

So there has been some discussion about changes that we made to the package importer for 5.0.

One of the more highlighted behaviours we changed was that packages with assets that have the same path as those existing in the project, but different GUID, now create a new asset besides the existing one.

People found this annoying as a lot of asset store developers were not releasing a consistent chain of updates from the same project. They were simply creating a new project with their stuff for every update. (This is bad practices, and we will be updating the asset store submission guidelines soon)

To address this and other usability issues with the package importer, 5.3 has had some changes to how packages are imported.

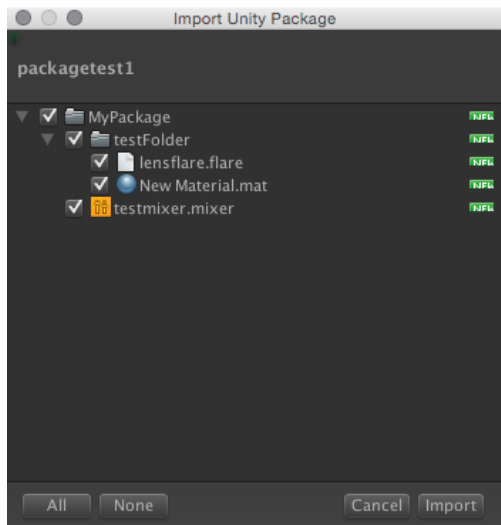
Firstly, the behaviour of overriding described above has been reverted to pre 5.0 behaviour, but will now warn users of conflicting GUID usages.

Furthermore we have added some additional functionality to the importer to try and make other common use cases a little easier.

One of the most important changes is the addition of a Re-Install option if unity can detect the presence of the package already installed. This is great for allowing asset store packages to be updated and not have old files that have been moved or removed in the package stick around in the destination project. This is an intrusive operation however, as it DELETES the package folder before installing the updated version.

The list of use cases are described below (please note that the icon clipping shown in these screen caps is fixed)

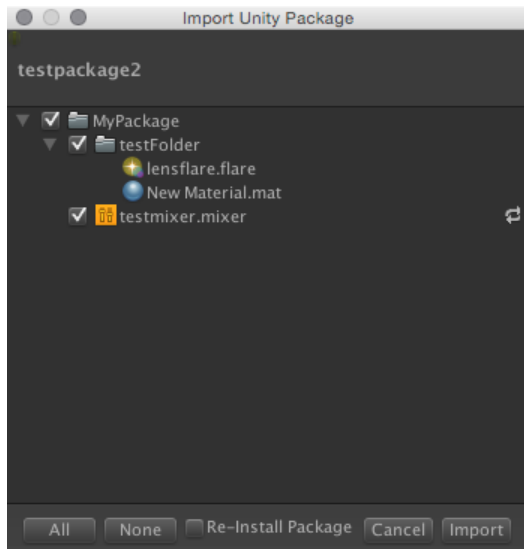
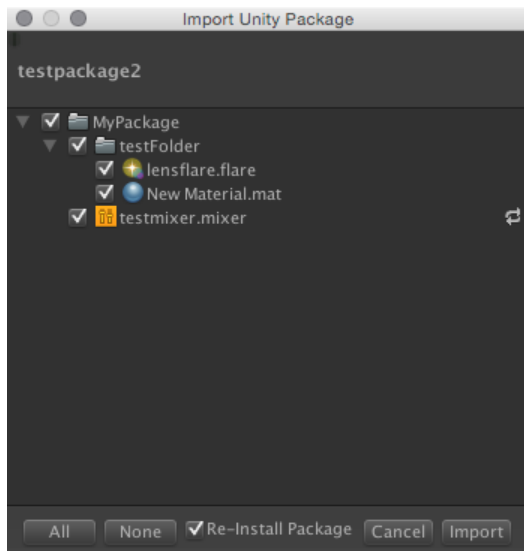
Initial Import



Here we show a basic package with some stuff in it being imported to an empty project

