

# EASY RENDER FARM

## USER GUIDE

# Content Guide

REQUIREMENTS .....	4
Unreal.....	4
Network settings.....	4
AWS Thinkbox Deadline .....	4
Important Configuration on AWS Thinkbox Deadline.....	4
EASY RENDER FARM INSTALLATION.....	6
ENABLING PLUGIN .....	6
CONFIGURATION .....	8
Creating config file.....	8
Edit global config .....	8
Edit local config overrides .....	8
Configuration File .....	9
Local Folder Settings.....	9
Remote Folder Settings .....	10
General Settings .....	11
Debug.....	11
SENDING RENDER JOBS .....	12
Render Output Folder .....	13
JOB Subfolders (How it works) .....	13
Path Replacement Example .....	14
{Format_Strings} Special Cases .....	15
Deadline Group .....	15
Priority .....	15
Frames per Task.....	15
Everything is ready!! .....	15
What's included into Job Package .....	15
RESTICTIONS .....	17
Rendering.....	17
Rendering World Partition not available yet.....	17
Physics simulations and particles are rendered differently on every Render Node	17

TROUBLESHOOTING..... 18  
Bad Render Result ..... 18

# REQUIREMENTS

## Unreal

Unreal Engine 5.0 or later

Unreal Movie Render Queue Plugin activated

## Network settings

All the render nodes need access to a shared folder on your network to share config and jobs. The network drive must be **mounted as a unit** like in windows R: or X:

**IMPORTANT:** The plugin doesn't allow direct network paths (UNC) by now

## AWS Thinkbox Deadline

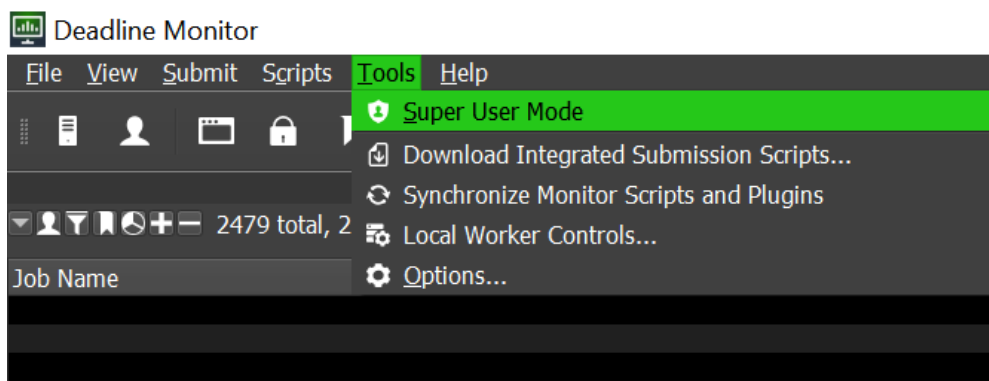
- AWS Thinkbox Deadline as repository installed in a Workstation
- AWS Thinkbox Deadline client installed on every Render Node

Please refer to [AWS Thinkbox Deadline Docs](#) to know more about installation instructions

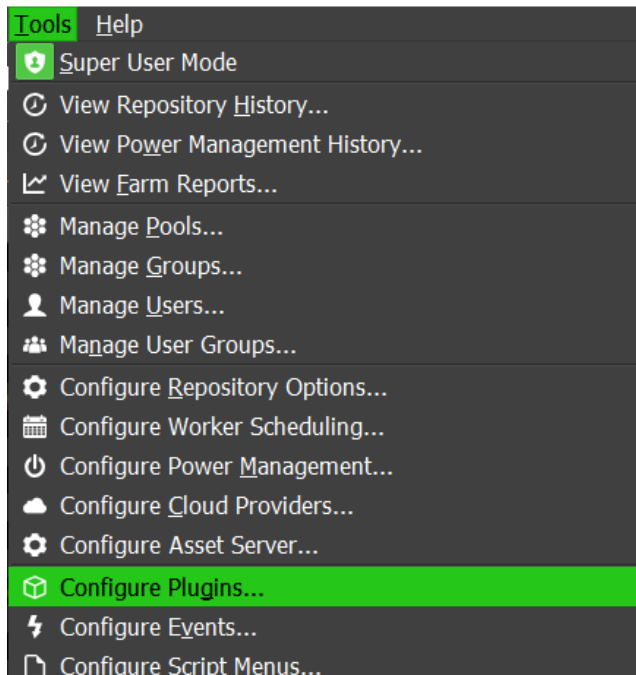
## Important Configuration on AWS Thinkbox Deadline

