Android™ Build Error Analyzer

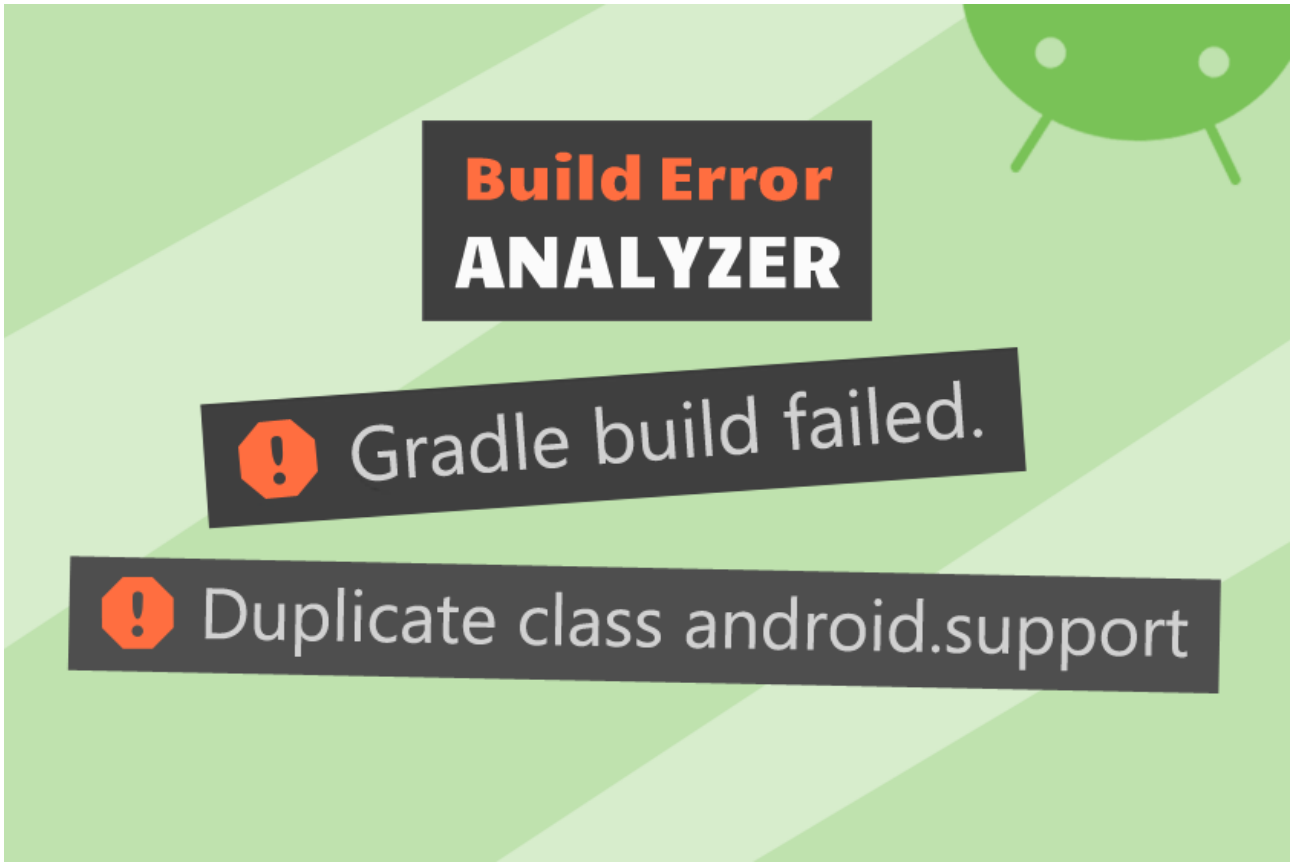


Table of contents

Requirements & Setup

Requirements

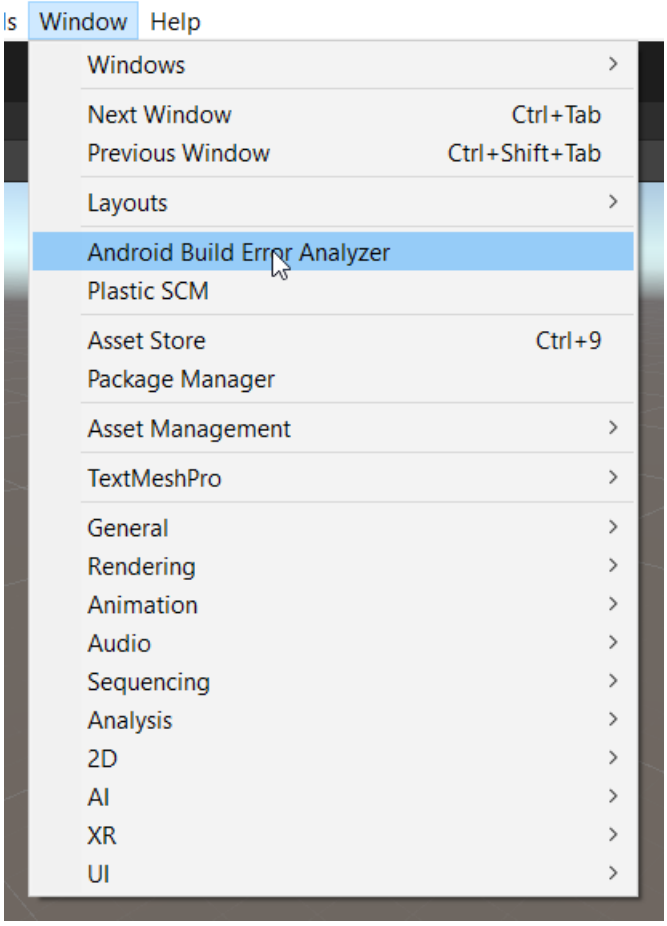
Unity 2021.3 or higher is required since that is the min version Unity allows for new assets in the store.

There is an older with less features that is compatible with Unity 2019.4+ and Unity 2020. If you need it then please write to office[at]kamgam.com

