

GUID Fixer - Manual

GUID Fixer



5f7201a12d95ffc...



a634b21d6de411...

You have conflicting GUIDs and need some new ones?
Some error has given you a GUID but you don't know what that belongs to?
You need a quick way to do some full text search on your Asset files?

Don't worry! GUID Fixer is here to help you out.

Table of contents

1. Prerequisites.....	2
2. Generating new GUIDs.....	3
3. Replacing an existing GUID.....	5
4. How to get a GUID.....	6
5. Copying missing GUIDs (Fixing Missing Scripts).....	7
6. Searching for GUIDs.....	9
7. Tricking GUID Fixer into acting as a full-text search engine.....	11
8. Swapping Scripts via GUID Change.....	12

1. Prerequisites

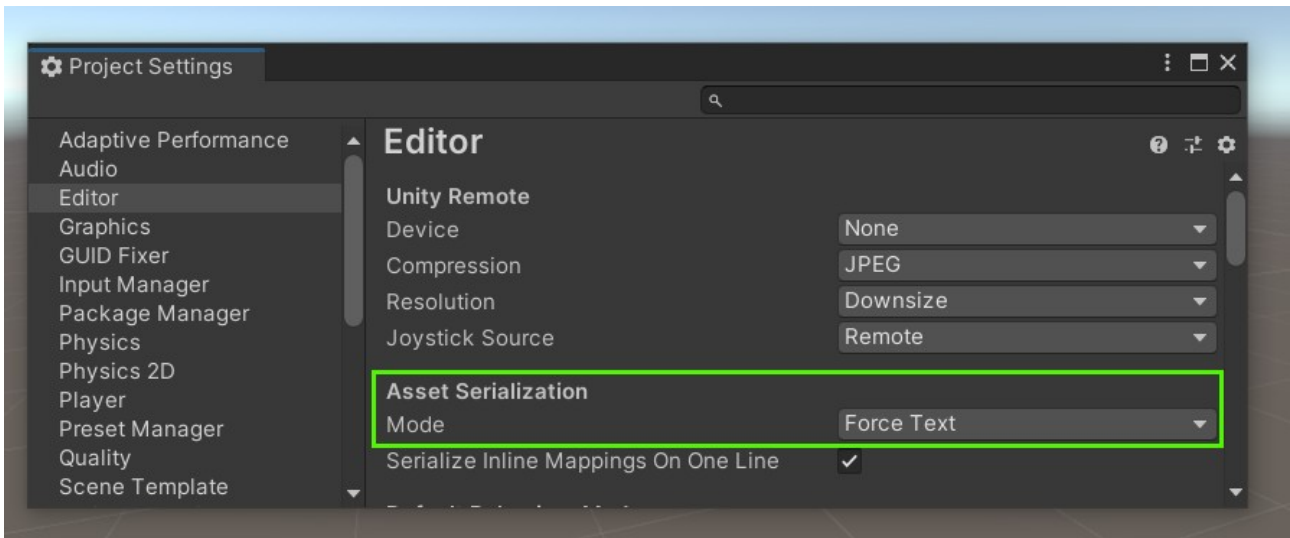
Please check that your Asset Serialization is set to „Force Text“.

To enable editing GUIDs your scene files have to be saved in a text format.

Unity is already configured to use a text format by default. Therefore no changes are required.

If you want to be extra sure then please check the settings under:

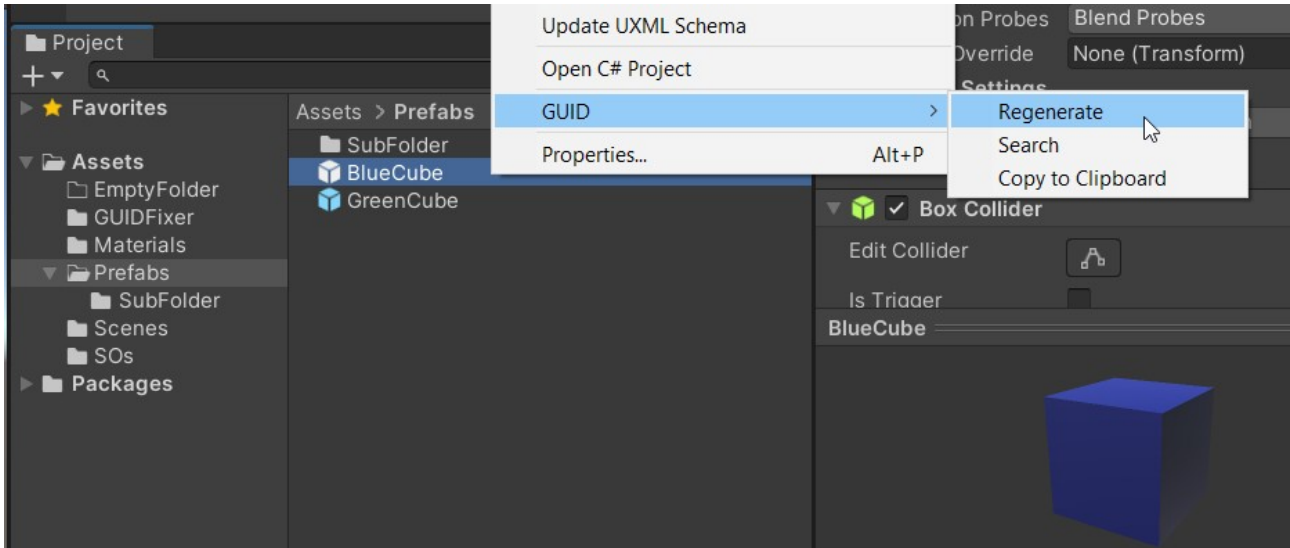
Project Settings > Asset Serialization Mode > Force Text



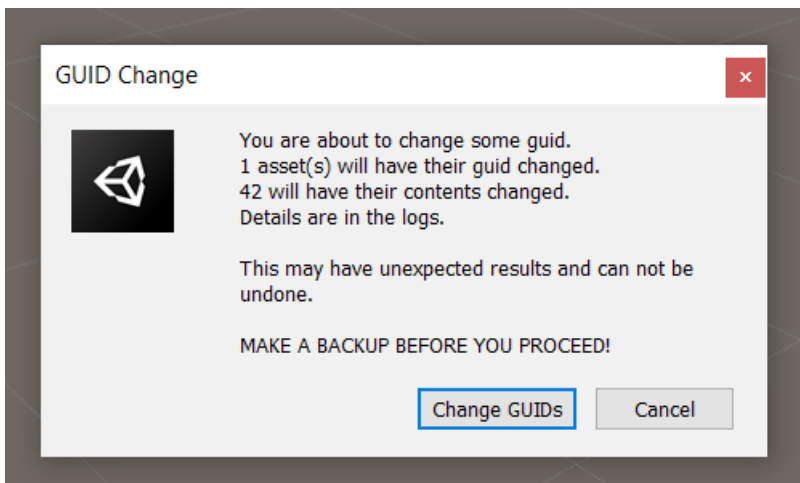
2. Generating new GUIDs

To generate new GUIDs simply **Right-Click > GUID > Regenerate**.

You can do this on any folders or files inside Assets. In this example we will only regenerate the GUID for one single Prefab (BlueCube):



A warning will appear as you are about to do some major changes to your project. Be sure to **COMMIT your changes before doing this** as Unity can not revert it.



After the change you will notice that the GUIDs in some files have changed.

