**GPS Location Provider**

**Network Location Provider vs GPS Location Provider**

* Network Location provider is comparatively faster than the GPS provider in providing the location co-ordinates.
* GPS provider may be very very slow in in-door locations and will drain the mobile battery.
* Network location provider depends on the cell tower and will return our nearest tower location.
* GPS location provider, will give our location accurately.

**Steps to get location in Android**

1. Provide permissions in manifest file for receiving location update
2. Create LocationManager instance as reference to the location service
3. Request location from LocationManager
4. Receive location update from LocationListener on change of location

**Provide permissions for receiving location update**

To access current location information through location providers, we need to set permissions with [android manifest](http://javapapers.com/android/android-manifest/) file.

<manifest ... >

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission. ACCESS\_COARSE\_LOCATION" />

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

</manifest>

* 1. ACCESS\_COARSE\_LOCATION is used when we use network location provider for our Android app.
  2. But, ACCESS\_FINE\_LOCATION is providing permission for both providers.
  3. INTERNET permission is must for the use of network provider.

**Create LocationManager instance as reference to the location service**

For any background [Android Service](http://javapapers.com/android/how-to-play-audio-in-android/), we need to get reference for using it. Similarly, location service reference will be created using getSystemService() method. This reference will be added with the newly created LocationManager instance as follows.

**Snippet**

locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);

**Request current location from LocationManager**

After creating the location service reference, location updates are requested using requestLocationUpdates() method of LocationManager. For this function, we need to send the type of location provider, number of seconds, distance and the LocationListener object over which the location to be updated.

**Snippet**

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);

**Java Program**

import android.os.Bundle;

import android.app.Activity;

import android.content.Context;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.widget.TextView;

import android.util.Log;

public class MainActivity extends Activity implements LocationListener{

protected LocationManager locationManager;

protected LocationListener locationListener;

protected Context context;

TextView txtLat;

String lat;

String provider;

protected String latitude,longitude;

protected boolean gps\_enabled,network\_enabled;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

txtLat = (TextView) findViewById(R.id.textview1);

locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);

}

@Override

public void onLocationChanged(Location location) {

txtLat = (TextView) findViewById(R.id.textview1);

txtLat.setText("Latitude:" + location.getLatitude() + ", Longitude:" + location.getLongitude());

}

@Override

public void onProviderDisabled(String provider) {

Log.d("Latitude","disable");

}

@Override

public void onProviderEnabled(String provider) {

Log.d("Latitude","enable");

}

@Override

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

Log.d("Latitude","status");

}

}

**XML files for layout and android manifest are as shown below**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<TextView

android:id="@+id/textview1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_centerVertical="true"

android:text="@string/hello\_world" />

</RelativeLayout>

**AndroidManifest.xml**

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.android.geolocationfinder"

android:versionCode="1"

android:versionName="1.0" >

<**uses-sdk**

**android:minSdkVersion="8"**

**android:targetSdkVersion="17" />**

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppBaseTheme" >

<activity

android:name="com. android.geolocationfinder.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**How to send latitude and longitude to android emulator**

* Open DDMS perspective in Studio (Tools 🡪 Android 🡪 Android Device Monitor).
* Select your emulator device
* Select the tab named emulator control
* In ‘Location Controls’ panel, ‘Manual’ tab, give the Longitude and Latitude as input and ‘Send’.

LocationManager

This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to fire an application-specified [Intent](http://developer.android.com/reference/android/content/Intent.html) when the device enters the proximity of a given geographical location.

You do not instantiate this class directly; instead, retrieve it through [Context.getSystemService(Context.LOCATION\_SERVICE)](http://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)).

**Example**

LocationManager **lm** = (LocationManager)getSystemService(***LOCATION\_SERVICE***);

**LocationListener (Interface)**

Used for receiving notifications from the LocationManager when the location has changed. These methods are called if the LocationListener has been registered with the location manager service using the [requestLocationUpdates(String, long, float, LocationListener)](http://developer.android.com/reference/android/location/LocationManager.html#requestLocationUpdates(java.lang.String, long, float, android.location.LocationListener)) method.

**Abstract Methods**

1. public abstract void onLocationChanged ([Location](http://developer.android.com/reference/android/location/Location.html) location)

Called when the location has changed.

There are no restrictions on the use of the supplied Location object.

1. public abstract void onProviderDisabled ([String](http://developer.android.com/reference/java/lang/String.html) provider)

Called when the provider is disabled by the user.

If requestLocationUpdates is called on an already disabled provider, this method is called immediately.

1. public abstract void onProviderEnabled ([String](http://developer.android.com/reference/java/lang/String.html) provider)

Called when the provider is enabled by the user.

1. public abstract void onStatusChanged ([String](http://developer.android.com/reference/java/lang/String.html) provider, int status,[Bundle](http://developer.android.com/reference/android/os/Bundle.html) extras)

Called when the provider status changes.

This method is called when a provider is unable to fetch a location or if the provider has recently become available after a period of unavailability.