- All folders are expanded and checked by default
- Each asset (including folders, which are assets) are marked as new

Basic Update



testpackage2 has an update to the testmixer, all other items are exactly the same as the previous version

We have two options here:

1. With Re-Install selected
2. Without

RE-INSTALLING IS THE DEFAULT OPTION IF AVAILABLE

- Note that items that are unchanged (lensflare and New Material), in the case of re-install,
- the user has the option to toggle them on or off. When re-install is not selected, there would be no
- change to the asset regardless, so we just show that its there without the toggle option.
- We also show an icon for stuff that has changed (on the right-hand side). Note that all changed assets have
- the toggle to update them or not.

A package is deemed re-installable if:

- Unity can place all assets within the package inside folders that are known to the package. What this means is that if I move a single asset from the installed package somewhere else, the parent folder of that asset is now outside of the known package structure, and I cant re-install
- The package has re-generated its GUIDs (bad) but I can place all assets under a single root folder that exists in the project too. This is to support legacy broken packages that have not been split up into the destination project.



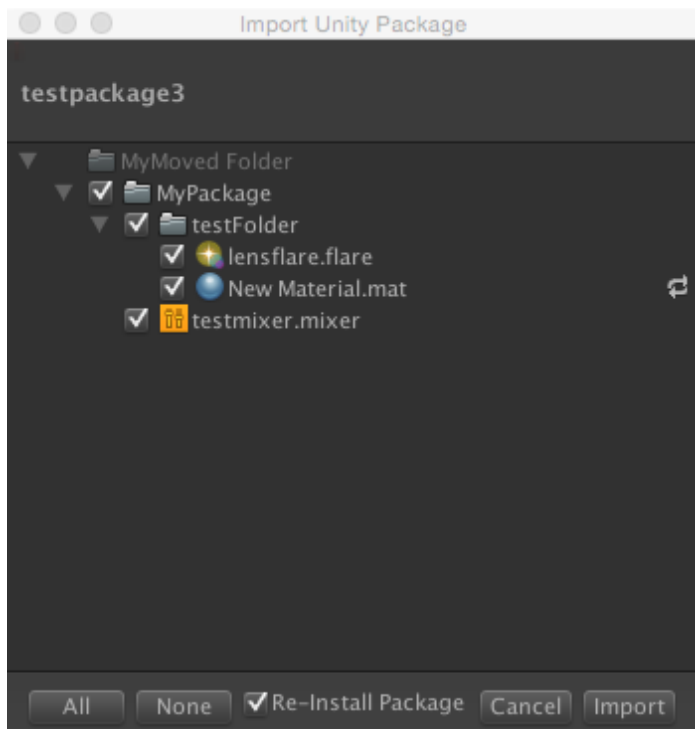
Re-Install?

Highlighted folders will be completely deleted first!
Recommend backing up your project first. Are you
sure?