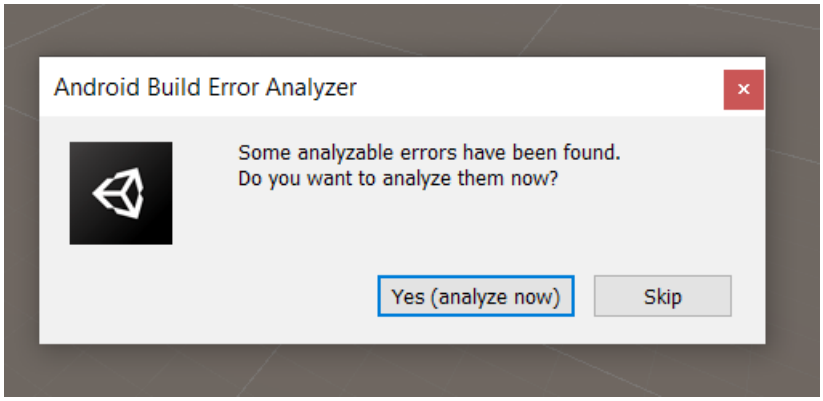
Opening the Analyzer Window

The „Build Error Analyzer“ window shows four buttons and an (initially blank) „results“ area.
There are two ways to open the Build Error Analyzer Window.

The first is to open it via the main menu: **Window > Android Build Error Analyzer**



The second one is the popup which is shown after a failed build.
(This can be disabled in the settings. It is enabled by default.)

