



setting up LightWave
screamernet



written and illustrated by matt gorner | 2003



Introduction:

It was years after I started using LightWave until the penny finally dropped with ScreamerNet! I read many tutorials, mostly of the 'do this' 'do that' format. These are all very well and good but I never understood **why** I was doing certain things or indeed how the ScreamerNet process worked exactly.

I'm a big believer in the fact that it's much easier to learn something if you understand **how** it works, which is why this tutorial is as long as it is!

It's also aimed at total beginners to ScreamerNet (and indeed LightWave) so please don't be offended if it sounds a little simplistic, it's really not meant to be condescending, it just explains everything in a lot of detail! Essentially it's the kind of tutorial I wish I had when I first started out!

Although it is written for Intel machines, some of the concepts explained will transfer to the Mac version, so it's still worth a read even you are using a Mac.

I hope you enjoy this tutorial, and indeed I hope it helps you setup ScreamerNet for your own rendering tasks!

Cheers
Matt Gorner
matt@creactive-design.co.uk

Contents:

Subject	Page
What is ScreamerNet?	1
How does it work?	1
The Command folder	1
The Content folder	2
Content organisation	2
Configuration files	3
Sharing the files	4
Setting up the master computer	8
Setting up the render nodes	13
Setting the save location	18
Rendering using ScreamerNet	19
Batch rendering on one machine	21
Setting up batch rendering on one machine	22
Setting up multi-processor machines	23
Troubleshooting and limitations	24



What is ScreamerNet?

ScreamerNet is LightWaves network rendering software. It allows you to use many computers over a network to render scenes quicker by dividing the workload. ScreamerNet can also be used to batch render multiple scenes on the same computer by queuing them up.

The ScreamerNet software has two parts to it, the first resides in the LightWave layout program, found under the 'Rendering->Network Rendering...' popup menu, this is the 'controller' part of ScreamerNet. The second can be found in the 'Programs' folder found in the main LightWave install, it's a separate program called 'LWSN.exe', this is the program that is run on the machines that are 'controlled.'

How does it work?

The concept is relatively simple, one of the computers on the network is designated the 'master' computer, this is usually the computer with the full LightWave install on it. The master controls the other computers or 'nodes' by passing 'render commands' to them, these commands tell the node which scene to load and which frame to render.

When a node has finished rendering a frame it passes a command back to the master saying it has finished and is ready for the next frame, and so on until all the frames have been rendered.

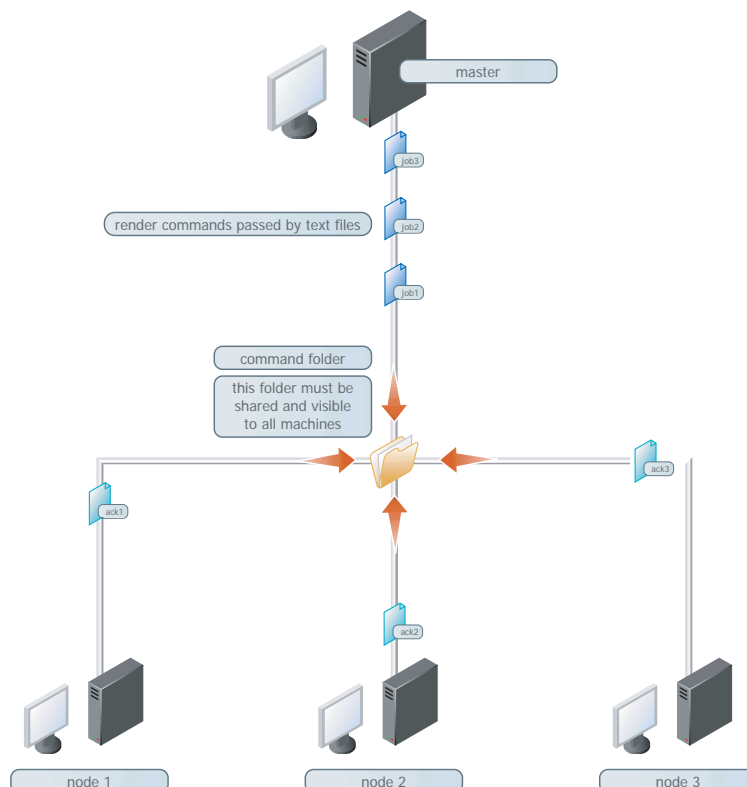
The Command folder:

These 'render commands' are not actually passed directly from computer to computer, what I mean by that is the LightWave program on the master computer isn't talking directly to the 'LWSN.exe' program on each node. They are talking to each other via a special folder - the 'command folder'.

This is an important folder that the master and all nodes must be able to see somewhere on the network. The reason for this is because the render commands are actually text files with the command inside, the text files are saved into the command folder with a unique name to separate one nodes commands from another. These commands are better known as 'job' files. Each job file has a number corresponding to the node it is designed for, so node 1 will be passed 'job1', node 2 will be passed 'job2' etc.

Likewise in order for the nodes to talk back to the master they too need to save files with unique names so the master knows which node is talking back to it. These 'returned commands' or acknowledgments, are known as 'ack' files, and like the job files the 'ack' files have a number to say which node has sent it, so node 1 will return commands to the 'ack1' file, node 2 to the 'ack2' file and so on.

This diagram helps illustrate this process:





We know that the master and nodes need to be able to find the command folder in order to 'talk' to each other, but the command folder only stores instructions on which scenes and frames to render, it doesn't actually pass any scene data to the nodes. So like the command folder the render nodes also need to know the location of the scenes content.

The Content folder:

This is where the 'content folder' comes into play. It stores all the objects and images used to make up the scene, as well as the scene file itself. Like the command folder all the nodes must be able to see this folder so that when told to render a frame from a certain scene, they know where on the network to look for it.

Content organisation:

Good organisation of content is essential if you want to make your life easier in LightWave, storing files all over your hard drive to make up one scene will just give you headaches, instead try to organise your LightWave projects in a logical manner. Before I go into this you must first understand how LightWave finds content, bear with me on this, it's long winded but worth a read!

Ignoring ScreamerNet for a second, start up LightWave layout and bring up the programs 'Options' panel (use the 'o' key or go to the popup menu 'Layout->Options->General Options . . .') The first setting you'll see is the 'Content Directory' input, this is where you tell LightWave where to look for any content files when it first tries to load anything. This location along with other settings is saved inside the LightWave preference file ('LW3.cfg') when you quit the program.