Please check your python executable is correctly configured into AWS Thinkbox Deadline:

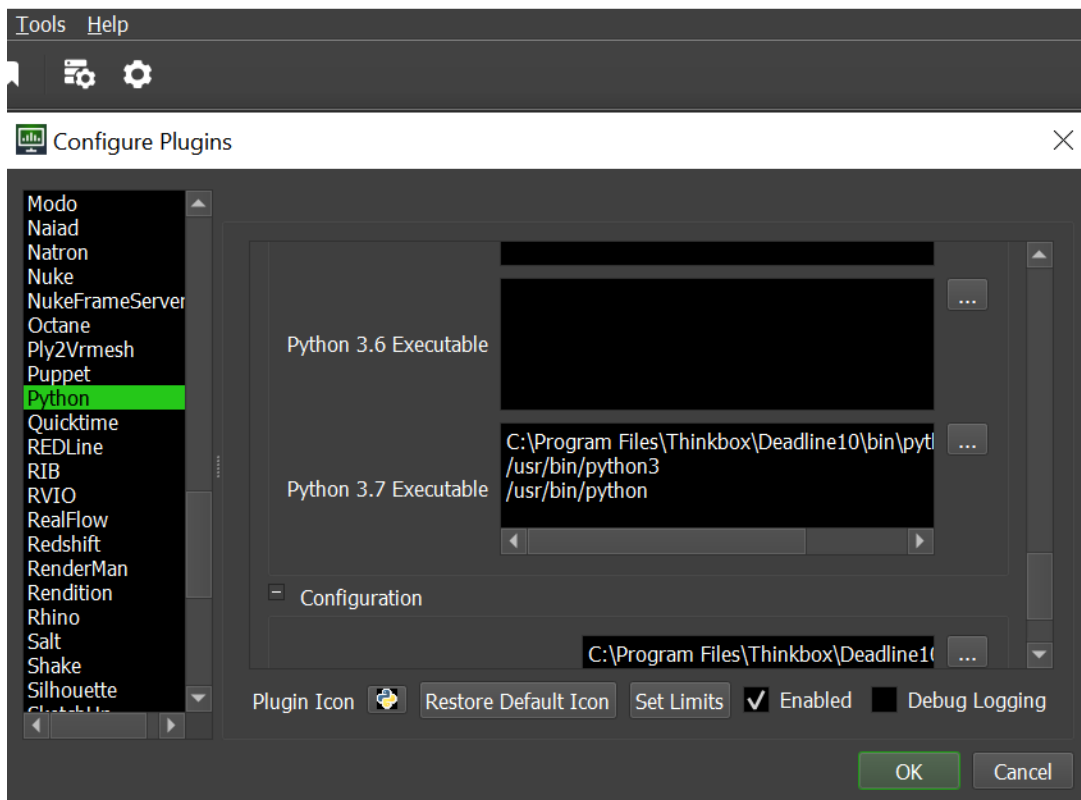
- 1- Enter in Super User mode



## 2- Go to Tools → Configure Plugins

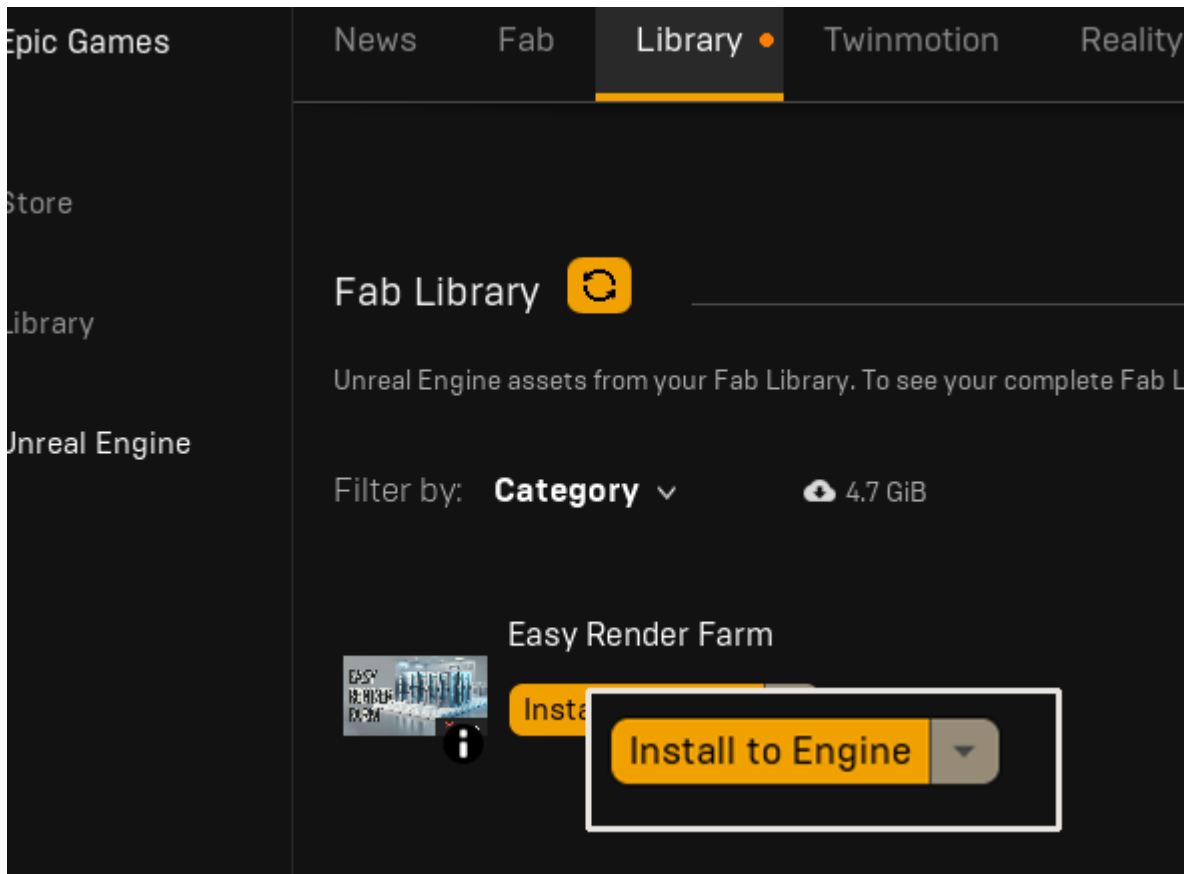


## 3- Look for Python into the left list, check the “Python 3.7 Executable” on the right areas and add the python installation path if missing



## EASY RENDER FARM INSTALLATION

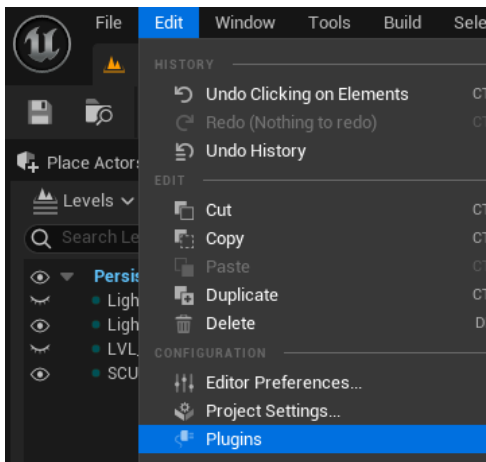
Installing EASY RENDER FARM using the “Install to Engine” button, placed in the Library Area in Epic Launcher and select the Unreal Engine version you want to install for