Changes made (double-click on file for diff):

Check: **All** None Unversioned **Versioned** Added Deleted **Modified Files** Submodules

Path	Extension	Status	Lines...	Lines ...
<input checked="" type="checkbox"/> GUIDFixer-2020-dev/Assets/Prefabs/BlueCube.prefab.meta	.meta	Modified	1	1
<input checked="" type="checkbox"/> GUIDFixer-2020-dev/Assets/Scenes/Level1.unity	.unity	Modified	13	13
<input checked="" type="checkbox"/> GUIDFixer-2020-dev/Assets/Scenes/Level2.unity	.unity	Modified	13	13
<input checked="" type="checkbox"/> GUIDFixer-2020-dev/Assets/Scenes/Level3.unity	.unity	Modified	13	13

Here are the changes made to the .meta file of BlueCube.prefab:

```
1 fileFormatVersion: 2
- guid: a634b21d6de41194295c0430ddd52d47
+ 2 guid: 12b952c241ab4d70aea28cf94d7441f4
3 PrefabImporter:
4   externalObjects: {}
5   userData:
6   assetBundleName:
7   assetBundleVariant:
```

And these are the changes made to each scene which contained that prefab:

```
416 --- !u!1001 &792251577649260702
417 PrefabInstance:
418   m_ObjectHideFlags: 0
419   serializedVersion: 2
420   m_Modification:
421     m_TransformParent: {fileID: 0}
422     m_Modifications:
-     - target: {fileID: 7922515776382865859, guid: a634b21d6de41194295c0430ddd52d47, type: 3}
+ 423 - target: {fileID: 7922515776382865859, guid: 12b952c241ab4d70aea28cf94d7441f4, type: 3}
424     propertyPath: m_RootOrder
425     value: 2
```

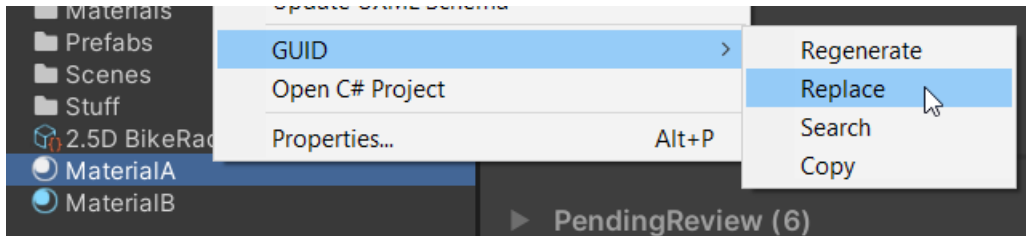
If you wish to regenerate the GUIDs of the whole project simply right click on the „Assets“ folder and choose **GUID > Regenerate**.

NOTICE 1: Only assets which are located under /Assets can be changed. Changing GUIDs in packages is not supported.

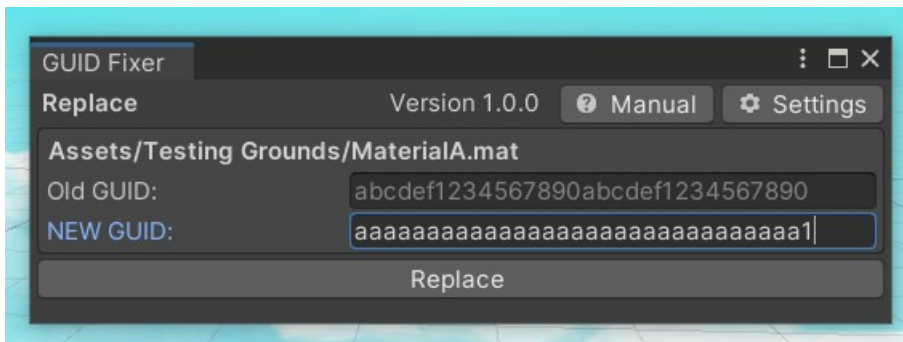
NOTICE 2: Be cautious with changing GUIDs of third party assets as you will most likely get duplicate files after an update. Unity considers identical files with different GUIDs as two different files.

3. Replacing an existing GUID

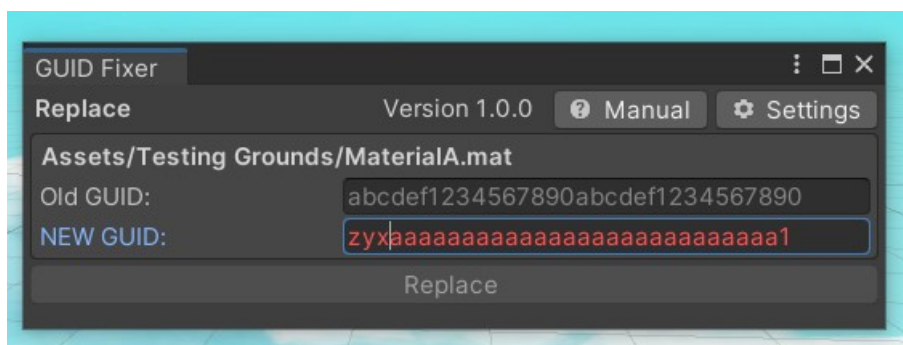
To replace a certain GUID with a custom GUID use **Right-Click > GUID > Replace**



Enter the new GUID.



Keep in mind that a valid GUID is a hexadecimal number with a length of 32 characters. The tool will help you to get it right by validating the GUID while you type it in.

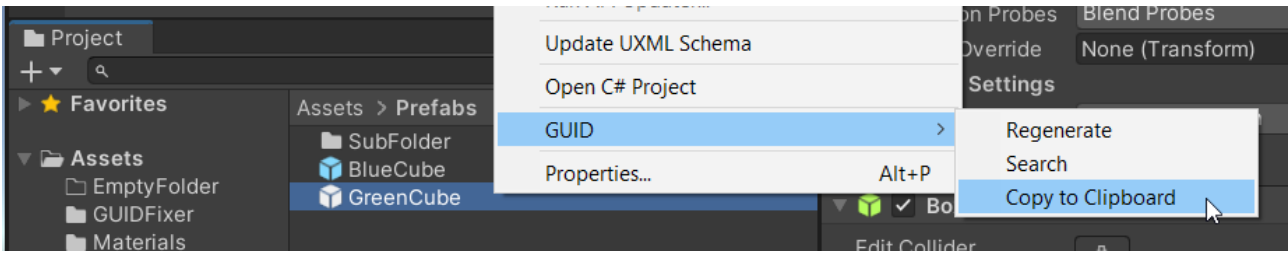


Finally click „Replace“ and your asset will have a new GUID.

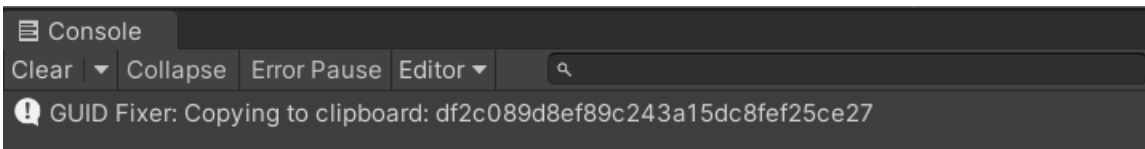
HINT: While the „Old GUID“ field is greyed out you CAN edit it, so you can use this as a GUID replacement form for any GUID, not just the selected file.

4. How to get a GUID

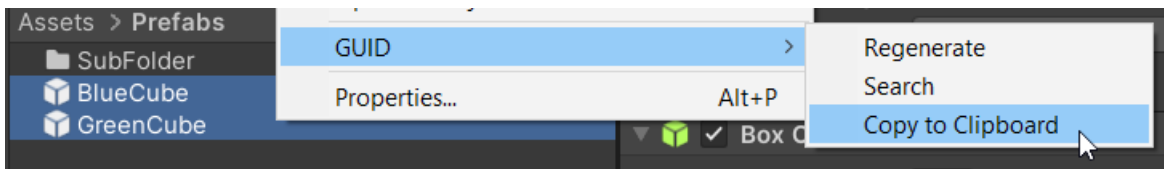
To get the GUID of an object use **Right-Click > GUID > Copy To Clipboard**.



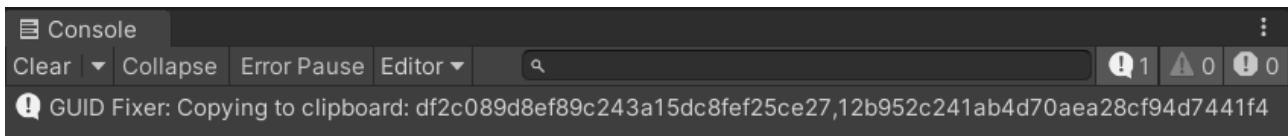
It will also be logged in the console:



You can do this with multiple objects too.

