

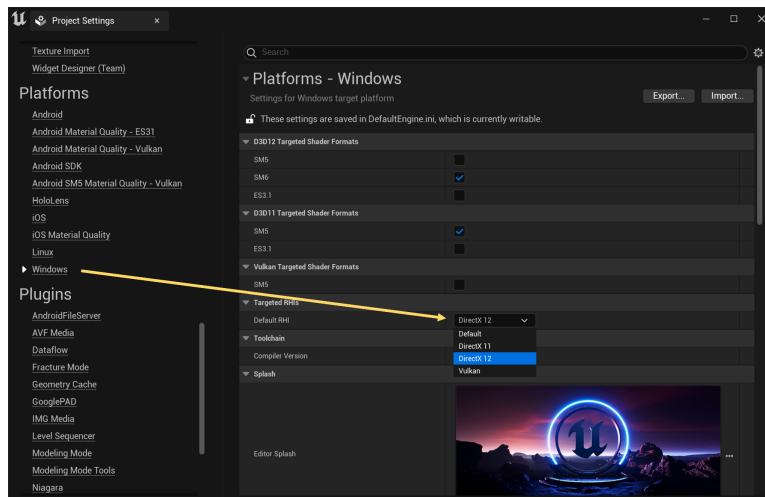


80 Level 3D Metasites Build Guidelines:

You would need to prepare the game build for upload, adhering to the following guidelines. In addition, the upload process is automated so ensure the folders are structured as described.

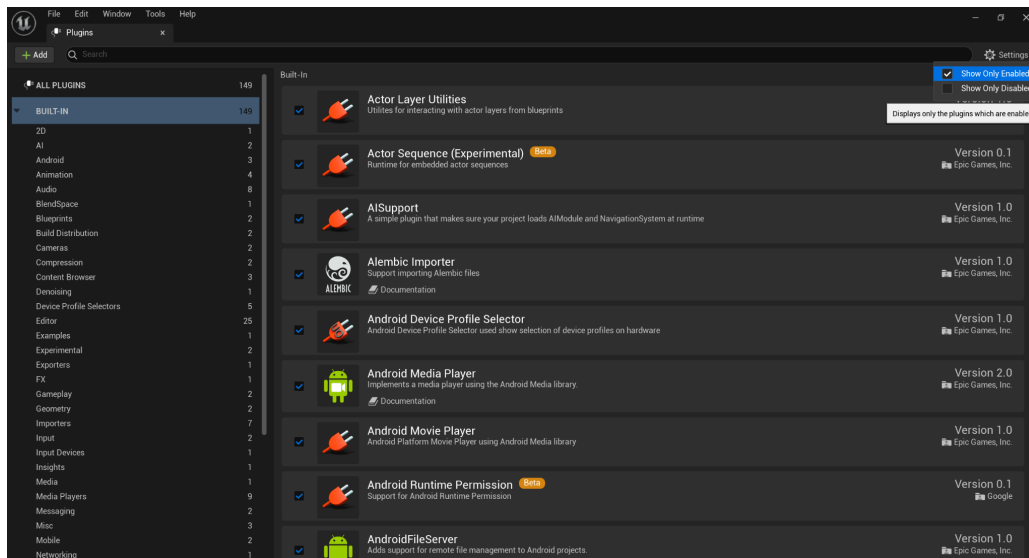
Build:

1. **.EXE build for 64-bit Windows**, supporting full screen mode. Ensure it's a **"Shipping"** build (and not "Development" etc.)
 - a. **How to create an .exe file:**
https://www.youtube.com/watch?v=BLXhZTK4HkY&ab_channel=UnrealEngine5Tutorials
 - b. **More settings:**
<https://docs.unreal-engine.com/5.1/en-US/packaging-unreal-engine-projects/>
 - c. **Windows SDK errors (if any):**
https://www.youtube.com/watch?v=EoldyhjW9e0&ab_channel=UnrealEngine5Tutorials
2. Game executable size must be less than or equal to **150 GB**. (See **Optimization Tips** below)
3. Projects must be in Unreal Engine **4.26 - 5.1** only.
4. Scene should contain the **"Player Start"** component, if users are supposed to walk around in the environment.
5. Scene should contain appropriate **collision boxes/invisible walls** to contain users and avoid clipping through the environment. (This is especially important if the scene is a "fly-through" experience, with the risk of users flying too high and seeing gaps in the environment.)
6. In Edit > Project Settings > Windows > Targeted RHIs, ensure **DirectX 11** or **DirectX 12** is selected. (see image below)