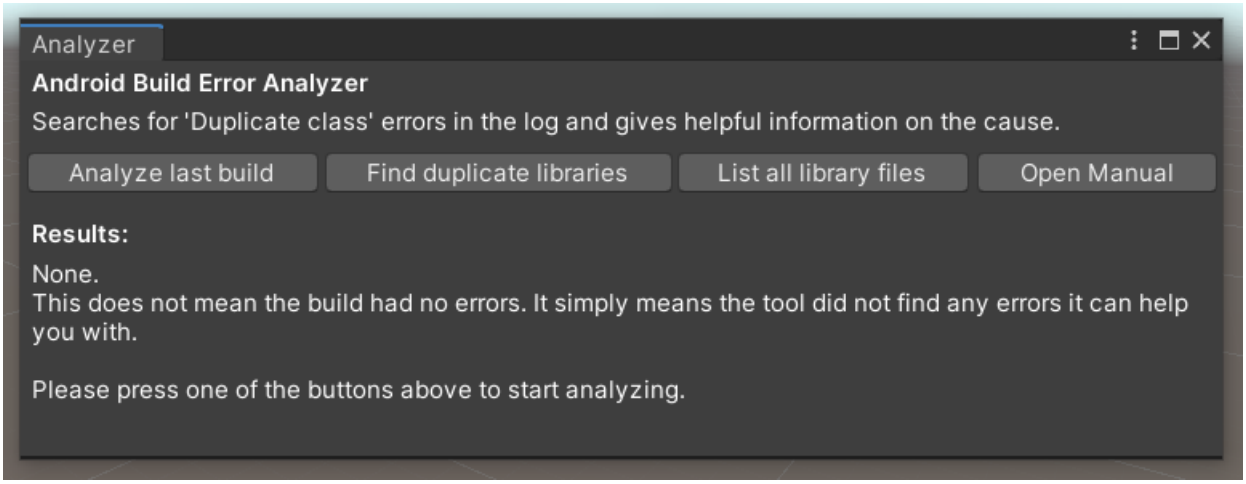


What do these buttons do?

The window shows four buttons to start different analysis.

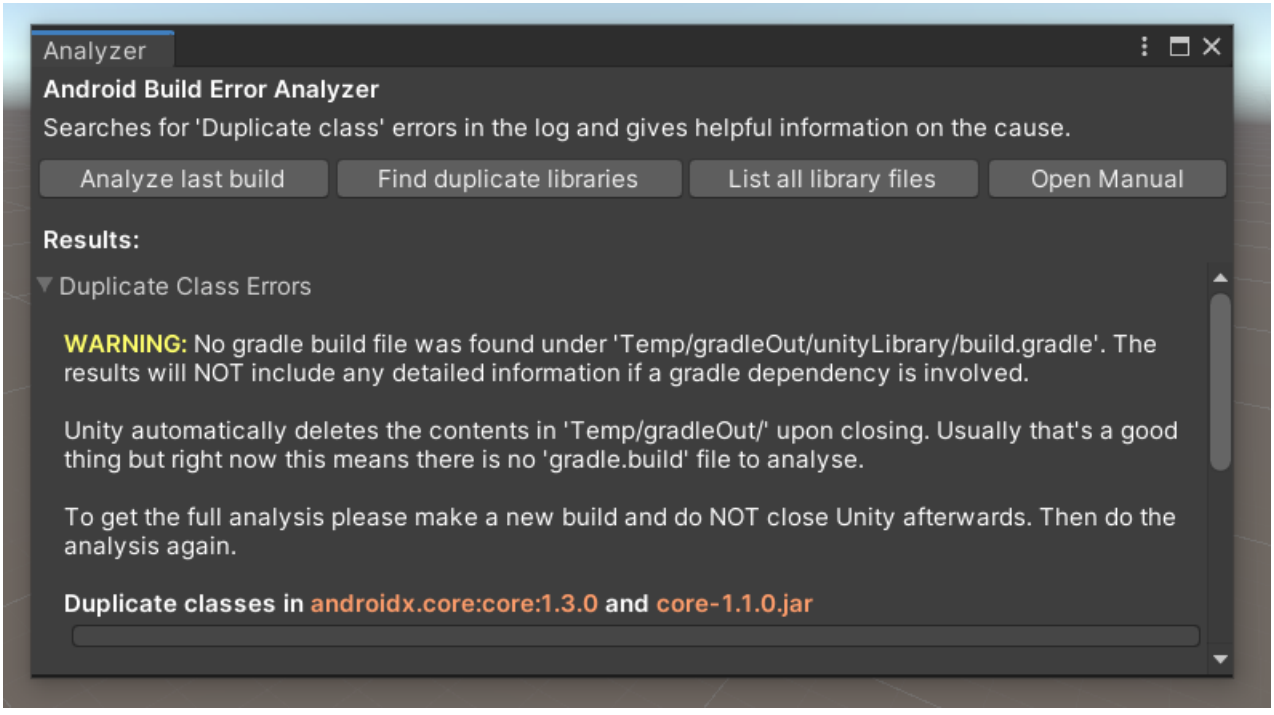
Analyze the last build

This button triggers the detailed analysis. It will take 10 to 60 seconds to do it all. It will search the whole project, packages and gradle templates for errors.

