RenderFarmer ver 0.91

A stand alone utility which manages the distributed rendering of Lightwave scenes.
RenderFarmer is a improved control program for Lightwave's Screamernet system

Program Written and designed by

Glyn Williams – Design & Coding
Martyn Bramall Original Coding
and Mike Powell – Windows MFC interface.

What is RenderFarmer

RenderFarmer is a stand-alone program replacing the *ScreamerNet* multi-machine rendering system built into Lightwave itself.

While using the same Node (LWSN) software and control method as *ScreamerNet* - *RenderFarmer* offers a number of advantages over the implementation built into Lightwave itself, Namely:

- **Automatic operation.**
- **Better and more responsive user interface.**
- **A log file.**
- **Named nodes**
- **Restore on start-up.**
- **User intervention during render**
- **Automatic identification of bad scenes**
- **Automatic identification of absent frames**

Automatic operation.

One of the main draw-backs of the Lightwave Screamernet software was the lengthy set-up process. Renderfarmer allows any artist to setup a scene for rendering with zero work.: *RenderFarmer* uses a special directory which is the called the **Scene-Queue** directory. Any lightwave scene file (LWS) copied into this directory at any time will automatically be queued for rendering.

Better and more responsive user interface.

RenderFarmer offers the user a large clear control panel which allows the operator a considerable amount of feedback. The screen shows the state of each of the nodes (Rendering machines) on the farm – as well as showing the a detailled report on progress of all of the scene files in the queue.

The panel shows the common Lightwave settings which give rise to mistakes: Namely render resolution, segment size, anti-aliasing settings and motion blur settings.

A log file.

RenderFarmer maintains a log-file showing the status of all machines on the farm as well as the progress of the scenes. This is useful in allowing remote checking on the progress of the system.

Hot-Plugging and hot-removal of rendering nodes.

Once the *RenderFarmer* is running - new rendering nodes can be added at any time. *RenderFarmer* will automatically detect them and give them work to do. - Similarly, nodes on the farm can be shut down at any time - perhaps if they are needed as workstations. This can be done without dropping frames. This means any spare workstations can be used all the time - even making use of lunch-breaks and coffee breaks. Making best use of all your processing resources.

Note. When a rendering node is taken off-line by a user, *RenderFarmer* has no way of knowing that this has occurred. It mistakenly believes the node is still rendering. so the RenderFarmer operator should use the Make-Off-line function on the main panel to tell the *RenderFarmer* that the node is no longer available.

Named rendering nodes.

RenderFarmer allows you to give each individual rendering node a unique name. So instead of NODE1 you can have a meaningful name like TonysAlpha or GlynsP90.

Restore on startup.

If when you reload *RenderFarmer* - it picks up the same list of scene-files to render from last time. Even if you delete the save-file, Renderfarmer will build a scene queue automatically from the contents of the **Scene-Queue** Directory

Change settings during render

You can make changes to render order and even modify scene files during the rendering process. You can stop and start the program without losing frames or forcing anything to re-render.

Automatic identification of missing frames.

RenderFarmer checks every frame through the rendering process. It scans scenes for missing frames. This way - if you submit a scene file to be rendered - which has already been partially rendered - only the missing frames are produced. *RenderFarmer* only considers a scene file complete when all rendered frames have been seen and accounted for.

Automatic Identification of Bad-scene files

Sometimes scene files contain incorrect save information which could cause the rendering process to fail. Renderfarmer will automatically spot such scenes (Usually within 30 frames) and reject the scene.

Installing the program

RenderFarmer is a simple Win32 executable requiring no DLLs or other special extras. Nothing else is added to your system configuration. Installation requires only copying the executable into a suitable directory and running it. No other files are required.

However setting up a complete working distributed rendering network is beyond the scope of this program. To work properly Renderfarmer needs to operate in an environment set-up for distributed rendering. If you have already set up a network using Lightwave's ScreamerNet software, then using Renderfarmer will be straightforward. If you have not yet done this, then we would advise you to follow the instructions in the Lightwave user guide.

Basic principles

Renderfarmer does not render anything. It farms out individual frames to rendering nodes. A node is a PC running the ScreamerNet software LWSN.EXE.

LWSN.EXE is a version of Lightwave with all the user interface cut out. All it does is load and render frames – under the control of another program.

For this system to work – all the rendering nodes must have access to the same files and see the Lightwave data in the same way. This means the creation of at least one common network drive. This drive needs to host all your Lightwave data – including:

1) Lightwave Scene files. LWS Files
2) Lightwave object files
3) All image maps referred to by Lightwave objects or scene files
4) The destination for rendered files.
5) The Screamernet Control directory

If you are just starting to use network rendering, you may have to re-organise your files and plan to work in a network friendly manner.

