

# Exclude From Build - Manual



1. Hides all unwanted files and directories before build.
2. Removes all unwanted game objects and components from the scenes during build.
3. Let's Unity finish the build.
4. Reverts step 1 and 2.

## ***Table of contents***

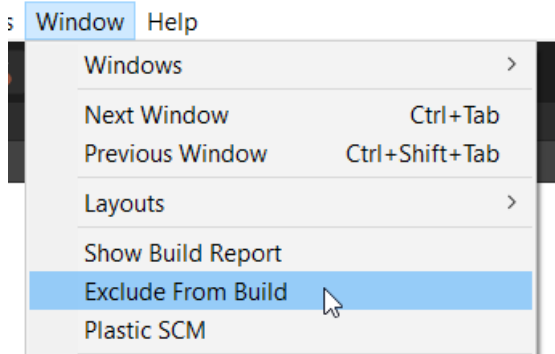
<b>Opening the ,Exclude From Build' Window.....</b>	<b>3</b>
<b>Groups.....</b>	<b>4</b>
Editing Groups.....	6
Headless & Dedicated Server Builds.....	7
Exclude based on Scripting Define Symbols.....	8
<b>Excluded Files and Directories.....</b>	<b>9</b>
How to exclude files and directories.....	9
Testing (,Start test' & ,Stop test').....	11
<b>Excluded Objects.....</b>	<b>13</b>
How to exclude GameObjects and Components.....	14
<b>Settings.....</b>	<b>16</b>
<b>Streaming Assets &amp; Build Reports.....</b>	<b>17</b>
<b>Packages &amp; Embedded Packages.....</b>	<b>18</b>

Excluding „Editor“ folders.....19

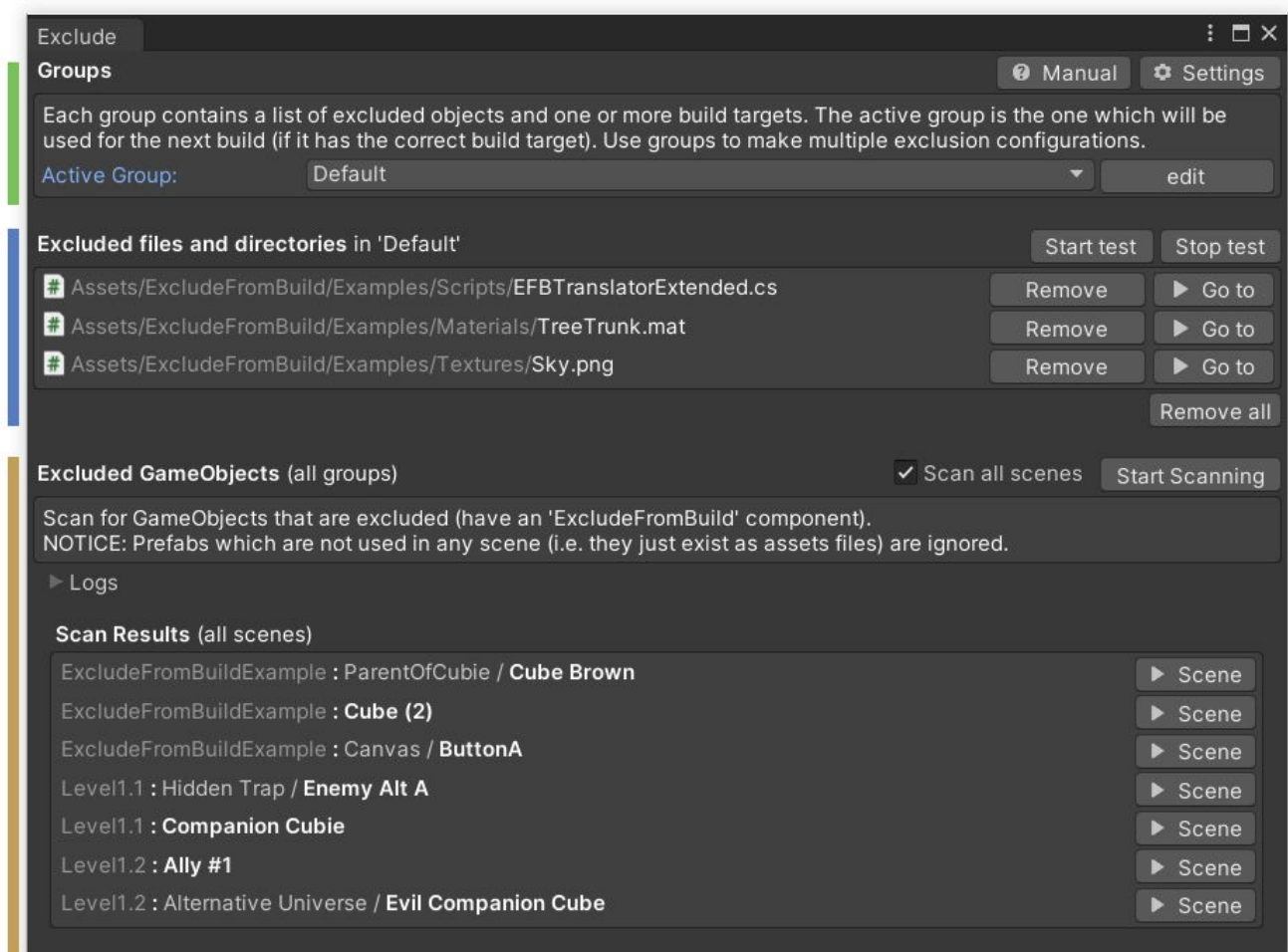
# Opening the ,Exclude From Build' Window

The window gives you an overview of the files, directories and components that are excluded from the build. It also allows you to edit your groups (more on that later).

To open it go to: **Window > Exclude From Build**

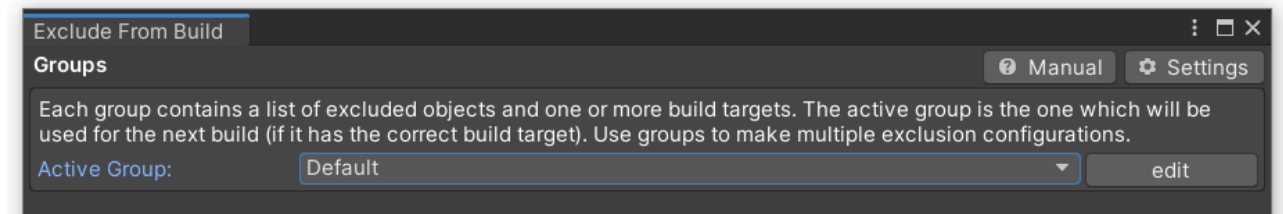


The window shows three sections ,Groups', ,Excluded files' and ,Excluded GameObjects'.