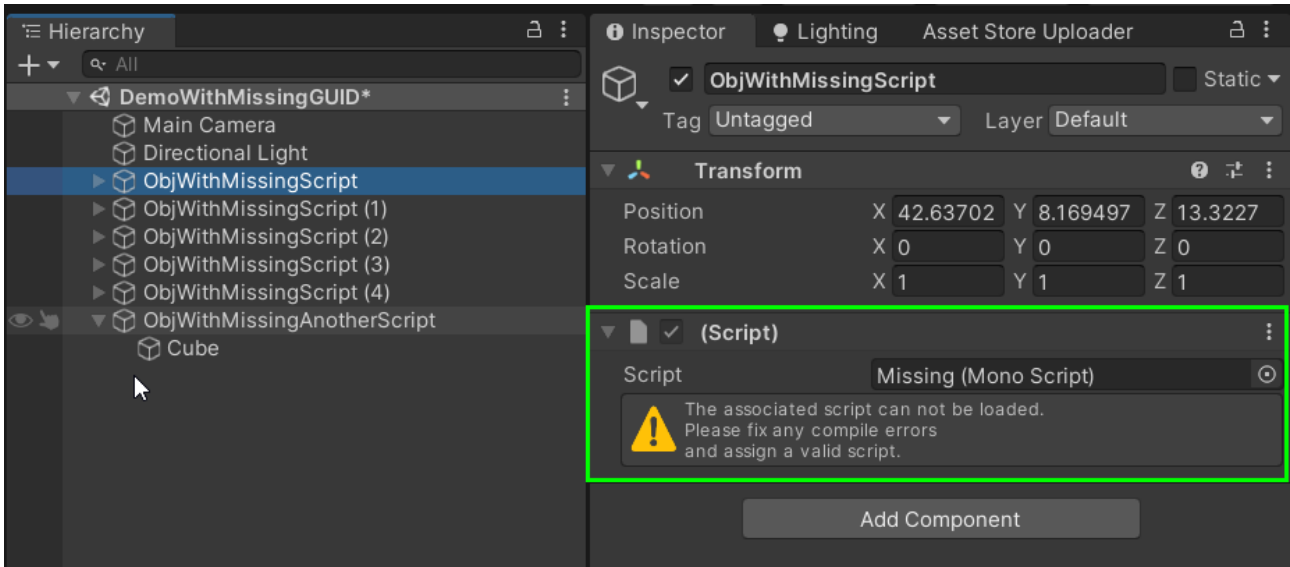


Their GUIDs will be comma separated.

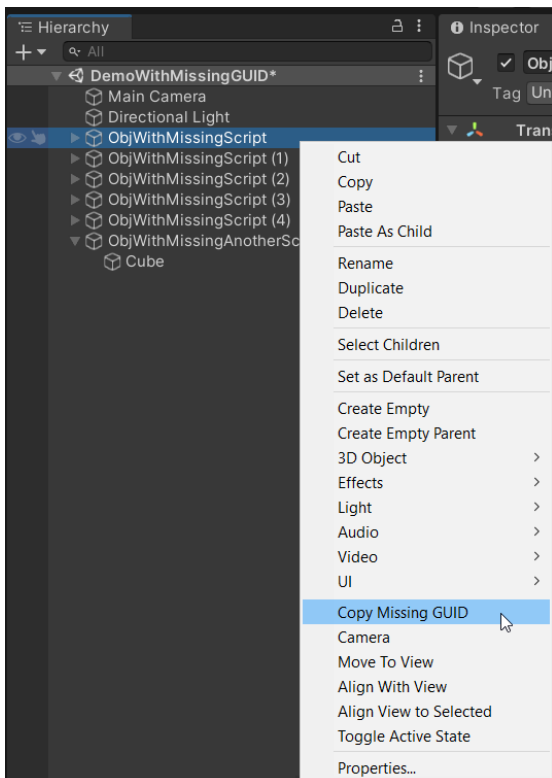


5. Copying missing GUIDs (Fixing Missing Scripts)

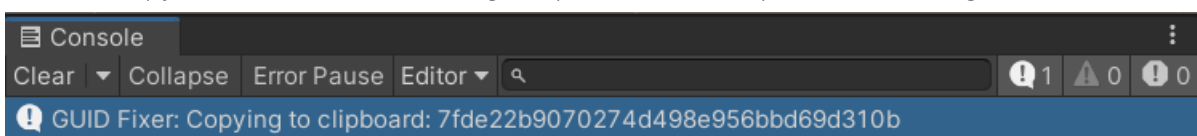
Imagine you have a script attached to many MANY game objects across multiple scenes and you want to fix it.



If you **Right-click** on the GameObject you can choose „Copy Missing GUID“.

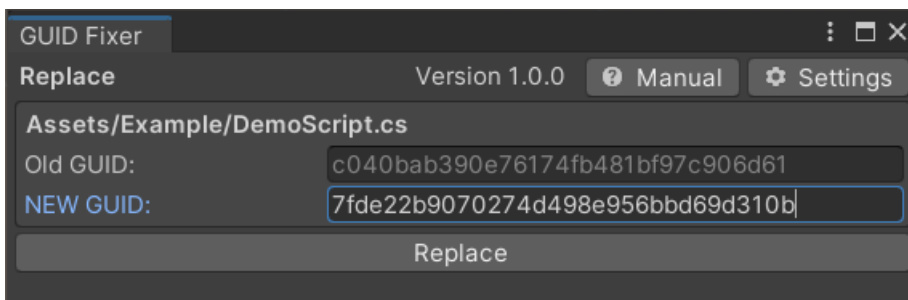
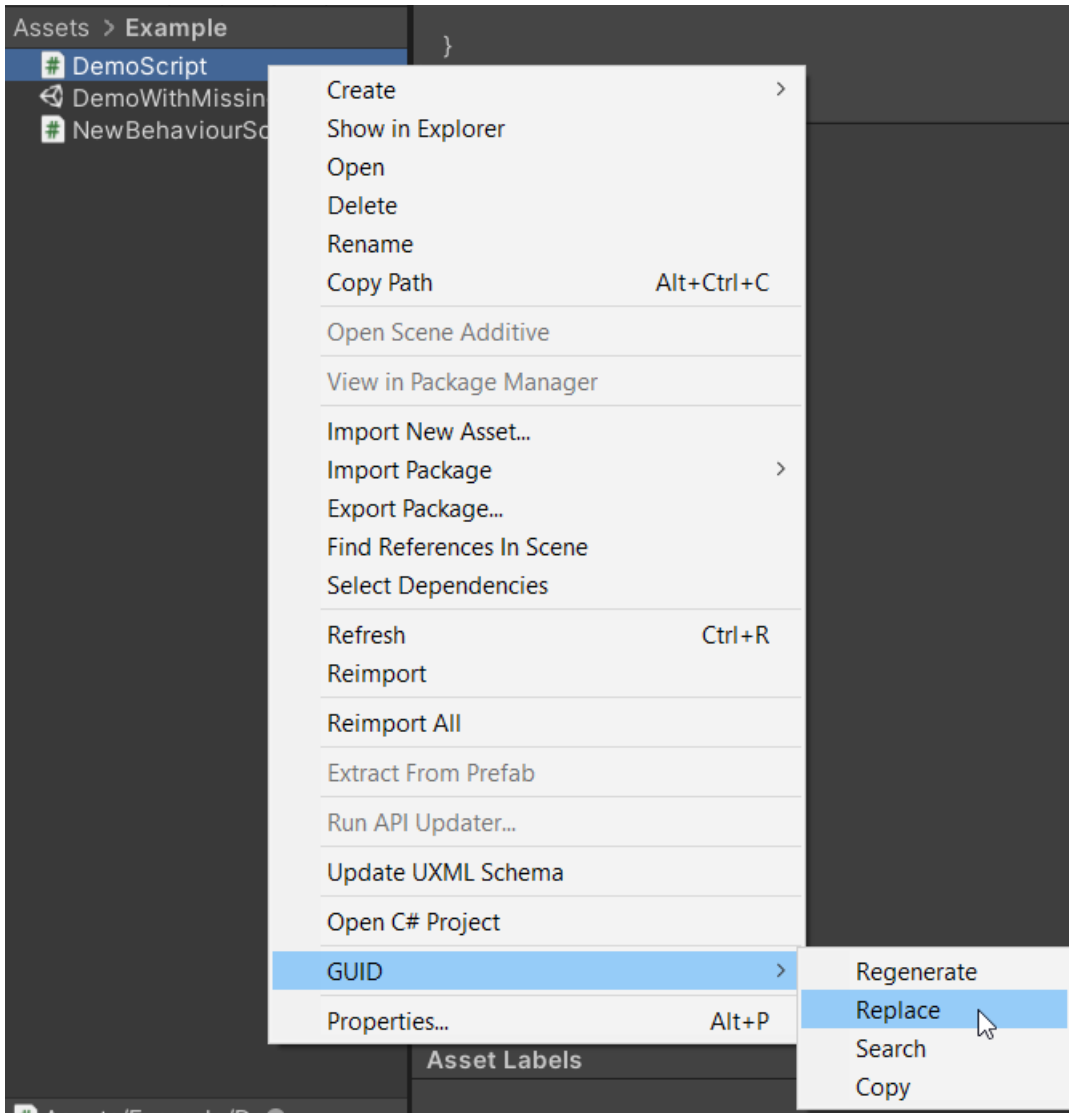