Cancel

Do It

Moved Update

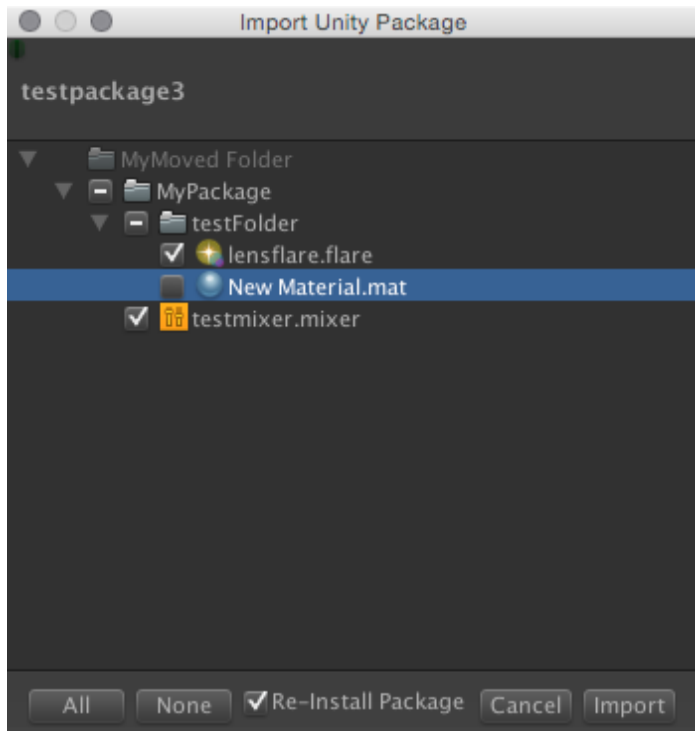


In this case I have moved the package into a different folder (MyMoved Folder) and then updated the package again

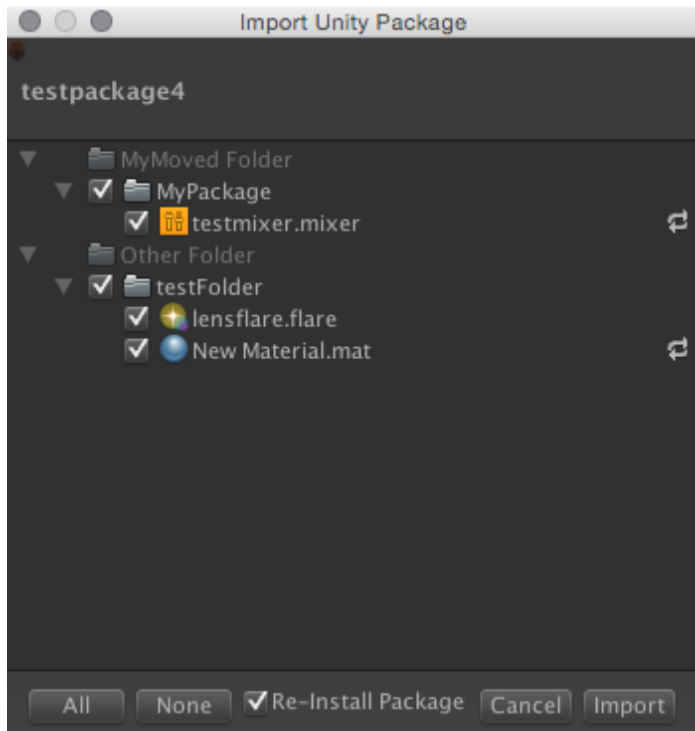
For this update, I've update the material.

- Note that I show the final destination folder of the package (greyed out) so that users can see where the destination has been calculated to.
- Re-Installing the package only deletes the folders that are in the package.

Package Re-Install



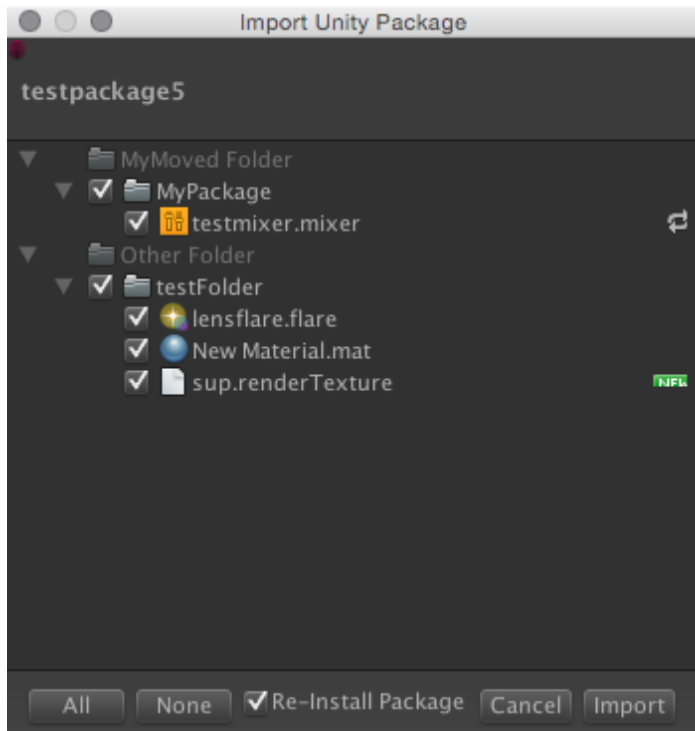
Here we allow package re-install, even though everything is exactly the same. We do this so that users can re-install the same package and choose to delete certain portions of the package. This is the same as basically deleting the assets manually from the Project Browser. But it would require more code to actually prevent it.



