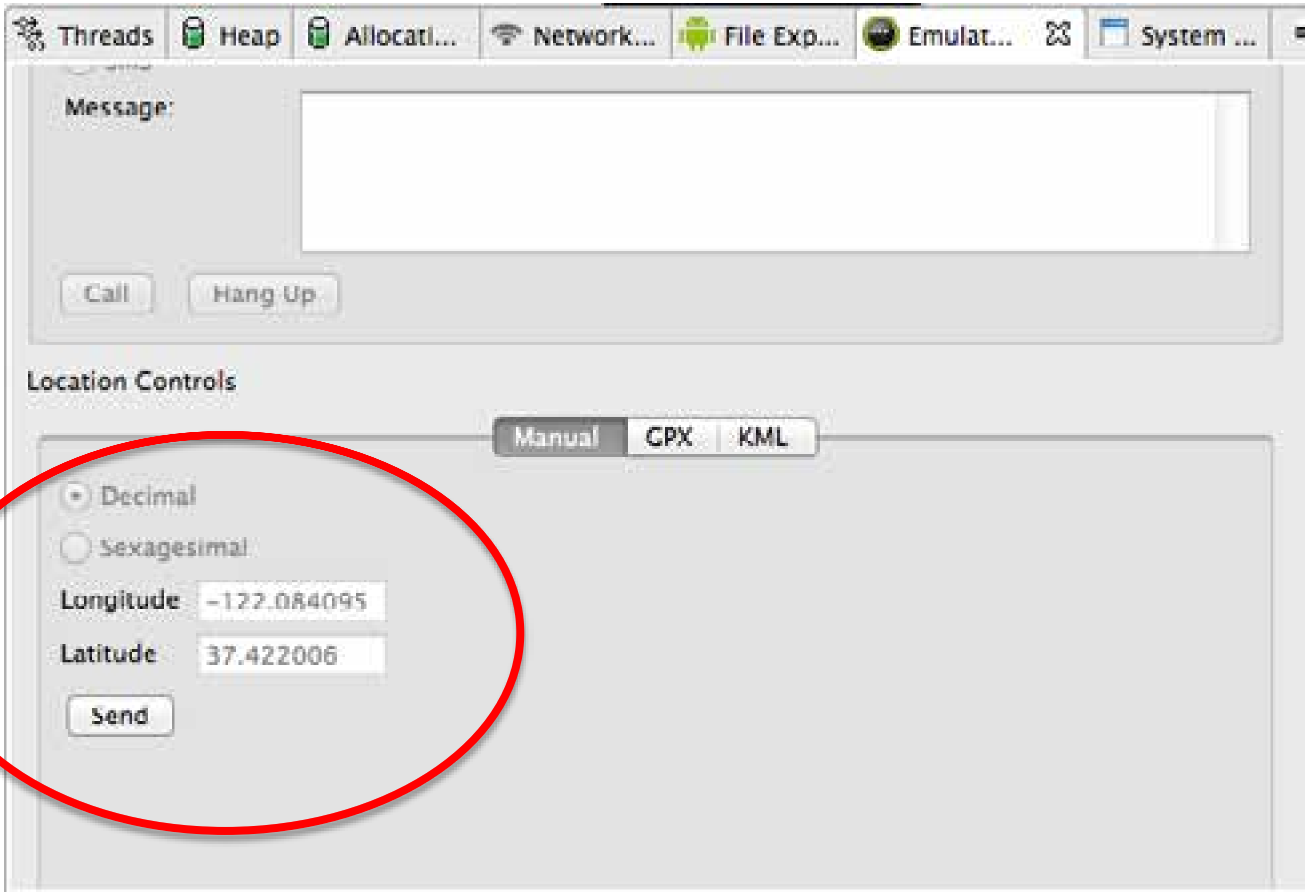
- on battery
- if movement stops for long periods
- under different usage conditions

Keep screen turned on

```
getWindow().addFlags(
   WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
```

Mock Updates (Emulator)

DDMS Emulator Control



● Decimal

○ Sexagesimal

Longitude: -122.084095

Latitude: 37.422006

Mock Updates (Device)

Android Manifest

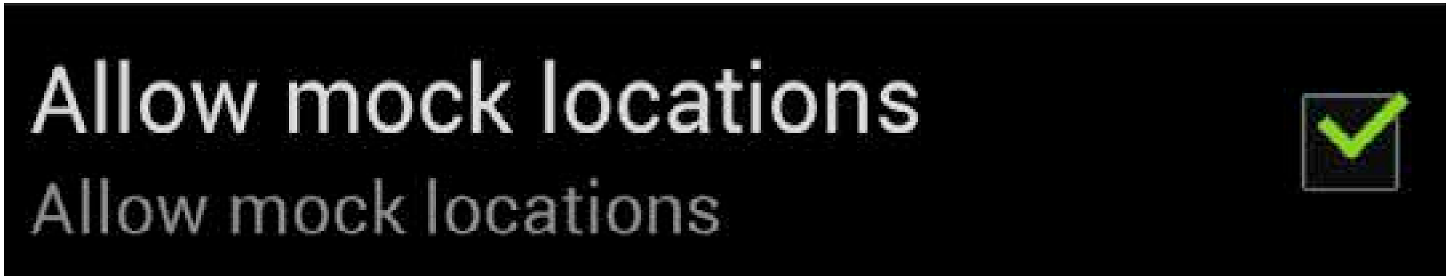
```
<uses-permission
android:name="android.permission.ACCESS_MOCK_LOCATION" />
```

Mock Updates (Device)