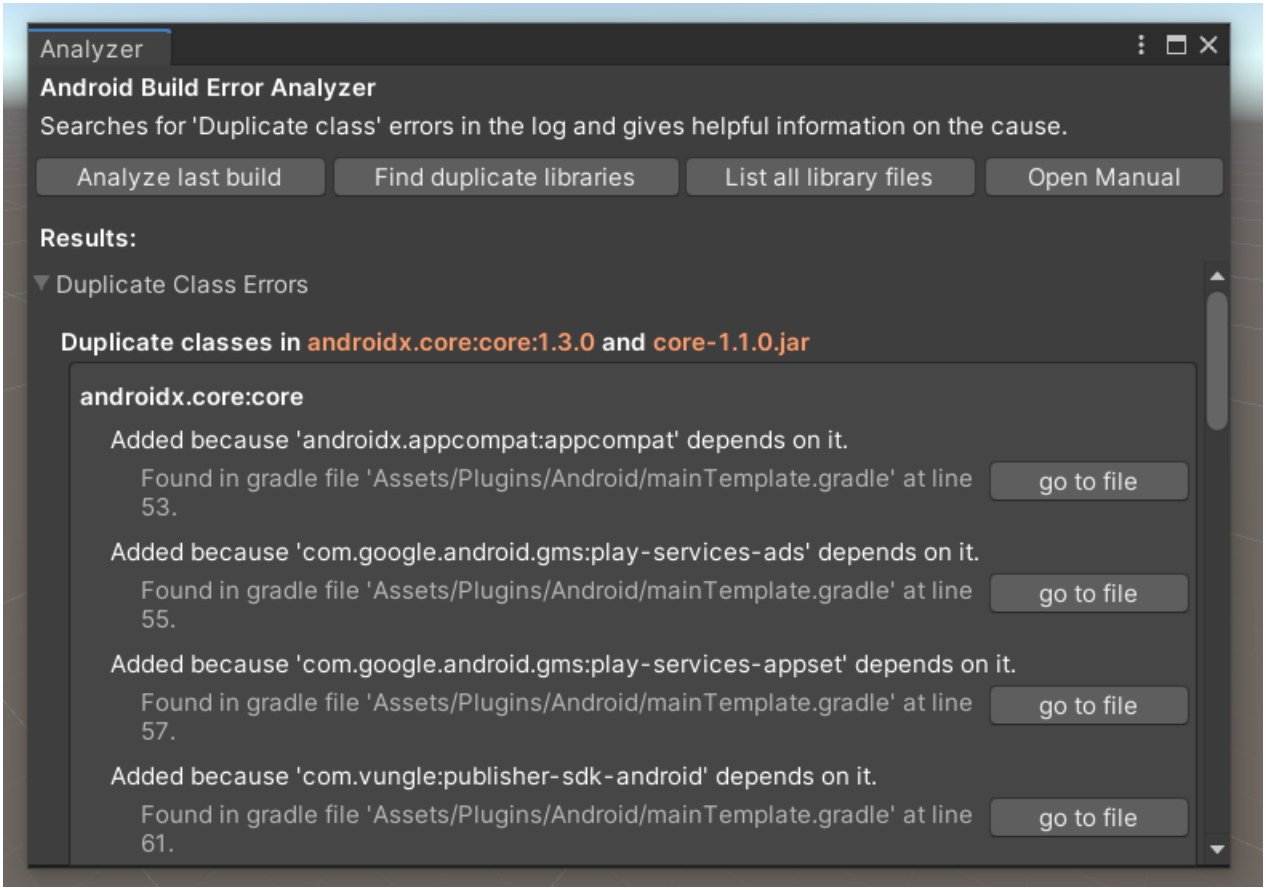


To give you details on the missing or duplicate libraries the analyzer requires some information from the last build. These files are usually temporary (deleted if Unity is closed). To get the best results you should do the analysis immediately after a build has failed.

A warning will be displayed if the last build data was not found.