# Groups

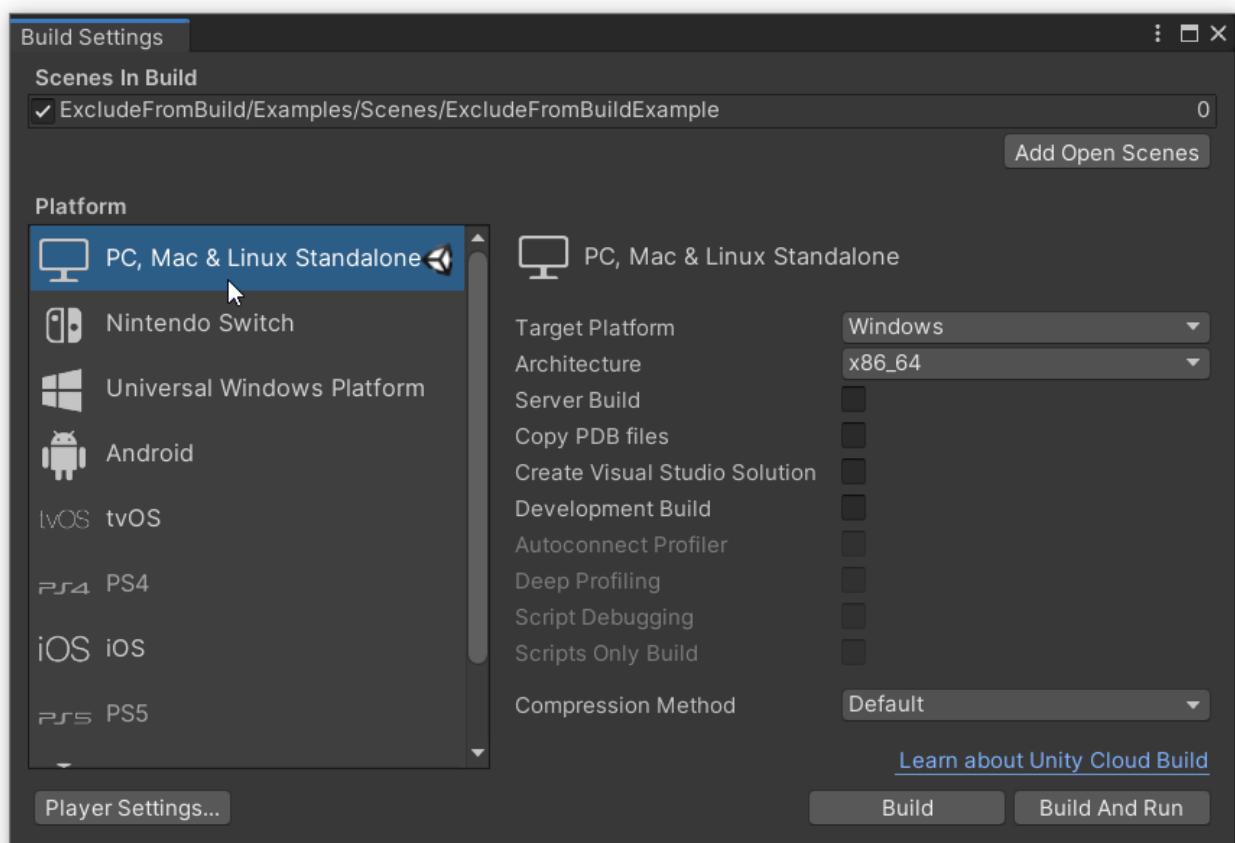
Imagine groups like folders containing information that belongs together.



Each group contains two things:

A) A list of files and directories to exclude from the build. They are listed in the **Excluded files and directories** section.

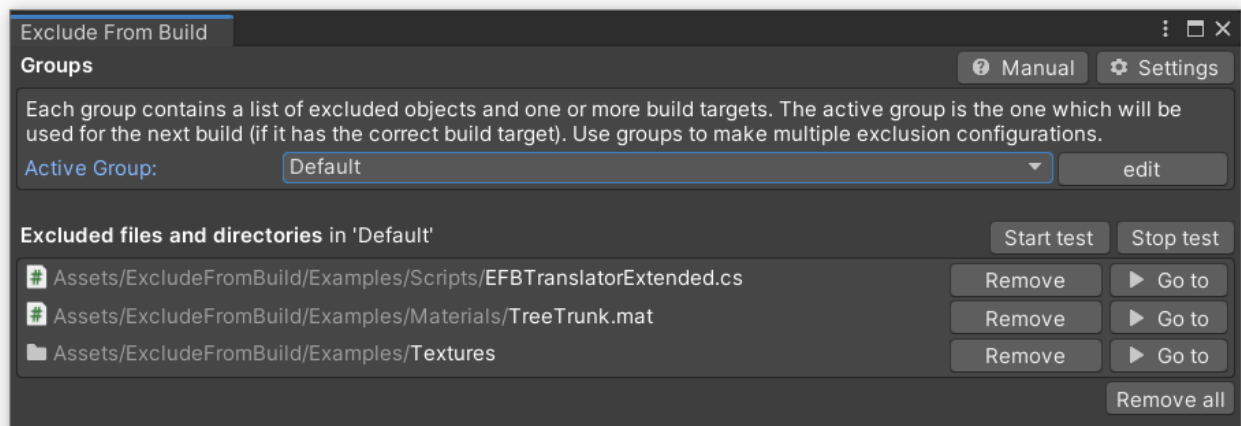
B) A list of one or more 'Build Targets' (also called 'platforms') like Android, iOS, Standalone, ... . You may remember seeing those on the left side of the 'Build Settings' Dialog.



Groups exist so you can vary what files you are excluding. Maybe that super high res texture should be in the Standalone build but not in the mobile build?

At build time the active group's build targets are compared to the platform you have chosen in the Build Settings. If they match then the group's excluded files and directories will be removed from the build. The same is done for all 'Exclude From Build' components in the scenes.

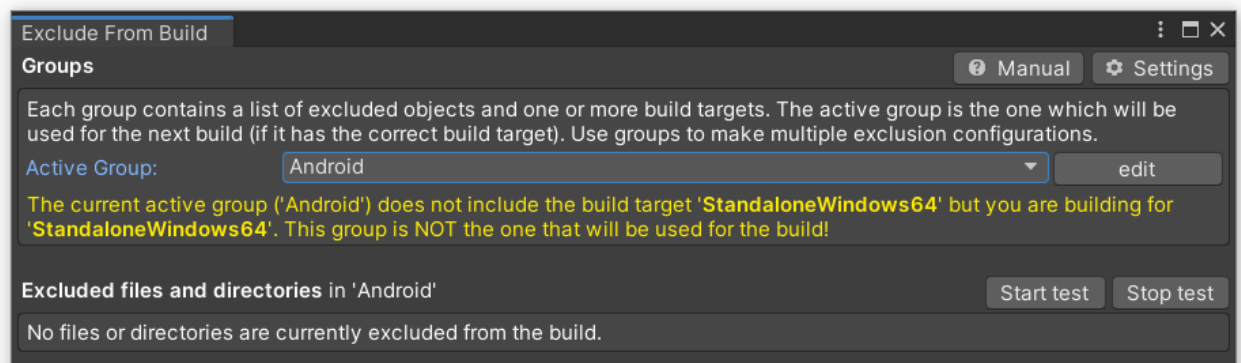
Changing the active group (the big dropdown) will change what files are excluded from your next build. It also changes what files are listed in the **Excluded files and directories** section. If you add or remove a file from the list then it will only be added to or removed from the active group.



The 'active group' setting exists because you may have multiple groups targeting the same platform.

A common example is an Android build for GooglePlay vs the Amazon Store. Both have the same platform (Android) but may need different files in the build.

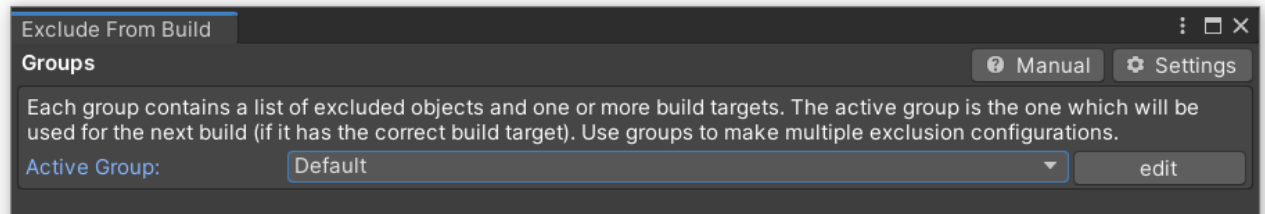
If the active group does not match the platform you have chosen in the 'Build Settings' then a warning will be displayed.



If you are just getting started with the tool then it's best to leave the groups alone and focus on the other parts first.