Android Manifest

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

Device Settings

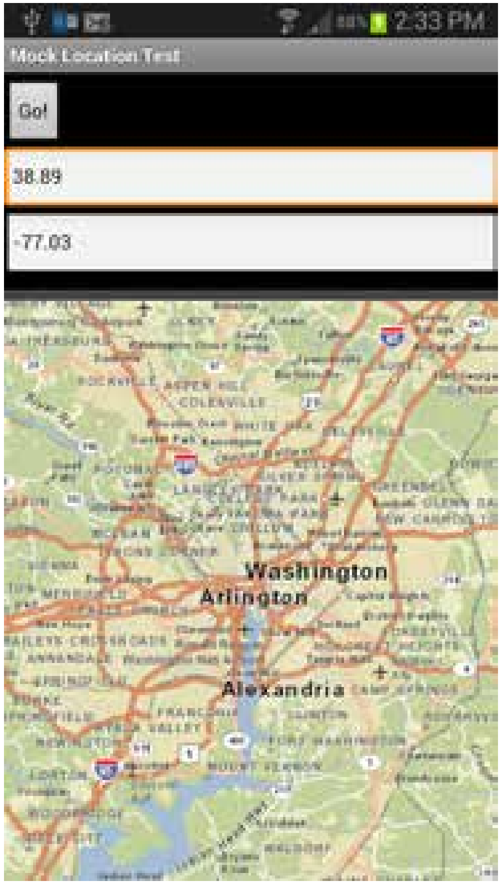


Allow mock locations ☑

Allow mock locations

Mock Updates (Device)

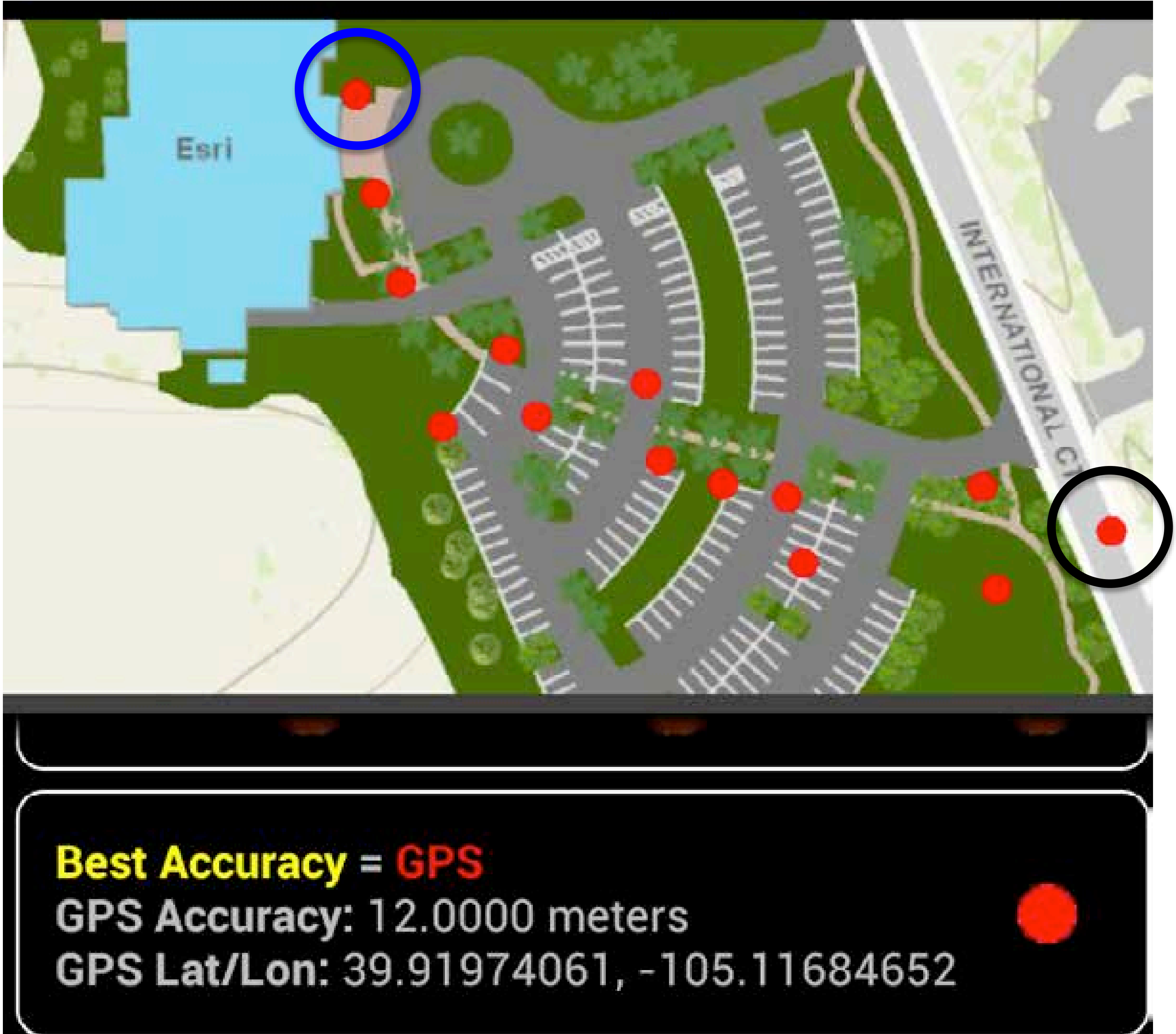
https://github.com/andygup/mock-location-test-android



Mock Location Test

38.89

-77.03



Best Accuracy = GPS
GPS Accuracy: 12.0000 meters
GPS Lat/Lon: 39.91974061, -105.11684652 ●

Privacy

Get legal advice

Allow for opt-out

Be clear about your privacy policy