This will copy the GUID of the missing script(s) into the clip-board and log them on the console.



Now you can use the „Replace“ method to change the GUID of any script to match the missing one. This will change the GUID of the existing script and effectively reconnect it.

NOTICE: make sure the script is not referenced by other objects. If they do then these references will need to be patched afterwards. This workflow is best for replacing broken references with a new script which does not yet have any other references.

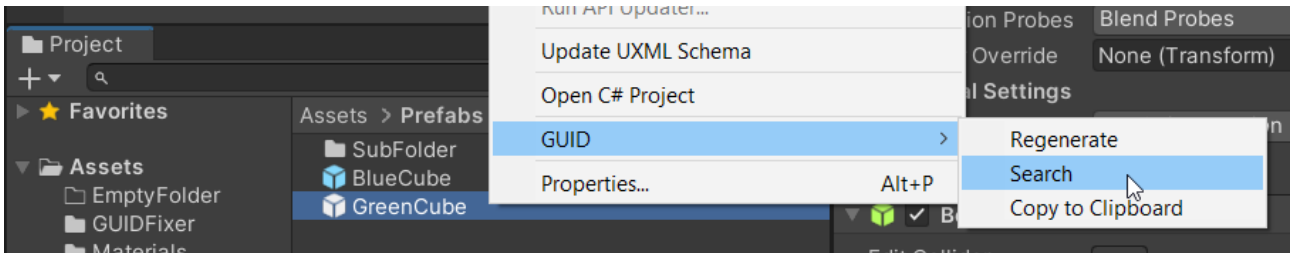


6. Searching for GUIDs

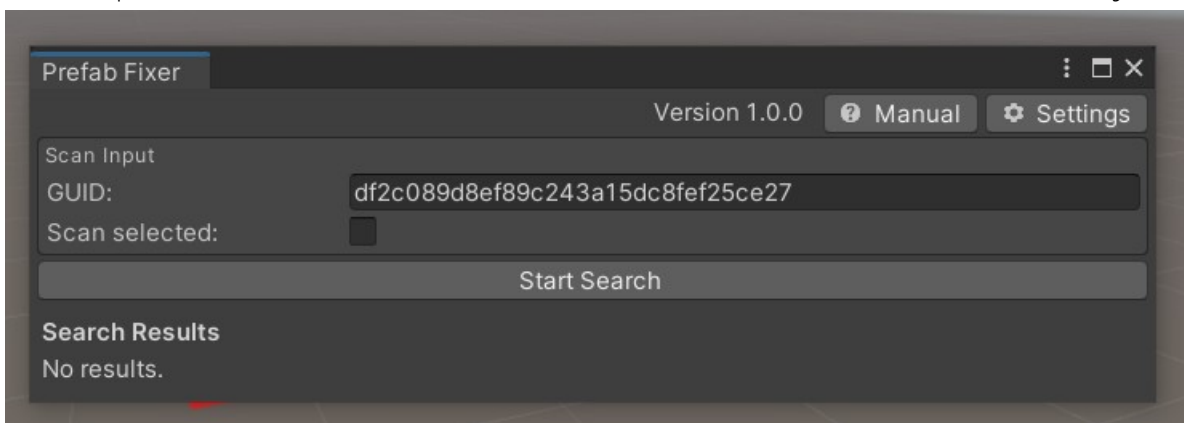
You might be hesitant to change a GUID (as you should be). To get a feeling for what a change would mean you can search for a GUID first. There are two ways to start a search, A and B.

A) Right-Click > GUID > Search

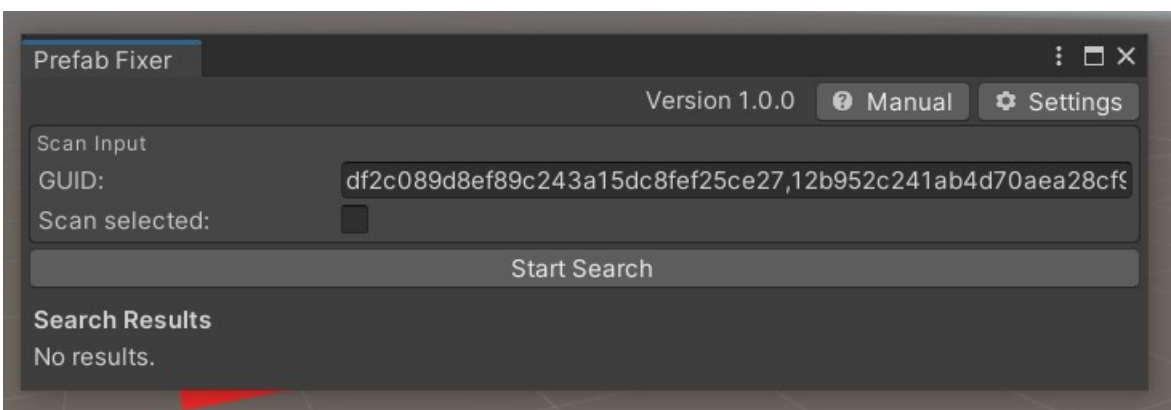
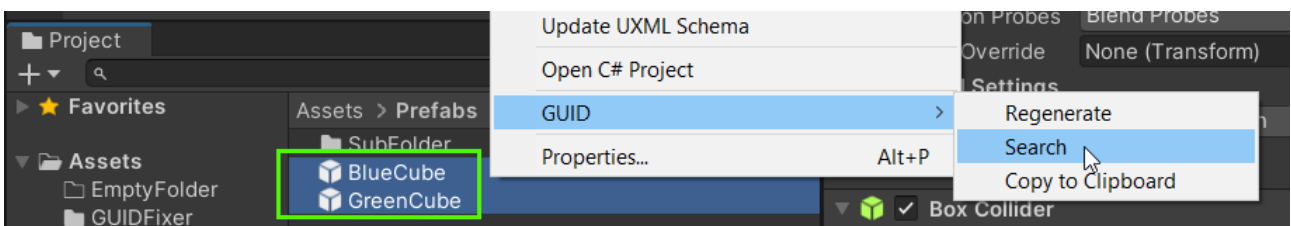
Let's say you want to see where the GUID of the „GreenCube“ Prefab is used.



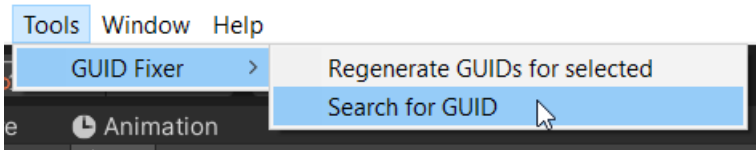
This will prefill the GUID field in the search window with the GUID of the selected object(s).