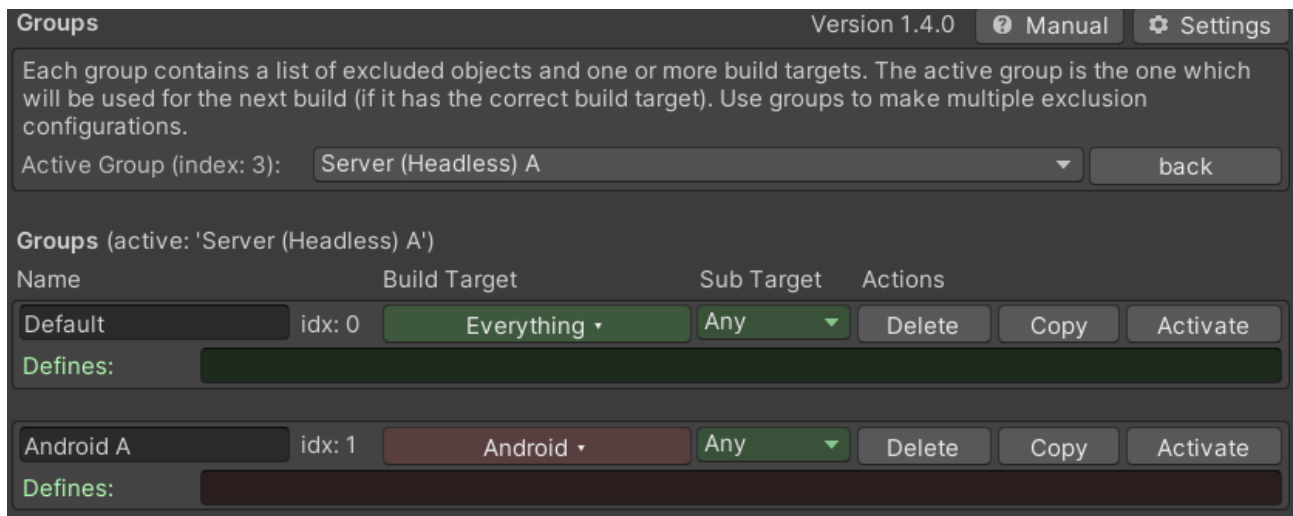
## Editing Groups

A single 'Default' group is added for you right from the start. This group includes all platforms. If you want to edit the groups then use the **,edit' button** on the right.

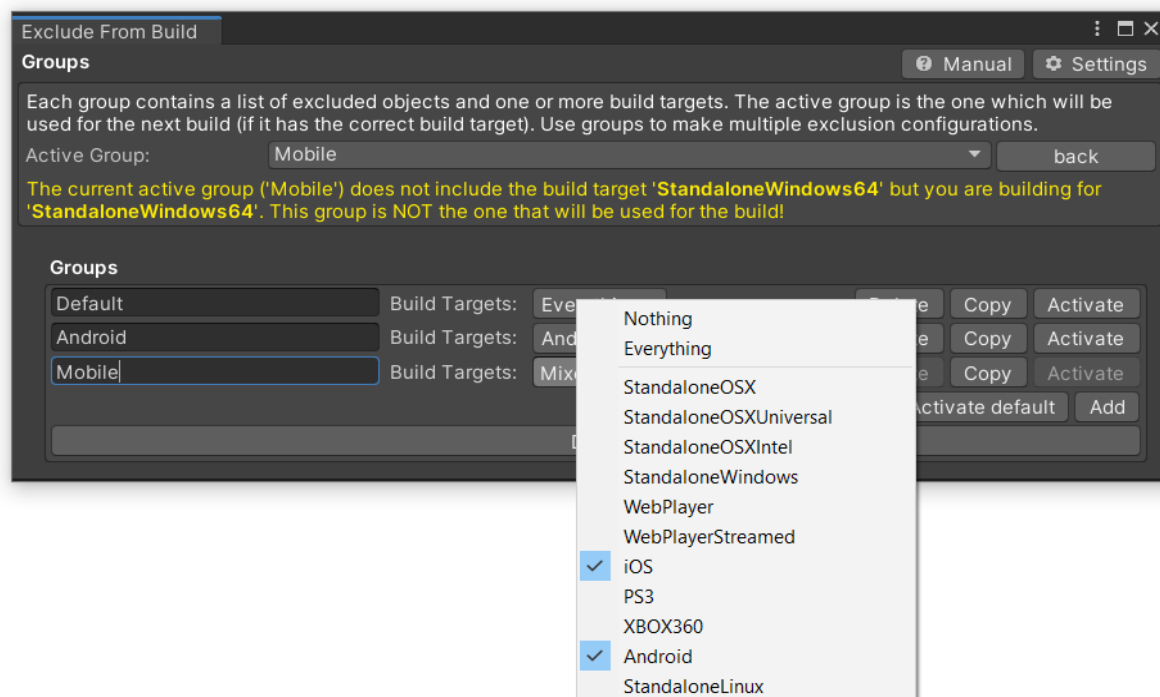


It will open the edit section. Here you can edit, copy and delete your groups.

In this example I have added one which is for Android only (see the Build Targets lists 'Android').



Remember: a group can be used for multiple build targets (aka platforms).



Once you are done hit the **,Save & Close' button** or the **,back' button** (both do the same).

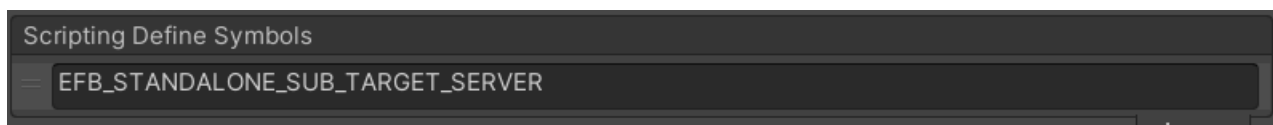
## Headless & Dedicated Server Builds

For server builds you can use the two extra options „Standalone Sub Target“ and „Defines“. Sadly the [Standalone Sub Targets](#) have only been added in Unity 2021.2+.

In Unity 2021.2+ you can set them via code:

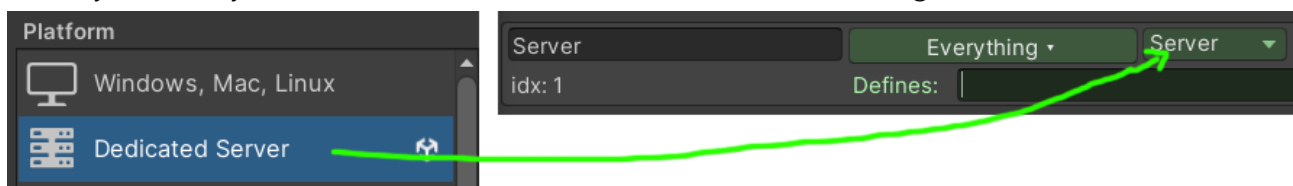
```
var buildPlayerOptions = new BuildPlayerOptions()
{
    subtarget = (int) StandaloneBuildSubtarget.Server,
    target = BuildTarget.StandaloneWindows64
};
```

In Unity 2021.1 or below you have to set a define to mark a build as a server (headless). The define is called „EFB\_STANDALONE\_SUB\_TARGET\_SERVER“ (see ExcludeFromBuildData.cs).



If you are switching from Unity 2021.1 or below to 2021.2+ then you can keep using the define. The logic for the newer Unity versions checks if the define is set OR if subtarget is set to StandaloneBuildSubtarget.Server.

In Unity 2021.2+ you can choose „Dedicated Server“ as the build target instead of the define.



You can also use the static SetActiveGroup0-19 methods of ExcludeFromBuildData. Here is an example of a batch file for Windows that builds a project with a certain group activated. For more info on how to use the commandline parameters of Unity go [here](#). Notice the „-executeMethod Kamgam.ExcludeFromBuild.ExcludeFromBuildData.SetActiveGroup%INDEX%“ part.

```
@echo off
set /p "INDEX=Exclude from Build group index: "

@echo on
"C:\Program Files\2019.4.35f1\Editor\Unity.exe" -batchMode -skipMissingProjectID -
skipMissingUPID -buildTarget StandaloneWindows64 -logFile log.txt -projectPath C:\dev\
ExcludeFromBuild\ExcludeFromBuild -executeMethod
Kamgam.ExcludeFromBuild.ExcludeFromBuildData.SetActiveGroup%INDEX% -buildWindows64Player
"./build-win/Main.exe" -quit
@echo off
type log.txt
pause
```



NOTICE: The groups are selected by index (order in the groups list, 0-19), not names:

## Exclude based on Scripting Define Symbols

You can add some required defines to your groups to limit what group is used. As with all the other settings the colors (red / green) indicate whether that group matches the current build settings or not. Only the group with „all green“ will be used for the build.

NOTICE: Defines are combined with AND logic. All specified defines need to exist for this to be a positive match.

