To package your Unreal Engine environment into a standalone executable (.exe file), follow these steps:

**Step 1: Prepare Your Project for Packaging**

1. **Save All Assets:**
   * Make sure all your assets, levels, and blueprints are saved and built.
2. **Build the Lighting:**
   * Before packaging, go to Build > Build Lighting Only to ensure all lighting data is baked and ready for the final build.
3. **Check Project Settings:**
   * **Project Name:** Ensure your project name is correct under Edit > Project Settings > Description.
   * **Maps & Modes:** Set your desired default map in Edit > Project Settings > Maps & Modes. This will ensure that the correct level loads when the .exe starts.
   * **Packaging Settings:** Under Edit > Project Settings > Packaging:
     + **Build Configuration:** Set to Shipping for the final release build.
     + **Build:** Make sure Full Rebuild and Use Pak File are checked.
     + **Include Prerequisites:** Check Include Prerequisites Installer to include the necessary runtime files for your application to run on other machines.
     + **Exclude Editor Content:** In Packaging, uncheck Include Editor Content in Package to reduce the size of the final build.

**Step 2: Package the Project into an Executable**

1. **Open the Project:**
   * Open your project in Unreal Engine.
2. **Package the Project:**
   * Click on File > Package Project > Windows (64-bit) (or select another platform if needed).
3. **Select Output Folder:**
   * When prompted, choose a location where the packaged project will be saved.
4. **Wait for the Packaging Process to Complete:**
   * The process might take a while depending on the size of the project and the complexity of the assets.
   * Monitor the Output Log for any errors or warnings during packaging. If there are critical issues, resolve them before attempting to package again.

**Step 3: Locate the Packaged Executable**

* After the packaging process is complete, navigate to the output folder you selected.
* Inside this folder, you will find a WindowsNoEditor directory (or a similarly named directory based on the selected platform).
* Inside this directory, locate the .exe file named after your project. For example, if your project is named MyProject, the executable will be MyProject.exe.

**Step 4: Distribute the Executable**

* The packaged folder (WindowsNoEditor or similar) contains all the necessary files for your Unreal Engine environment to run.
* You can now distribute this folder to others. To run the environment, simply double-click the .exe file.

**Additional Tips**

* **Testing:** Always test the packaged executable on different machines to ensure it runs correctly and all required assets are included.
* **Error Troubleshooting:** If packaging fails, check the Output Log for errors and warnings. Some common issues include missing files, incorrect project settings, or missing plugins.
* **Reducing File Size:** For smaller packages, remove unnecessary assets or lower the quality settings in the Project Settings.

By following these steps, you can successfully wrap your Unreal Engine environment into a standalone .exe file for easy distribution and use.

4o

**You said:**

Build the Lighting: Before packaging, go to Build > Build Lighting Only to ensure all lighting data is baked and ready for the final build. this option is not there

**ChatGPT said:**

ChatGPT

If the Build Lighting Only option is not available, it might be due to a few possible reasons. Here are some steps and troubleshooting tips to ensure you can build lighting before packaging your Unreal Engine project:

**1. Check the Level and Modes Toolbar**

* The Build button is typically located on the top toolbar of the Unreal Editor. If you can't find Build > Build Lighting Only:
  + Make sure you're in the **default viewport** and not in another editor mode (e.g., Material Editor, Blueprint Editor).
  + Look for the **"Build"** button on the toolbar in the main editor window. Clicking this will show a drop-down with options like Build All Levels, Build Lighting Only, etc.

**2. Open the Build Menu in the Level Editor**

* If the Build menu option is missing, try this:
  + Navigate to **Window > Developer Tools** in the Unreal Editor menu.
  + Ensure that **Build and Submit** is checked in the toolbar. This may add the Build button back to your toolbar.

**3. Lighting Build Options in the Level**

* If you still can't find the Build Lighting Only option, use the following workaround:
  + Go to **Window > World Settings** to open the World Settings panel.
  + In the **Lightmass** settings section, you can find various lighting build options such as:
    - **Force No Precomputed Lighting**: Ensure this is **unchecked** if you want to build static lighting.
    - Adjust other settings as needed to optimize your lighting build.

**4. Use the Lighting Build Shortcut**

* If the Build Lighting Only option is not available in the UI, you can still try using a keyboard shortcut:
  + Press **Ctrl + Shift + .** to build lighting only. This is the default shortcut for building just the lighting in the current level.

**5. Check Project Settings**

* Ensure that **Static Lighting** is enabled in the project settings:
  + Go to **Edit > Project Settings > Rendering**.
  + Under the **Lighting** section, ensure that **Allow Static Lighting** is enabled. If this is disabled, the editor might hide the Build Lighting Only option.

**6. Ensure There Are Static Lighting Actors**

* If your level does not contain any **Static** or **Stationary** lights (like point lights, spotlights, or directional lights with baked shadows), the Build Lighting Only option might not appear because there is no static lighting to bake.