Between updates, I've moved the sub folder "testFolder" into a separate root folder, and now I update with a new package, which has changed the material and the mixer

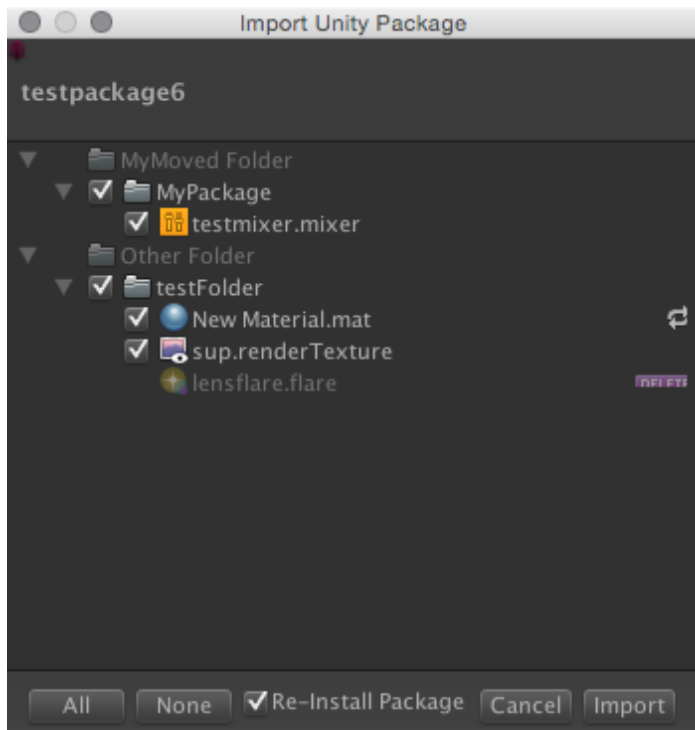
- Note that now we have 2 root folders that are destination folders
- Also note that we can actually re-install this package still, because all assets are under a known parent folder, its just that that parent folder has moved.

New Stuff - PATH PATCHING



A new render texture has been added in this update, and it is patched correctly into its proper location under testFolder. Also the mixer has changed