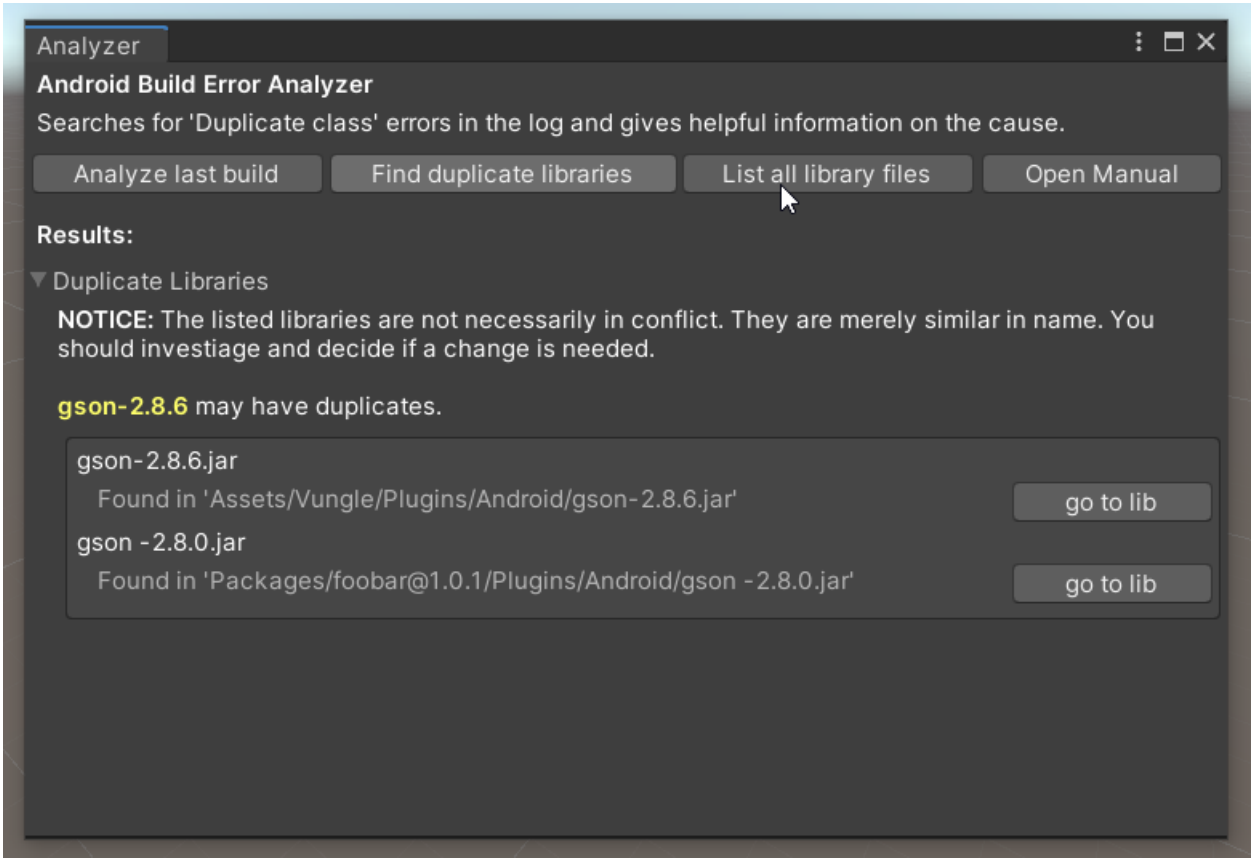
The results will give you an overview of why a certain library was included in the build and which other library it is in conflict with.