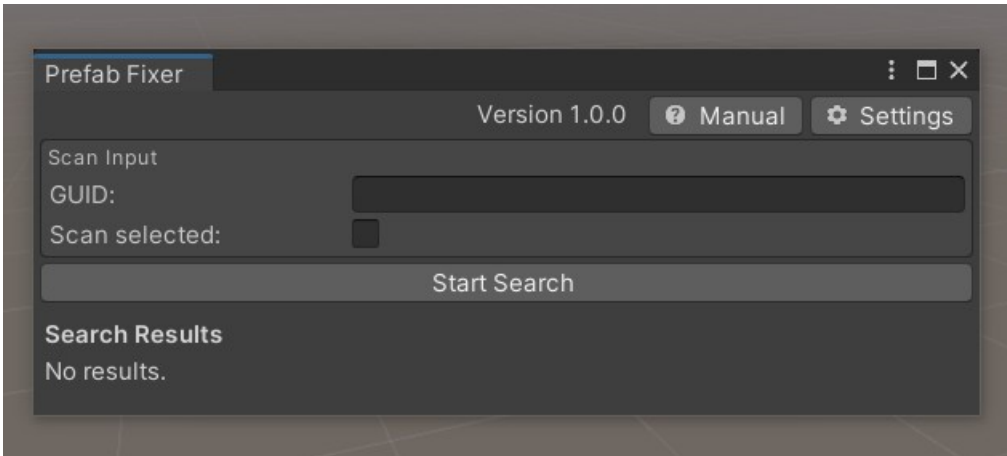
Remember: You can do all this with multiple objects too. The GUIDs will be comma separated.



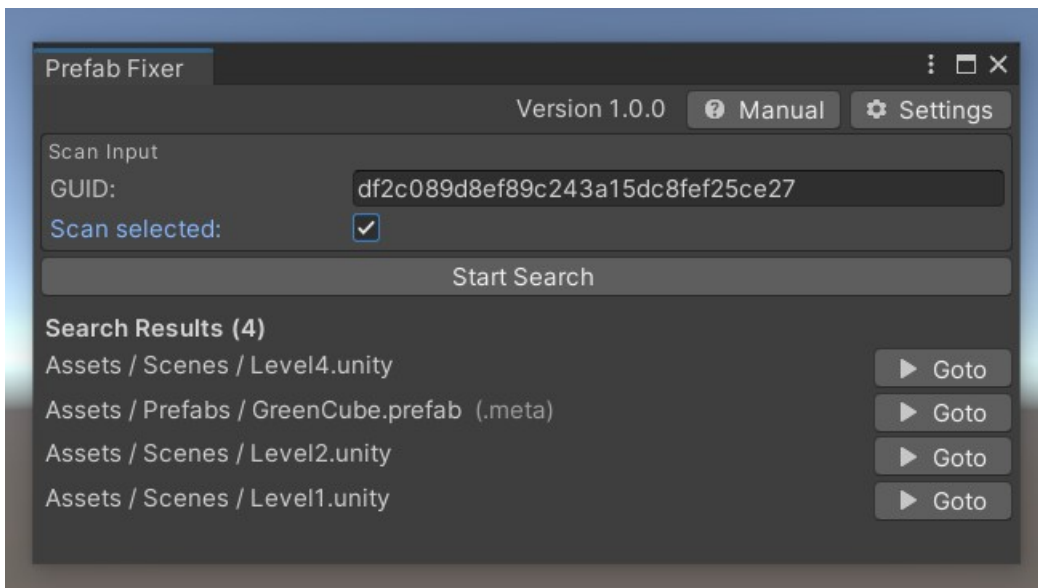
B) Tools > GUID Fixer > Search



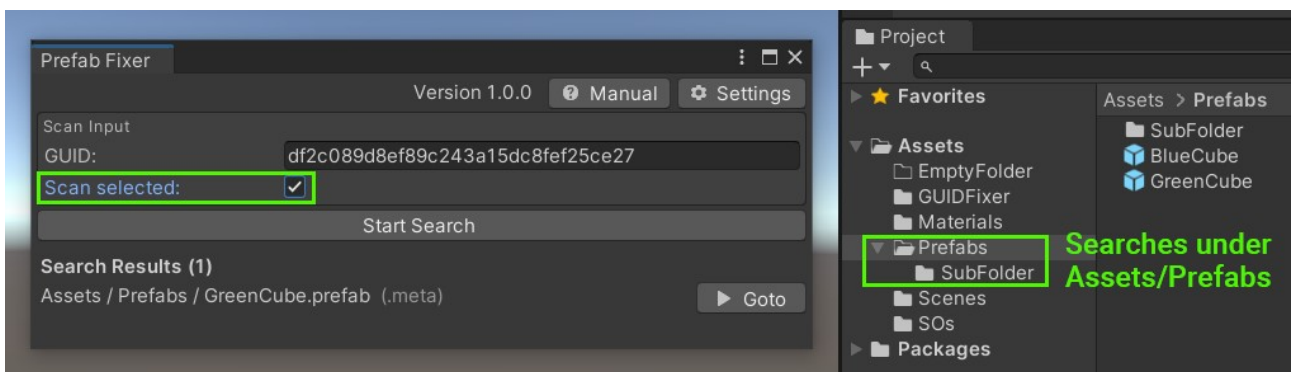
Doing it this way you will have to provide a GUID yourself.



By default the tool will search ALL assets under the „Assets“ directory.



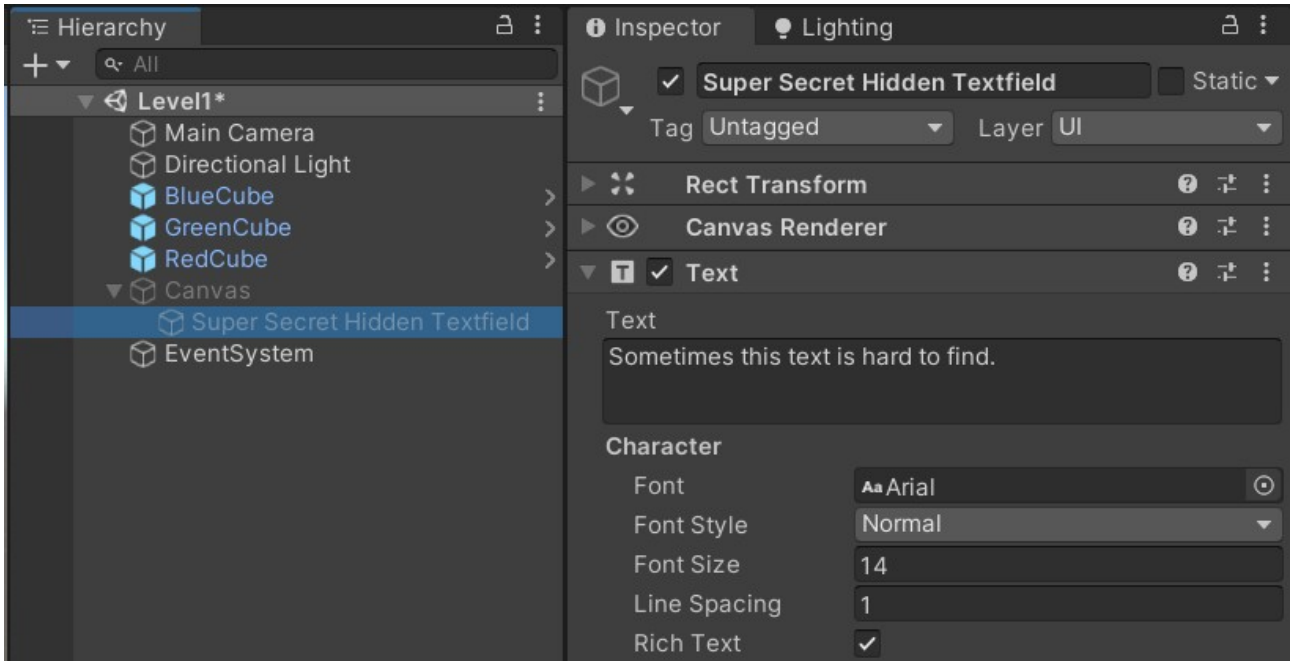
If you check the „Scan selected“ checkbox then only the current selection will be searched.