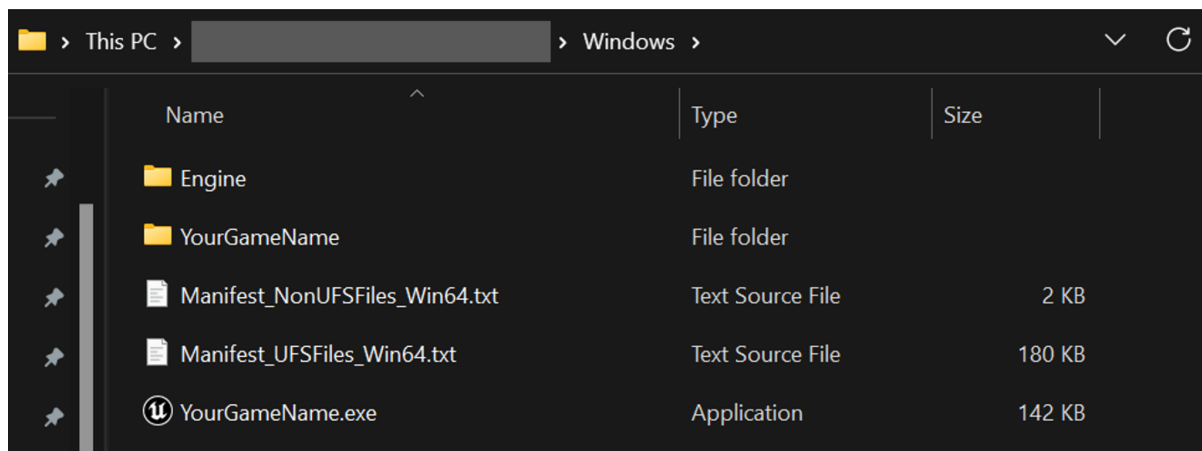


7. The build should include all third-party dependencies/plugins etc. that your project uses.
8. In Edit > Plugins, ensure you've **disabled any plugins** your project doesn't use. This will help make your build lighter. Look out for:
 - a. Android/VR/iOS related plugins
 - b. Networking related plugins (“network”, “online”, “TCP”, “UDP”)
 - c. Programming related integrations (“Visual Studio”, “CLion”, “Python”, “integration”)
 - d. Analytics related plugins
 - e. Go through the categories in the left sidebar.

If you disable any plugins, restart your project and create a build to **ensure everything still works well**. If something is broken, re-enable some plugin(s) you disabled and try again.



Once you've **tested your build** and are happy with it, please note that by default, the build .EXE will be contained in a folder named “Windows”. Example:



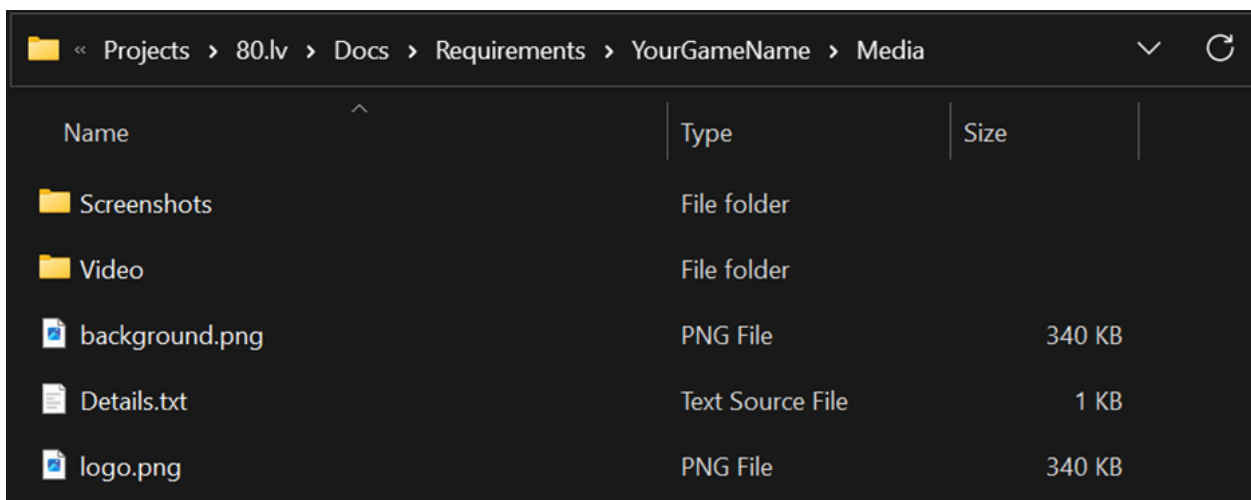
Rename this folder from “Windows” to “Build”.

