

# SHREE H. N. SHUKLA GROUP OF COLLEGES

*(Affiliated to Saurashtra University & Gujarat Technological University)*

## UNIT – 5

### Location Based Services and Google Maps

#### Deployment of applications

##### ➤ Get Started with Publishing

Start publishing on Google Play in minutes by:

- Registering for a Google Play publisher account
- Setting up a Google Wallet Merchant Account, if you will sell apps or in-app products.
- Exploring the [Google Play Developer Console](#) and publishing tools.

When you're ready, use the Start button to go to the Developer Console.

Register for a Publisher Account

---

1. Visit the [Google Play Developer Console](#).
2. Enter basic information about your **developer identity** — name, email address, and so on. You can modify this information later.
3. Read and accept the **Developer Distribution Agreement** for your country or region. Note that apps and store listings that you publish on Google Play must comply with the Developer Program Policies and US export law.
4. Pay a **\$25 USD registration fee** using Google Wallet. If you don't have a Google Wallet account, you can quickly set one up during the process.
5. When your registration is verified, you'll be notified at the email address you entered during registration.

Set Up a Google Wallet Merchant Account

---


# SHREE H. N. SHUKLA GROUP OF COLLEGES

*(Affiliated to Saurashtra University & Gujarat Technological University)*



If you want to sell priced apps, in-app products, or subscriptions, you'll need a Google Wallet Merchant Account. You can set one up at any time, but first review the list of [merchant countries](#).

To set up a Google Wallet Merchant Account:

1. **Sign in** to your Google Play Developer Console at
2. <https://play.google.com/apps/publish/>.
3. Open **Financial reports**  on the side navigation.
4. Click **Setup a Merchant Account now**.

This takes you to the Google Wallet site; you'll need information about your business to complete this step.

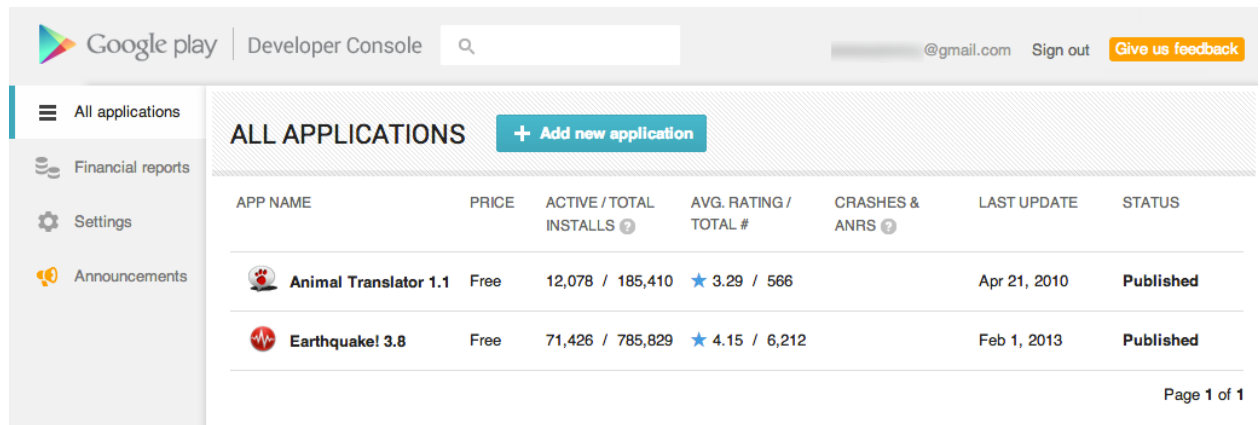
## Explore the Developer Console

---

When your registration is verified, you can sign in to your Developer Console, which is the home for your app publishing operations and tools on Google Play.

# SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



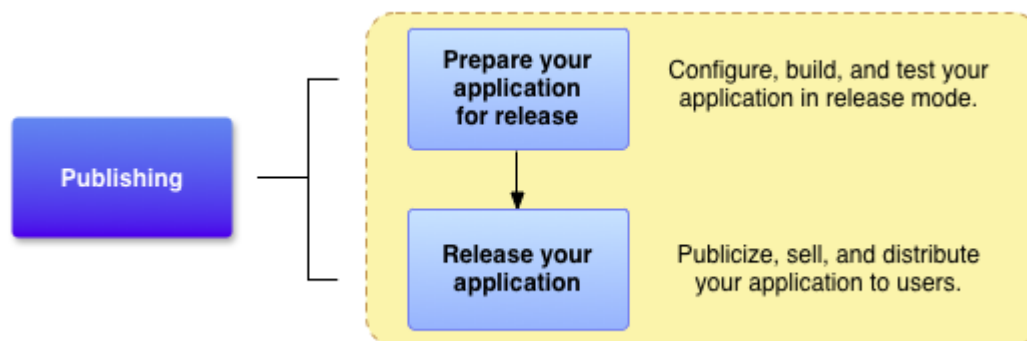
The screenshot shows the Google Play Developer Console interface. At the top, there is a search bar and a 'Give us feedback' button. The main content area is titled 'ALL APPLICATIONS' and includes a '+ Add new application' button. Below this is a table listing applications with columns for App Name, Price, Active / Total Installs, Avg. Rating / Total #, Crashes & ANRS, Last Update, and Status.