7. Tricking GUID Fixer into acting as a full-text search engine

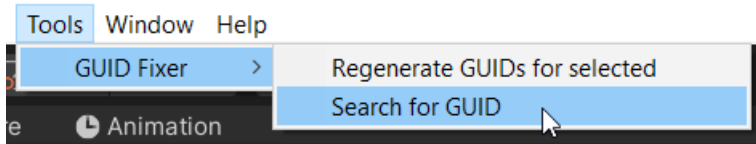
Since the GUID search is really just a full text search on all the text assets in your Assets/ directory you can use it to search for some other stuff too.

Let's assume you have this super secret textfield which you forgot where it is but you still know it has the text „hard to find“ in it.

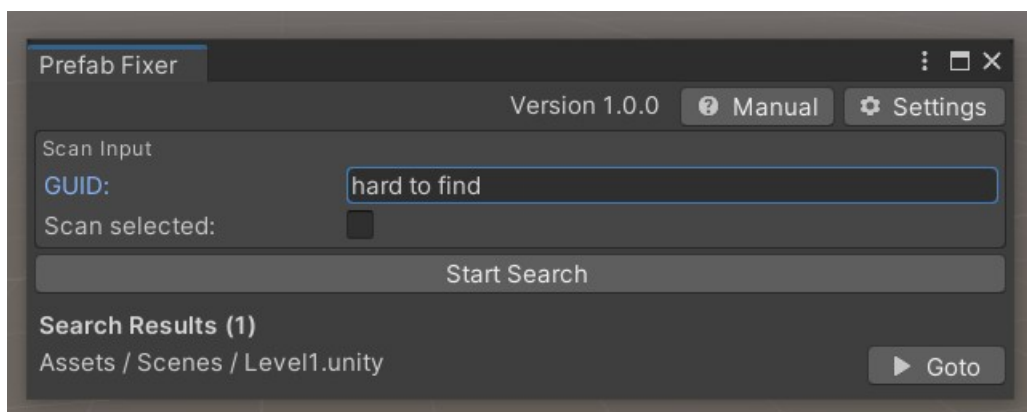


Try this:

First, we start a new search.



Second, we enter the text we are looking for („hard to find“) and Press „Start Search“.

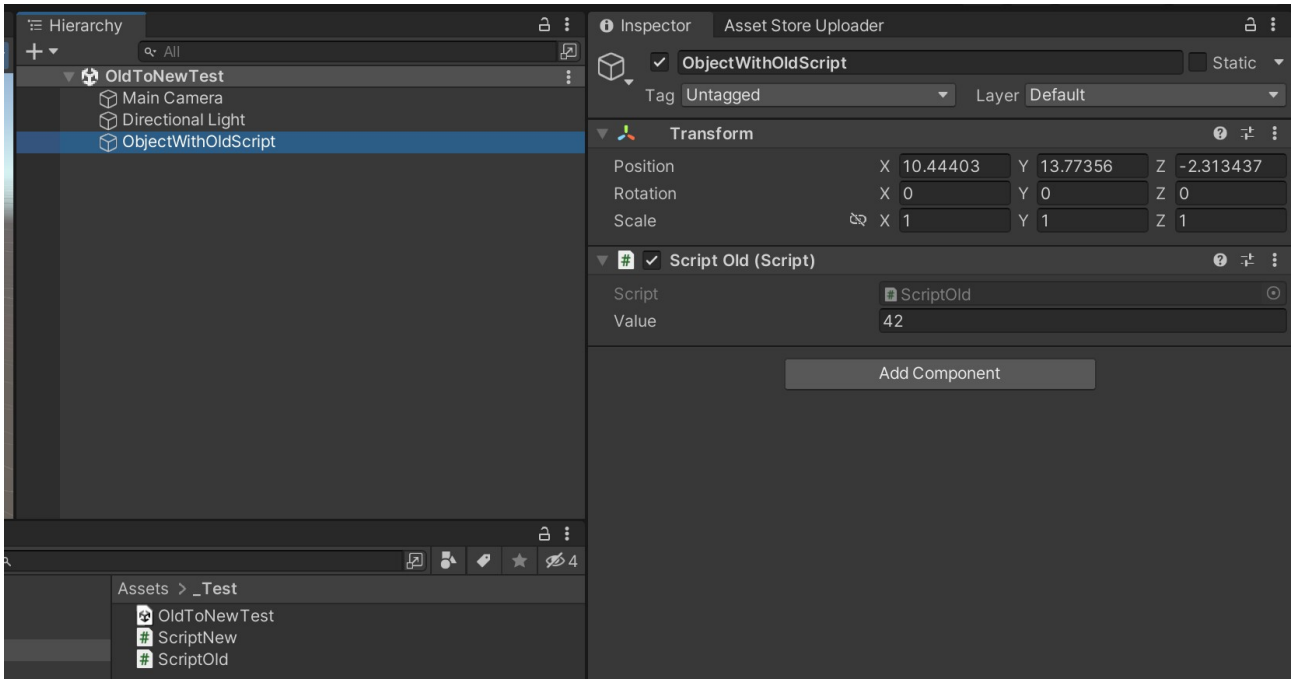


Now at least we know in which scene that textfield is in. ;-)

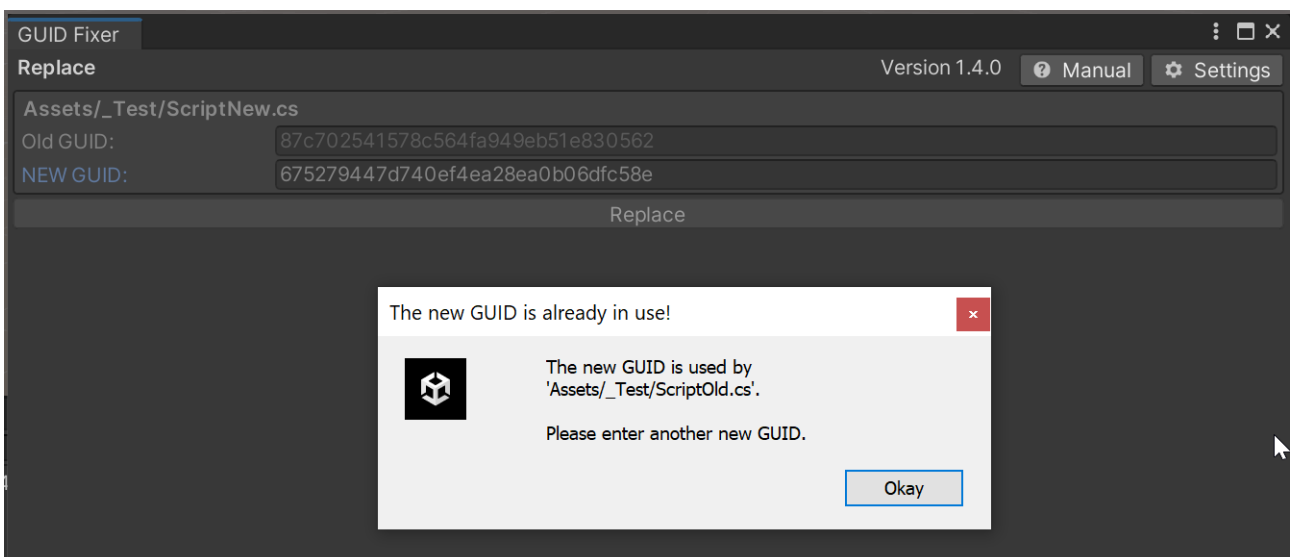
8. Swapping Scripts via GUID Change

One common use-case for GUID change is to swap out one script for another. Sometimes the old script is missing, sometimes it's not missing.

Let's assume this situation: we have a scene and an object with `ScriptOld` and a value of 42:



We also have a `ScriptNew` and what we want to do is swap out `ScriptOld` with `ScriptNew`. To do this we have to copy the GUID of `ScriptOld` to `ScriptNew`, yet if we try that directly the GUID tool will prevent this because the GUID is (still) used by `ScriptOld`.

