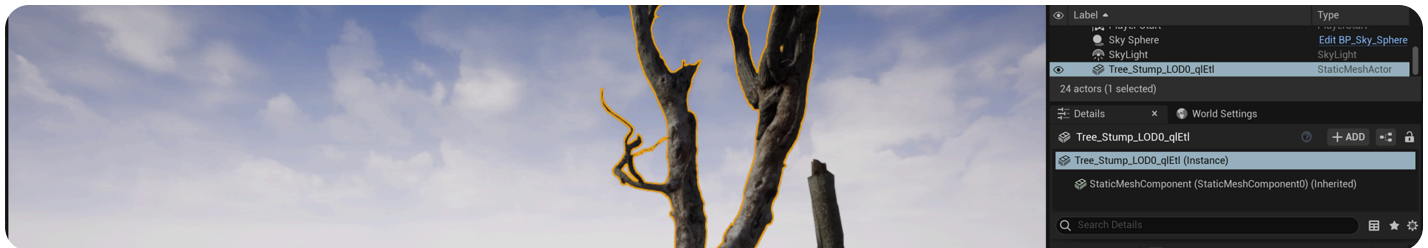


One File Per Actor

An introduction to the One File Per Actor System and how it can be used in your projects.



In previous versions of Unreal Engine, when you wanted to make changes to one or more Actors inside a Level, you needed to check the file out of source control. This locked other team members out of that file until your work was complete, slowing down the development process since only one person could work on a file at a time.

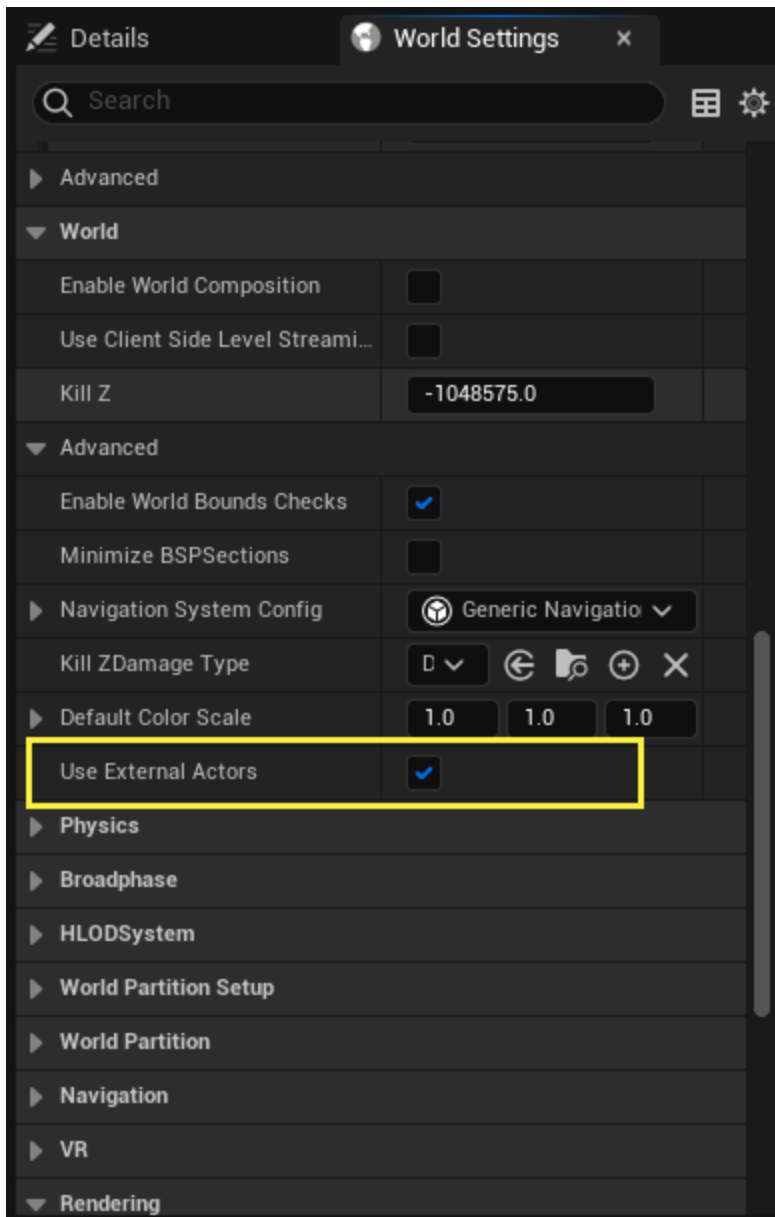
One File Per Actor (OFPA) reduces overlap between users by saving data for instances of Actors in external files, removing the need to save the main Level file when making changes to its Actors.



The One File Per Actor feature is only available in the Editor. All Actors are embedded in their respective Level files when cooked.

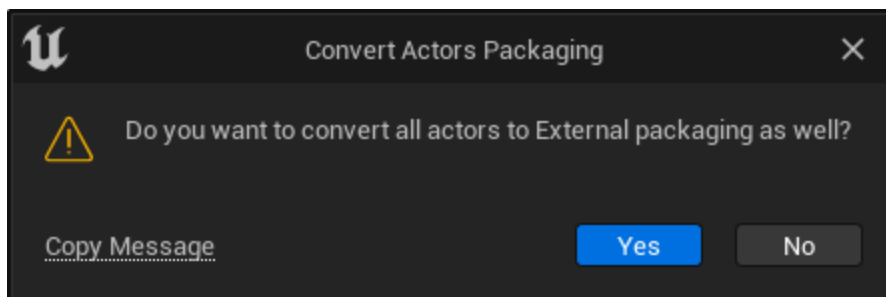
Enabling One File Per Actor

The One File Per Actor feature is enabled by default when using [World Partition](#). To enable OFPA in a non-partitioned world, follow the steps below:



Enabling OFPA for a Level.

1. Open the **World Settings** panel. From the main menu, go to **Window > World Settings**.
2. Navigate to the **World** section of the panel and click the checkbox next to **Use External Actors**.
3. You will be asked if you want to convert all Actors to external packaging. Click **Yes** to complete the conversion to OFPA.



Click Yes to convert all the Actors in the Level.

4. Save your Level.

Converting Sublevels

It is important to note that only the current Level will be converted to OFPA when you activate the **Use External Actors** option for your Level. To convert any of your sublevels, you must load them and enable the **Use External Actors** for each of them, as described above. Since this may be an issue if your Level contains several sublevels, you can use a commandlet to convert a Level and its sublevels automatically.

Convert Levels To External Actors

Executable -Tasks Target Map

```
UnrealEditor.exe -run=ConvertLevelsToExternalActorsCommandlet  
-nosourcecontrol -convertsublevels  
"Game/Maps/TestMaps/ExternalActors/MainMap"
```

A breakdown of the commandlet.

Follow the steps below to use the commandlet:

1. Open a Command Prompt window.

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\          >_
```

2. At the prompt, navigate to the location of the **UnrealEditor.exe** file. For example:

```
C:\Builds\Home_UE5_Engine\Engine\Binaries\Win64
```

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\          >cd c:\Builds\Home_UE5_Engine\Engine\Binaries\Win64
C:\Builds\Home_UE5_Engine\Engine\Binaries\Win64>_
```

3. Next, begin typing the command. Start with the name of the `.exe` file that will run the commandlet, `UnrealEditor.exe`.

4. Add the name of the commandlet and the following arguments:

- `-run="ConvertLevelsToExternalActorsCommandlet"` is the name of the commandlet.
- `-nosourcecontrol` tells the commandlet to not use source control.

- `-convertsublevels` tells the commandlet to convert any sublevels the map may have.

