


# Redirectors

Objects that redirect references for moved assets from their old location to their new location.



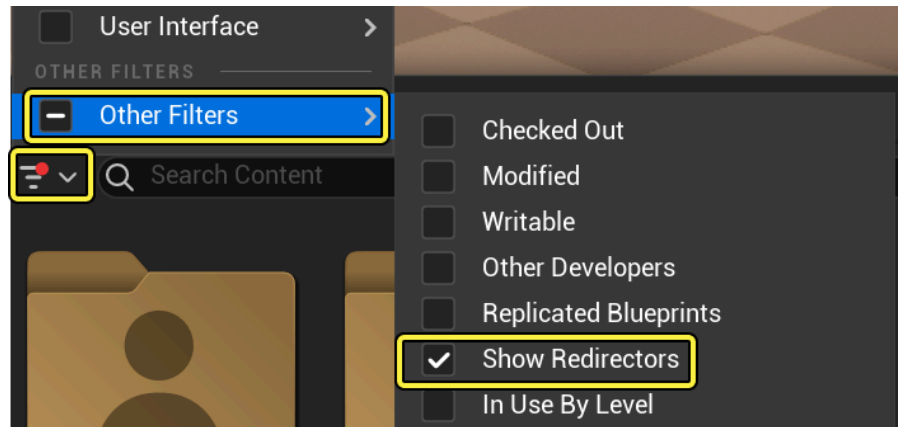
When you move or rename an asset in **Unreal Engine (UE)**, it leaves a **Redirector** in the asset's old location. Packages that are not currently loaded but that reference the asset will therefore be able to find it in its new location.

 Choosing a consistent naming scheme early in your project will avoid many of the problems experienced with Redirectors. Refer to [Recommended Asset Naming Conventions](#) for guidelines on how to name assets.

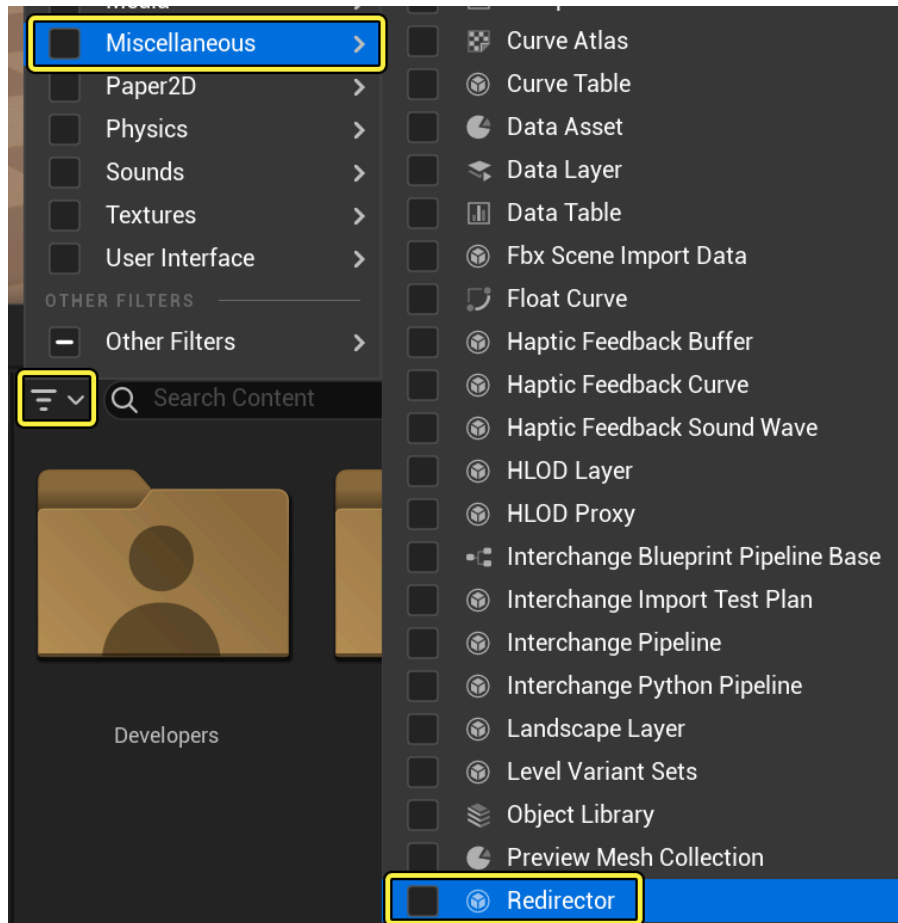
# Viewing Redirectors in the Content Browser

There are two filters for viewing Redirectors in the Content Browser:

- **Filter > Other Filters > Show Redirectors** will make Redirectors visible in the Content Browser, but won't filter out other types of assets.



- **Filter > Miscellaneous > Redirectors** will exclusively show Redirectors in the Content Browser, similar to how other filters work.



## Clean Up Redirectors in Unreal Editor

To remove Redirectors and force your asset references to re-target an asset's new location, right click a redirector and select **Fixup**. This will resave all packages that point to the redirector, and will delete the redirector if it was able to resave all things referencing it.

## Clean Up Redirectors Using the ResavePackages Commandlet

You can also run the `ResavePackages` commandlet using the `-FixupRedirectors` option to clean up all redirectors in your project. The following is an example command line:

```
1 UnrealEditor.exe <GameName or uproject> -run=ResavePackages -fixupredirects -autocheckout -projectonly -unattended
2
```

 Copy full snippet

This version of the command line will check out all files that need to be fixed up from your version control system, then clean up their redirectors on the user's local machine. After that, the user must submit them. `-autocheckin` can be used by an automated process and it will also check the files in for you.

## Caveats

## Renaming

If you create an object, rename it, and then create a new object with the same name as the original, an error will occur. This is because a Redirector was created when the first object was renamed, and a Redirector and a resource cannot have the same name.

## Dangling Redirector

There are a few known issues with Redirectors that can be reproduced as follows:

### Scenario 1

- Rename object A to B.
- Delete B.

- An error message will say that B could not be deleted because it is in use. This happens because the Redirector created during the rename will still point to B.

## **Scenario 2**

- Rename object A to B.
- Rename object B back to A.
- Delete A.
- The redirector that was created for the first rename will be destroyed, but a new one will be created at B. As a result, A will not be able to be deleted because it is being referenced.

Fixing redirectors from the editor or SavePackages before deleting should solve these problems.