APP NAME	PRICE	ACTIVE / TOTAL INSTALLS	AVG. RATING / TOTAL #	CRASHES & ANRS	LAST UPDATE	STATUS
Animal Translator 1.1	Free	12,078 / 185,410	★ 3.29 / 566		Apr 21, 2010	Published
Earthquake! 3.8	Free	71,426 / 785,829	★ 4.15 / 6,212		Feb 1, 2013	Published

Page 1 of 1

## ➤ Preparing for Release

Before you distribute your Android application to users you need to prepare it for release. The preparation process is a required [development task](#) for all Android applications and is the first step in the publishing process (see figure 1).



When you prepare your application for release, you configure, build, and test a release version of your application. The configuration tasks are straightforward, involving basic code cleanup and code modification tasks that help optimize your application. The build process is similar to the debug build process and can be done using JDK and Android SDK tools. The testing tasks serve as a final check, ensuring that your application performs as expected under real-world conditions. When you are finished preparing your application for release you have a signed .apk file, which you can distribute directly to users or distribute through an application marketplace such as Google Play.

This document summarizes the main tasks you need to perform to prepare your application for release. The tasks that are described in this document apply to all

# SHREE H. N. SHUKLA GROUP OF COLLEGES

*(Affiliated to Saurashtra University & Gujarat Technological University)*

Android applications regardless how they are released or distributed to users. If you are releasing your application through Google Play, you should also read [Publishing Checklist for Google Play](#) to be sure your release-ready application satisfies all Google Play requirements.

---

Publishing is the general process that makes your Android applications available to users. When you publish an Android application you perform two main tasks:

- You prepare the application for release.

During the preparation step you build a release version of your application, which users can download and install on their Android-powered devices.

- You release the application to users.

During the release step you publicize, sell, and distribute the release version of your application to users.

Usually, you release your application through an application marketplace, such as [Google Play](#). However, you can also release applications by sending them directly to users or by letting users download them from your own website.

Figure 1 shows how the publishing process fits into the overall Android [application development process](#). The publishing process is typically performed after you finish testing your application in a debug environment. Also, as a best practice, your application should meet all of your release criteria for functionality, performance, and stability before you begin the publishing process.



## Canvas

The Canvas class holds the "draw" calls. To draw something, you need 4 basic components: A Bitmap to hold the pixels, a Canvas to host the draw calls (writing into the bitmap), a drawing primitive (e.g. Rect, Path, text, Bitmap), and a paint (to describe the colors and styles for the drawing).

# SHREE H. N. SHUKLA GROUP OF COLLEGES

*(Affiliated to Saurashtra University & Gujarat Technological University)*

## Summary

---

### Nested classes

enum [Canvas.EdgeType](#)

Constant values used as parameters to `quickReject()` calls.

enum [Canvas.VertexMode](#)

## Paint

The Paint class is a special class in Android. You can think of its object as a "brush" using which you can draw on the canvas. Canvas object allows you to perform **drawing** operations on it, e.g. draw lines, circle, text and Bitmaps.

## Geocoding

A **class** for handling **geocoding** and reverse **geocoding**. **Geocoding** is the process of transforming a street address or other description of a location into a (latitude, longitude) coordinate. ... The **Geocoder class** requires a backend service that is not included in the core **android** framework.

## MapOverlays

A ground **overlay** is an image that is fixed to a **map**. Unlike markers, **groundoverlays** are oriented against the Earth's surface rather than the screen, so rotating, tilting or zooming the **map** will change the orientation of the image. Ground **overlays** are useful when you wish to fix a single image at one area on the **map**

# SHREE H. N. SHUKLA GROUP OF COLLEGES

*(Affiliated to Saurashtra University & Gujarat Technological University)*

## GoogleMap

This is the main class of the Google Maps Android API and is the entry point for all methods related to the map. You cannot instantiate a GoogleMap object directly, rather, you must obtain one from the `getMapAsync()` method on a MapFragment or MapView that you have added to your application.

Note: Similar to a View object, a GoogleMap can only be read and modified from the Android UI thread. Calling GoogleMap methods from another thread will result in an exception.

You can adjust the viewpoint of a map by changing the position of the camera (as opposed to moving the map). You can use the map's camera to set parameters such as location, zoom level, tilt angle, and bearing. For more information, see [Camera](#) and [View](#).