

NearTime Manual - 5.3.2.3

1. Introduction

Welcome to the NearTime Manual. In this document, we will be discussing how to prepare your project with NearTime and interacting with its webservice. For any questions about NearTime, please email support.mo-sys.com.

1.1 Prepare your Project for NearTime

The Unreal Engine is a highly flexible tool that gives the scene designer the option to alter thousands of settings to produce the perfect environment as well as providing the tools to create additional custom settings. As a result, it is not feasible for NearTime to manipulate the scene beyond configuring the desired render output options.

In order to ensure NearTime will produce the expected render of your scene, there are a few requirements and checks you can perform.

NearTime will render the selected level sequence against a specified level map. This map must open in a state that is appropriate for rendering. Common issues include: - Needing to update lighting depending on shot - e.g. changing light intensity/direction to simulate different times of day. - Having scene actors that should be hidden in some shots. - Having scene actors in different states in different shots - e.g. a computer monitor on/off.

We recommend placing actors that need to change into sublevels, one for each of their different states. Then, creating a master level for each required combination that loads your main content level plus the appropriate set of sub levels for your variable content.

Each required sub level will need to have its streaming type set to always loaded rather than blueprint. This ensures the level is correctly loaded when rendering outside of the editor.

It is also important to ensure that each frame of your level sequence will render the same. This should be regardless of whether it is rendered alone or as the 37th frame of a sequence, for example. In particular, pay attention to any animations you have in the scene. If you have used Unreal's internal random number functions to introduce randomness to the scene - such as creating blinking lights on a console - this should work as intended. However, any custom work will need to be further tested.

It is not currently possible for level sequences submitted to NearTime to reference, for instance, animation data held in a companion Unreal asset (except under specific conditions). As a result, it is necessary that all information needed to render a take must be in a single level sequence file.

2. Test Your Render

We recommend testing your render by checking out the project from Perforce on a new machine. This is because it is how the project will be loaded into NearTime. This will help identify any missing files or broken references. For example, if a fully specified path has been used to reference an asset rather than a relative path, which is a common issue with OCIO colour profiles.

You can test the render output by using the Movie Render Queue to produce a Remote render. Just select Window > Cinematics > Movie Render Queue.

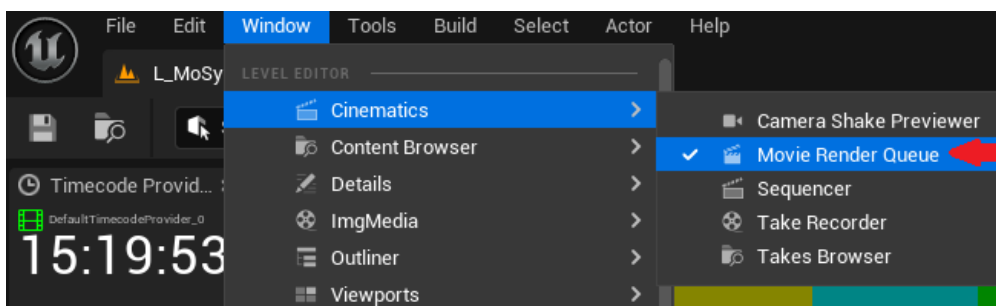


Figure 1. Movie Render Queue Location

Then, to produce it, add a sequence to your jobs and select Render (Remote).

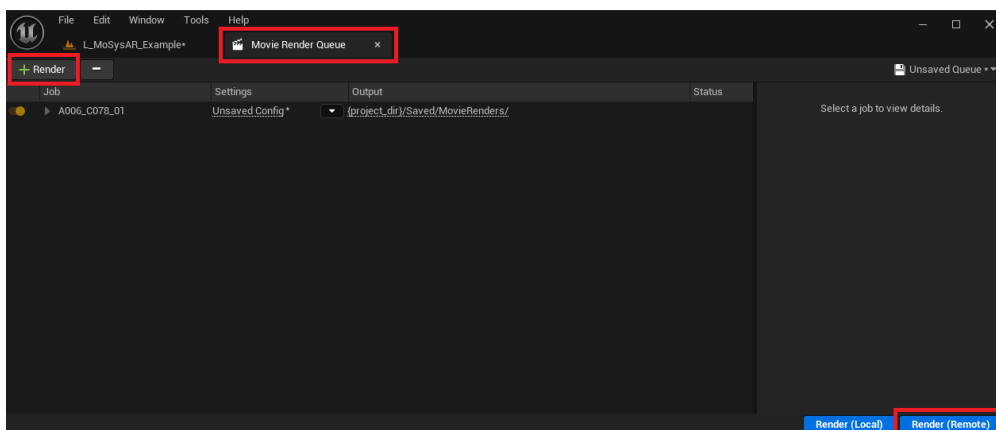


Figure 2. Render Local or Remote

This will start a separate Unreal Engine instance outside of the editor in a similar way to NearTime. Please be aware it is important to use the Remote option. Whilst in theory there should not be any differences between the output produced by Local or Remote renders, this has not always been the case in the past.

We suggest you render out several frames at full quality in one Movie Render Queue task to ensure things look as you expect. Once you are happy with the look, you should re-render each of those frames, one at a time and compare these to the original renders to ensure the output is consistent.

With all these steps successfully completed, the project should now be tested and the selected level map should work in NearTime, with no modifications necessary.