Find duplicate libraries

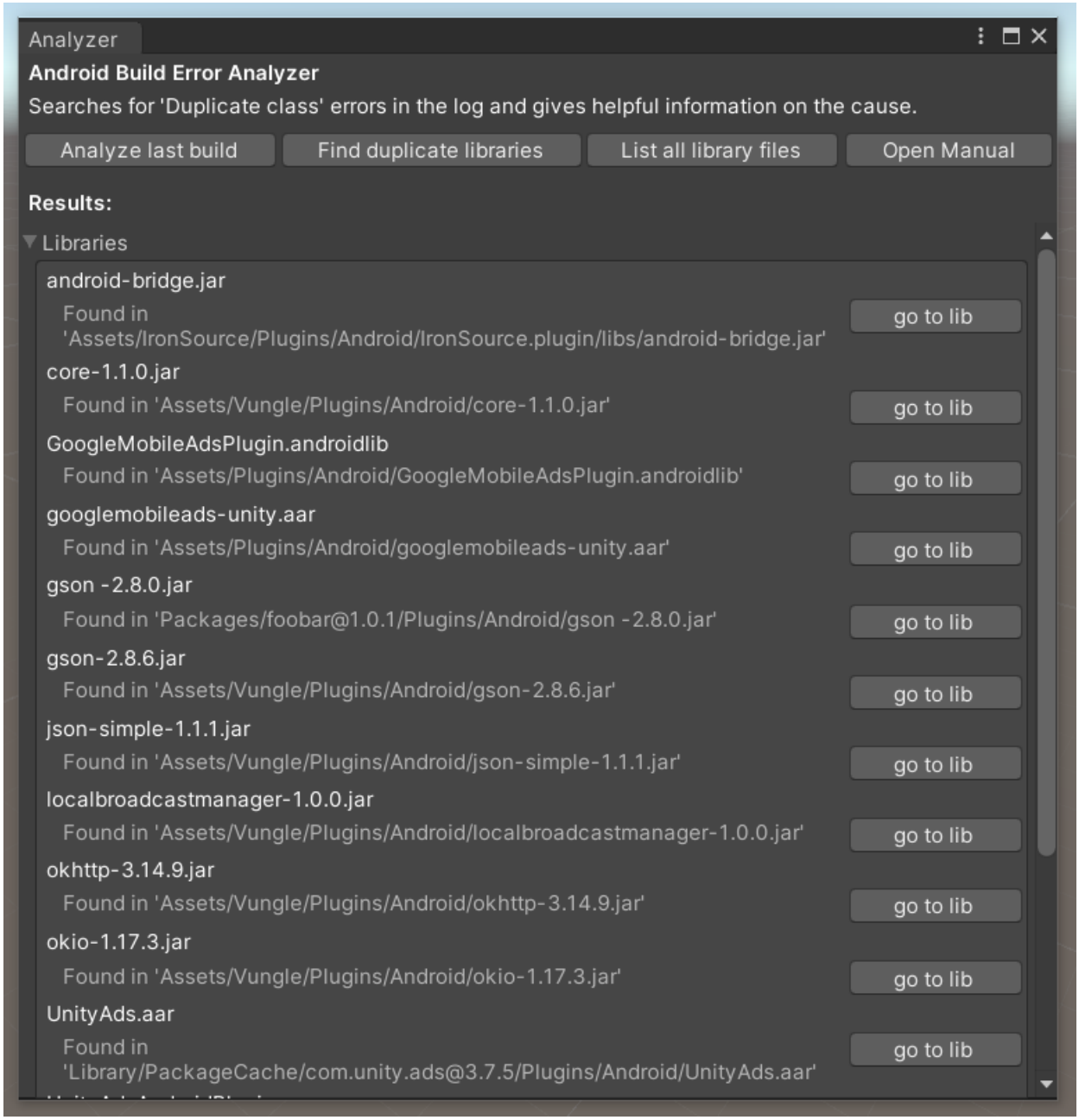
This button triggers a search for library files (.jar, .aar, .androidlib) in your project. It then compares the found libraries and displays a list of libs which have similar names (i.e. they are likely to cause conflicts).

Notice: the results are not guaranteed to cause errors. They are just a guess by the analyzer.