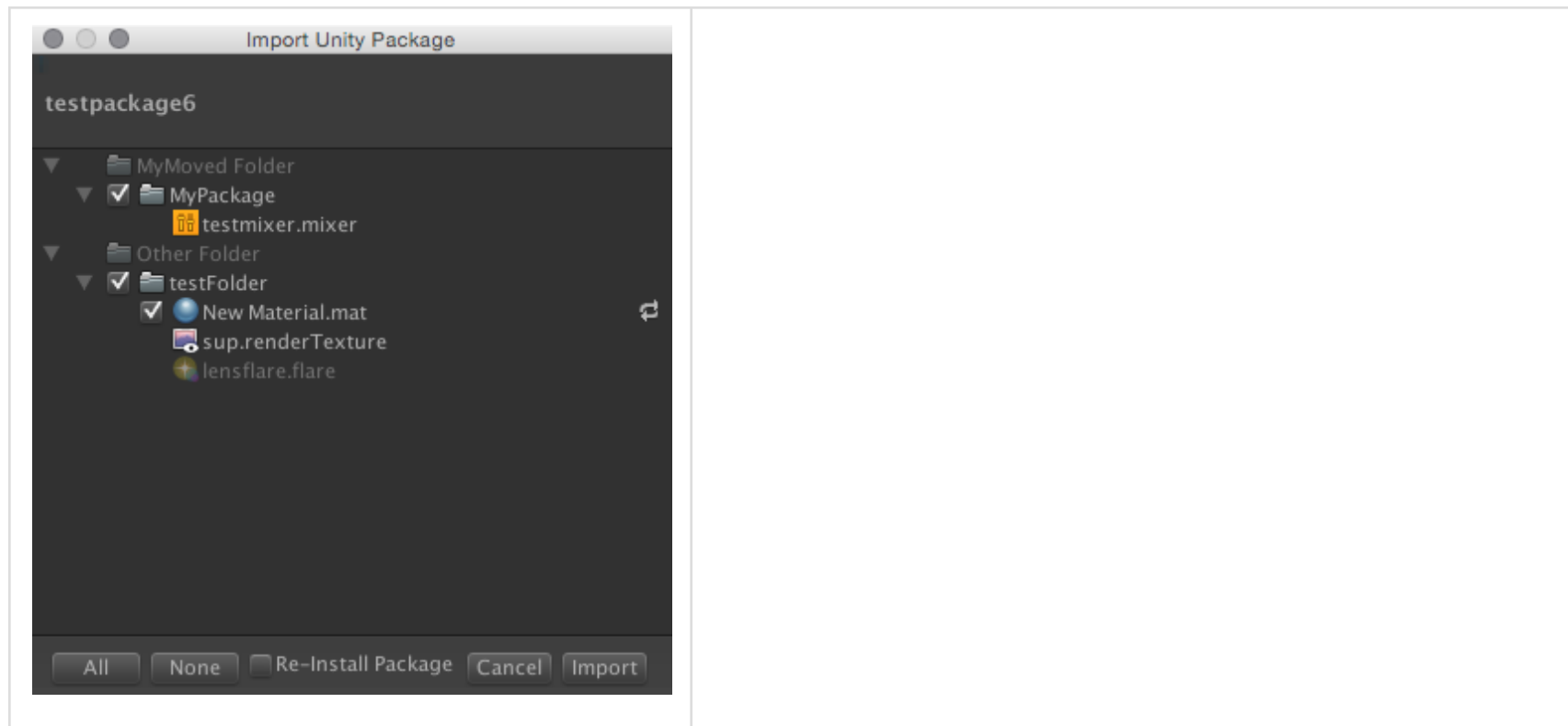
Removal From Package



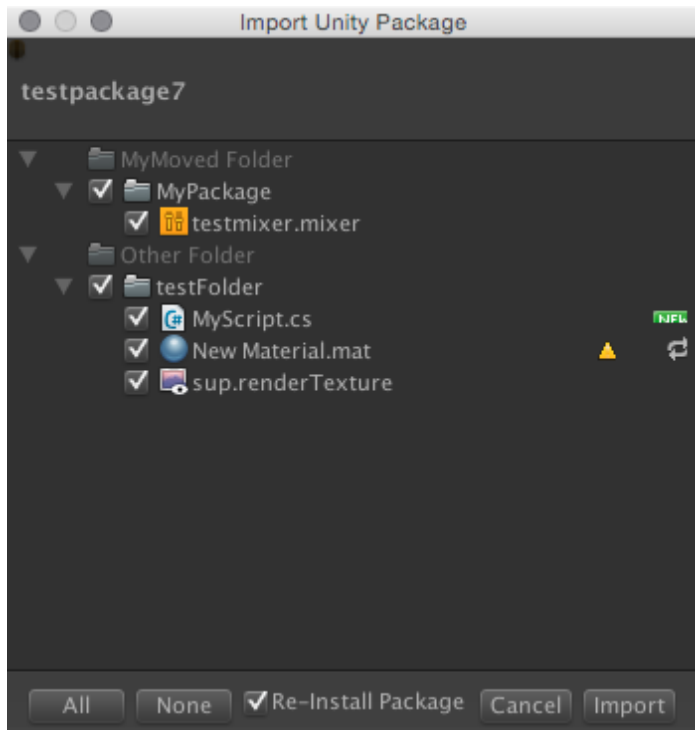
This new update to the package has updated the material, but also deleted the lens flare