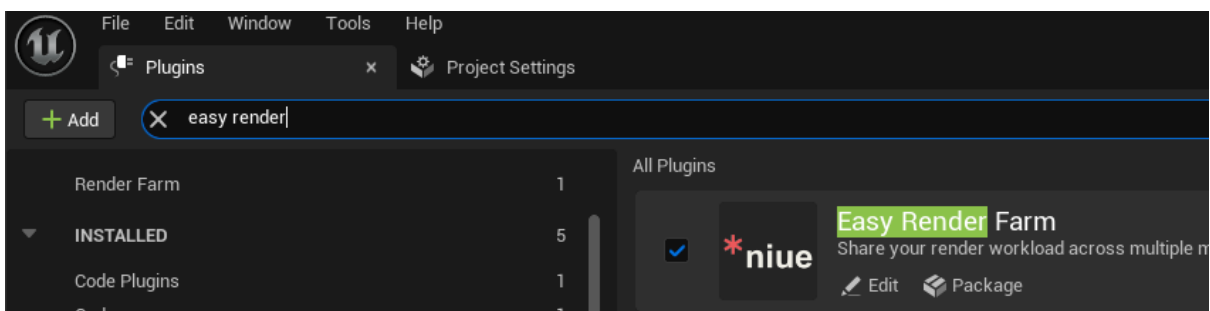
## ENABLING PLUGIN

Using EASY RENDER FARM into your project requires to activate, “Easy Render Farm” and “Movie Render Queue”

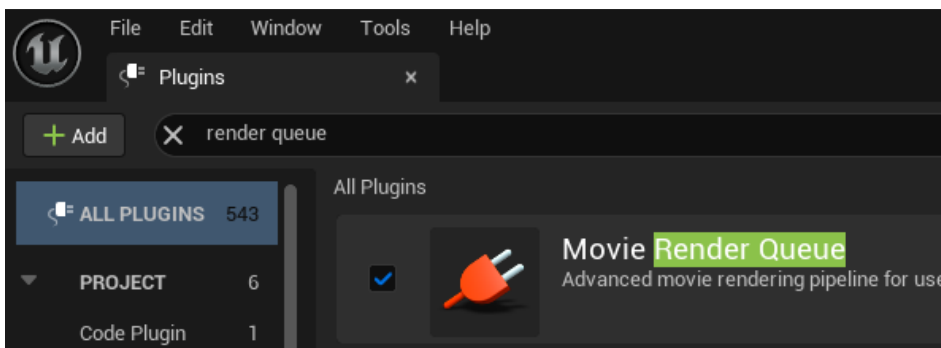
Enter in the Unreal Editor Plugins window and activate



### EASY RENDER FARM Plugin



### Movie Render Queue



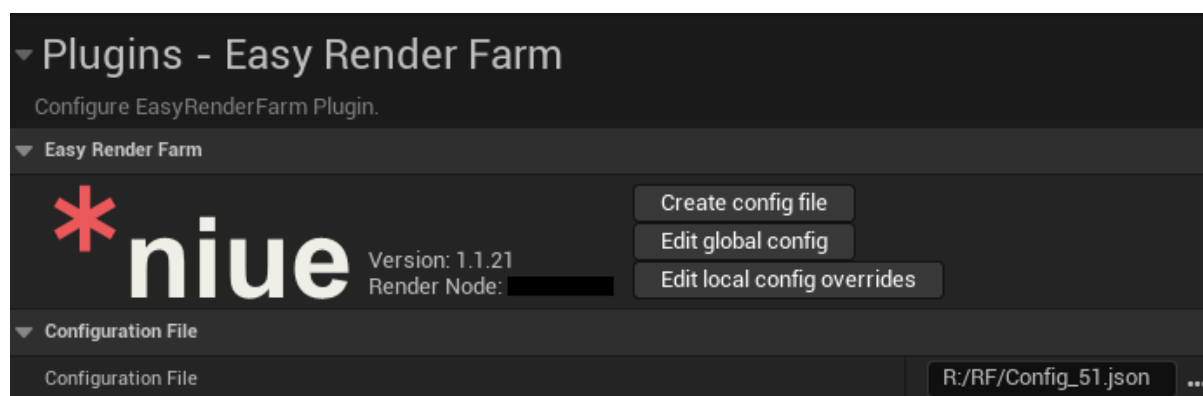
**Important:** Restart the editor after activating both plugins