Example of a defines not matching combo:

Name	Build Target	Sub Target	Actions		
Default	Everything ▾	Player ▾	Delete	Copy	Activate
idx: 0	Defines:	ASDF			

Scripting Define Symbols

Example of a defines matching combo:

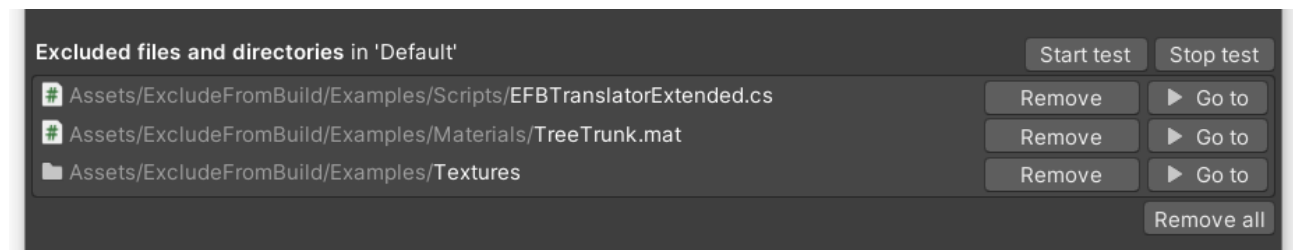
Name	Build Target	Sub Target	Actions		
Default	Everything ▾	Player ▾	Delete	Copy	Activate
idx: 0	Defines:	MY_CUSTOM_DEFINE			

Scripting Define Symbols
MY_CUSTOM_DEFINE



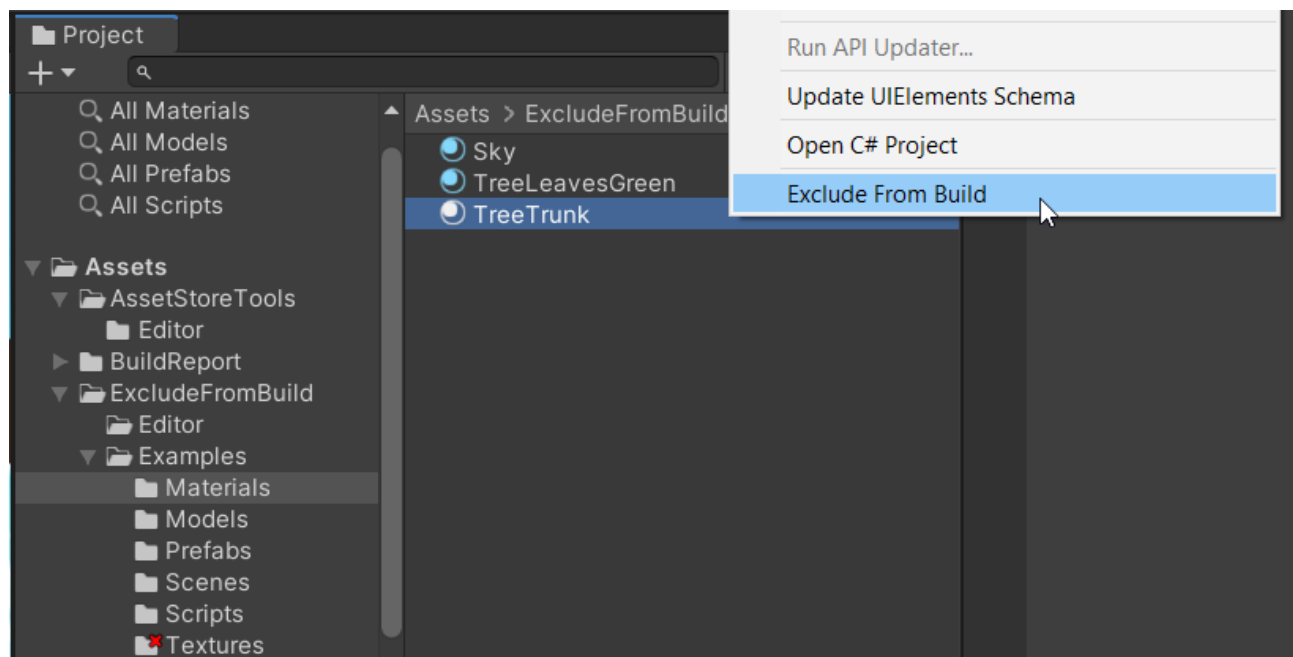
## Excluded Files and Directories

This section lists all the excluded files for the active group (,Default' in this case). You can easily remove files from the exclusion list by pressing the **,remove' button**. The **,go to' buttons** will lead you to the file in the Project window.

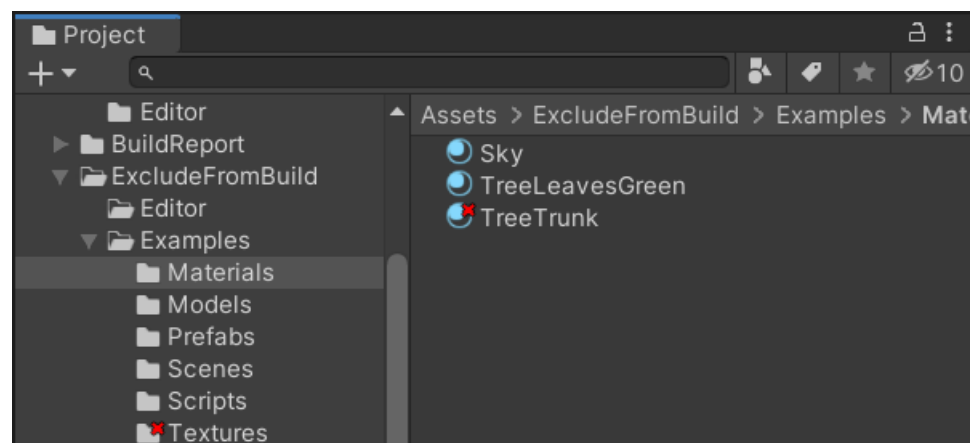


## How to exclude files and directories

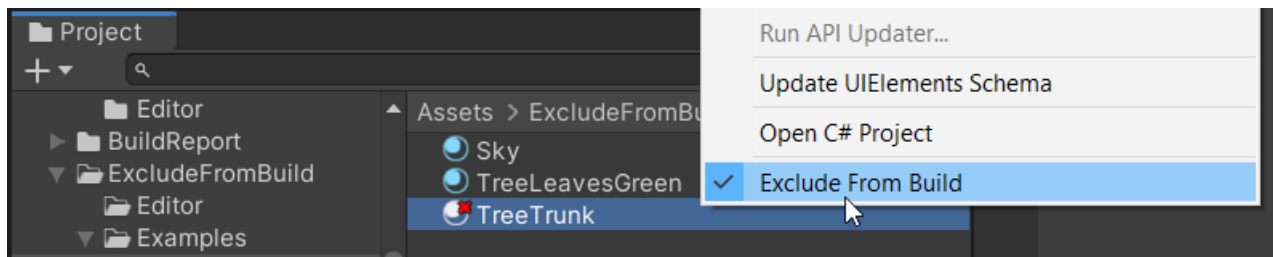
Easy, simply **right-click on the file** or directory and choose **,Exclude From Build' . Done.**



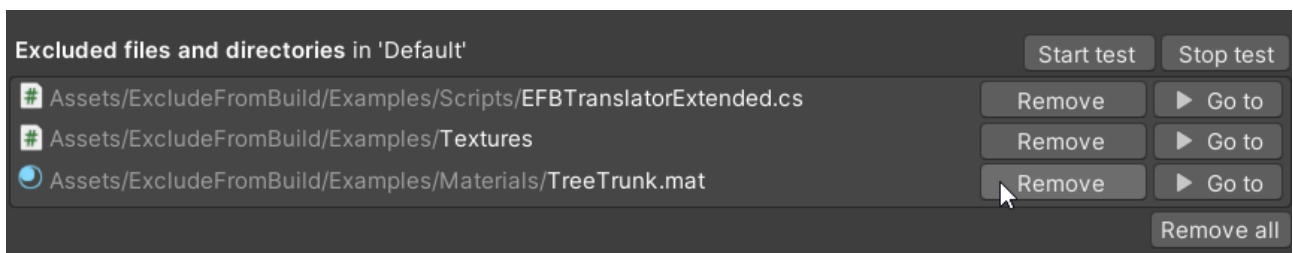
Once an object is excluded it will show a little red icon (you can disable this in the settings).