Info about artist and project:

NOTE: All images here should be png or jpg, or they will get ignored by the upload process.

1. In a file named **"Details.txt"**, the title and description of the project, the artist's name and website.
2. Artist logo image - 100x100 png or jpg
3. 4-6 screenshots - 1920x1080 (16:9, png or jpg). Put them in a folder named **"Screenshots"**.
4. Page background image - 3840x2160 (16:9, png or jpg)
5. (Optional) Video - 16:9 webm. Put it in a folder named **"Video"**.

All these files and folders should be inside a folder named **"Media"**. Example:

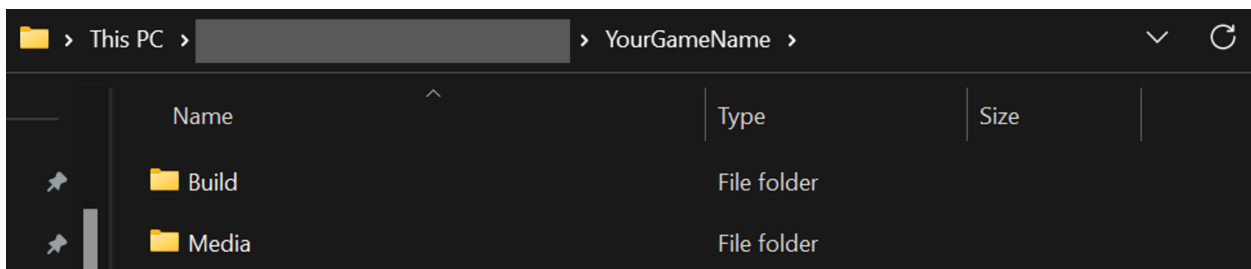


Final folder:

At the end, the final shareable zip should contain 2 folders:

1. **"Build"** folder
2. **"Media"** folder

Zip this entire folder and prepare a download link.



Upload process:

Phew! Now you are ready to upload!

1. Head over to <https://80.lv/cloud-experience>
2. Fill out the form.
3. Sit back and relax, we'll get back to you!

Step 3

Sign up for an 80 Level creator account by completing the form below

First Name	Last Name
<input type="text"/>	<input type="text"/>
Experience Title	Email Address
<input type="text"/>	<input type="text"/>
Build Link	Media Kit* Link to .zip
<input type="text"/>	<input type="text"/>
Experience Description	
<input type="text"/>	

insert the same zip download link in both boxes

Legal Agreement:

Once we receive your submission, you will receive a legal agreement from us. You need to sign it and send it to us so we can begin uploading your work to the cloud.

Optimization tips:

It is important to keep the executable size as small as possible to crank out the maximum performance from the cloud. Bonus, it helps you quickly iterate builds and iron out issues.

1. Here are some useful recommendations from Unreal staff itself:
<https://dev.epicgames.com/community/learning/courses/eER/unreal-engine-technical-guide-to-linear-content-creation-production/4G4r/unreal-engine-optimizations>
2. The “Size Map” and “Reference Viewer” utilities can help you quickly reach and identify large assets and modify them, or remove them entirely if not being used.
 - a. Size Map:
<https://dev.epicgames.com/community/learning/courses/zRn/working-with-assets-in-unreal-engine/r4y7/unreal-engine-size-map>
 - b. Reference Viewer:
<https://dev.epicgames.com/community/learning/courses/zRn/working-with-assets-in-unreal-engine/b8Ln/unreal-engine-reference-viewer>
 - c. Note: Packaging your game does already remove assets not being used in your level. However, these tools are still a good way to get an early warning of large and potentially useless assets.
3. Another way to reduce the build size is to “migrate” your final levels to a blank project. This way, assets used only by old, unused levels get stripped out.
<https://forums.unrealengine.com/t/how-to-include-only-used-assets-in-your-package/278247>
4. (ADVANCED ONLY) Unreal boasts of a huge array of tools to help us monitor (profile) the performance of our project. The full list can be found here:
<https://docs.unrealengine.com/4.27/en-US/TestingAndOptimization/PerformanceAndProfiling/>

Thank you from the team at 80 Level.

You can reach out to dev@80.lv if you have questions.