## CONFIGURATION

Go to “Project Settings” and search Easy Render Farm into the Plugins Area.

### Creating config file

The first time you configure the plugin you will need to create a new configuration file by clicking on “**Create config file**” Button



Choose a path you want to save the global file. Be sure to save it in a folder where all the Render Nodes have access. Once you create it you can use this file on your other render nodes

After choosing the path you will be able to edit this global configuration from any Render Node or create a specific local configuration for a Render Node

### Edit global config

If you want to edit the global configuration of the Render Nodes, click on “**Edit global config**” button, it will show you the same window you use after choosing the path using the “**Create Config file**” button

### Edit local config overrides

Not every Render Node has to be configured the same way as others, or maybe you want to use a refurbished workstation to be part of your render farm, for these cases we must use “**Edit local config overrides**” configuration button

Local configuration is a “per machine” override of the global configuration values just for the Render Node on which you are editing the file

## Configuration File

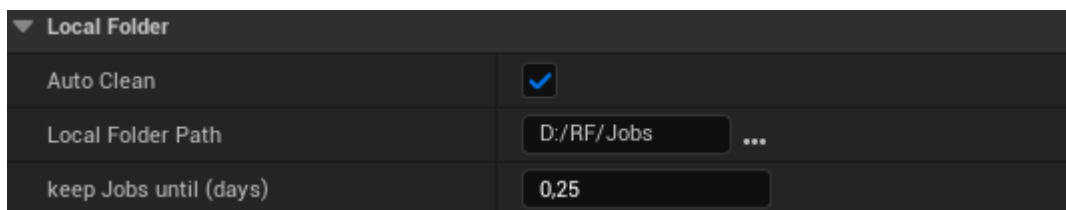


**Config File Path:** Full path of current the config file.

Config file must be saved in a shared folder of the network to share the settings between render nodes

**It is recommended that all machines have the same Unreal Engine installation path** (and the same version installed). If not so, please check per-machine configuration on this document

## Local Folder Settings



Configuration for your Local Machine temp folder to store incoming render jobs files

- **Auto Clean (Local):** Activate to clean Local Folder deleting Jobs prior to "keep Jobs until" field
- **Local Folder Path:** Path to the local folder where the Jobs will be copied locally to be rendered
- **keep Jobs until (days):** Number of days the jobs will be archived in your Local Folder

## Remote Folder Settings

Remote Folder	
Remote Network Folder	R:/RF ...
Auto Clean	<input checked="" type="checkbox"/>
keep Jobs until (days)	0,25



Area of Network folder to store render jobs to share between Render Nodes

- **Remote Network Folder:** Path to the shared Remote Folder to distribute Job files. All Render Nodes need access to this folder in the network.

**IMPORTANT:** By now, the plugin needs the network folder mounted as a unit on your machine as “Remote network folder”

- **Auto Clean:** Activate to clean Remote Folder deleting Jobs prior to "keep Jobs until" field
- **keep Jobs until (days):** Number of days the jobs will be archived in your remote folder

## General Settings

General Settings	
Default Deadline Group	ue
Full Path to 7Zip exe	C:/Program Files/7-Zip/7z.exe ...
Full Path to Deadline exe	C:/Program Files/Thinkbox/Deadline10/bin/deadlir ...
Deadline Python Version	3.7
Full Path to Unreal Engine exe	D:/Epic Games/UE_5.3/Engine/Binaries/Win64/Unr ...
Unreal launch Parameters	11 Array elements  