There is 2 options here for the user:

1. If they want to re-install the package, it will show that the local asset will be deleted! This allows users to visualise exactly what they are going to lose if they decide to re-install.
2. If re-install is not selected, then the local asset will not be touched, it is still shown for the user to visualise what is happening in their local version.



GUID Troubles

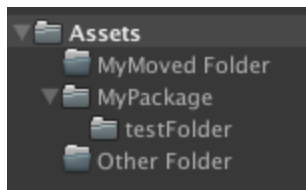
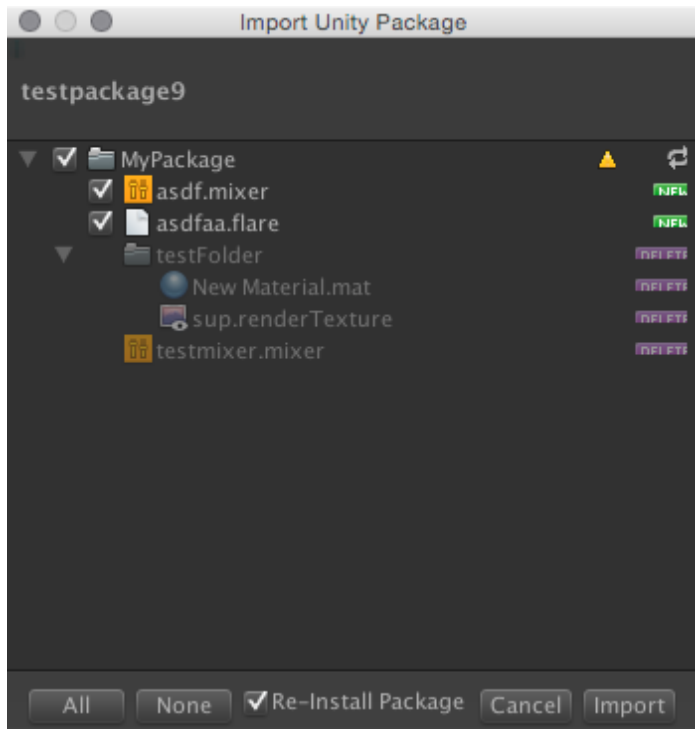


For this new package update, the package maker has deleted the New Material, but then decided they want it back, and simply created a new one. A new script has also been added.

- A warning is shown to let the user know that the material does not match the one that is in the same path (in terms of GUID), and that the material is different to the local version.
- When importing, the material will be overridden with the one in the package, including changing the GUID.
- It is important to warn the user, because any links they have to this material in their project that are not part of the package will be broken, because the GUID does not match.
- Note that we break the links between project and package, because there could be scene and prefab links within the package, and it would be harder for users to try and fix those links than the ones between the package and their stuff.

⚠ Overriding existing asset with new GUID from package, this may break links within your project to package assets (Assets/Other Folder/testFolder/New Material.mat)

Complete GUID and Package Redo



Here we have completely create a new package with new assets in it, but the package name is the same as the previous one (MyPackage). Note all GUIDs are different.

In the first image we see that I cannot detect that the package has been moved into different folders, and there is no way for me to detect that, so I simply try to install it as a new package.

The second image shows me moving everything back to the original install location, so that the package lives under "MyPackage".

Now the import window shows that we can re-import this package, because even though the root folder has a different GUID, its still in the standard install location, so we give the user the option.

Note that the package contents here is completely different too, so all the old stuff will get deleted and the new assets will be installed instead.