This is a snapshot summary of all the files used on a typical network setup. Examples of those files marked in bold have been included in the Renderfarmer ZIP.

File Usage

Directory	**File**	**Notes**
R:\Scrmnet	LWSN.EXE	The Scrmnet directory should include all the files needed to support LWSN.EXE
R:\Scrmnet	**LW.CFG**	
R:\Scrmnet	ERBUFIO.DLL	
R:\Scrmnet	HIIPUNV.DLL	
R:\Scrmnet	LW_ERR.ENU	
R:\Scrmnet	ERCORE.DLL	
	(others	
R:\Scrmnet	**SnNode1.BAT**	A Batch file to set render node 1 running
R:\Scrmnet	**SnNode2.BAT**	A Batch file to set render node 2 running etc.
R:\Scrmnet\Command	(Dir)	This directory is where the node command files are written.
R:\Renderfarmer	**RenderFarmer.EXE**	The program itself
R:\Scenes\ToBeRendered	(Dir)	The Directory where pending scene files are stored
R:\Scenes\ToBeRendered	Example.lws	This scene file is queued to be rendered.
R:\Scenes\ToBeRendered	RenderFarmer.QUE	This file is used by Renderfarmer for housekeeping and preserving its state when the program exits.
R:\Scenes\ToBeRendered	**Logfile.TXT**	Renderfarmer's main Log file.
R:\Scenes\ToBeRendered\HasBeenRendered	(Dir)	The place where deleted scene files go when you delete them. (they are not really deleted)
C:\WinNT	**Renderfarmer.INI**	Configuration info for the Renderfarmer

Directories

The ScreamerNet command directory.

Renderfarmer (and Lightwave) send commands to the Screamernet Nodes in a very simple way. Renderfarmer writes small text command files in a common directory. The nodes respond by writing similar small files in the same directory. In our example set-up the command directory is R:\Scrmnet\Command

---

The SceneQueue directory.

In our example set-up this directory is named ToBeRendered. This is a good name for it. Any Lightwave scene file (LWS) saved here will automatically be placed at the end of the rendering queue. When an artist wants to see a file rendered, they can save their file normally and then save another copy in ToBeRendered. That is how easy queuing a file up. Alternatively someone can use explorer to drag a file into the directory. Any valid scene-file placed in this directory will get rendered.

You can also delete a scene file – and that will remove the file from the queue. One benefit of this model is that scene files can even be added from home over a remote link!

The HadBeenRendered directory

This is a sub-directory of the SceneQueue (ToBeRendered) directory. If you elect to remove a scene-file using the Renderfarmer software, the program will copy the file into this directory (rather than deleting it outright). Sometimes artists can be forgetful and place their only copy of the scene file in the SceneQueue. This way there should always be a copy.

The ScreamerNet Command directory

Renderfarmer works in just the same way as ScremerNet II in that it controls nodes by reading and writing tiny one-line command files held in a common directory. Not surprisingly RenderFarmer needs to know where this directory is.

ScreamerNet Command - directory for instance

```
R:\ScrmNet\Command
```

The Image Save Path

RenderFarmer actually looks for rendered images appearing on the disk to check on the progress of the Rendering process. For that reason - RenderFarmer needs to find out exactly where images are being rendered to.

The LWS scene files only contains the last part of the image save path. RenderFarmer needs the first part of the save path to complete

Finally RenderFarmer needs the path fragment needed to complete the save-path for images produced.- this is usually the same as LW.CFG content path.

For instance opening a LWS scene file with a text editor - the scene contains the save path line such as:

```
SaveRGBImagesPrefix MOVIES\Briefings\Mis02(RoutPat)\m2
```

this is only part of the path. If those images are written to

```
R:\MOVIES\Briefings\Mis02(RoutPat)\m2
```

then RenderFarmer needs the missing part of the path .(R:\) to complete the save path.

Important Files.

Renderfarmer.QUE file

This file holds temporary workspace for Renderfarmer. For instance, whether the program is currently running. If Renderfarmer crashes on loading for any reason or claims to be already running – then quit the program, delete this file and start again. The only data you'll lose is the scene queue order.