- **Default Deadline Group:** Name of the Deadline Render Group you wish to send your render to.
- **Full Path to 7Zip, Deadline, and Unreal Engine exe:** Full path to the executable of each of these programs needed to run the EASY RENDER FARM plugin
- **Deadline Python Version:** Number of Python version (format #.#) from the Deadline Manager --> Tools --> Configure Plugins --> Python
- **Unreal Console Parameters:** List of command parameters used on launching UE in Render Nodes

## Debug

Debug	
Copy Every UE Log	<input checked="" type="checkbox"/>

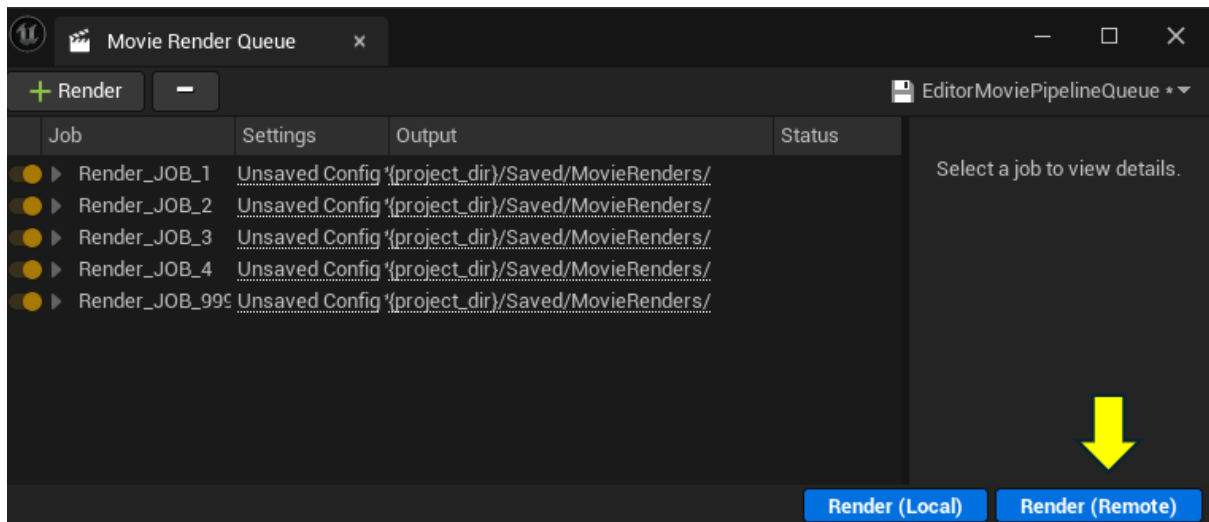
- **Copy Every UE Log:** If true, Render Job will store a log file in the Remote Folder, in Logs subfolder

Press **'Save'** to generate a JSON file with your selected configuration.

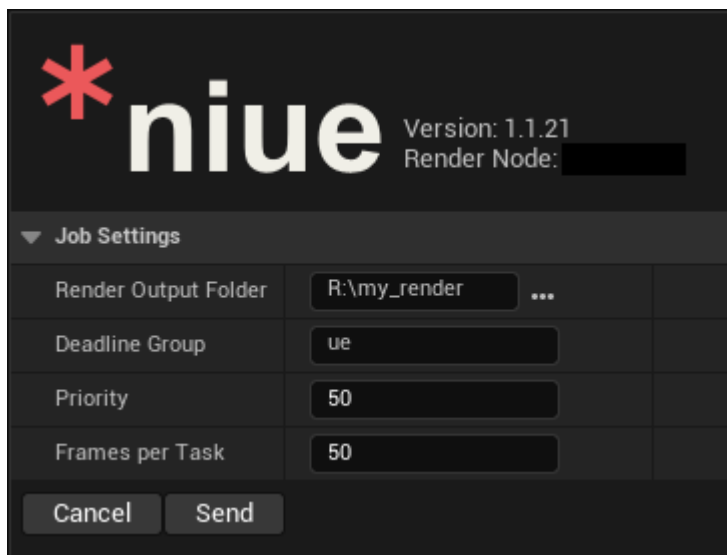
## SENDING RENDER JOBS

Sending Jobs using Easy Render Farm maintains the same workflow as rendering local frames using Render Queue regarding its settings in the configuration.

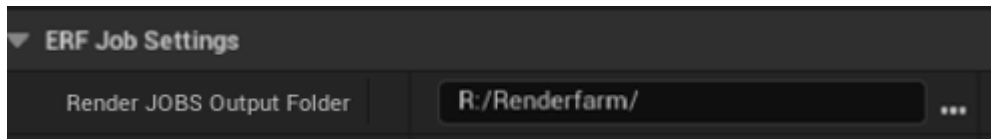
When your render jobs are configured, just click on “Render Remote” button on the Movie Render Queue



A new window will appear to define your desired Render Job Setting



## Render Output Folder

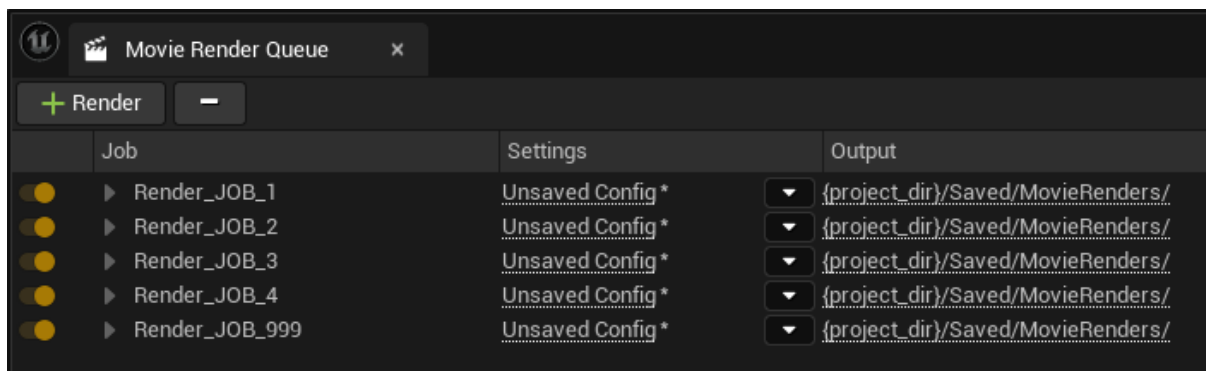


The folder where all the rendered files will be placed. All the render Nodes must have access to this folder over the network.

The Job Output Folder will be used as partial override of the path set in Movie Render Queue settings, allowing you to maintain the UE dynamic folder structure based on {Format\_Strings} but in a different Drive and folders. Please read carefully next section for a better understanding

## JOB Subfolders (How it works)

As you may know, you can create as many render Jobs as needed into the Movie Render Queue



*Screenshot from Movie Render Queue job list*

You can also set a different output folder structure for each of them into the Job Configuration, “Render Output Folder” field sets the base path on with all the shot will be stored using the “Output Directory” configuration field of each job

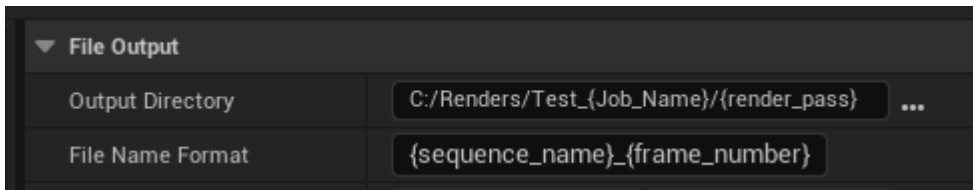
The Job Output Directory path will be replaced until the path has Unreal {Format\_Strings} and it will be joined to the Render JOBS Output Folder

This will ensure all the Jobs you send in a try will be stored in the same place but following their individual naming rules