If you want to include it again then the process is the same. **Right-click** and choose **Exclude From Build**. The checkmark next to the menu entry indicates whether it is excluded or not.



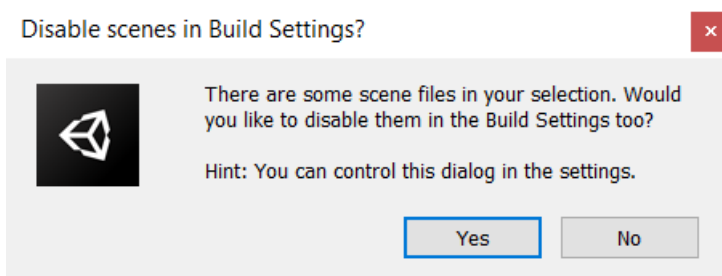
You can also remove it in the exclude window by using the 'Remove' button.



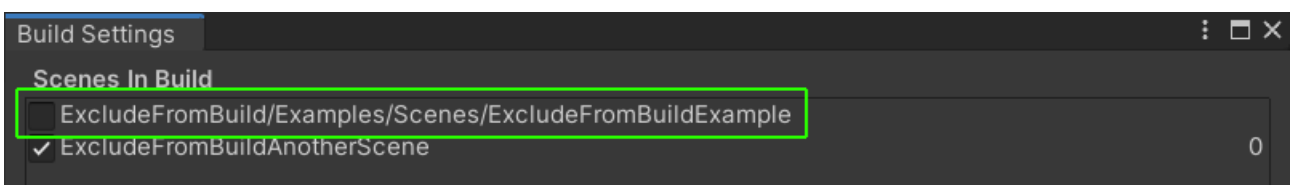
If you wonder how to exclude game objects in scenes or components then read on in the „Excluded Objects“ section below.

**HINT:** You can also exclude scene files (simply right-click on them).

Once you exclude a scene file that is also part of the build scenes list you will be asked if you want to disable it in that list.



If a scene is in the BuildSettings scenes list then excluding will not remove it but only disable it.



In the build that has the same effect as removing it from the list. We do it this way because then in the build window you can still see what scenes you had included.

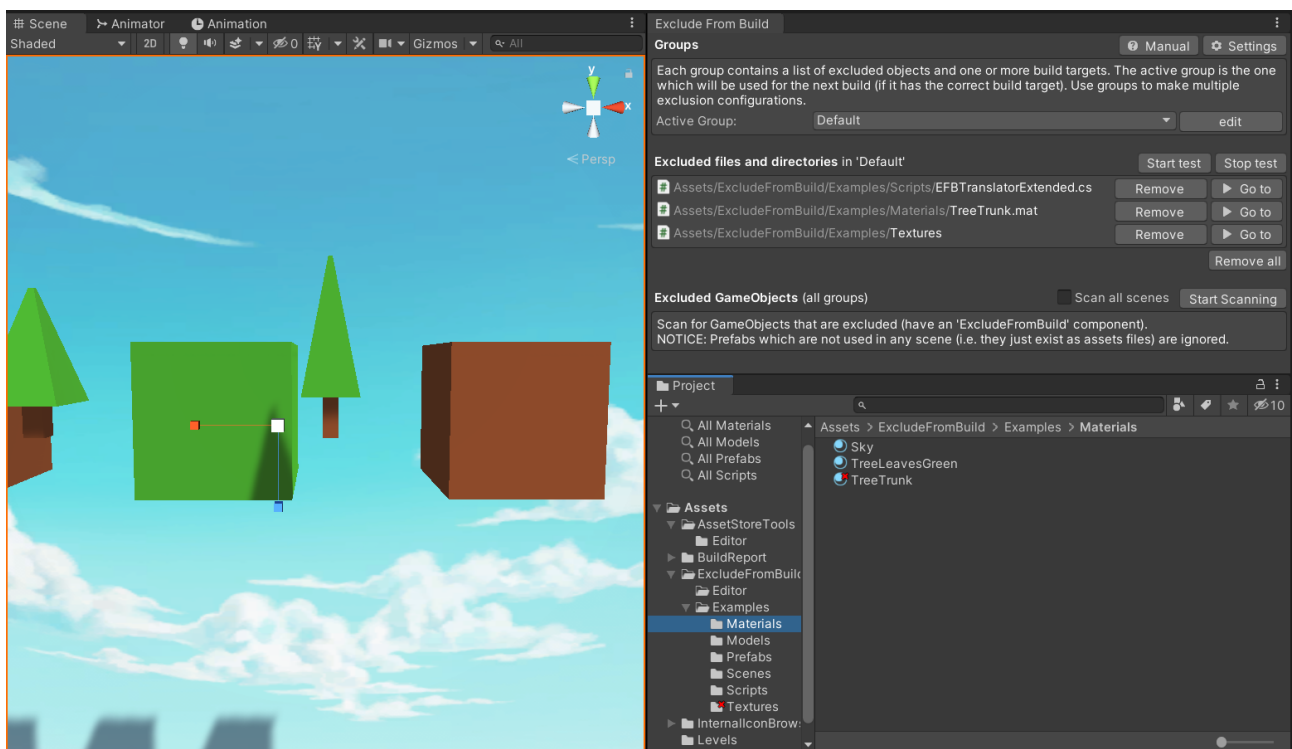
## Testing (,Start test' & ,Stop test')

It would be cumbersome having to make a build every time you wish to see how the excluded files affect the project. To make this easier you can use the **,Start test' button** (WAIT, don't click it just yet).

Starting the test will execute the same logic on the files and directories that is used to exclude them at build time. Meaning it will HIDE them from Unity by temporarily renaming them with a `,~'` at the end. For this reason it is very important that you **DO NOT COMMIT ANYTHING TO YOUR VERSION CONTROL SYSTEM WHILE TESTING!**

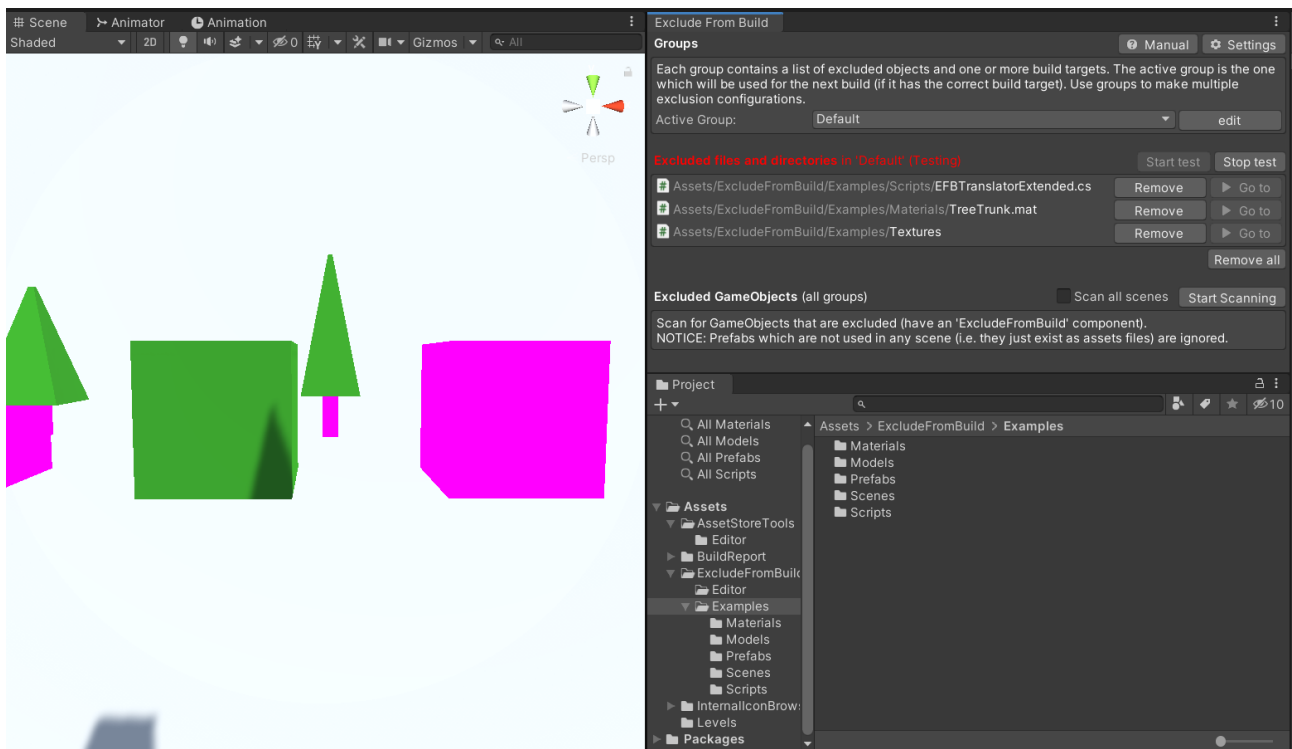
If you have any SCRIPTS excluded then this will trigger a recompile (may take quite some time).