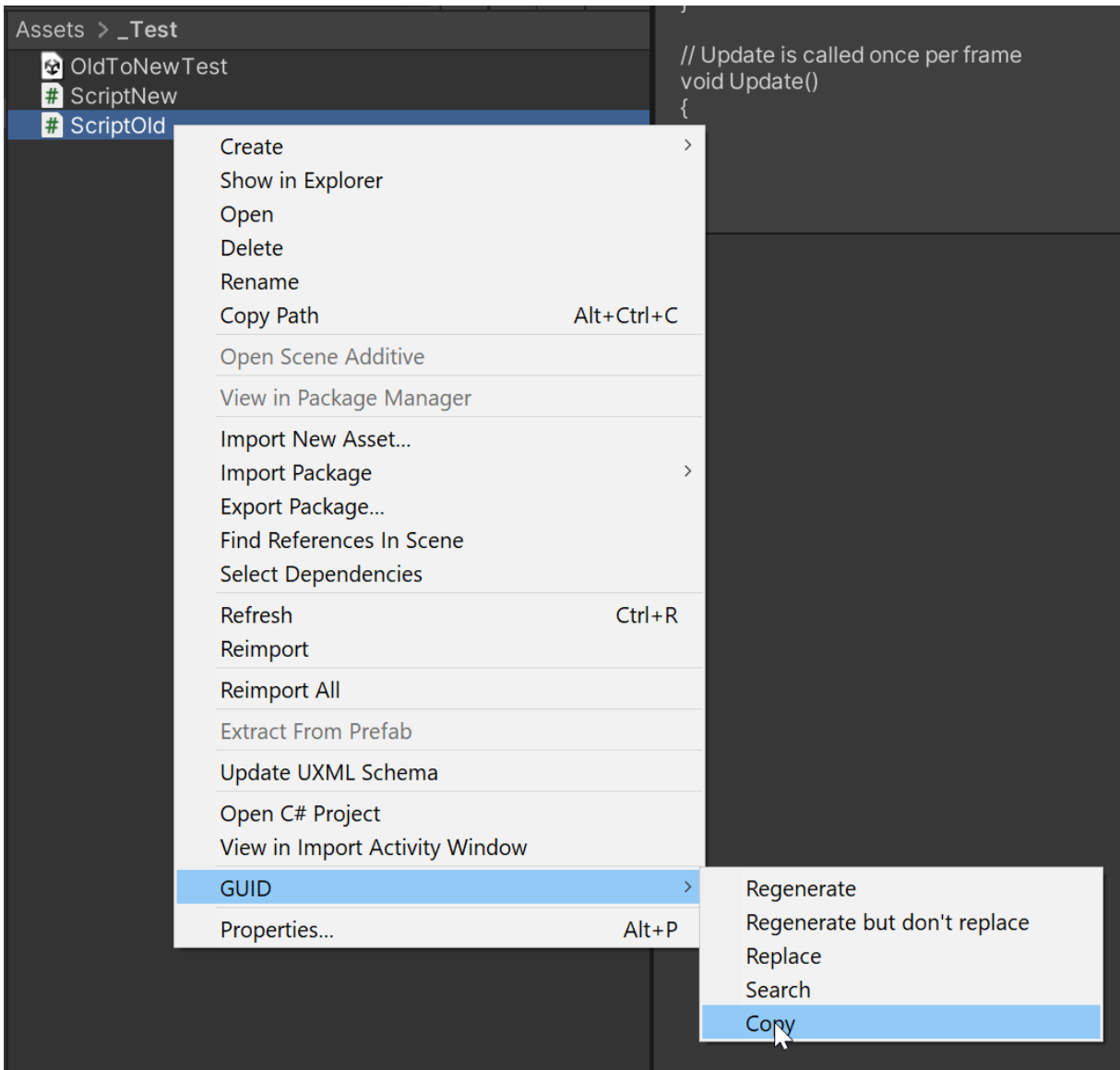


To work around this we have to

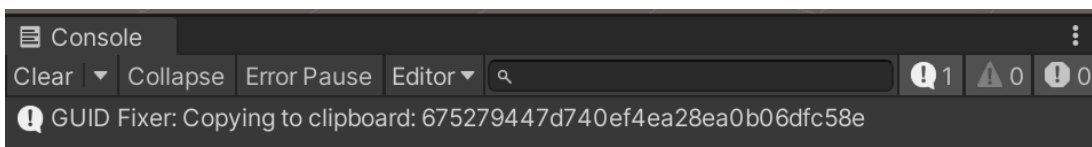
- 1) Copy the current GUID of the old script (and save it somewhere)
- 2) Generate a new GUID for the old script but without updating its references in the scene.

3) Then we can replace the old GUID with the new one since the old one is no longer used (we changed it in step 2).

Step 1: Copy the GUID of the old script.

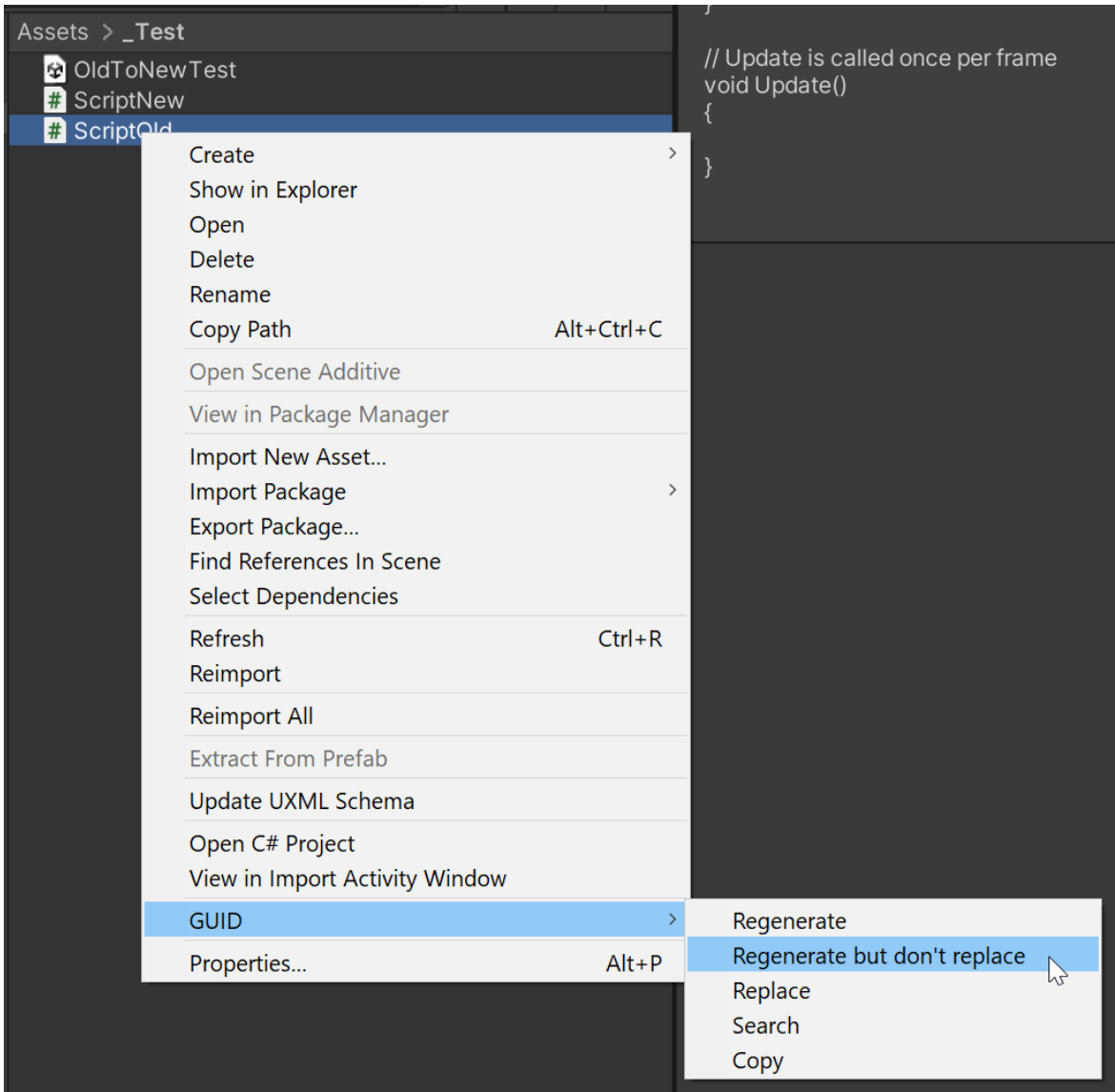


HINT: The copied GUID is also logged in the console. So if you don't clear the console you can always copy it from there.