Renderfarmer.INI

This file holds the config information for Renderfarmer and lives in your Windows directory. It is a simple text file and is easy to change yourself.

```
[Scene_Queue_Dir]
PATH=R:\scenes\ToBeRendered
```

(This is the path to the scene queue directory – it has to be on a shared drive.

```
[Screamer_Command_Dir]
PATH=R:\Scrmnet\command
```

(This is the path to the usual Screamernet command directly. Renderfarmer needs to find it)

(NEW NEW NEW NEW NEW)

```
[Config_Dir]
PATH=R:\Scrmnet
```

(This is the location of the LW.CFG file that the Screamernet Node use)

```
[Installed_Nodes]
No_Of_Nodes=9
```

(This is obvious)

(NEW NEW NEW NEW NEW)

```
[Free_Space_Minimum]
Megs=50
```

(This line controls free space checking on a network drive ie: R: The program will halt if there is less than the space specified here. In some instances over-filling a drive can result in a crash. Specifying a value of Zero will turn off free space checking)

```
[Node_Names]
Node1=Node_1
Node2=Node_2
Node3=Alpha1
Node4=Mike_P133
Node5=Matt_PP
Node6=Old386
Node7=Server
Node8=Glyn_P133
Node9=Guest
```

(Node names can be anything. They need not be the network name of the machine. Just ensure that Alpha1 runs the correct SN.BAT program to start)

```
[Render_Method]
Method=0
Scenes_At_Once=1
```

(Don't mess with this stuff)

```
[Options]
Beep_error=1
```

```
Beep_compl=1
Auto_fix=1
Cycle_time=1
```

(Don’t mess with this stuff)

Node Set-up batch files

Each node needs to run LWSN.EXE with a different set of parameters. We suggest using a different batch file for each node to start LWSN running.
For instance **SNNode1.bat** for node 1 and **SNServer.BAT** for the server.

Node 3 – SNAlpha1.BAT=

```
lwsn -2 r:\scrmnet\command\job3 r:\scrmnet\command\ack3
```

Node 8 SNGlyn133.BAT=

```
lwsn -2 r:\scrmnet\command\job8 r:\scrmnet\command\ack8
```

Using the program

The first time the program is executed – the program checks to see if the config file: RENDERFARMER.INI is present in your Windows directory. If it is not there – the program prompts you to create the file.

Once the program is running - the main control panel screen appears.

Farm Status Panel

This panel shows the status all the rendering nodes in the farm.
The main panel shows the name of the node alongside the nodes status, the time it has elapsed since that state was entered and if a scene is being processed, the name of the scene file and the frame number.

Each node reports a status:

- Offline - means the node is not currently available to the farm. Offline nodes are repeatedly tested to see if they are available - if a node is available - this status will turn into Ready.
- Ready - means the node is ready for work.
- Loading - means that a scene file is being loaded by that node
- Rendering means that a node is currently rendering a frame.
- Retiring - means the node will complete whatever it is doing and then quit LWSN at the earliest opportunity.

Farm Panel Controls:

GO! Button

When you run the program - we discovered most users like to have the opportunity to review, change priories and check scenes before setting the farm to rendering. When you finally want the farm to start rendering press the GO! Button.

MakeOffline.

If a node becomes unavilable for any reason- for instance, the workstation is required for something else and the workstation user has stopped LWSN when running - Select the node in question, by clicking on the name and press - the “make-offline” button. The system will note this and hand out the frame being processed to another node.

If the node should become available once more - RenderFarmer will automatically pick it up.

Retire

Retire forces a rendering node offline and will cause the LWSN program running on that node to terminate.

Queue Panel

The Queue panel shows the list of scenes currently in the scene-queue directory. The scenes are ordered as follows:
Top) Rendered and complete scenes
Next) Currently rendering scenes
Bottom) Remainder of the scene queue/

Each scene file is accompanied by information summarising - first frame last frame, frame step, anti-aliasing, image size etc.

The progress column shows the progress of the render (as a percentage) and lists frames which are complete. When all of the frames in a scene are checked this becomes “Complete”
The last column shows the save-path for images. When the save path is incorrectly set-up, this can cause Renderfarmer to fail.
Scene files which do not have a valid save path - or ones which are incorrect in some way - are flagged as such and will not be passed on to the rendering process.

Queue Panel Controls:

Re Scan Scene
Select a scene and press this button.
Renderfarmer will scan the scene file and report which frames are present.

Re Scan Queue.
This button re-scan every scene file in the queue.
Pressing the Re-Scan button cancels all such assumptions and checks all the frames in the queue from scratch. Doing this is pretty harmless if no frames were dropped.- although on some networks checking the presence of all frames in a long sequence can take quite a while.

Add Scene
Pressing this button brings up a common file selector box allowing the user to select a new LWS file to add to the rendering queue.
NOTE This can also be done by simply copying or saving a copy of the LWS file into the scene-queue directory.

Boost to Top
Select a scene and press this button to queue the scene to render immediately. (It will be shifted to the top of the queue)

Move Up
Move a selected scene one notch up the queue

Move Down
Move a scene down one.

Mark as Done.
Consider this scene as complete and ignore it. It will be placed in the completed section. To un-do this. Re-scan the scene.

What the program does.

(at first)
While waiting the program will check each scene file in the queue in turn and scan it for frames. It does this non-intensively in spare moments, and it may take several minutes to process an entire queue.

(after pressing go)

RENDERFARMER PARTICLE SYSTEMS

Each scene file in this directory is rendered in turn, from the first frame to the last. Each frame is handed to the first vacant rending node.
The queue file is continuously scanned for new scene files and also any files which are deleted. A scene file saved over a previous one (with the same name) will be placed at the back of the queue. Renderfarmer will never delete rendered frames.
Unlike *ScreamerNet* - this software can only render scene files placed in the scene-queue directory.

At the moment there is a limit of 300 scene files in this directory - at this time - exceeding this limit will probably cause the software to fail.

The software has a 30 node limit. – We have not tested it with more that 10 nodes.

The software hands out frames in batches of 15.
If a node reports that it has processed a frame and the frame is proved to be absent then two things happen.

1) That node is forced to re-load the scene
2) The scene file is marked as problematic. 16 problems and the scene file is retired.

Setting up a single-node Render Farm.

Read the Lightwave manual section on ScreamerNet and first ensure that Lightwave ScreamerNet is working with a single rendering node. If you have already done this then move on to using the program.

This section suggests a way of setting up ScreamerNet which we have found to be effective.

We would advise the following way of working

Create a Screamernet directory - ie. C:\NEWTEK\SCRMNET

and within that create a command directory C:\NEWTEK\SCRMNET\COMMAND

Within C:\NEWTEK\SCRMNET
place the following files:

```
LWSN.EXE - The node software
```

a copy of

```
LW.CFG - Lightwave’s config file
```

the DLLs

```
ERBUFIO.DLL
ERCORE.DLL
ERUTILS.DLL
HIIP.DLL
HIIPUNV.DLL
MSVCRT20.DLL
```

and in this directory - you should create a BAT file to run the SN.EXE program.

Call it RUNNODE.BAT

It should look like this:

```
lwsn -2 c:\newtek\scrmnet\command\job1 c:\newtek\scrmnet\command\ack1
```

When you run the RUNNODE.BAT - you should see a text window appear.

Run the program - and use the config dialog box to set-up the paths to the scene-queue directory, the ScreamerNet control directory and edit the RenderFarmer.ini file to set up the image-path fragment.

Setting Up a Multi-Machine Render Farm

Using a multi-machine set-up is very similar to using the single user set-up with one important distinction.- All objects, images and save paths must be in ***shared*** directories. Each and every node on the machine must be able to see the same data and on the same drives as all the other machines. If a single machine has a different world view then the network render will fail. The shared directories must have identical paths on all machines.

We recommend setting up a single, large, shared network-drive which appears as the same drive letter on all the machines. In our office a very large drive on the server appears as drive R: (R for Art) - This drive holds *all* the maps, objects, and scene files for all the shots. In addition this drive also houses the ScreamerNet command directory and the scene file queue

Using BATs to run LWSN

When running the rendering node software on the machines - each node needs a unique start-up command. The easiest way to do this is to have a different BAT file set up for each machine.

For my machine I have a unique BAT file: R:\SCRMNET\SNGLYN.BAT which looks like this.

```
lwsn -2 c:\newtek\scrmnet\command\job8 c:\newtek\scrmnet\command\ack8
```

And Matts's machine has a BAT file R:\SCRMNET\SNMATT_PP.BAT

```
lwsn -2 c:\newtek\scrmnet\command\job5 c:\newtek\scrmnet\command\ack5
```

and so on for each machine which is to appear in the *Render Farm*

Technical Support

This program receives exactly the right amount of technical support you'd expect from a free program. ie. none.

This program comes without guarantees. It may not work. It may not work in the way you wish. As a company we have no resources to correct errors or modify the code to your requirements. Sorry.

It is unlikely that any new features will be added to this program.

DO NOT EMAIL ME unless you have exhausted every option. At the very least, you should make sure that your network set-up works with the built-in Lightwave Screamernet farm. If it does not work with Newtek's software it will certainly not work with ours.

However, I will attempt to answer sensible technical questions if I can.

Mark EMAIL Subject Renderfarmer
And send to GlynWilliams@particle-systems.com

Price

This program is released free and without charge – and can be freely distributed without our consent. It should not be changed in any way.

For anyone who is interested, the source code is for sale for a sensible price.

If you do use this program, and feel it is of benefit to you, then please **go out and buy a copy of I-War** (or any other Particle Systems game). We'll get some royalties that way, and hey, you get a free game in the process! The game is stuffed full of cool Lightwave rendered stuff.