

# Preparing Scenes for NRC rendering

- [Introduction](#)
- [Render checklist](#)
  - [Assets](#)
  - [Simulation](#)
  - [Output](#)
- [UNC Paths](#)

## Introduction

Preparing a scene for distributed rendering requires a certain rigor that is not necessary when all assets are local to the machine rendering them. While Layout will complain if it can't find an asset or plugin and allow user interaction to find the right file, rendering on a network is not interactive. You must adhere to LightWave's content directory structure, and elements that use random numbers need to be fixed (baked) to ensure identical rendering across different machines. Also, since you will be relying on machines that are not the one you are sitting in front of, you need to be sure that not only is everything accessible across a network, but that it can be written to as well as read from. NRC is not going to ask if it can't find something, it will plow ahead assuming what you did was intentional.

## Render checklist

Go through the scene before you "send it to the farm" to be sure that:

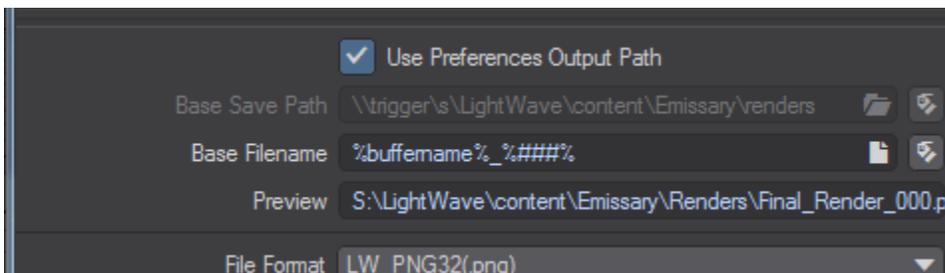
### 1. Assets

All assets used for scenes need to be accessible to all nodes rendering. At the simplest, this entails using the [Package Scene](#) tool to make sure all your scene assets are contained within a content directory, then moving that content directory to somewhere accessible to all nodes on the network.

### 2. Simulation

All simulations must be baked. By simulation, we are referring to anything in the scene that requires random numbers, including [radiosity solutions](#) and dynamics (Bullet, FX dynamics, including particles, and Flocking). To bake simulations, refer to the documentation for those sections and using MDDs or other Vertex Caching mechanisms.

### 3. Output



All output paths need to be correct. The output path should be manually defined to ensure correct operation for a network render. Don't leave it to the default **Use Preferences Output Path**. If you are rendering with a Mac-based farm, Unix-style paths will be used. If rendering on a mixed farm, Unix paths will be translated internally to UNC paths for the Windows machines in the farm.

Make sure that your output format is what you want at the end of the render - network rendering cannot write animations, you need to be sure that your scene length is correctly set and the right buffers are being saved to the paths indicated.

## UNC Paths

For Windows users, the best way to ensure network discoverability is to use UNC path conventions. Render networks using purely Windows machine *can* use drive mapping, but this is prone to failure if all drives are not mapped the same way. UNC paths are, like drive mapping, unique to Windows. However, NRC can translate UNC paths to Unix-style paths internally, meaning that they are ideal for cross-platform rendering. UNC paths are created starting with a double backslash, then the machine name, drive letter and directory, like this example:

```
\\machine\drive\directory\subdirectory\file.ext
```

A Windows machine will understand such a path implicitly. Mac OS X machines using NRC will see the following:

```
/Volumes/drive/directory/subdirectory/file.ext
```

This enables cross-platform communication between Windows and Mac OS X.