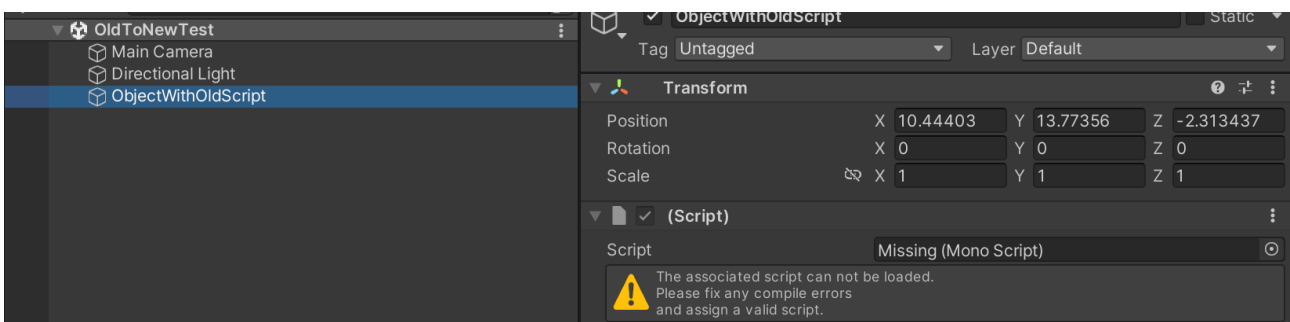


Step 2) Generate a new GUID for the old script without updating it's references in the scene.

Now we generate a new GUID for the old script. IMPORTANT: Do it with the „but don't replace“ method as otherwise the tool would update all the GUID references in all files (we don't want that).



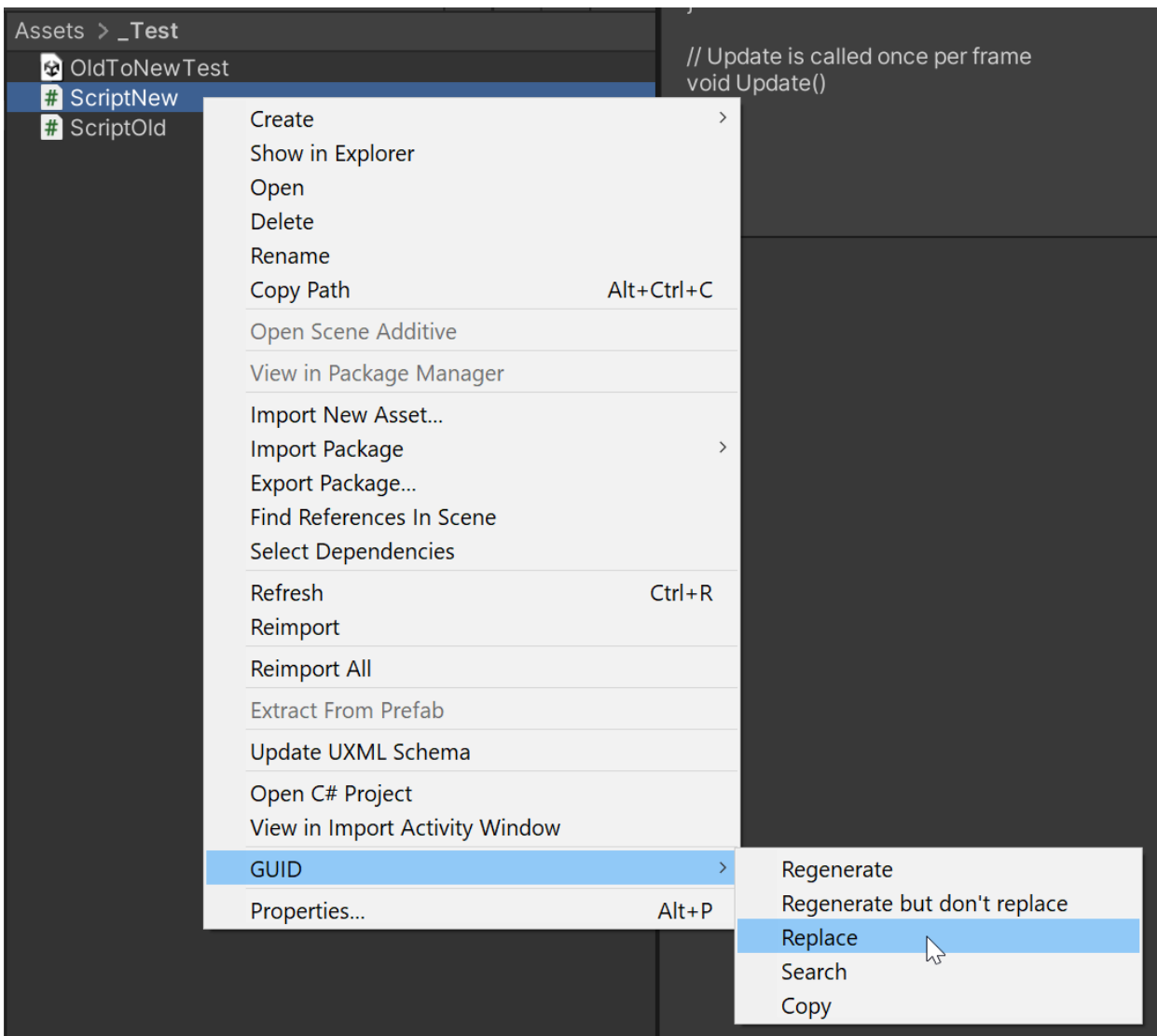
This will BREAK the script references (since we did not replace them). This is exactly what we want (we will make them point to the new script in the next step).



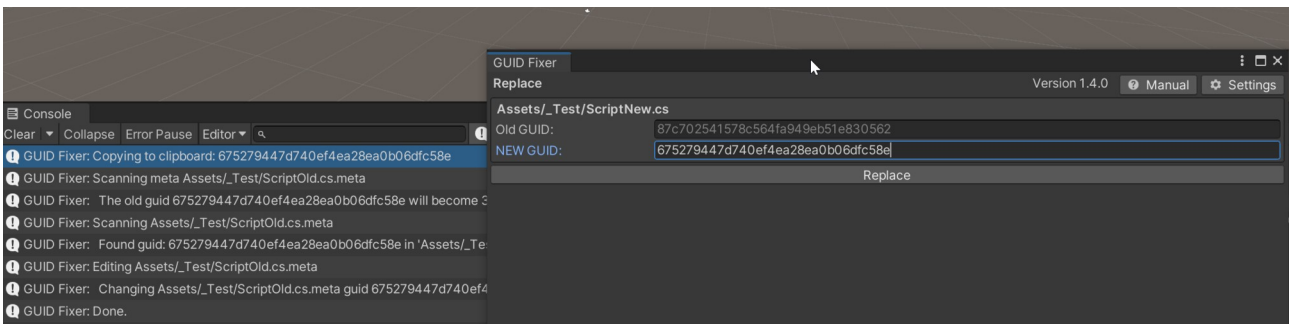
HINT: This is also where the fixing missing scripts workflow would start.

Step 3) Replace the old GUID with the new one

Now all that is left to do is change the „ScriptNew“'s GUID to the one we copied in step 1.



HINT: You can copy the old GUID from the log if it's no longer in your clipboard.



Finally we see that the script on the object is now „Script New“ instead of „Script Old“:

