



Unreal + Wwise

howest
university of applied sciences

Meltem Ozcelik (Concept: Amandine Gerard) / **Daniil Shashenkov** (Concept: Tooth Wu) / **Joao Desager** (Concept: Tan Zhi Hui)

Unreal

Make 100% sure we have everything we need installed correctly.

Unreal 5.5.2, **with** source:

Unreal Engine 5.5.1 Installation Options

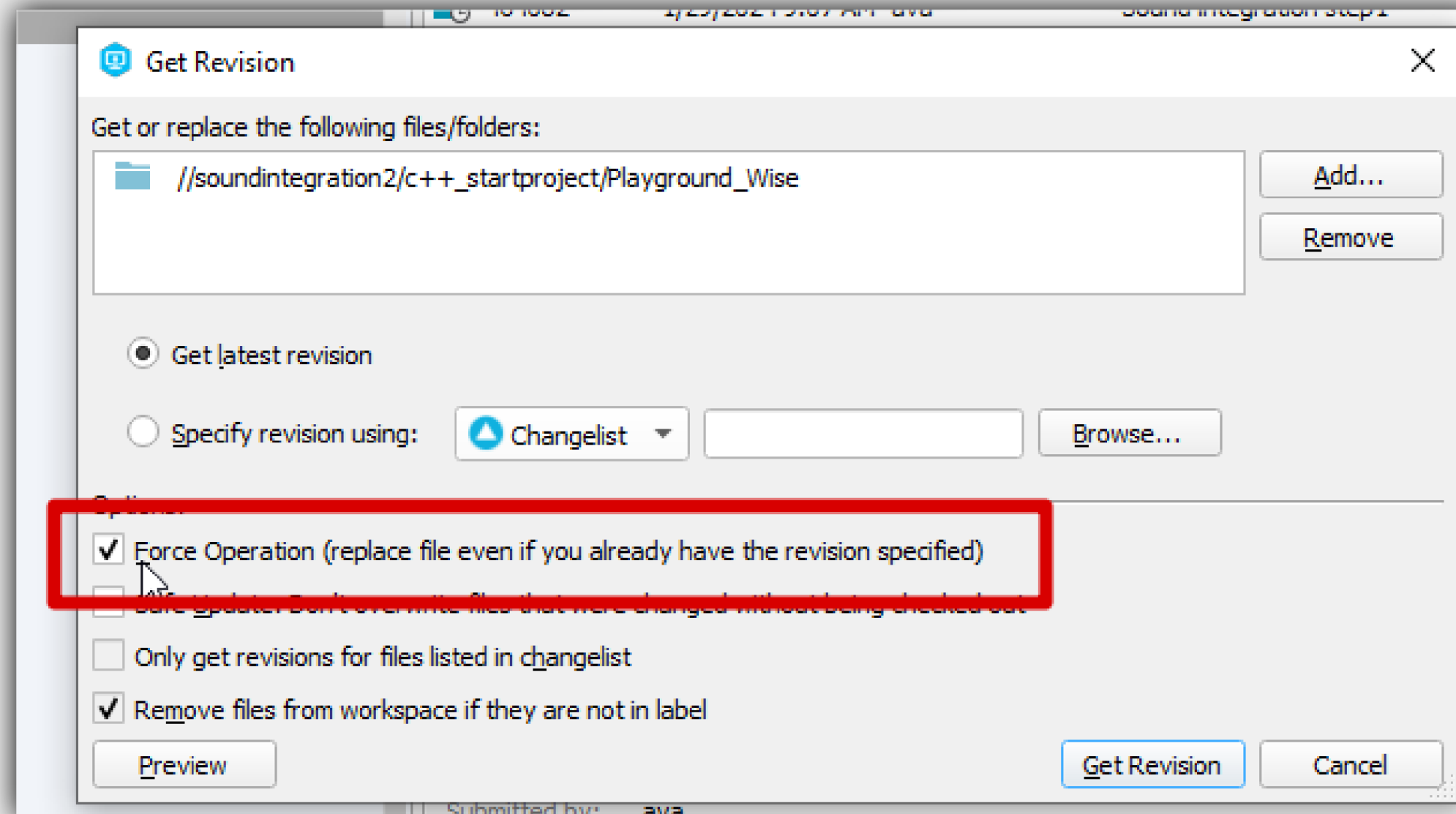
Core Components <small>(Required)</small>	34,28 GB	<input checked="" type="checkbox"/>
Starter Content	641,71 MB	<input checked="" type="checkbox"/>
Templates and Feature Packs	3,77 GB	<input checked="" type="checkbox"/>
Engine Source	385,04 MB	<input checked="" type="checkbox"/>
Editor symbols for debugging	38,36 GB	<input type="checkbox"/>
▼ Target Platforms		

And **without** any other platforms:

Unreal Engine 5.5.1 Installation Options

Editor symbols for debugging	38,36 GB	<input type="checkbox"/>
▼ Target Platforms		
Android	8,82 GB	<input type="checkbox"/>
IOS	3,27 GB	<input type="checkbox"/>
Linux	15,90 GB	<input type="checkbox"/>
TVOS (requires IOS)	6,17 GB	<input type="checkbox"/>

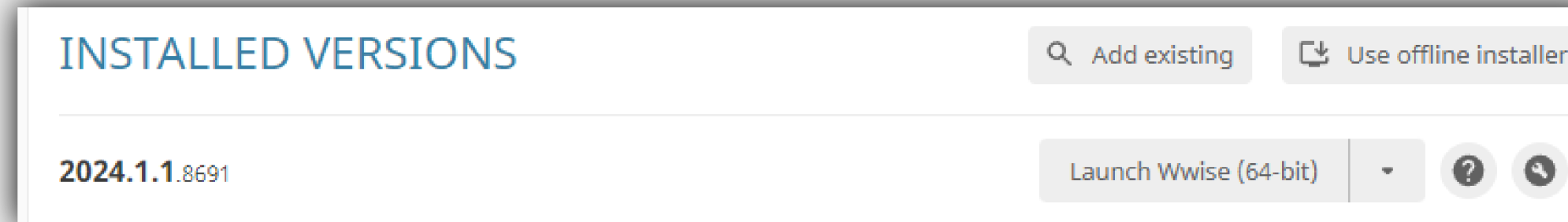
- Delete your current project from your computer and force-get it back from performe.



- This is to illustrate that you can always remove everything and try again.

Wwise

Make sure to have the 2024.1.1 version of Wwise

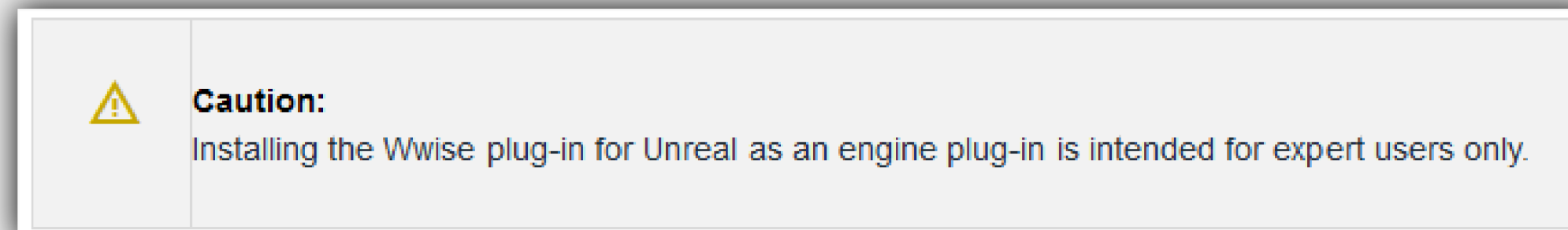


Integrating Wwise can be done in two ways

1. As a plugin local to the game
2. As a plugin for the engine.

The first option is the "easiest" one, however the downside is that the plugin is a massive 5GB addition to your project. This is not even possible if the project is using git as a versioning system.

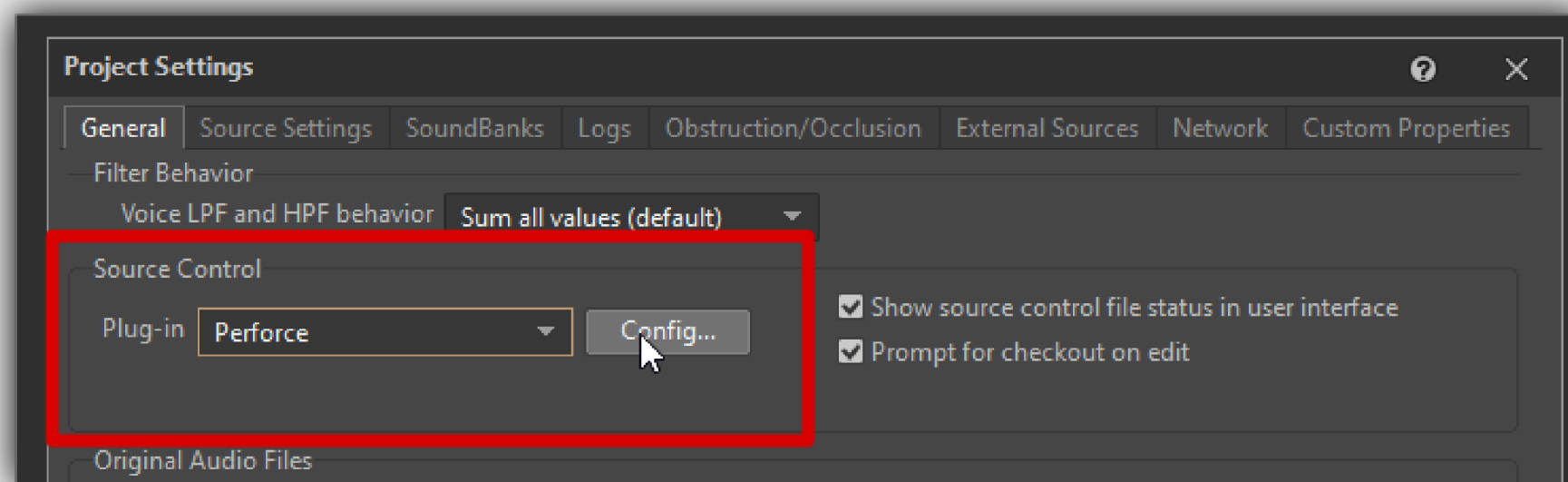
The second option is the one from hell



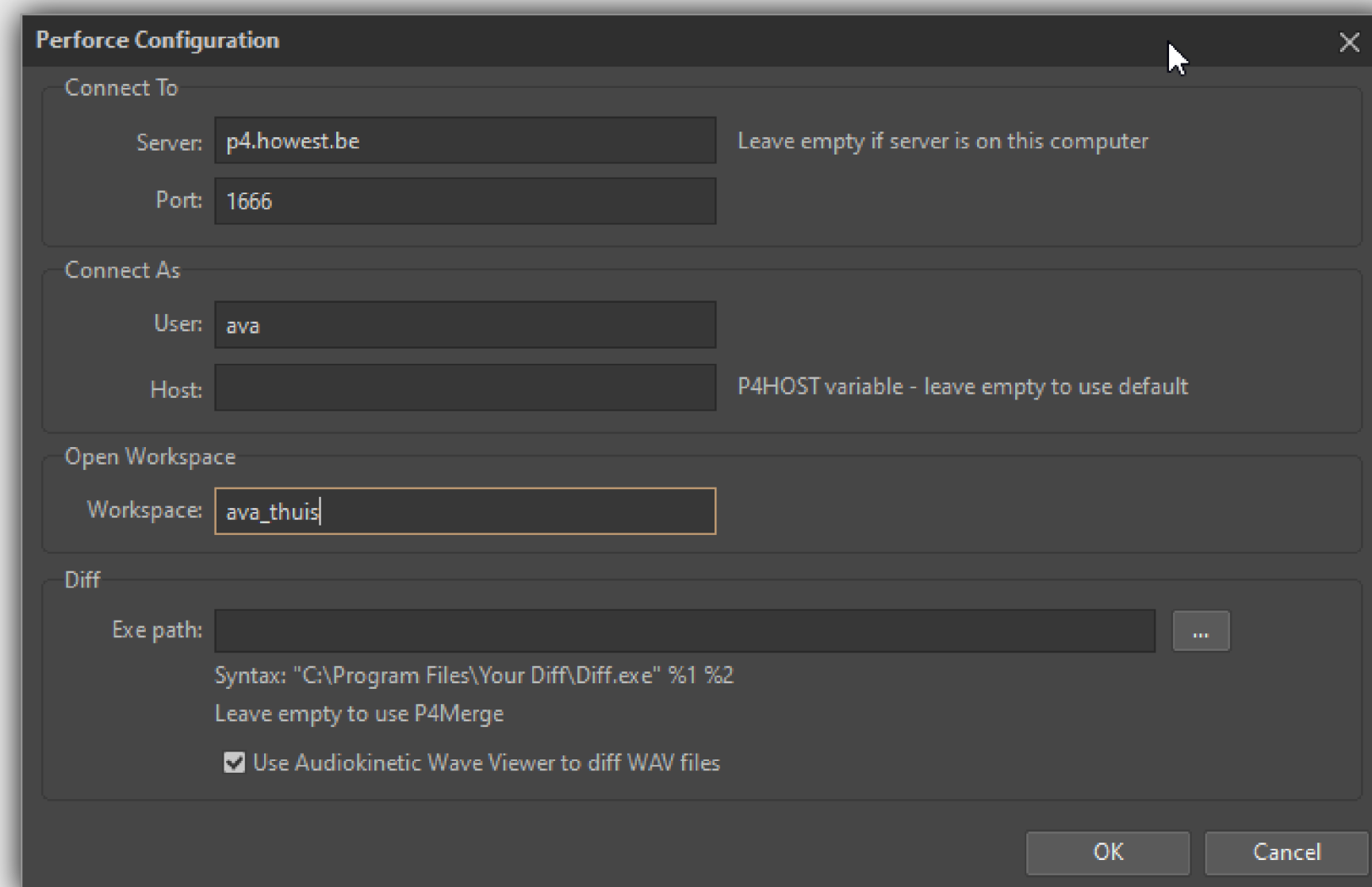
But the best option for your peers and versioning system.

Let's start with a new Wwise project. Create it inside your perforce folder next to the folder of your Unreal project (**not** inside it).

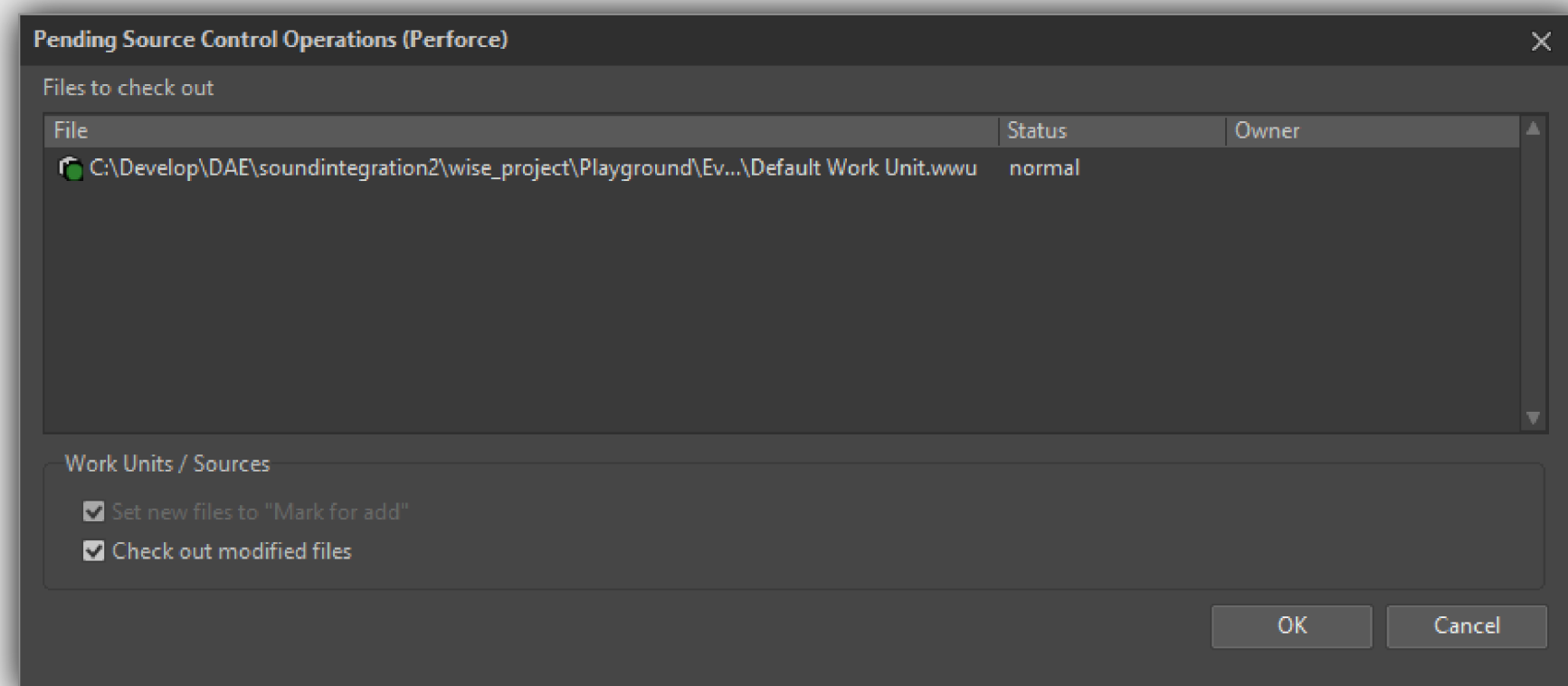
Right at the start, open project settings and setup the perforce connection:



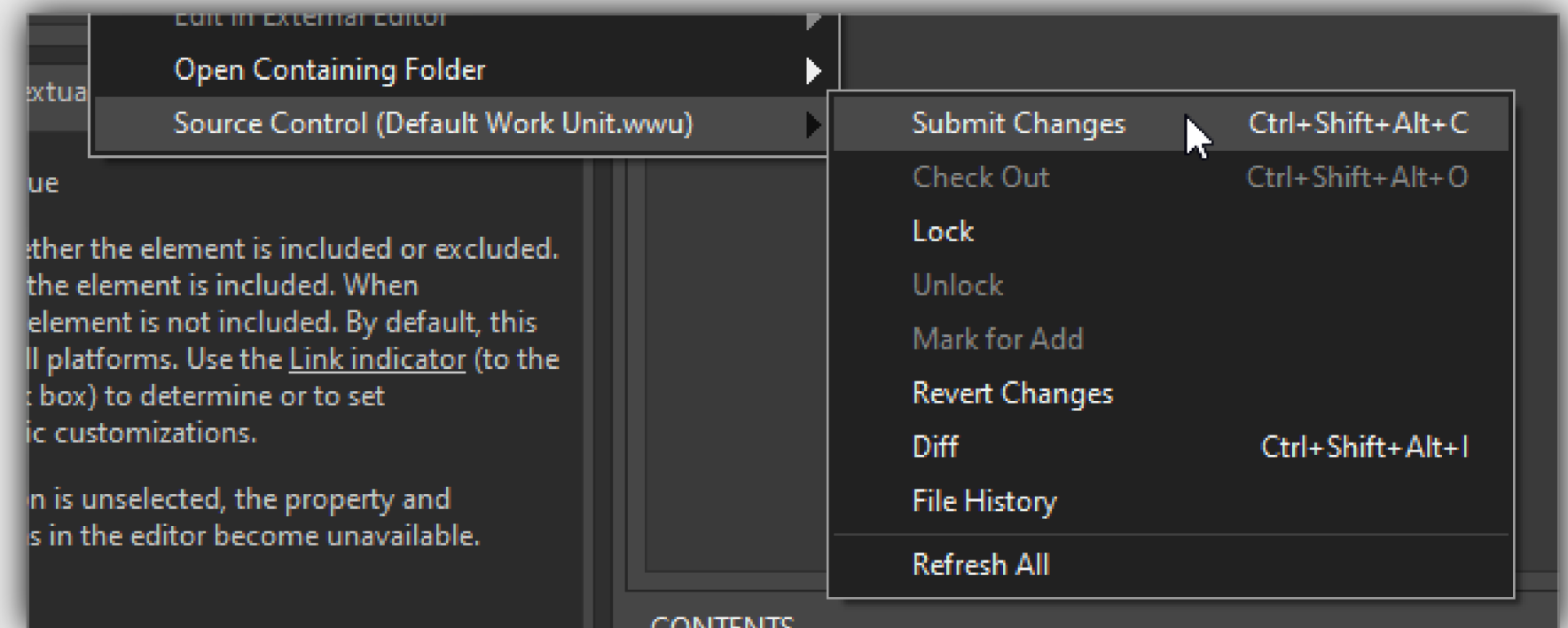
Enter the details



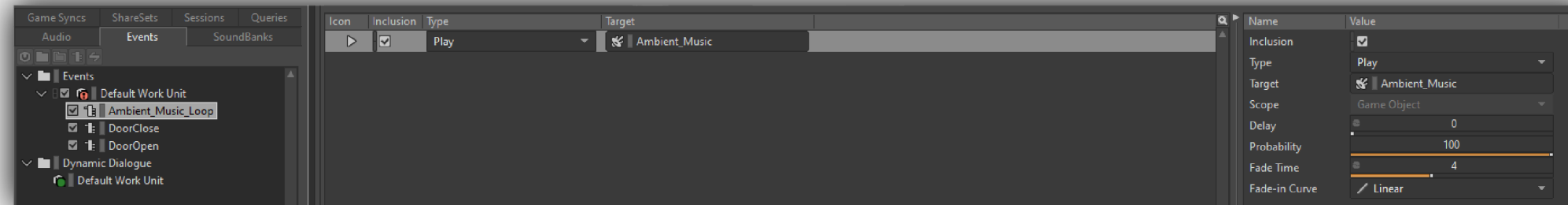
If you now make a change to an asset Wwise will prompt you to check the file out



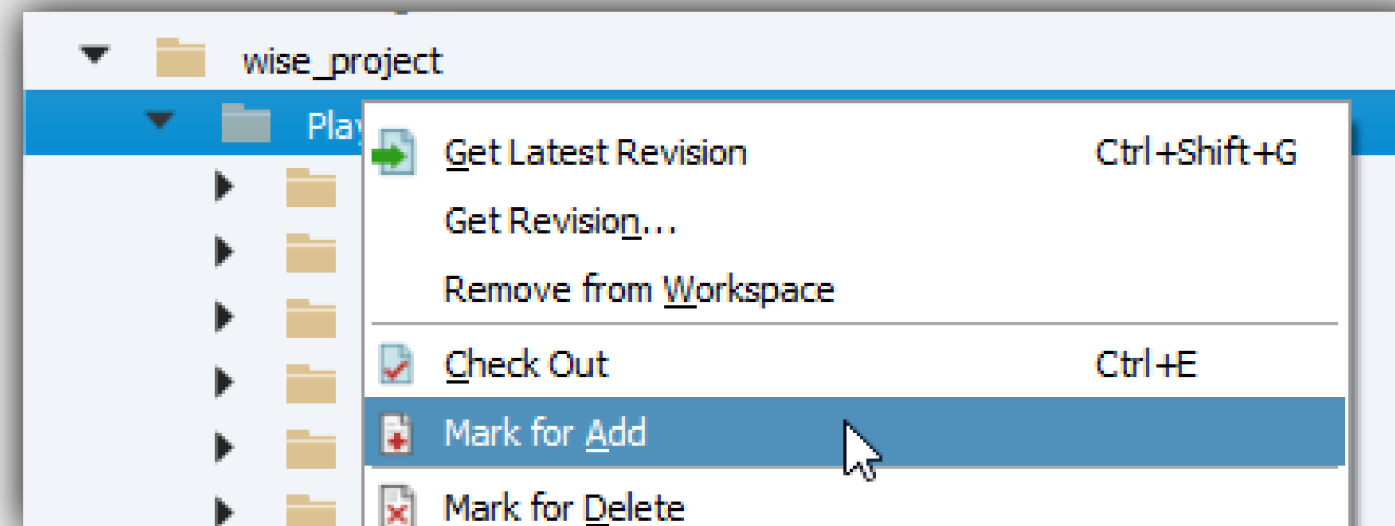
If you right click a work unit, you get some options extra for perforce



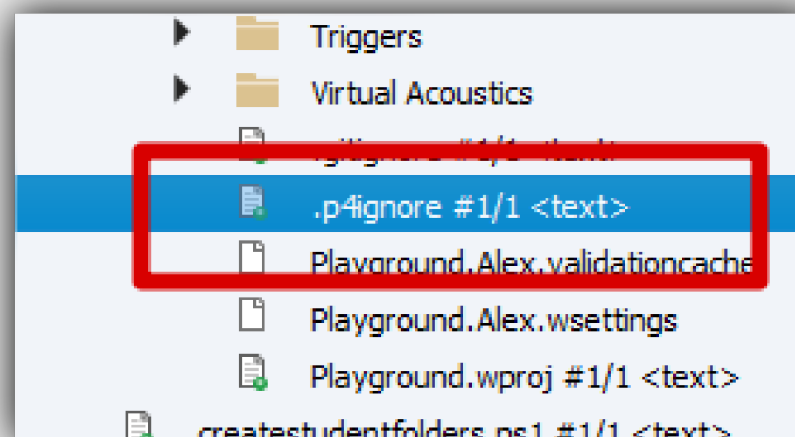
For this lab, add the `Ambient_Music_Loop` event again as you did in Audio Implementation 2



Save everything and submit to perforce. You can right click the project folder in P4V and select "Mark for Add"

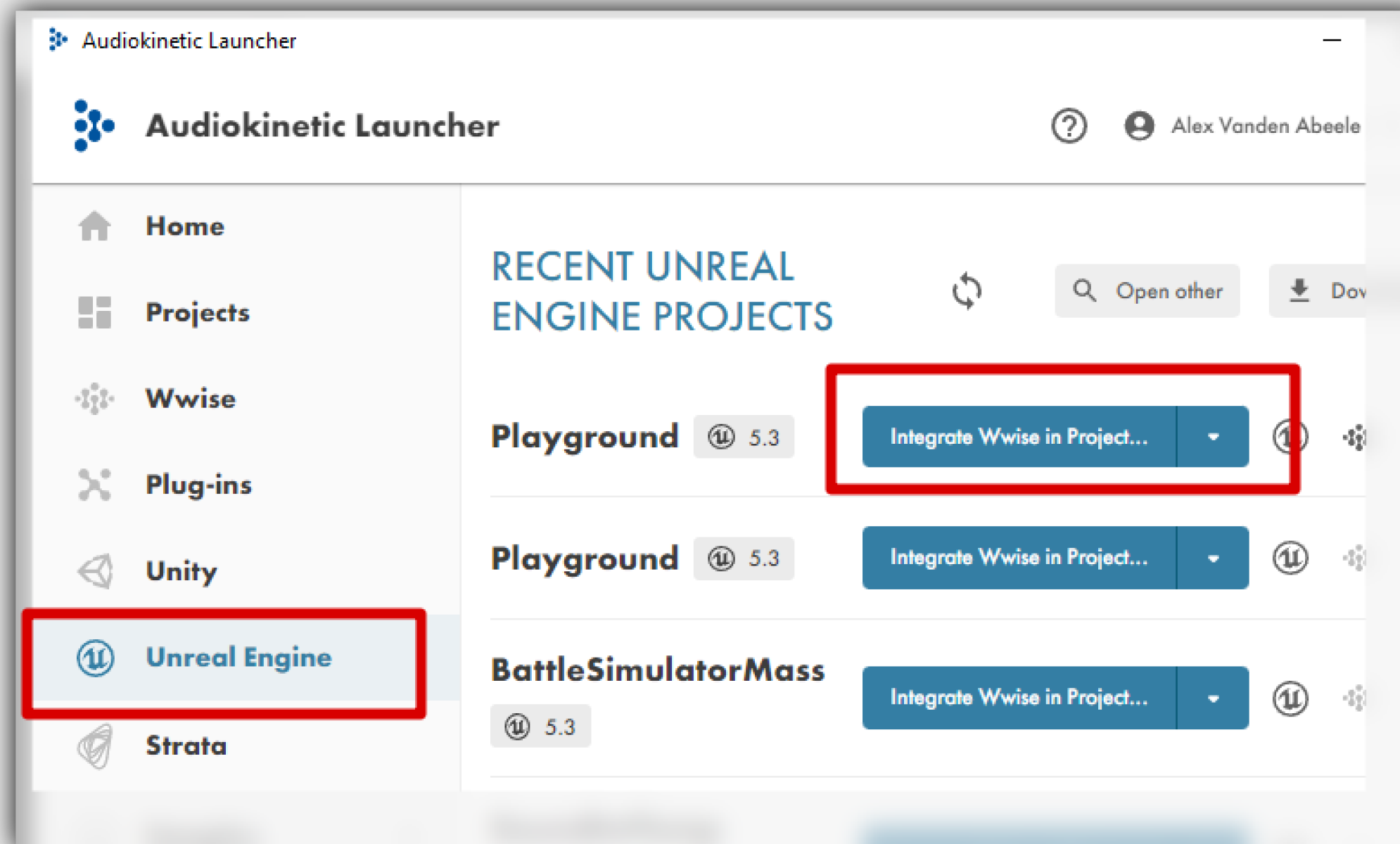


Only a few files will be added, thanks to the .p4ignore file, feel free to check out its contents



Wwise as local plugin

In the Wwise launcher, find your copied project and click "Integrate Wwise in Project..."



Wwise as local plugin

- Make sure to use the same Wwise version (2024.1.1)
- Make sure everything is blue, if a setting is incorrect, some red notification will tell you what is wrong.
- Make sure to select your Wwise project. If you don't it gets generated inside the Unreal project, but that is not an ideal place for it.
- Before you integrate, **checkout the uproject file** and everything in the Config folder on perforce.

INTEGRATE WWISE IN PLAYGROUND [Learn more](#)

5.3

Integration version

Latest > Version **2023.1.0.8367.2849**

Make sure to consult the platform requirements.

Wwise SDK

*The Wwise SDK will be used to copy **platforms** to your project.*

Wwise Unreal Integration supports **Android, Linux, Mac, iOS, tvOS, PS4, PS5, Switch, Windows, WinGC, UWP, XboxOne, XboxOneGC and XboxSeriesX.**

The "include", "x64_vc160" and "x64_vc170" folders are mandatory.

Copy Wwise SDK files from:

C:\Program Files (x86)\Audiokinetic\Wwise 2023.1.0.8367\SDK

The following platforms were detected:

- Microsoft/Windows/Visual Studio 2019
- Microsoft/Windows/Visual Studio 2022

Modify Wwise 2023.1.0.8367

If you wish to add new platforms in your Unreal Engine project, you must add them first in Wwise.

The Wwise SDK for these platforms will be copied for each one detected.

*The Wwise SDK will be used to copy **plug-ins** to your project.*

If you wish to add new plug-ins in your Unreal Engine project, you must add them first in Wwise.

Add Plug-ins to Wwise 2023.1.0.8367

Wwise Project

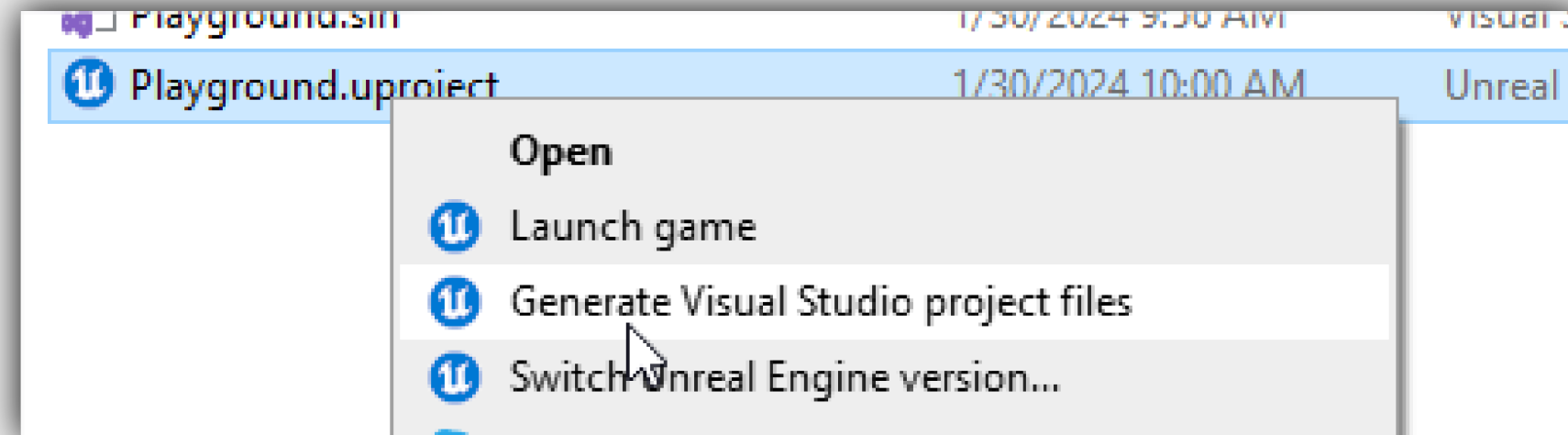
Wwise project path

C:\Develop\DAE\soundintegration2\wise_project\Playground\Playground.wproj

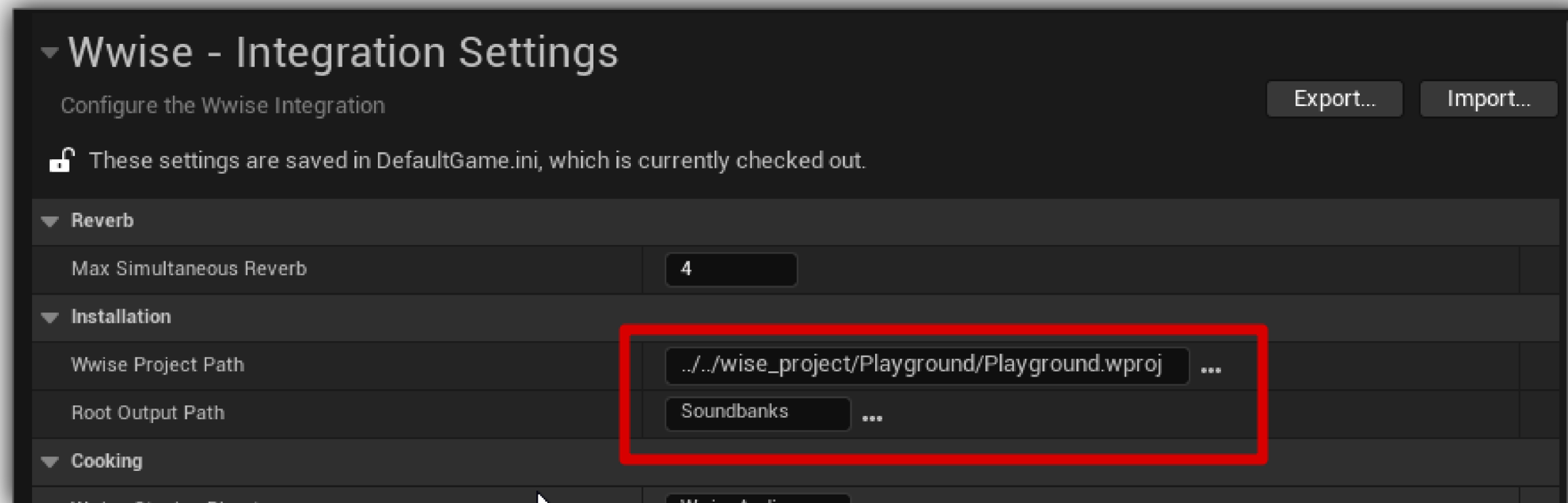
https://www.audiokinetic.com/en/library/wwise_launcher/?source=InstallGuide&id=integrating_wwise_into_an_unreal_project

Wwise as local plugin

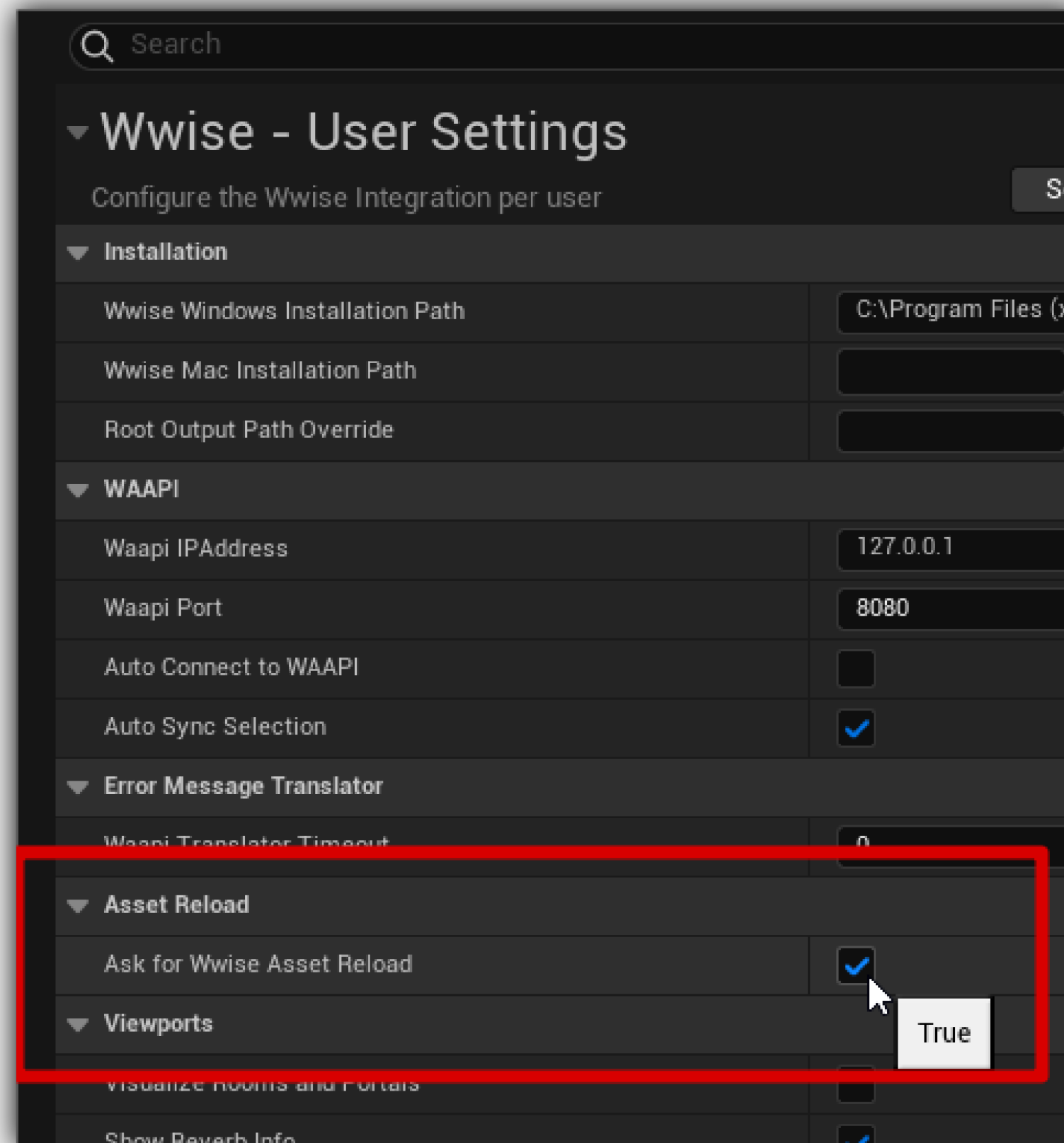
After the integration is done (it takes quite some steps) right click the uproject file and generate the Visual Studio project files



Open up the visual studio project, hit play. In Unreal in the project settings, select the correct paths:

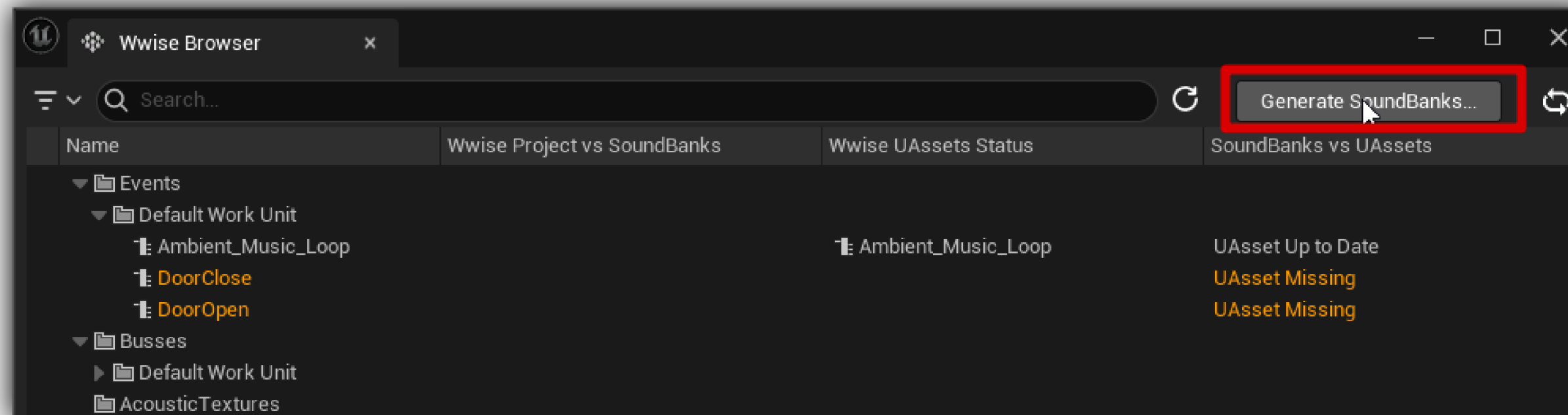


Don't forget to enable asset reloading:

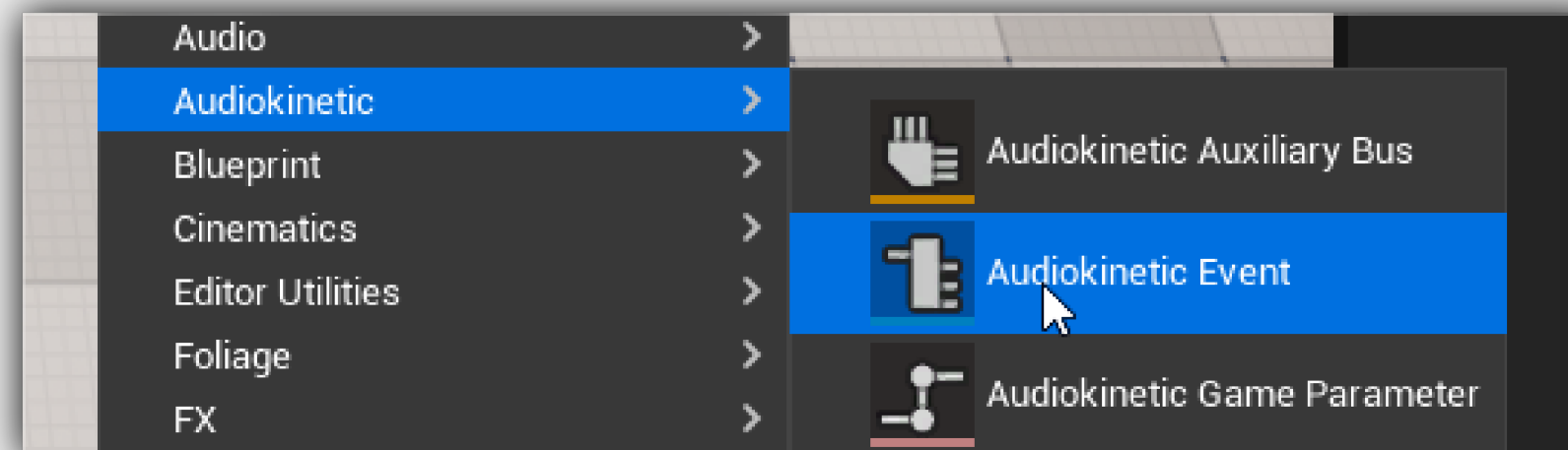


Wwise as local plugin

Via `Window` -> `Wwise browser` you can now generate the soundbanks:

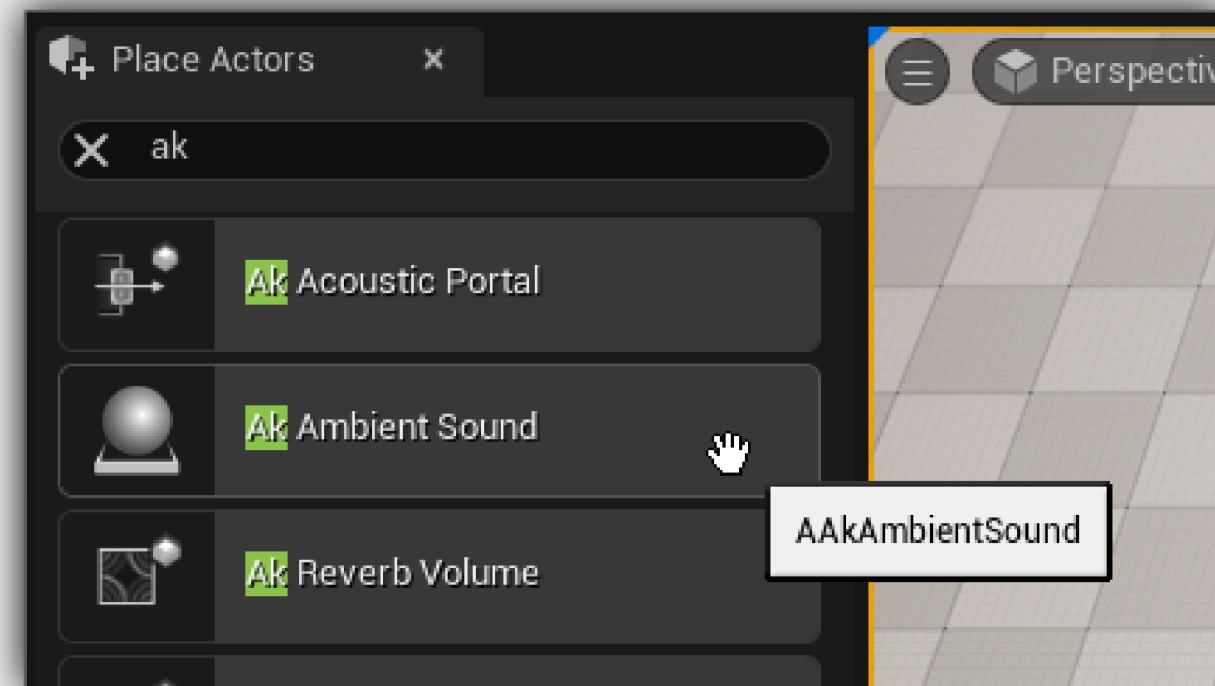


We don't have a UAsset yet, so create one with the same name `Ambient_Music_Loop` :

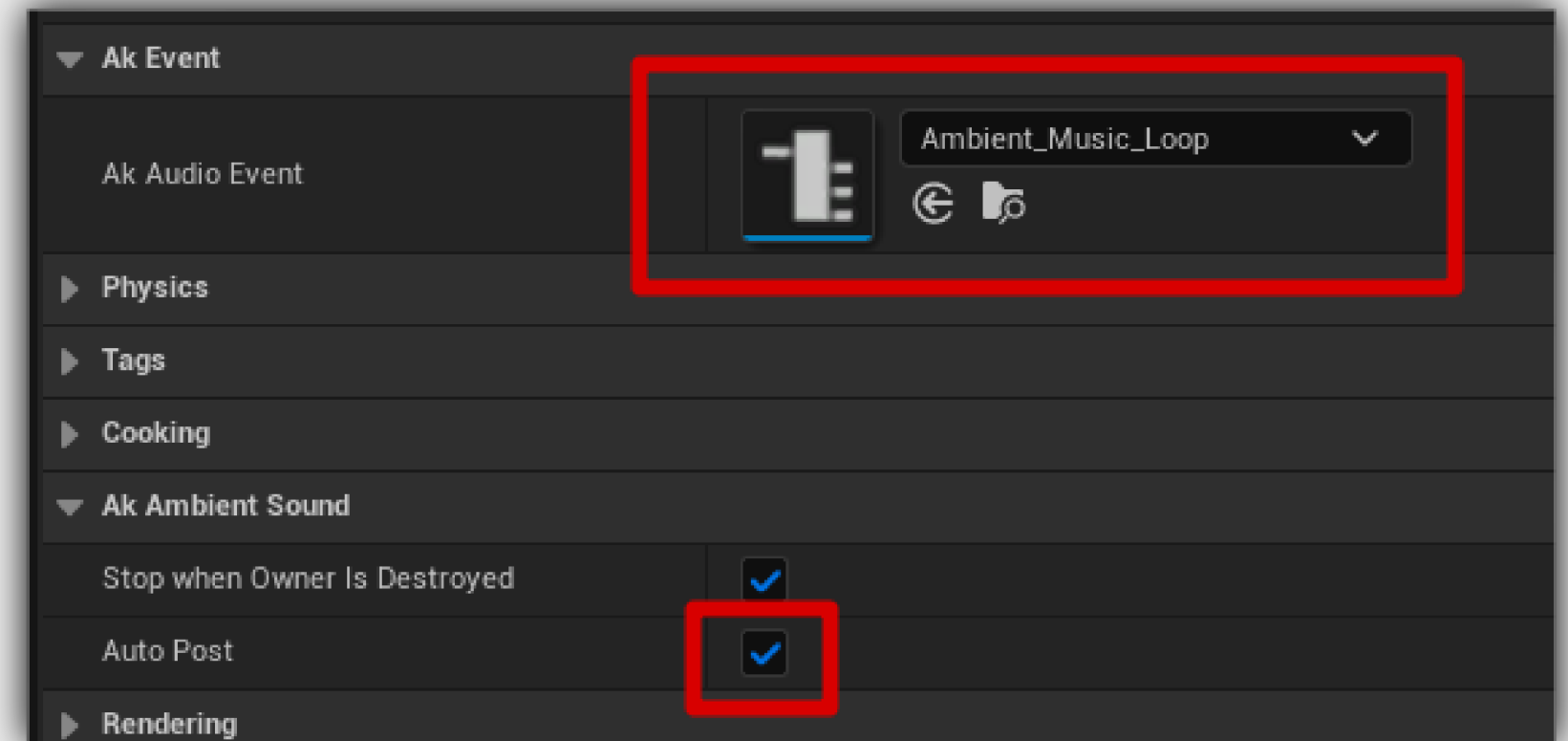


Wwise as local plugin

From the "Place actors" window, drag an AkAmbientSound actor in the scene



And set the event



And the music should start playing when we hit Play.

Try packaging a shipping build to make sure everything is ok.

Wwise as local plugin

Now perforce. If people want to sync and use our work, we need to submit it.

- We need to submit all changes to the project file and DefaultGame.ini
- We need to submit the Soundbanks folder
- We need to submit the Plugins folder that contains the Wwise plugin.
 - But that folder is +5GB large!
 - Don't do this yet!

We don't need the *.pdb files to be used by others, so we can exclude those. Add a .p4ignore file in the Wwise folder with this content:

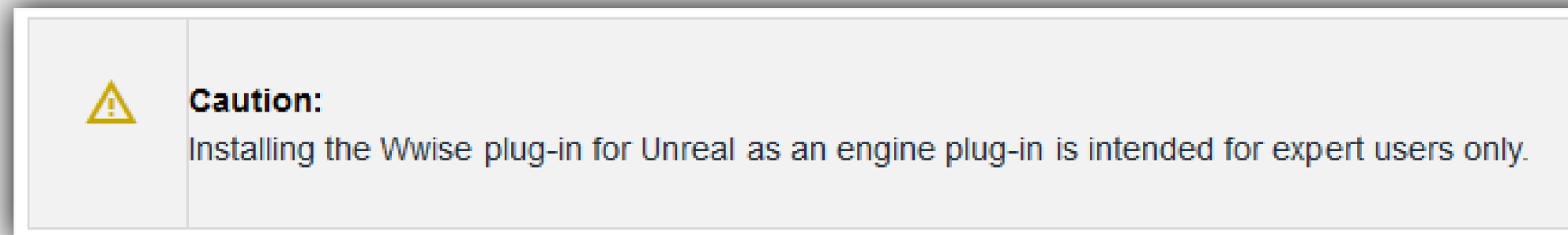
```
*.chm  
*.pdb  
*.exe  
  
!/ThirdParty/**/*.*dll  
!/ThirdParty/**/*.*lib
```

This reduces the commit of the plugin to 2.6GB (which is still very large) - **wait, don't do this yet.**

Wwise as an engine plugin

Since we have a working plugin now, we can actually make an engine plugin out of it and use that one instead. You can simply distribute this plugin to your colleagues who install it in their engine as well and there will be no need to put the plugin in the project anymore. Plus all other project can now also start using Wwise.

However



So tread carefully! Make a backup of your project so you can always revert back if something went wrong.

Let's make you experts. Close Unreal.

<https://dev.epicgames.com/community/learning/tutorials/qz93/unreal-engine-building-plugins>

Wwise as an engine plugin

The documentation says

If you will be working on multiple Unreal projects and Wwise will be used in each of them, you might want to install the Wwise Unreal Integration as an engine plug-in.

"will be used in each of them" is generally not true for your peers. They mention this because the plugin is enabled by default. Let's change that, in Wwise.uplugin:

```
"EnabledByDefault": false,
```

This way, every project will have to switch it on to be able to use it (which is what you want).

If you restart the engine, the plugin will be disabled, so enable it in the Plugins settings, or add it to your uproject Plugins list

```
,  
{  
  "Name": "Wwise",  
  "Enabled": true  
}
```


Wwise as an engine plugin

Now we can build. Find the location where you installed Unreal 5.5.2, for me that is

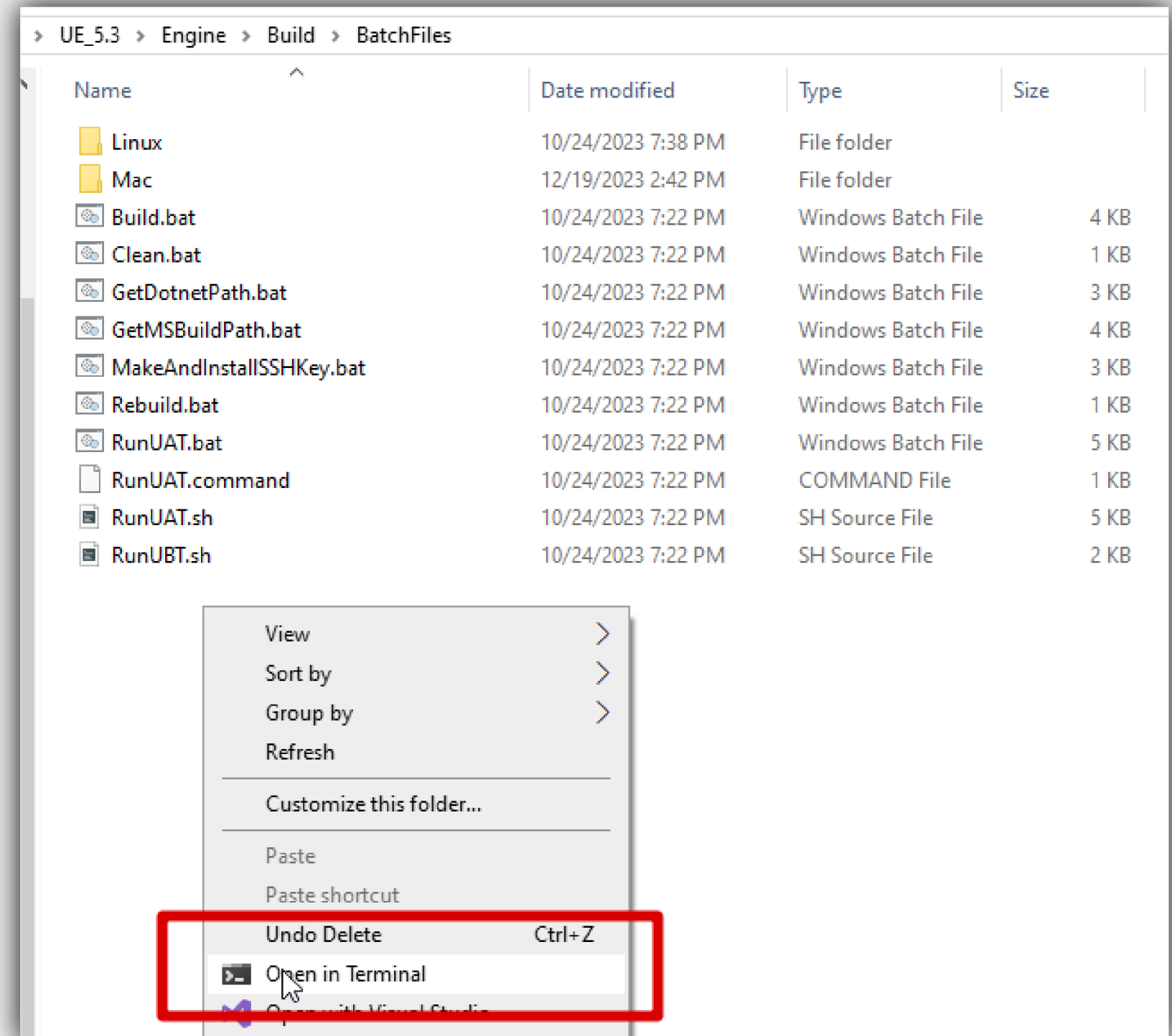
`C:\Program Files\Epic Games\UE_5.5`

Go to the folder

`Engine\Build\BatchFiles`

And open a Terminal there (right-click in the folder)

If you don't have that option I recommend installing it via the Windows Store. You can choose powershell too if you want.



Wwise as an engine plugin

Wwise used to be one plugin, now it is three: **WwiseSoundEngine**, **Wwise** and **WwiseNiagara**.

- WwiseSoundEngine contains the low level stuff and the SDK
- Wwise is the original plugin and contains all functionality we have
- WwiseNiagara has some specific classes to be used in the Niagara system.

We'll need to build at least the first two.

Run this command (adjust paths accordingly + make sure it's all on one line)

```
.\RunUAT.bat BuildPlugin -plugin="C:\PathToProject\Plugins\WwiseSoundEngine\WwiseSoundEngine.uplugin"  
-package="C:\Temp\WwiseSoundEngine" -TargetPlatforms=Win64
```

Fingers crossed, but if all goes well, this will build the first plugin plugin. It takes a while, go get a coffee.

...

Wwise as an engine plugin

In the folder you specified for the `-package` parameter (`C:\Temp\WwiseSoundEngine` in the example), you now have the compiled plugin.

Copy the `ThirdParty` folder from the local plugin to your build plugin. You now have the final plugin. You can zip the content and distribute it as you like.

- To install the plugin into Unreal, simply place that build WwiseSoundEngine folder in the `Plugins\Marketplace` folder of your engine.
 - If the folder `Marketplace` does not exist yet, create it.

Now the second plugin:

```
.\RunUAT.bat BuildPlugin -plugin="C:\PathToProject\Plugins\Wwise\Wwise.uplugin"  
-package="C:\Temp\Wwise" -TargetPlatforms=Win64
```

Go grab two coffees.

Wwise as an engine plugin

Once done, copy the resulting `Wwise` folder also to the `Plugins\Marketplace` folder.

If you want the WwiseNiagara plugin too repeat the steps we did before but now with `WwiseNiagara` instead of `Wwise`.

Zip the contents of the Marketplace folder and you can distribute in your team.

Finally, you can now remove the Plugins folder from your project, since we don't need the local plugin anymore. All people working on the same project need to install the plugin though.

And **done**, you don't need to submit a 5GB folder full of files via source control, but you'll have to place a zip somewhere in a shared folder.