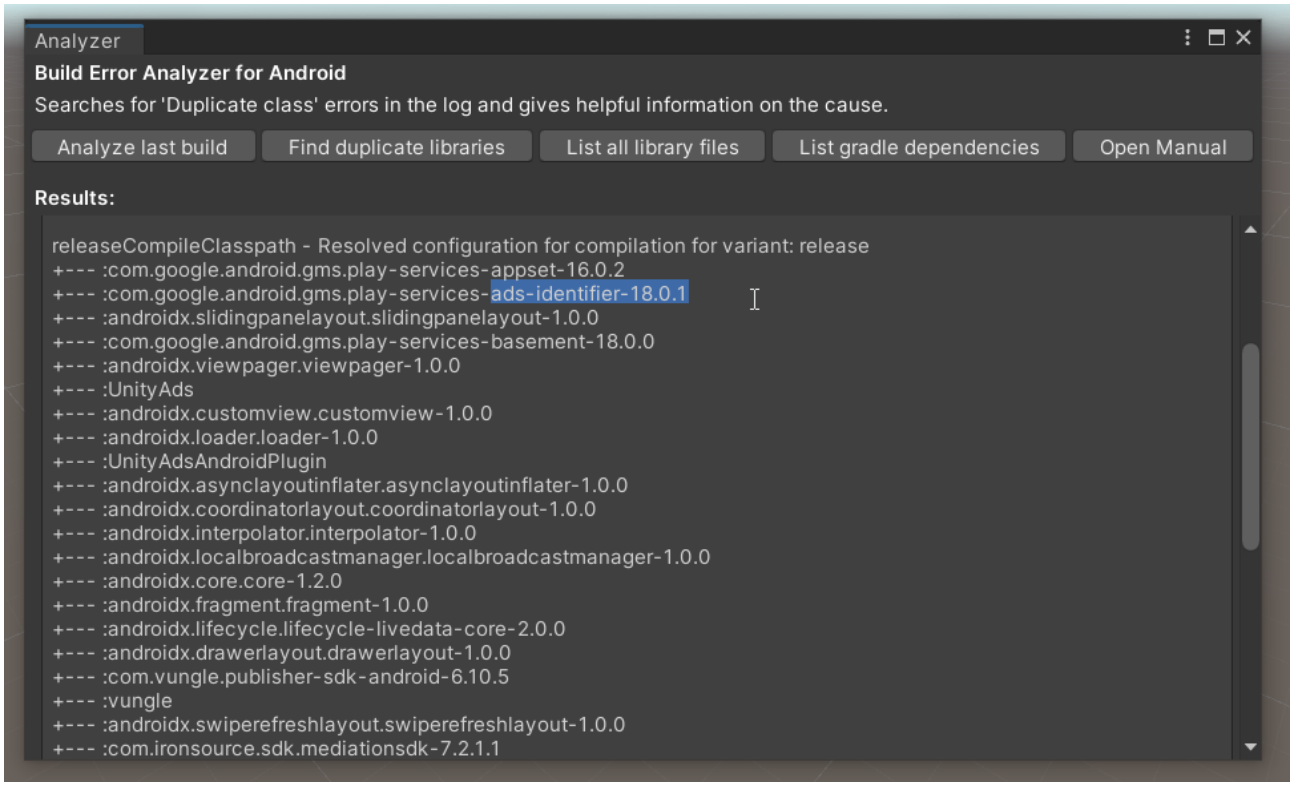
List all libraries

This button starts a search for all library files (.jar, .aar, .androidlib) in your project. The results are then listed in alphabetical order. It is useful to get a quick overview of which libraries exist within the project.