**7. Build All Levels**

* If you cannot find a way to build lighting only, you can use Build All Levels as an alternative:
  + Click the Build button in the toolbar and choose Build All Levels. This will build geometry, lighting, and other level elements.

**Summary**

If the Build Lighting Only option is missing, ensure that:

* You're in the correct editor mode (Level Editor viewport).
* Static lighting is enabled in the project settings.
* You have static or stationary lights in the level.
* Use the keyboard shortcut Ctrl + Shift + . as an alternative.

If you still cannot find the option, using the Build All Levels option is a reliable fallback before packaging your project.

4o

**You said:**

how to get this option "Exclude Editor Content: In Packaging, uncheck Include Editor Content in Package to reduce the size of the final build."

**ChatGPT said:**

ChatGPT

To **exclude editor content** from your packaged build in Unreal Engine and reduce the final size of your .exe file, follow these steps:

**Step 1: Open Project Settings**

1. In the Unreal Engine editor, go to the top menu bar and click on **Edit**.
2. From the dropdown menu, select **Project Settings**.

**Step 2: Navigate to Packaging Settings**

1. In the **Project Settings** window, scroll down the left panel and find the **Platforms** section.
2. Under **Platforms**, select **Windows** (or the platform you are packaging for), and then choose **Packaging**.

**Step 3: Find the "Include Editor Content" Option**

1. In the **Packaging** settings, look for an option called **Cook Only Maps** and **Additional Non-Asset Directories to Copy**. These settings control the content included in the final packaged game.
2. Look for **Include Editor Content in Package** (or similar wording depending on your Unreal Engine version):
   * Uncheck **Include Editor Content in Package**. This will ensure that only the necessary game content is packaged, excluding any editor-related assets, tools, and files.

**Step 4: Additional Optimization Settings**

While you're in the Packaging settings, consider adjusting the following options for further size reduction:

1. **Use Pak File:** Enable this option to pack your assets into a single file, which helps in reducing the overall package size.
2. **Full Rebuild:** Enable this option to ensure a fresh build, which can help clean up unused assets.
3. **Exclude Unused Assets:** In the **Project Settings > Packaging** section, you can enable **Exclude Unused Assets** to further reduce the package size.
4. **Cook Only Maps:** Use this option to specify only the maps (levels) that you want to include in the build.

**Summary of Changes**

* **Uncheck Include Editor Content in Package:** This option is found under **Project Settings > Packaging** and helps exclude unnecessary content from the final build.
* **Optimize Packaging:** Consider other options such as **Cook Only Maps** and **Use Pak File** for further optimization.

By unchecking the **Include Editor Content** option and adjusting the packaging settings, you'll be able to reduce the size of your packaged project.

4o

**You said:**

Here is the screenshot for project settings . help me run this step "Exclude Editor Content: In Packaging, uncheck Include Editor Content in Package to reduce the size of the final build."

**ChatGPT said:**

ChatGPT

In Unreal Engine's recent versions, the **"Include Editor Content in Package"** option has been either renamed or adjusted to focus on excluding unnecessary content automatically. However, the option to exclude editor content now largely relies on other settings that control which assets are packaged. Here's how you can achieve a smaller build size without explicitly finding the older "Include Editor Content in Package" checkbox:

**Steps to Exclude Unnecessary Editor Content**

1. **Use "Cook Only Maps":**
   * In the **Project Settings** window, go to **Packaging**.
   * Look for the **"Cook Only Maps"** option. Enable this option.
   * This setting ensures that only the specified maps (levels) and their directly referenced assets are packaged, which helps exclude editor-related content.
2. **Exclude Unused Assets:**
   * In **Packaging** settings, find and enable **"List of maps to include in a packaged build"**.
   * Click the **"+"** button to add the maps that you want to include. This helps Unreal determine which assets to cook and package, excluding any unused or editor-only content.
3. **Adjust Packaging Settings:**
   * In the Packaging section:
     + **Use Pak File:** Ensure this is checked. It compresses the package into a single .pak file, reducing the build size.
     + **Full Rebuild:** Check this option to ensure the packaging process uses a clean build.
     + **For Distribution:** Enable this to create a distribution-ready build that doesn't include extra debugging symbols or editor content.
4. **Disable Debug Files:**
   * Make sure **"Include Debug Files"** is unchecked. Debug files are used for debugging purposes and are not needed in the final build.
5. **Blueprint Nativization (Optional):**
   * Set **"Blueprint Nativization Method"** to **"Disabled"** if you want to skip converting Blueprints to native code, which can reduce build size. However, this might affect performance.
6. **Use Additional Command-Line Options (Optional):**
   * When packaging the project, you can use command-line options in the **Advanced Packaging Settings** to further control content inclusion.

**Summary**

* The exact option to "Include Editor Content in Package" may not be present in the newer versions of Unreal Engine. However, you can achieve a similar effect by using options like "Cook Only Maps," listing specific maps to include, and adjusting other packaging settings.
* By following these steps, you should be able to reduce the size of the final build, effectively excluding unnecessary editor content.

4o