Before the test:



During the test:

Notice the missing 'Textures' folder and the brown 'TreeTrunk' Material is also gone. The tool will show a red warning while testing (so you don't forget to turn it off).



NOTICE: The exclusions on components or game objects within scenes are NOT affected by the test. You can control it for each 'Exclude From Build' component via the 'Test In Play Mode' setting. Testing components outside of Play Mode is not supported at the moment.

Again: Don't forget to stop the test before continuing your work.

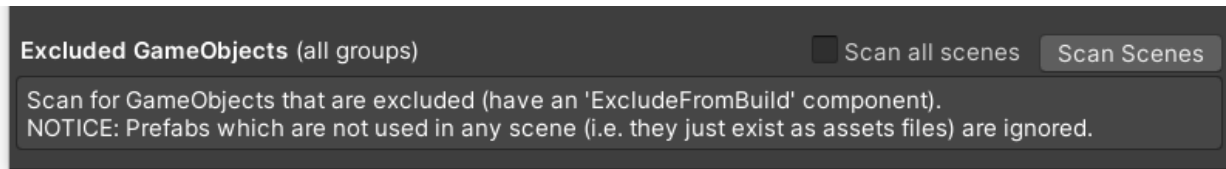
The tool checks at every start if there still is a test running and warns you (asks you to stop it).

If you exclude script files then getting compile errors while testing is to be expected. Fix them, otherwise you will also get these errors while building too.

## Excluded Objects

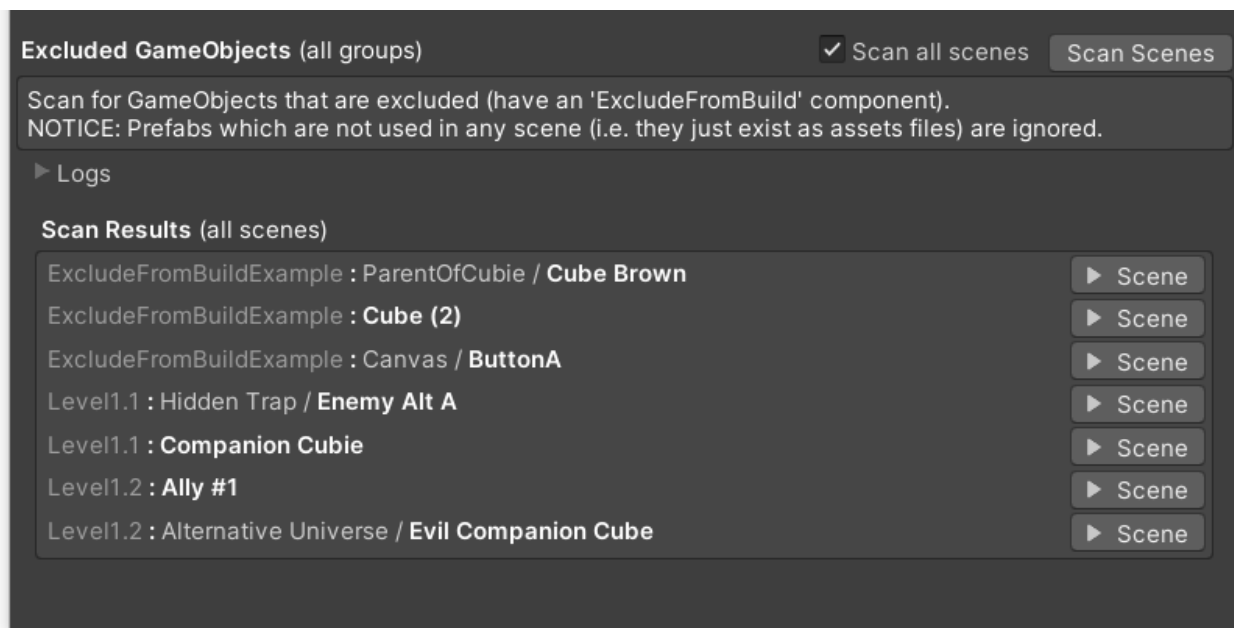
This section allows you to scan scenes for game objects which do have an 'Exclude From Build' component attached.

By default only the currently opened scenes are scanned. To scan all scenes check the **'Scan all scenes'** box and hit the **'Scan Scenes'** button.



Scanning may take a while as every single scene in the project is loaded and every game object is checked.

Once it's done it will display a list of game objects and a path (scene : path in scene).

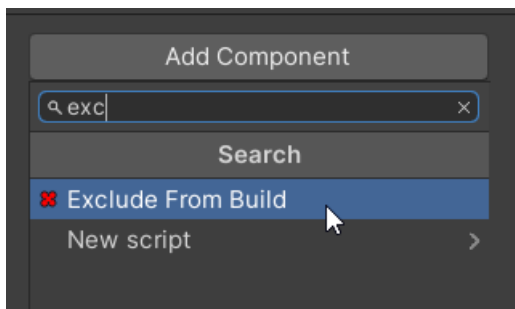


The scan process does not take the 'Groups' setting into account. It lists all found objects.

The scan process does not take unused Prefab assets into account. It scans prefabs within scenes just fine but if you have a Prefab which is not used in any scene then it won't show up in the results (i.e. it only scans scene assets and nothing else).

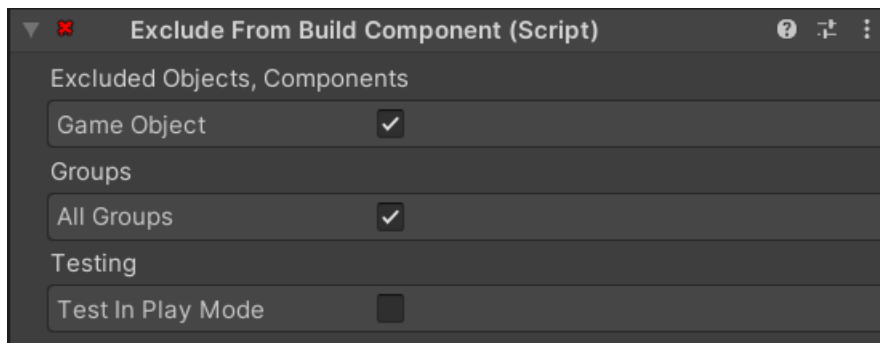
## How to exclude GameObjects and Components

To exclude game objects or even single components you will have to add the **Exclude From Build' Component** to the game object.



You may be concerned about them bloating your scenes. Don't worry, these components are editor only. They will all be removed along with all the other excluded files and directories.

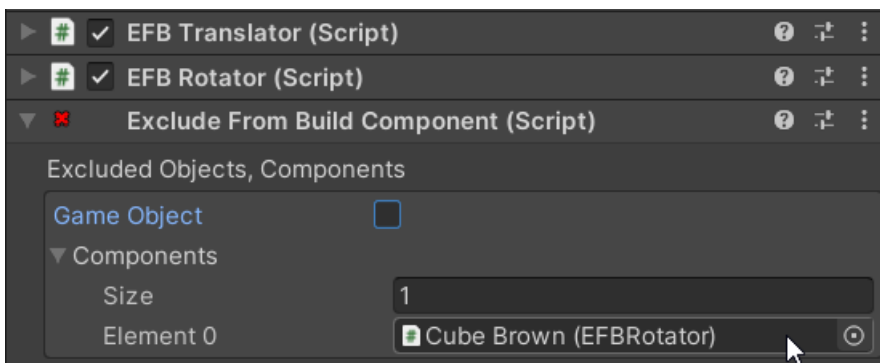
The component has three sections to control the behaviour (A, B, C)



### A) Excluded Objects, Components:

By default it is set to exclude the whole game object (,Game Object' checked).