List gradle dependencies

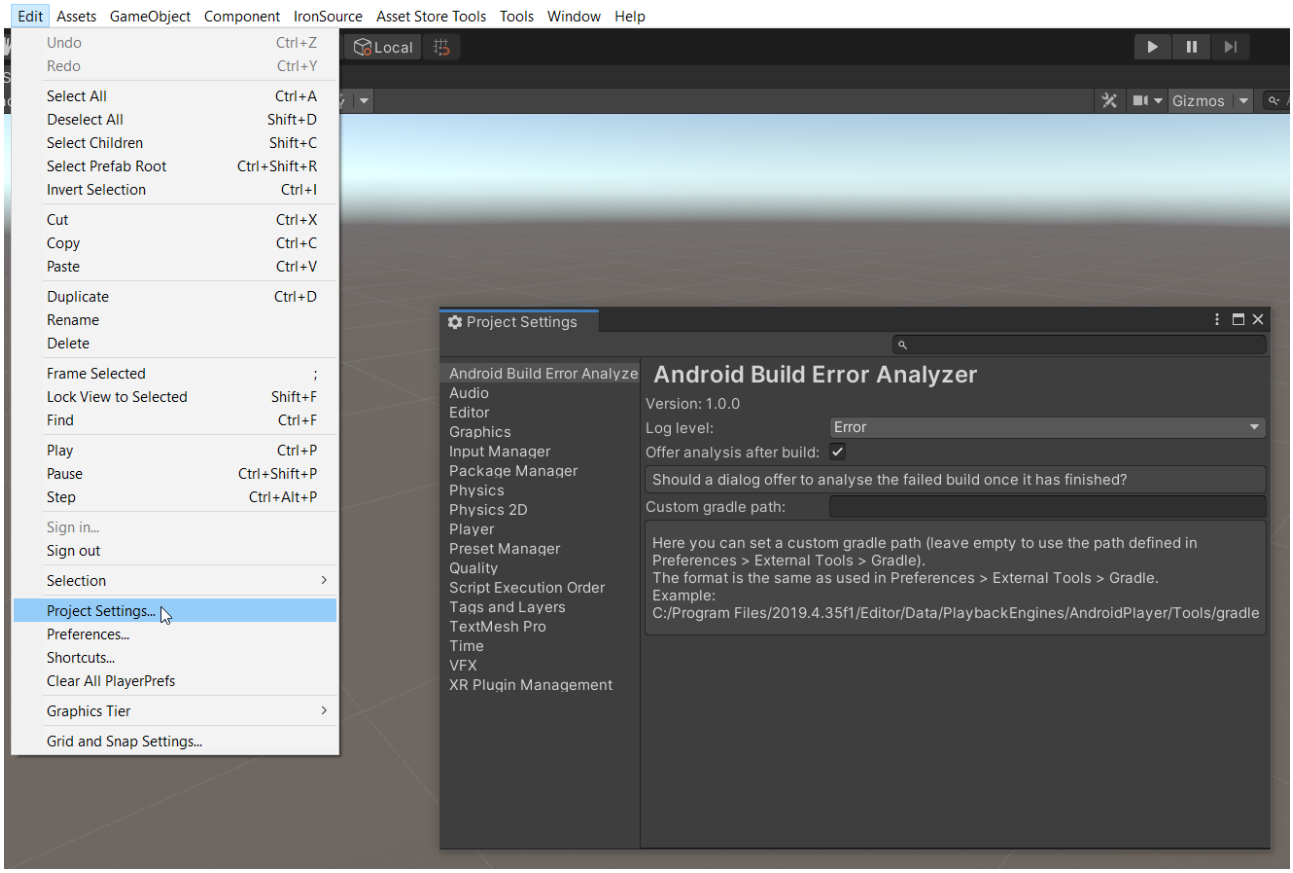
You can use this to list all the dependencies available in the gradle build log. Please notice that these are only accurate while Unity is opened AFTER a build. If you close Unity it sadly does sometimes also clean up the build remnants (gradle build results).



This may help you to find out where a certain library that is added by gradle may come from.

Settings

The settings are stored in a Scriptable Object located under Assets/. You can access them via the Project Settings.. menu.



Frequently Asked Questions

I have a duplicate class error and the analysis result. Now what should I do with this information?

1) Make a backup or commit to your VCS:

First step would be making sure you have everything committed to your version control system (I'd recommend GIT). That way you can revert any changes easily.

2) Find Libs:

Second step would be to check if the "Find duplicate libraries" option shows any duplicate library files. If yes, then find the newest one (that one you keep). Remove all the other (older) ones.

3) Analyze Build:

If no duplicate lib files are found then try the "Analyze last build" option. If that does not find anything either then it's an error the tool does not recognize -OR- you are exporting as an Android Studio project.

Android Studio project exports are not supported since Unity has no access to the logs produced in Android Studio (it's a separate program). In that case please try to build directly from Unity instead.

If it finds something then the reason is probably a conflicting gradle dependency. Most likely that's caused by some third party asset (Advertisement SDKs are a common culprit). Try disabling each third party asset and make a new build. Do that until the error is gone. Once you know which one causes the problem then check if they are modifying your gradle template imports and resolve any conflicting imports. The External Dependency Manager might help you there (in case you have not yet tried it).

Please understand that the tool is no auto-fix solution. What it does is try to get you the info you need to solve the problem. You still have to resolve the conflicts yourself as it's just impossible to know all the library combinations.

Can I use this to analyze Android Studio builds?

TLDR: No, not really.

Android Studio project exports are not supported since Unity has no access to the logs produced in Android Studio (it's a separate program). Please try to build directly from Unity instead to fetch the errors. Once you have resolved them you can return to Studio exports.