Setup Google Play Services SDK

To develop an app using the Google Play services APIs (/reference/gms-packages.html), you must download the Google Play services SDK from the SDK Manager (/tools/help/sdk-manager.html). The download includes the client library and code samples.

To test your app when using the Google Play services SDK, you must use either:

- A compatible Android device that runs Android 2.2 or higher and includes Google Play Store.
- The Android emulator with an AVD that runs the Google APIs platform based on Android 4.2.2 or higher.

Ideally, you should develop and test your app on a variety of devices, including both phones and tablets.

Install the Google Play Services SDK

To install the Google Play services SDK for development:

1. Launch the SDK Manager.
   - From Eclipse (with ADT), select **Window > Android SDK Manager**.
   - On Windows, double-click the `SDK Manager.exe` file at the root of the Android SDK directory.
   - On Mac or Linux, open a terminal and navigate to the `tools/` directory in the Android SDK, then execute `android sdk`.
2. Install the Google Play services SDK.

   Scroll to the bottom of the package list, expand **Extras**, select **Google Play services**, and install it.

   The Google Play services SDK is saved in your Android SDK environment at `<android-sdk>/extras/google/google_play_services/`.

3. Install a compatible version of the Google APIs platform.

   If you want to test your app on the emulator, expand the directory for **Android 4.2.2 (API 17)** or a higher version, select **Google APIs**, and install it. Then create a new AVD (/tools/devices/index.html) with Google APIs as the platform target.

   > **Note:** Only Android 4.2.2 and higher versions of the Google APIs platform include Google Play services.

4. Make a copy of the Google Play services library project.

   Copy the library project at `<android-sdk>/extras/google/google_play_services/libproject/google-play-services_lib/` to the location where you maintain your Android app projects.

   If you are using Eclipse, import the library project into your workspace. Click **File > Import**, select **Android > Existing Android Code into Workspace**, and browse to the copy of the library project to import it.

Set Up a Project with the Library

To set up a project to use the Google Play services SDK:

1. Reference the library project in your Android project.

   See the Referencing a Library Project for Eclipse (/tools/projects/projects-eclipse.html#ReferencingLibraryProject) or Referencing a Library Project on the Command Line (/tools/projects/projects-cmdline.html#ReferencingLibraryProject) for more information on how to do this.

   > **Note:** You should be referencing a copy of the library that you copied to your development workspace—you should not reference the library directly from the Android SDK directory.

2. If you are using ProGuard, add the following lines in the `<project_directory>/proguard-project.txt` file to prevent ProGuard from stripping away required classes:

```
-keep class * extends java.util.ListResourceBundle {
    protected Object[][] getContents();
}
```

Once you have the Google Play services library project added to your app project, you can begin developing features with the Google Play services APIs (/reference/gms-packages.html).

Ensure Devices Have the Google Play services APK

As described in the Google Play services introduction (/google/play-services/index.html), Google Play delivers service updates for users on Android 2.2 through the Google Play Store app. However, updates might not reach all users immediately.

> **Important:** Because it is hard to anticipate the state of each device, you must *always* check for a compatible Google Play services APK before you access Google Play services features. For many apps, the best time to check is during the `onResume()` (/reference/android/app/Activity.html#onResume()) method of the main activity.

Here are four scenarios that describe the possible state of the Google Play services APK on a user's device:

1. A recent version of the Google Play Store app is installed, and the most recent Google Play services APK has been downloaded.
2. A recent version of the Google Play Store app is installed, but the most recent Google Play services APK has *not* been downloaded.
3. An old version of the Google Play Store app, which does not proactively download Google Play services updates, is present.
4. The Google Play services APK is missing or disabled on the device, which might happen if the user explicitly uninstalls or disables it.

Case 1 is the success scenario and is the most common. However, because the other scenarios can still happen, you must handle them every time your app connects to a Google Play service to ensure that the Google Play services APK is present, up-to-date, and enabled.

To help you, the Google Play services client library has utility methods to determine whether or not the Google Play services APK is recent enough to support the version of the client library you are using. If not, the client library sends users to the Google Play Store to download the recent version of the Google Play services APK.

> **Note:** The Google Play services APK is not visible by searching the Google Play Store. The client library provides a deep link into the Google Play Store when it detects that the device has a missing or incompatible Google Play services APK.

It is up to you choose the appropriate place in your app to do the following steps to check for a valid Google Play services APK. For example, if Google Play services is required for your app, you might want to do it when your app first launches. On the other hand, if Google Play services is an optional part of your app, you can do these checks if the user navigates to that portion of your app:

1. Query for the status of Google Play services on the device with the `isGooglePlayServicesAvailable()` method, which returns a result code.
2. If the result code is `SUCCESS`, then the Google Play services APK is up-to-date, and you can proceed as normal.
3. If the result code is `SERVICE_MISSING`, `SERVICE_VERSION_UPDATE_REQUIRED`, or `SERVICE_DISABLED`, then call `getErrorDialog()` to display an error message to the user, which allows the user to download the APK from the Google Play Store or enable it in the device's system settings.