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\          > cd c:\Builds\Home_UE5_Engine\Engine\Binaries\Win64

c:\Builds\Home_UE5_Engine\Engine\Binaries\Win64>UnrealEditor.exe -run="ConvertLevelsToExternalActorsCommandlet -nosource
control -convertsublevels
```

5. Finish the command with the location of the Level:

`"/Game/Maps/TestMaps/ExternalActors/MasterMap"` In this example, **Game** is the name of your project and **MainMap** is the name of the map you wish to convert.

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

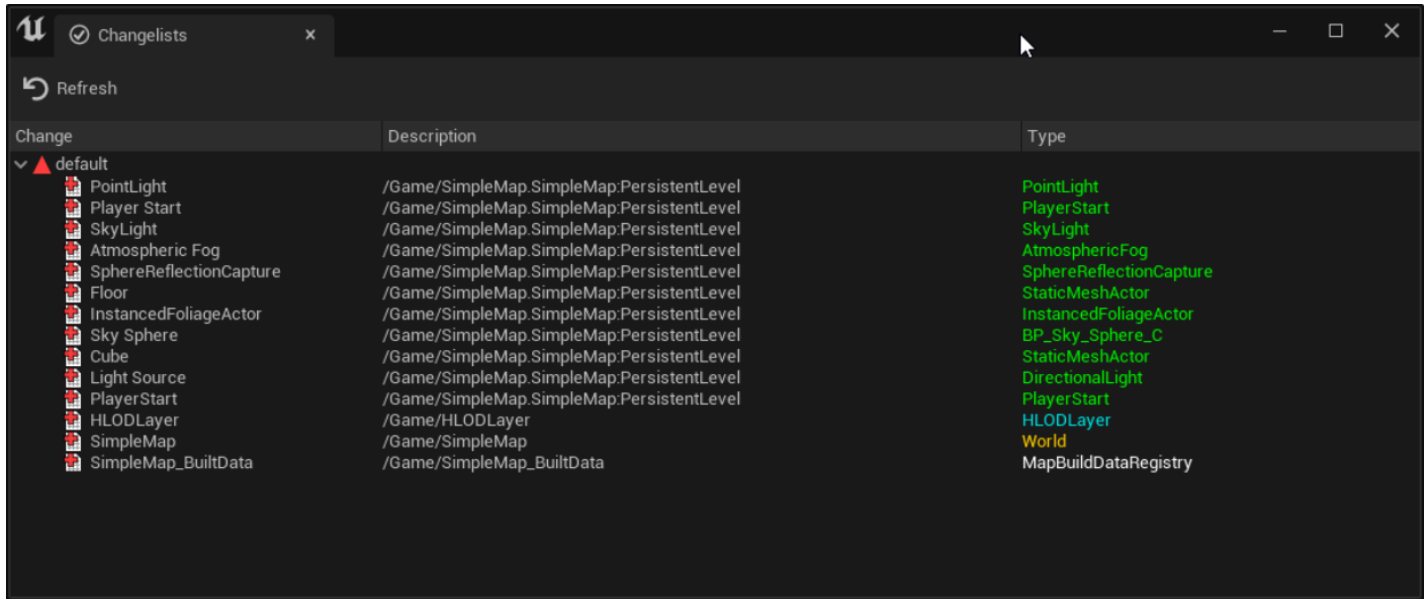
C:\Users\          >cd c:\Builds\Home_UE5_Engine\Engine\Binaries\Win64

c:\Builds\Home_UE5_Engine\Engine\Binaries\Win64>UnrealEditor.exe -run="ConvertLevelsToExternalActorsCommandlet" -nosour
cecontrol -convertsublevels "/Game/Maps/TestMaps/ExternalActors/MainMap" _
```

6. Press **Enter** to run the commandlet. It will then convert the Level and any sublevels to OFPA.

Using OFPA With Source Control

While working in your source control application, you will notice that external Actor file names are encoded. To address this issue, you can view and validate the contents of a changelist before you submit it using the **View Changelist** window.



An existing changelist for a level that uses OFPA.

This window displays existing changelists and the files they contain. For external Actors, you can see the Actor names, Level paths and Asset types, rather than the encoded filenames.

With changelist support built into the Editor, you can validate the contents of your changelists before you submit them to source control. This is required due to the added complexity that OFPA brings to your projects. For example, a user can check out multiple files and only submit some of them, possibly leaving dangling references.

For more information on using source control in Unreal Engine, see [In-Editor Source Control](#).



When using OFPA, content and Actor files should be submitted to source control from within the Editor.



Changelist support is only available when using Perforce as your source control provider.