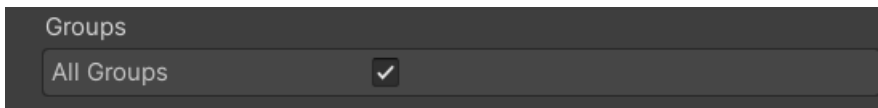
If you want to only exclude one or more components then uncheck ,Game Object' and drag the components into the ,Components' list.



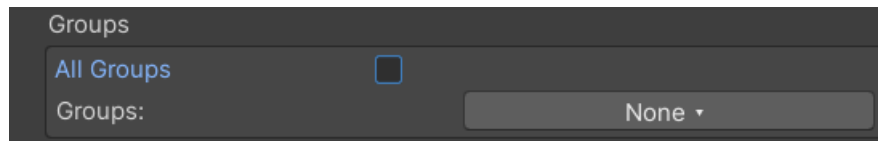
## B) Groups

If you are not yet familiar with groups then please go read Section 1 (Groups) first.

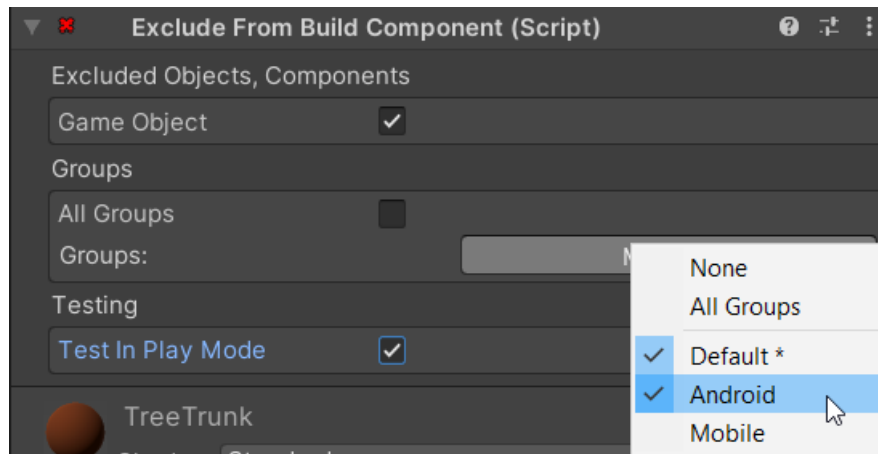
By default the exclusion will happen for every group („All Groups“ is checked).



If you want to bind this exclusion to a specific group then you can disable „All Groups“ and choose one or more groups. By default it is set to „None“, which means it would never be executed.



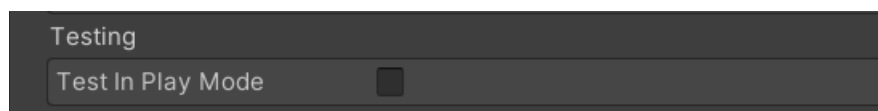
To give you a little help the currently active group is marked with a star ,\*' at the end.



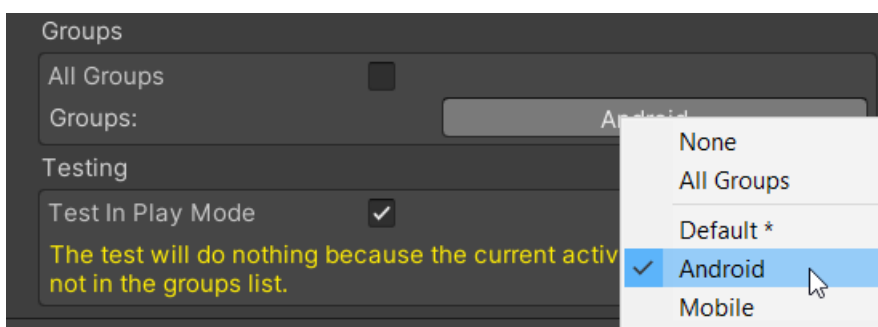
Remember: it's not a dropdown but a multi-select list.

## C) Testing

Testing is only supported in Play Mode. If you enable it and hit ,Play' then the exclusion logic will be executed in OnAwake().

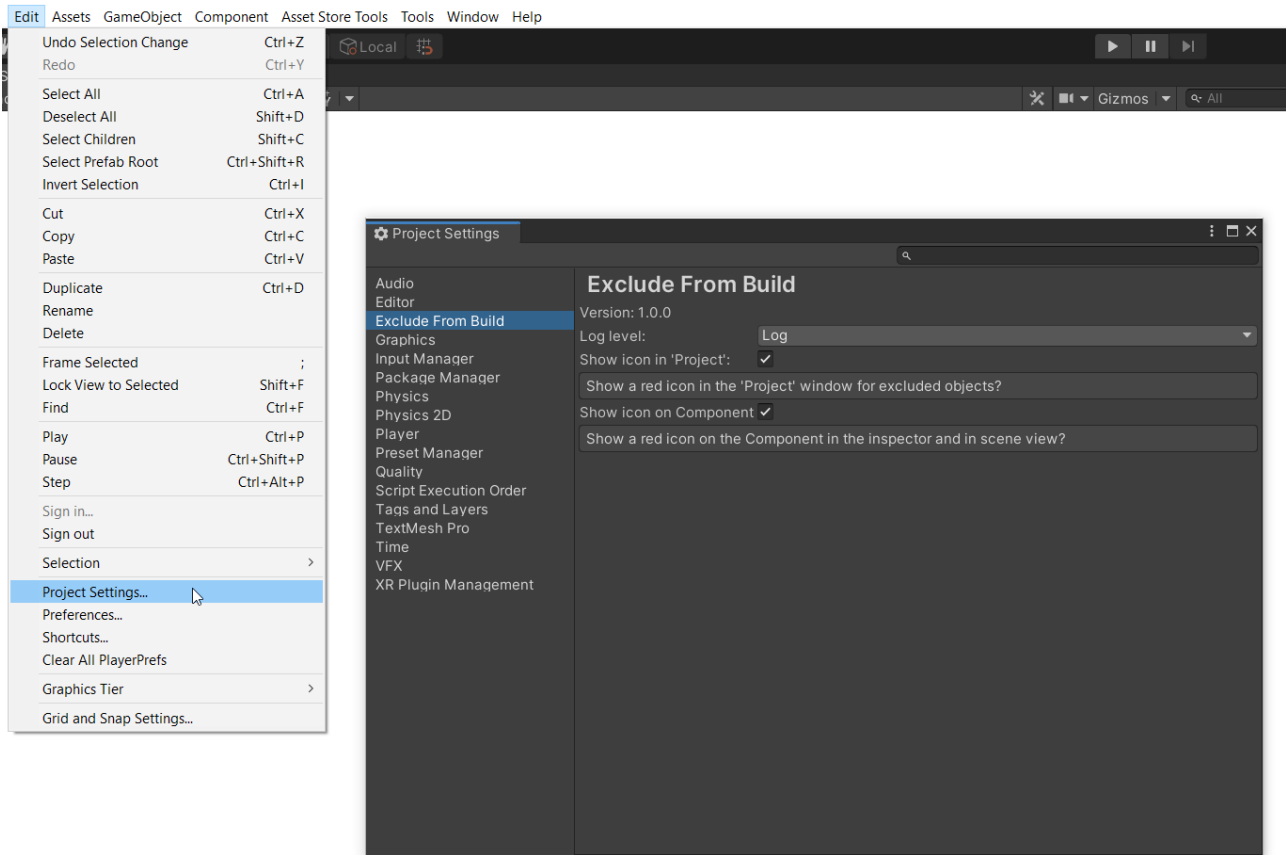


If the active group is not included in the selected groups then a warning will be displayed.



# Settings

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





# Streaming Assets & Build Reports

If you are excluding files which are in the /StreamingAssets directory then you may have noticed that they are still listed in some Build Report tools. Don't be alarmed, the **StreamingAssets are in fact NOT in your build**. Just test it yourself. Here is how:

- 1) Make a build and write down the total final size (in MBytes)
- 2) Copy a very big file (100+ MBytes) into Assets/StreamingAssets.
- 3) Make a new build and check the final size. It should have increased by a lot.
- 4) Now add the StreamingAssets folder (or the big file) to the excluded files list.
- 5) Make another build and check the final size. It should be small again because the big file in StreamingAssets is no longer included.