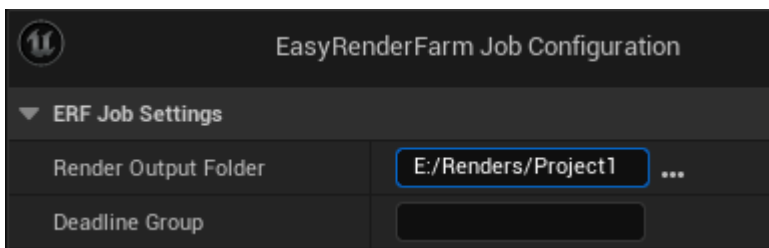
A typical example for this behavior could be a Render Passes workflow

### Path Replacement Example

**MRQ → Output directory = C:/Renders/Test\_{Job\_Name}/{render\_pass}**



**Easy Render Farm → Render JOBS Output Folder = E:/Renders/Project1**



Sending the job will replace the MRQ Output Directory path until first {String\_format} and Render JOBS Output Folder will be this way:

**“C:/Renders” will be replaced by “E:/Renders/Project1”**

The rest of the MRQ Output Directory will be used as subfolders

**“Test\_{Job\_Name}/{render\_pass}”**

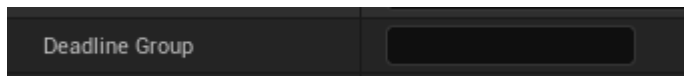
Giving the final JOB path **“E:/Renders/Project1/Test\_SQ01/Beauty”**

## *{Format\_Strings} Special Cases*

**{project\_dir}** → Instead of the Project Directory Path we use the name of the project to avoid local Path errors

**{time}** → Distribute rendering will use the Time you send the Jobs to the Render Farm to maintain the frame name coherence no matter the moment of the frame packages were render

### Deadline Group

A dark-themed UI element consisting of a label 'Deadline Group' on the left and a text input field on the right.

Name of the Deadline Group to render the Job, if empty the default Deadline Group of the Configuration file will be used

### Priority

A dark-themed UI element consisting of a label 'Priority' on the left and a text input field on the right containing the number '50'.

Priority of the render job inside the Deadline manager. Higher priority number will render before render jobs with lower number

### Frames per Task

A dark-themed UI element consisting of a label 'Frames per Task' on the left and a text input field on the right containing the number '50'.

Render Jobs will be split in smaller packages to distribute the Job to Render Nodes in separate pieces. Frames per Task defines the number of frames within every package

Everything is ready!!

Click on “**Send**” button to create and send the Render job to Deadline

Wait until Easy Render Farm finishes packaging your project and afterwards check the Deadline Monitor to view your render progress

## What’s included into Job Package

Easy Render Farm will pack these elements and send them to the Render Farm:

- Level you defined into Render Queue Render Job and the assets referenced into the scene, including the Project Plugins used
- Sequencer selected in the Render Queue Render Job
- Project configuration

Un-referenced assets will not be included because package size and performance

## RESTICTIONS

### Rendering

Rendering World Partition not available yet

At the moment, World Partition Levels are not able to be rendered over Easy Render Farm

Physics simulations and particles are rendered differently on every Render Node

Distributing render packages on different machines means different render processes start, that's why you will need a **deterministic** functions or values to ensure the same calculations at the same time on every machine. Another way to solve it is to bake the animations before sending the render

Using World Position Offset systems like wind or materials using Time Node will make different rendering results on different machines.

In case of Materials based on Time like node, you can use a Parameter Collection driven by your Level Sequence along the timeline. E.g. setting keyframes at 0 and 20s and using those seconds as its value.

If you are using any procedural actors try to use a fixed Seed to be sure you are not getting a random seed every time you render the sequence

## TROUBLESHOOTING

### Bad Render Result

To ensure a successful rendering process there are checks you can do before sending the job to the Render Farm

- 1- Use Play to check all required levels are loaded
- 2- Make a preview playing your sequencer with the Camera Cut activated

If you find everything right, you can download the Job Package File previously sent to the server, uncompress it and check how it works as a normal Unreal project. You will find what's not correctly configured or missing in your project