Below you can see a screenshot of a [build report tool](#) which I use a lot (it's great). As you can see it lists some files from Assets/StreamingAssets.

That's not a bug in the tool but a consequence of how the ExcludeFromBuild works. It hides the StreamingAssets during the build and restores them immediately afterwards. The build report however simply assumes that the files under Assets/StreamingAssets are always included in the build. Usually that's a fine assumption but with ExcludeFromBuild it's actually not true.

The screenshot shows the 'Build Report' window of the 'Build Report Tool v3.9.3'. The interface has a dark theme and includes a toolbar with 'Get Log', 'Open', 'Save', 'Options', and 'Help & Info' buttons. The main content area is divided into tabs: 'Overview' (selected), 'Project Settings', 'Size Stats', 'Used Assets', and 'Unused Assets'. The 'Overview' tab displays the following information:

- Report generation took:** 00:00:01.3211835
- Total Build Size:** 54,73 MB
- Streaming Assets** are the largest, taking up **59.98%** of the build

Below this, there are two sections for the 'Top 10 largest in build':

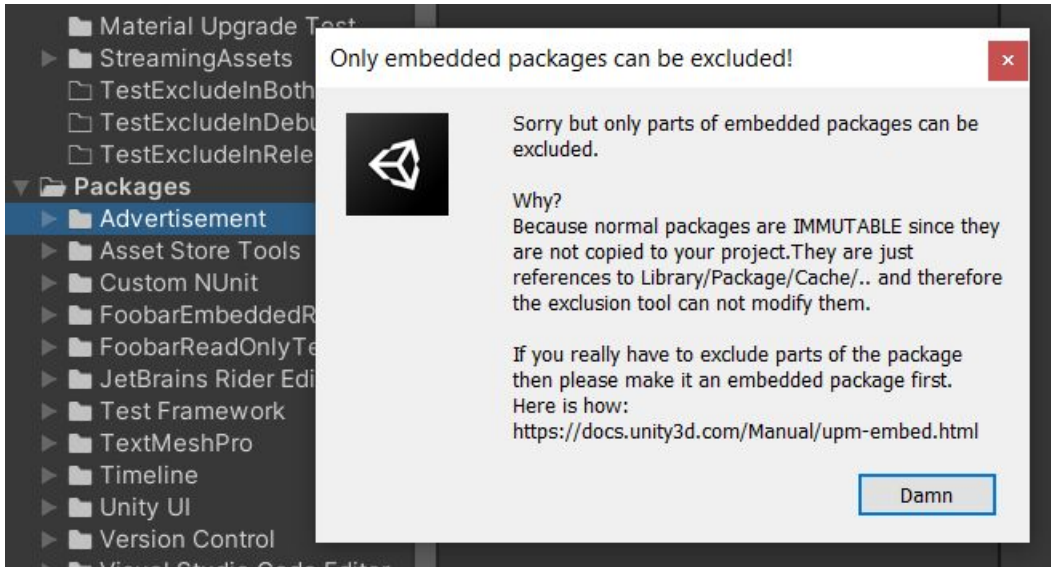
Top 10 largest in build:	
Refresh	
1. StreamedSky 1.png	1,95 MB
2. StreamedSky 10.png	1,95 MB
3. StreamedSky 2.png	1,95 MB
4. StreamedSky 3.png	1,95 MB
5. StreamedSky 4.png	1,95 MB
6. StreamedSky 5.png	1,95 MB

And two sections for the 'Top 10 largest not in build':

Top 10 largest not in build:	
Refresh	
1. ExcludeFromBuildManual.pdf	762,01 KB
2. BuildReportWindowDark.guiskin	204,33 KB
3. BuildReportWindow.guiskin	203,86 KB
4. Sky.png	169,65 KB
5. Sky.png	169,65 KB
6. AssetStoreTools.dll	136,5 KB

# Packages & Embedded Packages

If you try to exclude a package then you may stumble across this dialog:



Why is this?

Well, normal packages are supposed to be IMMUTABLE. You can not change them. They are not copied to your project. They are references to Library/Package/Cache/.. . The exclusion tool honors this Unity convention and thus does not allow you to change the content of referenced packages.

However, if you really have to exclude parts of the package then the workflow is the same as if you really need to modify the package: Make it an embedded package.

Here is how: <https://docs.unity3d.com/Manual/upm-embed.html>

Once your package is embedded you can exclude any folders or files within that package.

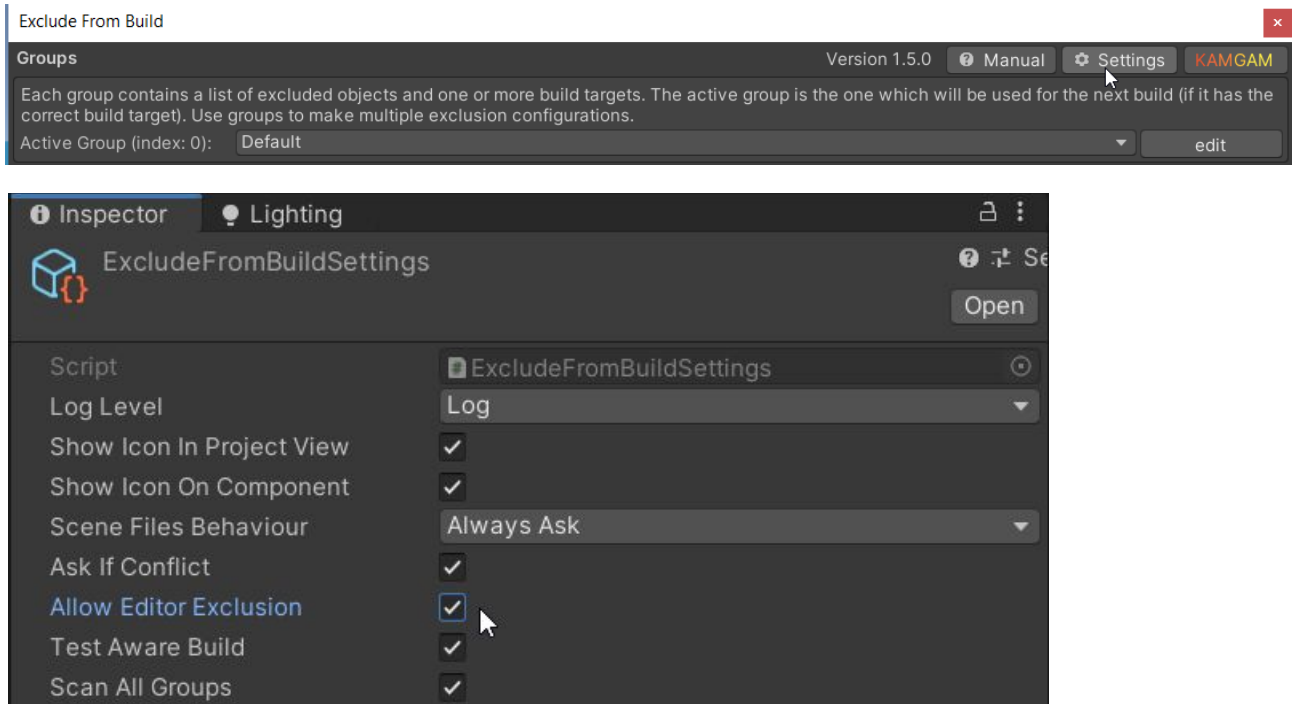
**NOTICE:** You can not exclude a packages' root folder, ever. The reason is that the Unity process locks those folders and thus, no-one can modify them as long as Unity is running.

# Excluding „Editor“ folders

Usually the exclusion tool will not allow you to exclude „Editor“ directories since it is not necessary (those are not included in the build anyways).

However, if a script in an „Editor“ folder contains a hook for the pre-build process then it can sometimes be necessary to exclude the „Editor“ folder too to avoid the creation of such hooks.

To do that you first have to enable editor folder exclusion in the settings:



Once that option is enabled you can exclude „Editor“ folders too.

I recommend using this in combination with the „test-aware“ build workflow (exclude first, wait for recompile, then start the build).