If you load the 'LW3.cfg' preference file into a text editor (it's just a text file really) among many other entries you will find these:

```
DirectoryType Scenes C:\LightWave\Scenes
DirectoryType Objects C:\LightWave\Objects
DirectoryType Surfaces C:\LightWave\Surfaces
DirectoryType Images C:\LightWave\Images
DirectoryType Previews C:\LightWave\Previews
DirectoryType Motions C:\LightWave\Motions
```

These are additional search locations added by LightWave which it will scan when you try to load anything, for example if you tried to load an object, LightWave will look in 'C:\LightWave\Objects' first. The reason it is looking in 'C:\LightWave' is because that is where the content directory has been set. If you were to set your content folder in your preferences to 'D:\Somewhere\on\my\harddrive' the entries above would be changed automatically by LightWave to:

```
DirectoryType Scenes D:\Somewhere\on\my\harddrive\Scenes
DirectoryType Objects D:\Somewhere\on\my\harddrive\Objects
DirectoryType Surfaces D:\Somewhere\on\my\harddrive\Surfaces
DirectoryType Images D:\Somewhere\on\my\harddrive\Images
DirectoryType Previews D:\Somewhere\on\my\harddrive\Previews
DirectoryType Motions D:\Somewhere\on\my\harddrive\Motions
```

LightWave takes where you set the content directory and adds those sub-directories, if they don't exist LightWave uses only the content directory you set, in the case above 'D:\Somewhere\on\my\harddrive'.

The point is LightWave **expects** to find these sub-directories within the content directory. Taking this into account, when you save a scene (which is also a text file you can read) LightWave saves the location of any items in the scene file **without** the content directory part of the location. For example if you had a content folder set to 'C:\Program Files\LightWave3D 7.5\Content' and in your scene you had loaded an object called 'final_object.lwo' which was stored in a folder called 'My Objects' and then a sub-folder called 'Final Versions', the scene file would have an entry that loads the object from disk like this:

```
LoadObjectLayer 1 My Objects/Final Versions/final_object.lwo
```

The LightWave program knows the top level location of the content directory (taken from the 'LW3.cfg' file) and the scene file knows the rest. There is a reason for this, an important reason! Let's imagine LightWave didn't have a content directory preference and instead it saved the full location of any objects, images etc. with the scene file, so in our example above it would be 'C:\Program Files\LightWave3D 7.5\Content\My Objects\Final Versions\final_object.lwo' if you were to give that scene to someone else to work on, or you wanted to work on a different computer, unless you copied the files into exactly the same directory structure, LightWave would complain about missing files.

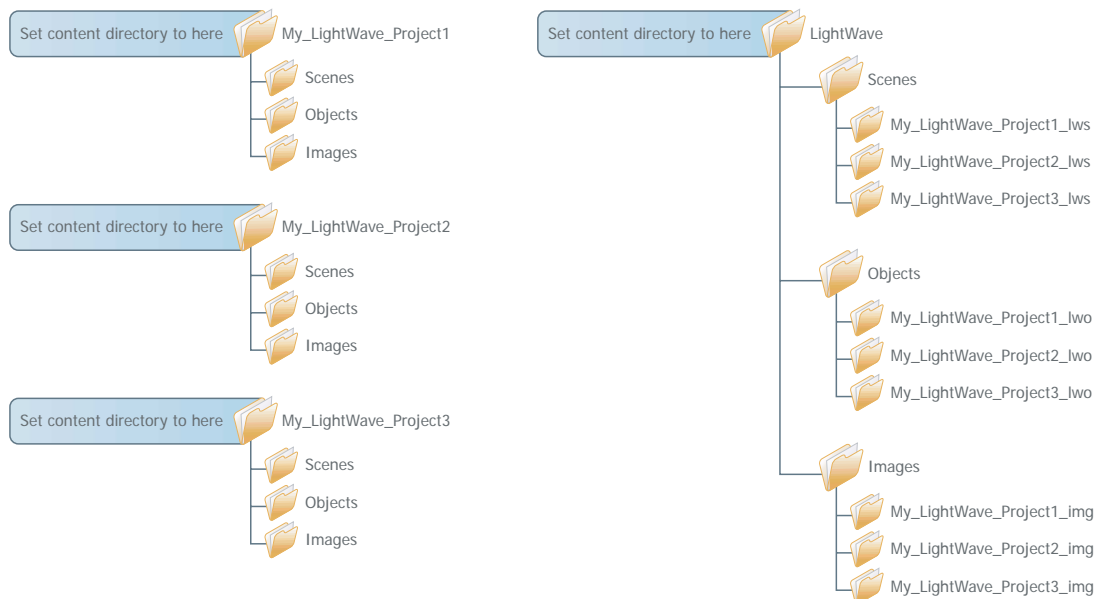


This is because LightWave is trying to load the files using the locations from the original computer which may not match the computer it is trying to load from. Hard-coding file locations isn't a good idea, which is why LightWave doesn't work this way.

So what is the best way to organise content files? Well that depends on how you work with your files. The favoured method is to have a folder with the project name and inside that the 'objects', 'scenes' and 'images' folders. You would then set your content directory to the top level folder before trying to load anything. This method is great if you constantly work on a project in various places or several people are working on it and it needs to be passed around, because everything needed for the scene is in a self-contained set of folders.

However if you are the sole user and you work from one machine, this method can be a pain when working on several projects. This is because you have to keep setting your content directory to each project before loading, this is shown more clearly in the diagram below.

Personally I find it easier to organise my content files as shown on the right. The benefit of doing it this way is that you only have to ever point your content directory to one folder, it's essentially a re-worked version of the first method, it is still portable, just have to collect items for a scene in several places, but if you don't move your content around much I find this a better way of organising content files.



If none of that made any sense don't worry, as far as ScreamerNet is concerned it doesn't matter how you store your files, as long as it can find them, but it is good practice to organise your files logically.

Another very important fact to remember when saving scenes, objects and textures is that spaces in file and folder names can cause havoc with ScreamerNet. I find this a little backward as computers have been able to handle filenames with spaces for some time now, but ScreamerNet doesn't like them, so don't use them! Instead use the underscore symbol '_' or a hyphen '-'.

Back to ScreamerNet, let's recap. We've learned so far that in order for the master computer to 'talk' to the nodes they all must be able to see the command folder, as they use this to pass messages to each other. The nodes must also be able to see the content folder, as this stores all the files needed for rendering, so the content folder holds the files for rendering, the command folder stores the instructions on what to render.

Configuration files:

As well as the content and command folders, there is a third item the nodes must also be able to find; **plugins**. Any plugins you've used to create effects in your scene must also be found on the network, for example if you've used Sasquatch to create hair on a character, or a motion plugin to move an object, indeed LightWave itself uses plugins to load and save files, so even if you haven't used any FX plugins ScreamerNet still needs to know where they are to save files after it has rendered them. All these plugins are separate files that need to be loaded by the 'LWSN.exe' program, just as LightWave did when you created your scene.

When you think about it, if the ScreamerNet program needs to know the location of all the plugins you have installed, you would need to type the location of every single one! That would be a lot of typing! The way LightWave deals with this is by storing the location of all the plugins in a configuration file. When you scan your plugins LightWave is 'writing down' where they all are, it then saves them to a



configuration file, this file is called 'LWEXT3.cfg' (LightWave EXTensions). So in order to tell ScreamerNet where all your plugins are all you need do is tell it where the 'LWEXT3.cfg' file is on the network.

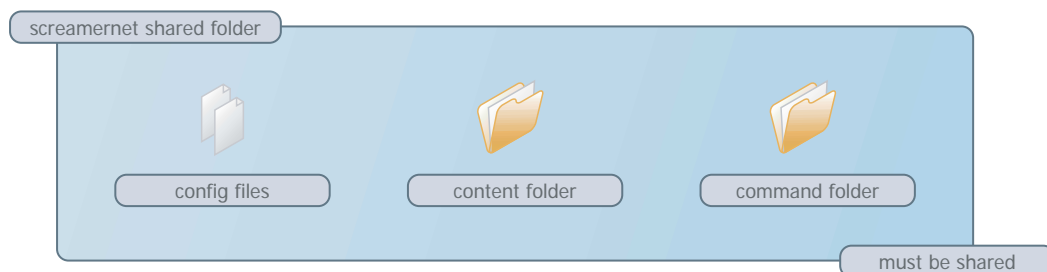
There is another config file ScreamerNet needs to find in order to work; 'LW3.cfg'. As we mentioned earlier this is the LightWave layout configuration file, it stores a whole bunch of settings but most importantly, it stores the location of the content folder we mentioned earlier.

Let's recap, here's a breakdown of what ScreamerNet needs to be able to find on a network in order to run:

Needs to find:	Needed for:	Information stored in:
Command Folder	Master computer to 'talk' to render nodes	All nodes are told this at initialisation time
Plugins	Plugins used in the scene and for file saving	LWEXT3.cfg
Scene Content	All objects, images and scene files	LW3.cfg / nodes also told this at initialisation time

Technically these items don't need to be in one place, they don't even need to be in the same building, or even the same country! As long as ScreamerNet can find them they could be anywhere. However, for the sake of simplicity it makes sense to group these items in one place; a **shared network folder**.

Sharing the files:



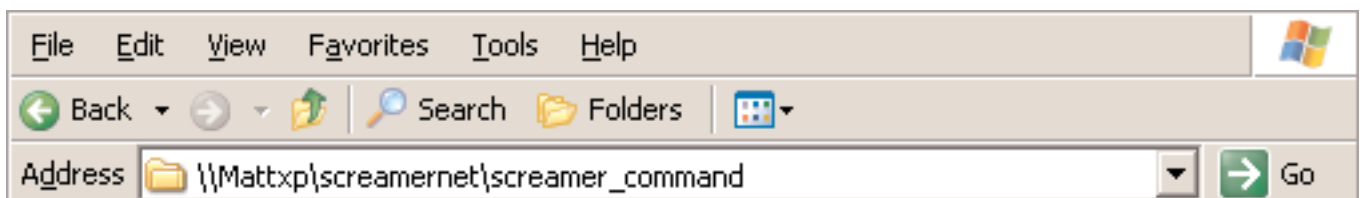
There are many methods of physically networking Windows machines together but because there are so many variables to consider - too many for this tutorial, I'm going to assume that you already have a Windows network up and running, if not the Internet is your best resource, or your computer administrator if you have access to one.

Sharing files across a network obviously means you are going to refer to locations using pathnames. As far as ScreamerNet is concerned there are two main methods for doing this. The first is using **UNC** or **Universal Naming Convention**, the other is using **Drive Mapping**. They ultimately do the same thing but differ very slightly.

UNC naming:

This is the simpler method of the two. UNC naming is exactly as it sounds, it's a standard naming convention for finding drives or folders on a network.

Whenever you browse through folders using Windows, the 'address' or 'network path' can be seen in the address bar (the screenshot below may differ depending on your Windows version).



In the example above it tells us that the computer name on the network is 'Mattxp' and on that computer is a folder (which has to be shared to be seen) called 'screamernet'. Inside that folder is another called 'screamer_command'.

That is essentially it! Locating folders / files for ScreamerNet using this method requires us to type the location of the shared network folder, kind of like the address for your house. So long as you can see the computers, drives and folders you are trying to access on the network, it's very easy to work out the network path by simply looking at it in the address bar as shown above.



Drive Mapping:

This technique was favoured by a lot of the ScreamerNet tutorials I read, I could never understand why as it takes more time to setup and can be confusing.

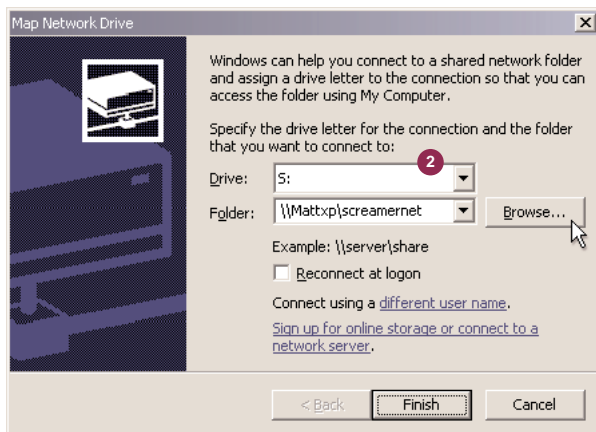
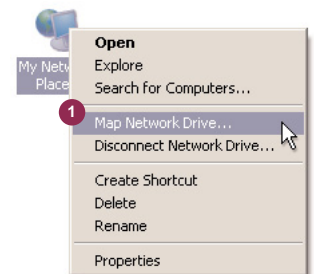
Drive mapping is the process of replacing a network path to a folder with a letter. Imagine we had a folder on a computer on a network, and it's pathname was something like '\\my_computer\documents\excel\timesheets\2003'. You could substitute all that with a letter, 'T:' for example. Then whenever you accessed 'T:' drive it would take you straight to the folder '2003'.

Sounds great doesn't it! Well not quite. If we were to use this method for ScreamerNet we would have to map the path to the shared ScreamerNet folder to a letter on **each** computer. The problem arises when a node computer tries to access a mapped drive on a computer that is switched off or unavailable, it will complain that the path can no longer be found.

This can be a problem if you are setting up ScreamerNet in an office using your colleagues computers, especially if they don't know what the error message means! They will soon tell you about it!

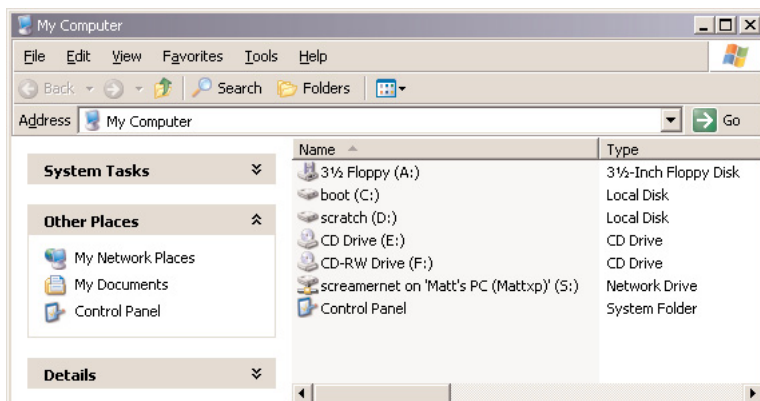
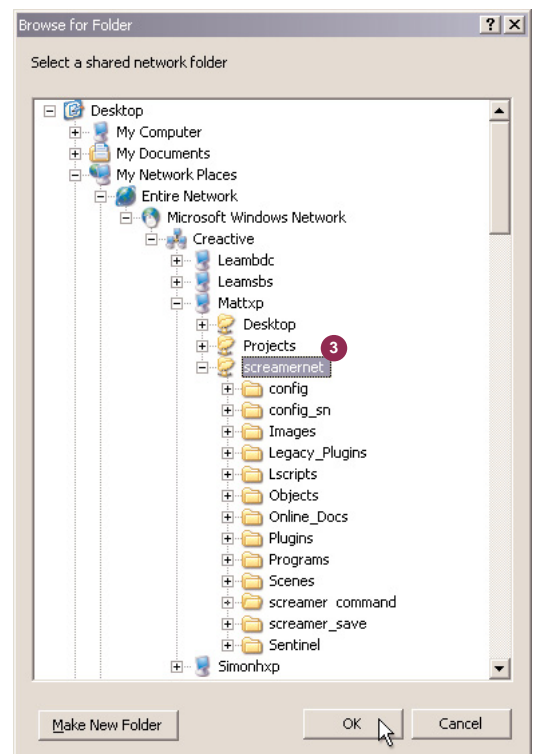
However if you are intent on using this method for resolving pathnames, here's how to map a letter to a pathname. I won't be using this method for this tutorial as it offers no benefits over UNC naming that I'm aware of.

1) If you right-mouse click on the 'My Network Places' icon (again this will differ depending on the version of Windows you are using, if in doubt as to where the option is, do a search in Windows Help). There will be an option called 'Map Network Drive . . .'. Clicking the option will invoke another window.



2) The 'Drive' popup menu is where you set which letter you want to assign, in our ScreamerNet example 'S:' seems to makes sense. Once a letter is selected you can then set the path to map the letter to in the 'Folder' popup menu. Alternatively you can browse the network for the folder like so (3):

Once done click the 'Finish' button. If you now look in 'My Computer' you will see a new 'drive' called 'S:' which points to the folder you specified. Any reference to the 'S:' drive will go straight to that folder.





Okay, back to ScreamerNet and sharing the files it needs.

We know what ScreamerNet needs to be able to find, now is the time to figure out where to put these files. By default LightWave stores most of its files in one folder, usually found on the main hard drive in a folder called 'LightWave'. If yours is different from this don't worry.

I say most files because Windows steps in here and interferes a little! If a program is written to conform with Windows protocol, it will store its preference files in user specific folders, the idea is that many people using one machine can have their own preferences and setups, technically this is a good idea but can cause problems with ScreamerNet if you're not sure which preference files LightWave is using, to avoid confusion either use one set of preference files or 'point' LightWave to the right files when it starts up - more on this later.

The location of these preferences depends on which version of Windows you are using, but it will either be in 'C:\WinNT\Profiles\WhichUser' or 'C:\Documents & Settings\WhichUser' if in doubt do a search for 'LW3.cfg' and see where they are.

The content files are also usually stored within the main LightWave folder, but again these can be moved anywhere you like as long as you tell LightWave and ScreamerNet where the new location is.

To make this tutorial the same for all users I'm going to suggest that the LightWave content, config files and command folder are all stored in the main LightWave folder (on the master computer). If you don't want to do this just make sure you replace all pathnames to these items with the new location you've decided to put them.

Before we start sharing these files, you will need to create some new folders in your LightWave directory. Below is a diagram showing the folder structure for a typical LightWave install (left of diagram) and the other showing the new folders you need to create.



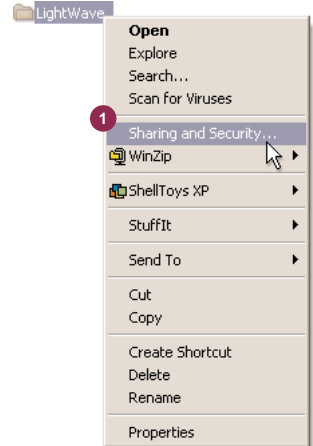
The new folders are 'screamer_command', 'screamer_save', 'config' and 'config_sn', respectively these are the command folder, a folder to save the rendered frames to, the regular LightWave config folder and a config folder specifically for ScreamerNet use.

Okay, we now have place-holder locations for all the items ScreamerNet needs, they are all in one place (as suggested in a previous diagram) all we need to do now is share the folder across the network so all the computers can see it.

The process for this may differ slightly depending on which version of Windows you are using, I'm writing this on Windows XP Professional, but the principles are the same for nearly all versions of Windows.

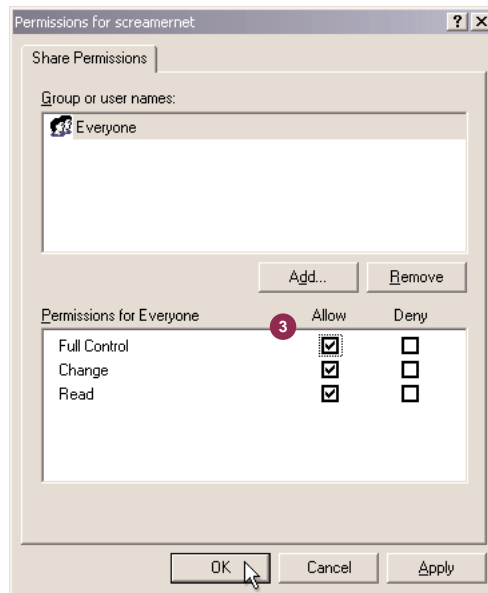
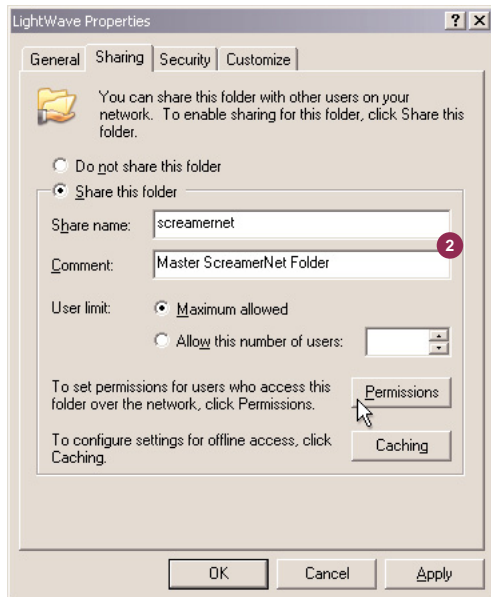


1) Locate your main LightWave folder on the master computer and right-click the mouse to bring up the context sensitive menu. In the list there should be a 'Sharing and Security' option, if not select the 'Properties' option and find the sharing tab.

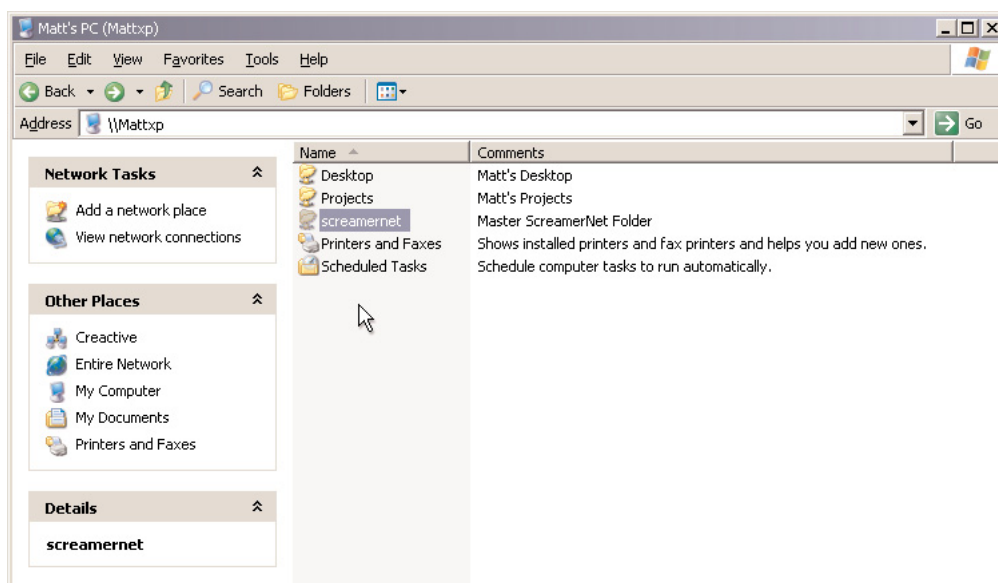


2) You should now be presented with another window. Under the 'Sharing' tab there will be an option to turn on sharing for that folder. Once activated you will now be allowed to enter a name in the 'Share name' box. You can call this anything you like but something short and succinct like 'screamernet' is all that is needed, and just to be safe make sure it's all **lowercase** with **no spaces**.

3) While you're in there make sure you set the 'Permissions' for the folder to 'Full Control' so that any computer (or render node) can read from and write to it.



Now if you browse your Windows Network you should be able to see the folder you've just shared.





Setting up the master computer:

Now we have all the folders in place, and accessible across the network, we can begin to setup the computers.

Setting up the master is a little more involved than setting up render nodes because this is the computer with the main LightWave install on it, which means it's the computer that will normally be used for editing scenes, creating objects, texture maps etc.

As we've mentioned earlier LightWave uses config files to tell it where certain things are. The pathnames to these items will be local to the computer they reside on. This would cause problems for ScreamerNet as it needs the pathnames to be 'network-aware'.

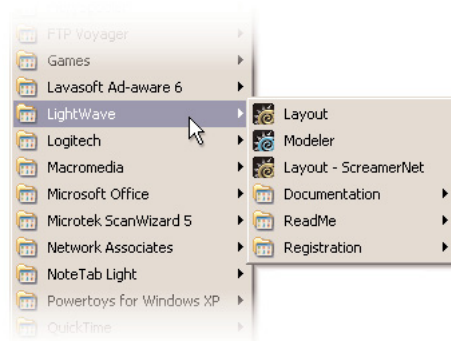
What this means is that we will need two sets of config files; one set for when we are working with LightWave as normal, and another set for when we are using ScreamerNet.

This is the reason why we created the two folders 'config' and 'config_sn' earlier.

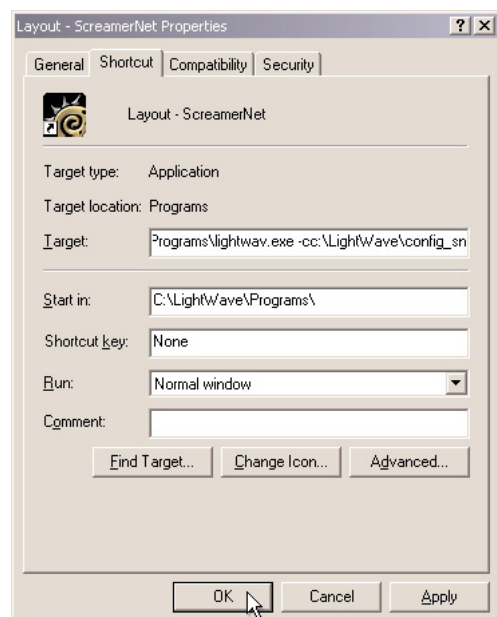
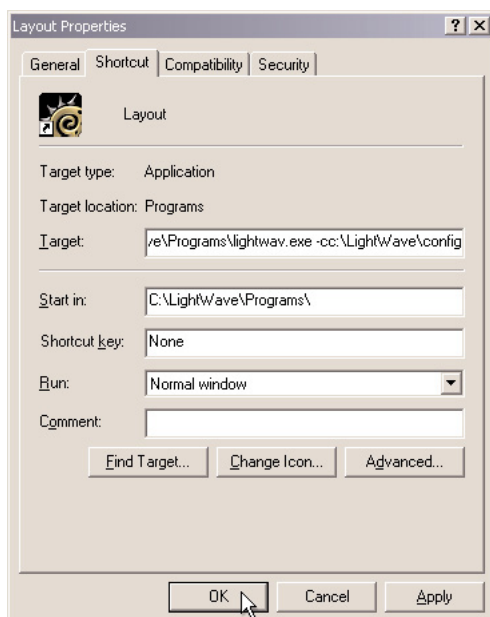
By default LightWave won't know anything about these folders because we created them, so we need to tell LightWave to use them. The easiest way to do this in Windows is to 'point' LightWave to the correct set of config files when it loads.

Fortunately this is easy to do in Windows using shortcuts. Windows shortcuts can pass information to a program as it loads by typing commands into the 'Target' box. To locate this box you'll need to find the existing shortcut to LightWave Layout in your 'Start Menu' or create some new ones from scratch in the usual manner.

You'll need to create two shortcuts to the LightWave Layout program, one for regular LightWave use and one for ScreamerNet use.



Once you've created the two shortcuts, right mouse click on them and bring up the properties for each one. You should now see windows like the ones below (but without the commands, obviously I have this setup already!)





Find the box called 'Target' and enter the following lines:

For normal LightWave use

```
C:\LightWave\Programs\lightwav.exe -cc:\LightWave\config
```

For ScreamerNet use

```
C:\LightWave\Programs\lightwav.exe -cc:\LightWave\config_sn
```

The first part '`C:\LightWave\Programs\lightwav.exe`' is to do with the shortcut, it's simply a path to the LightWave Layout program, the second part is the bit we're interested in '`-cc:\LightWave\config_sn`' the '`-c`' switch tells LightWave you want to tell it where the config files are kept, and '`c:\LightWave\config_sn`' is the path to where the config files are. Note there is **NO** space between the '`-c`' and '`c:\LightWave\config_sn`'.

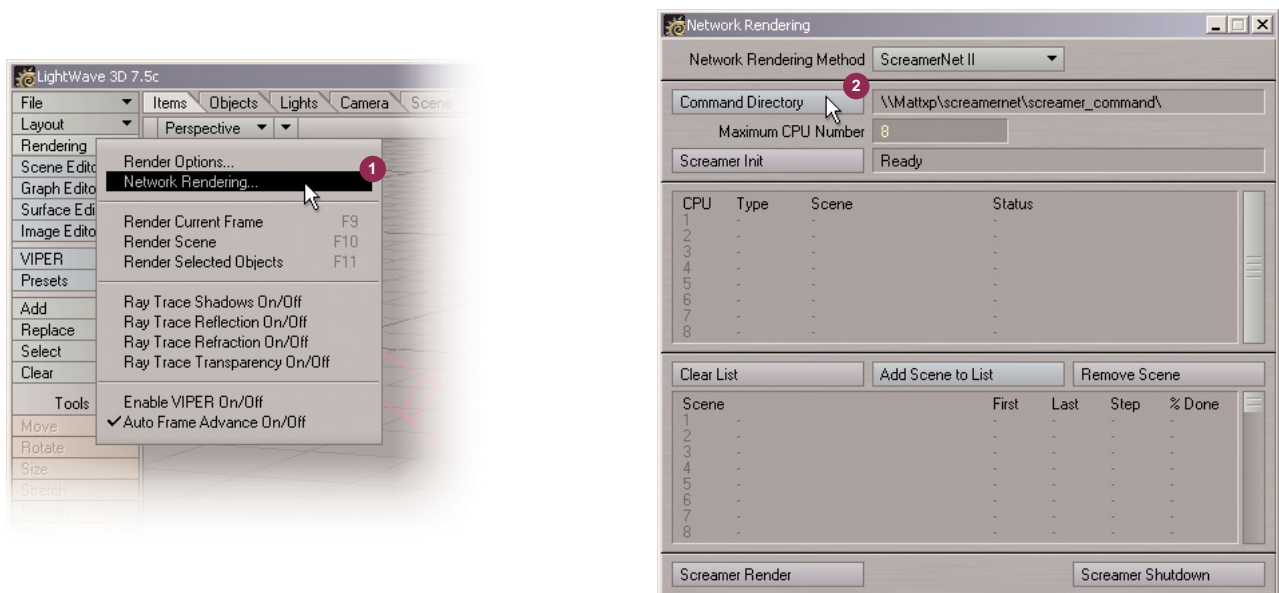
We don't need these paths to be 'network-aware' as they are not used by ScreamerNet, they simply point LightWave to the files that do contain the information ScreamerNet needs.

Now that we have the shortcuts for LightWave pointing to the correct config files, you need to actually copy your existing config files to **both** the 'config' and 'config_sn' folders, so that any custom menus, colour settings, plugins you've added etc. are loaded when you run LightWave. As mentioned earlier these config files may be located in different places depending on your version of Windows, simply do a search for 'LW3.cfg', once found copy the files 'LW3.cfg' and 'LWEXT3.cfg' into the two config folders you've created.

Okay, now everything is in place run the version of LightWave Layout that will be using the ScreamerNet config files.

We need to setup a few things in this version of LightWave so that the settings are saved to the ScreamerNet version of the config files when you quit the program.

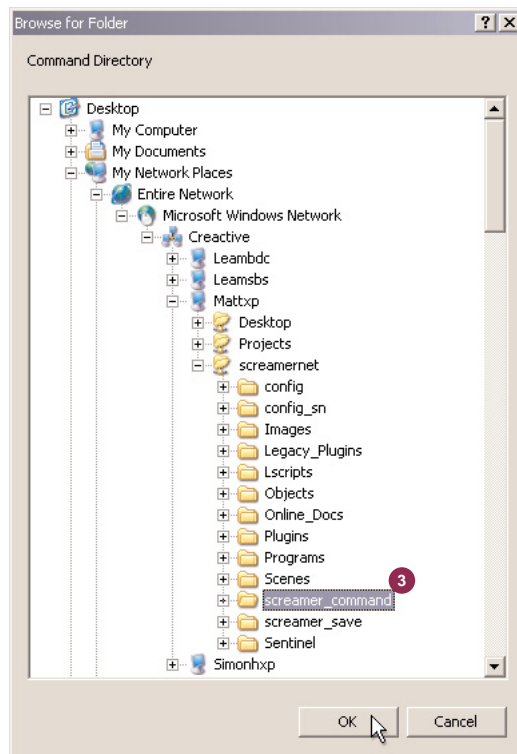
1) The first item we need to set is where the command folder is located on the network. This option is found in the 'Network Rendering' dialog box under the 'Rendering->Network Rendering . . .' popup menu.



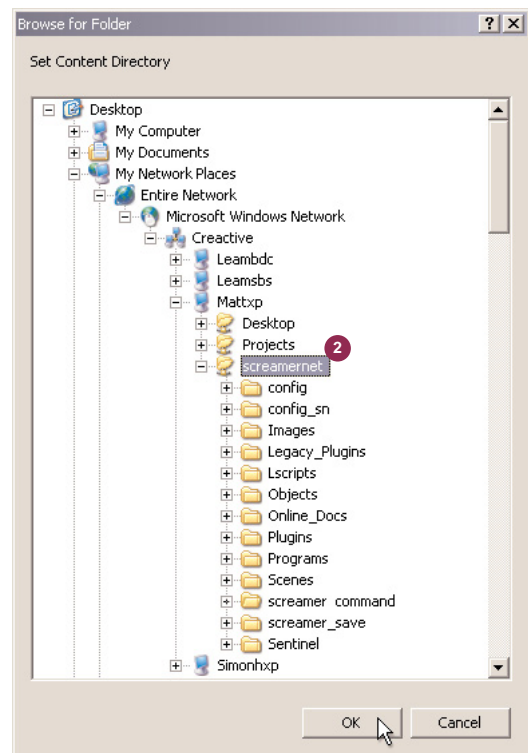
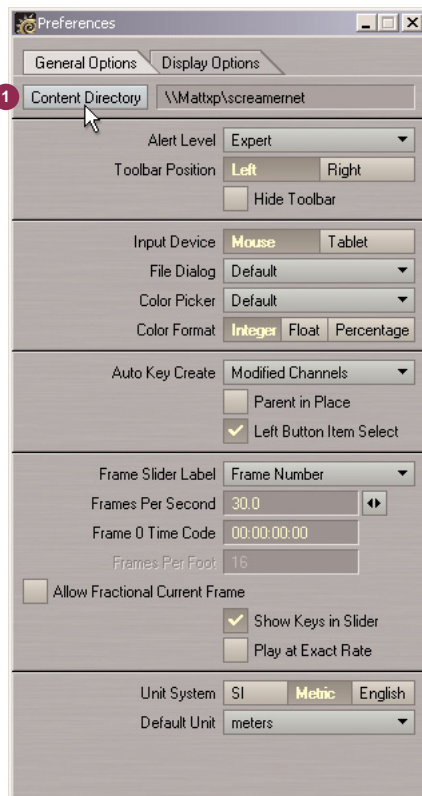
2) Clicking the 'Command Directory' button will bring up another window in which you can locate the shared network folder on the master machine, if you are using the names as suggested in this tutorial it will be called 'screamer'.

3) You should be able to expand the 'screamer' folder to view the folders inside, if not check that you have access privileges to do so. If you can see the folders, locate the 'screamer_command' folder and click okay.

Whenever you browse for the 'screamer' shared folder like this make sure you browse from 'Entire Network' down. If you added 'screamer' as a network place shortcut and used that it resolves to a local pathname which causes problems, I learned the hard way!



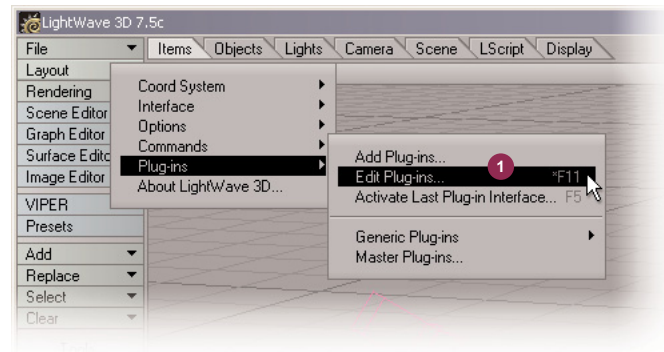
Next item we need to set is the content directory. This is found under the 'General Options' panel (press the 'o' key). Click the 'Content Directory' button (1) and locate the 'screamernet' shared network folder again (2). Select the 'screamernet' folder and click okay.



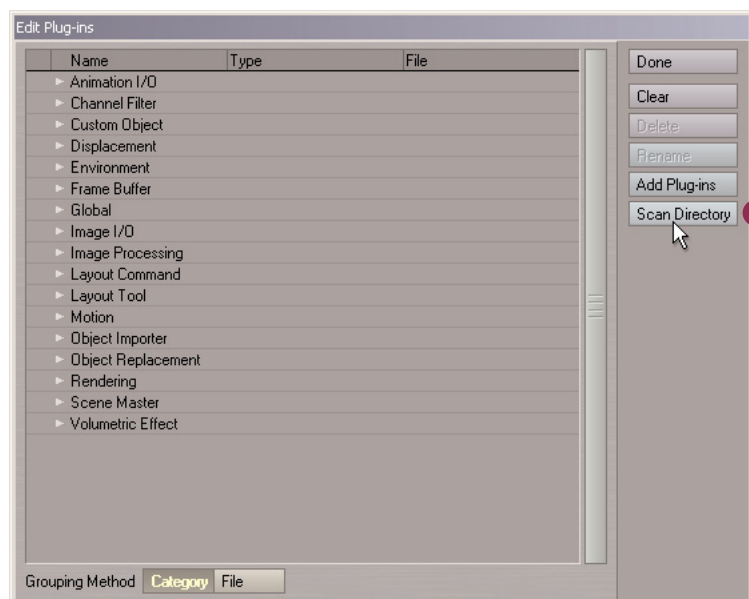


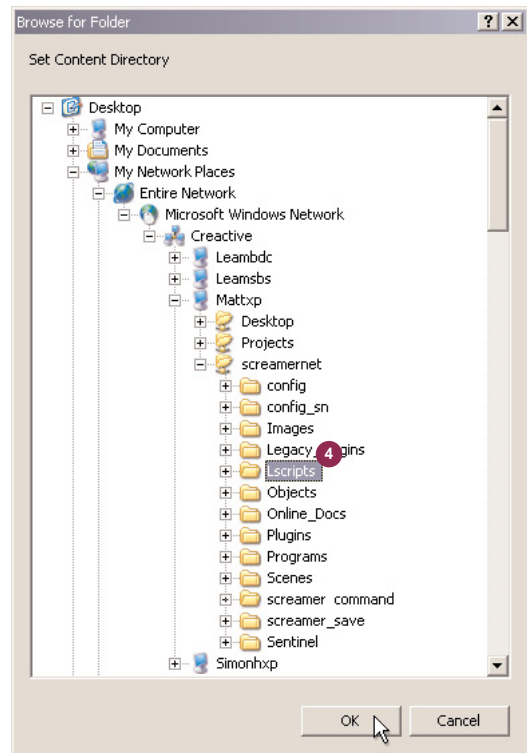
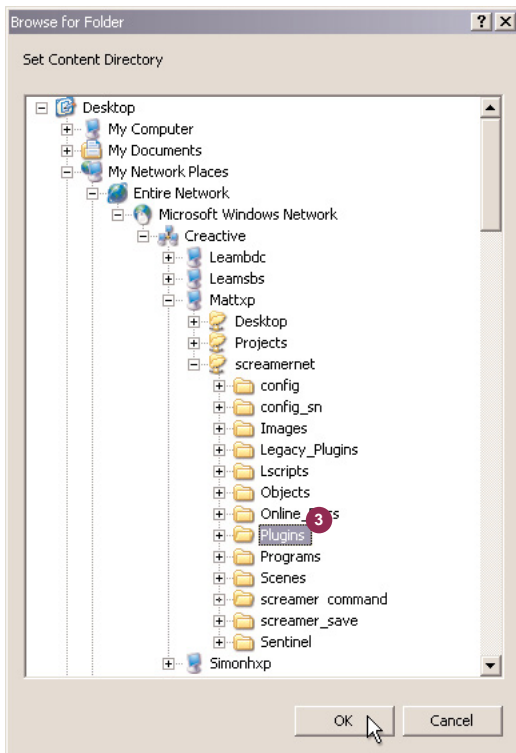
Almost done! The last thing we need to set up is the location of the plugins on the network. Some tutorials tell you to open up the 'LWEXT3.cfg' file and perform a search and replace on the pathnames, you could do it that way, but it's easier (and less error prone) to re-scan your plugins using the shared network path location.

1) The option to scan your plugins can be found under the 'Layout->Plugins->Edit Plugins . . .' popup menu (or ALT + F11 key).



2) This will bring up the 'Edit Plugins' window where you will find the option to 'Scan Directory'.





Click the 'Scan Directory' button and locate the 'Plugins' folder (3) on the shared network, do this again and locate the 'Iscripts' folder (4). That's the master computer setup!

To recap we created two shortcuts to the LightWave layout program. Each shortcut pointed to a different set of config files, one for normal LightWave usage, the other ScreamerNet use.

We then copied the existing config files into the 'config' and 'config_sn' folders. Once done we loaded the version of LightWave pointing to the ScreamerNet config files.

Next we setup the content folder, the command folder and re-scanned the plugin list. The reason for this is to save the 'network-aware' pathnames for each of these items into the ScreamerNet config files.



Setting up the render nodes:

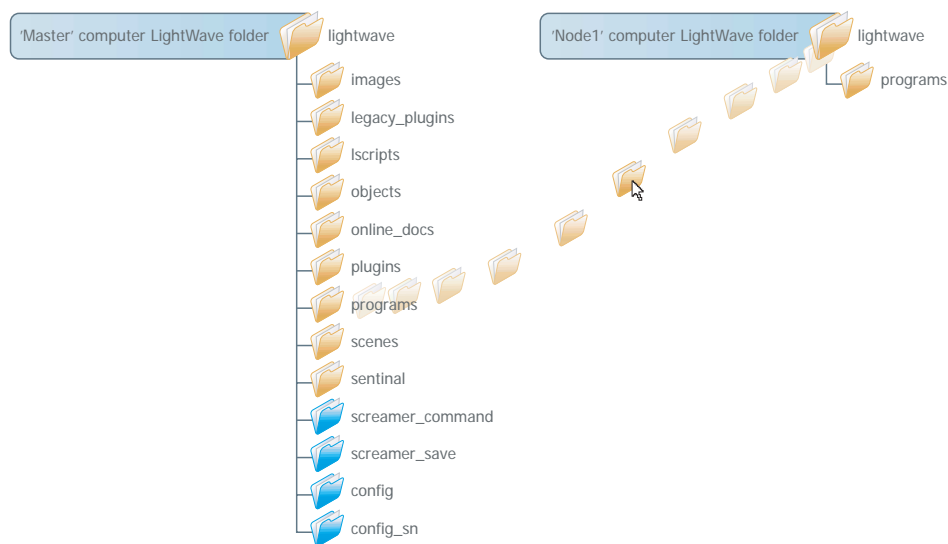
Setting up the nodes is relatively simple compared to the master computer because you can setup one node and then simply copy the same set of files to the other nodes changing only a few pieces of text.

At this point you need to decide whether your master computer itself is a render node. When ScreamerNet is up and running the master computer isn't doing anything more than passing commands around, it's certainly not doing any rendering, so unless you plan to do some work while the other computers are rendering you could make the master a render node too.

Whether you decide to use your master computer as node 1 is down to you, the process of setting up the master as a node is exactly the same as if it were a totally different computer, it's just running both LightWave and the 'LWSN.exe' program.

Before we start with the detailed stuff the first thing you'll need to do is copy some files from the master computer onto the hard drive of each computer that will be a render node.

Create a folder called 'LightWave' on the main hard drive (usually the 'C:' drive) of each node computer (obviously we don't need to do this on the master as it should already have LightWave installed). Now copy the entire folder called 'Programs' from the master computer into the 'LightWave' folder you've just created **on each node**.



The last part of setting up a node requires a little explanation. If you were unfortunate to remember computers in the days of MS-DOS you'll be familiar with running programs via a command line. Essentially MS-DOS didn't have a lovely GUI with icons for programs so you had to type the name of the program to run it, along with any 'commands' the program needed after it's name.

Well nostalgia is back! Because despite the 'lovely' Windows XP interface some programs still require you to run them via the command line, and 'LWSN.exe' is one of them!

When you run the 'LWSN.exe' program it needs to know a few things at runtime; where the config files are stored, where the content directory is located and which 'job' and 'ack' number it is going to be in order to distinguish itself from the other render nodes.

Having to type all this information for each node everytime you needed to run ScreamerNet would be a pain, fortunately Windows allows you can create a text file that acts as a 'stored' command line. This text file has a special name; a 'batch' file. It has the file extension '.bat' but really it's just a plain text file.

To create a render batch file open up a text editor, Notepad is fine and type the following red text, replacing any reference to '\\Mattxp\screamernet' with the network path to your own shared folder we setup earlier.



```
echo "LightWave ScreamerNet Node 1 Initialisation . . ."  
cd c:\lightwave\programs\
```

```
LWSN -2 -c\\Mattxp\screamernet\config_sn -d\\Mattxp\screamernet\ \\Mattxp\screamernet\screamer_command\job1  
\\Mattxp\screamernet\screamer_command\ack1
```

(The two lines above starting with LWSN should all be on **ONE** line).

Now save the file (as a text file) with the extension '.bat', and place it inside the 'LightWave' folder of the computer that will be node 1. I've called my batch file 'sn_init_1.bat' but you can call it anything you like.



sn_init_1.bat

The render batch file script may look complicated to users un-familiar with MS-DOS, but it's pretty straightforward once broken down, so let's do that.

```
echo "LightWave ScreamerNet Node 1 Initialisation . . ."
```

This has nothing to do with ScreamerNet whatsoever, it's just a little text that is echoed to the screen to tell you which node number the batch file has been assigned. You could leave this out if you so wished.

```
cd c:\lightwave\programs\
```

Changes the current directory to the 'Programs' directory on the main hard drive so that the next line knows where to find the 'LWSN.exe' program. If you have installed LightWave somewhere else then change this line to reflect where you have put it.

```
LWSN -2
```

Runs the LightWave ScreamerNet program. The '-2' sets up which 'mode' ScreamerNet will run in. '-2' basically says we're accepting commands from the master computer on which scene / frame to render. There is a -3 mode which allows you to specify the scene and the start / end frames, but this mode defeats the point of using several computers to render a scene as the -3 only allows you to pass it one scene. Basically you'll never need to use -3 mode!

```
-c\\Mattxp\screamernet\config_sn
```

This command tells 'LWSN.exe' where the config files are kept on the network, the '-c' is the 'switch' that says the following text is the path to the config files. If you have your config files elsewhere now is the time to tell ScreamerNet where they are. Remember 'LWSN.exe' needs to know the location of the config files in order to find the location of the plugins, the paths to which are stored in the 'LWEXT3.cfg' file. Note: It's important that there is **no** space between '-c' and the pathname after it.

```
-d\\Mattxp\screamernet\
```

This command tells 'LWSN.exe' where the content directory is on the network, the '-d' is the 'switch' that says the following text is the path to the content directory. I've never quite understood why you need to tell ScreamerNet this as it should be stored in the 'LW3.cfg', I guess it's just in case you want to store them somewhere else. Again if your content directory is not in the shared network folder, replace '\\Mattxp\screamernet\' with the path to where you put the shared network folder containing these items.

```
\\Mattxp\screamernet\screamer_command\job1
```

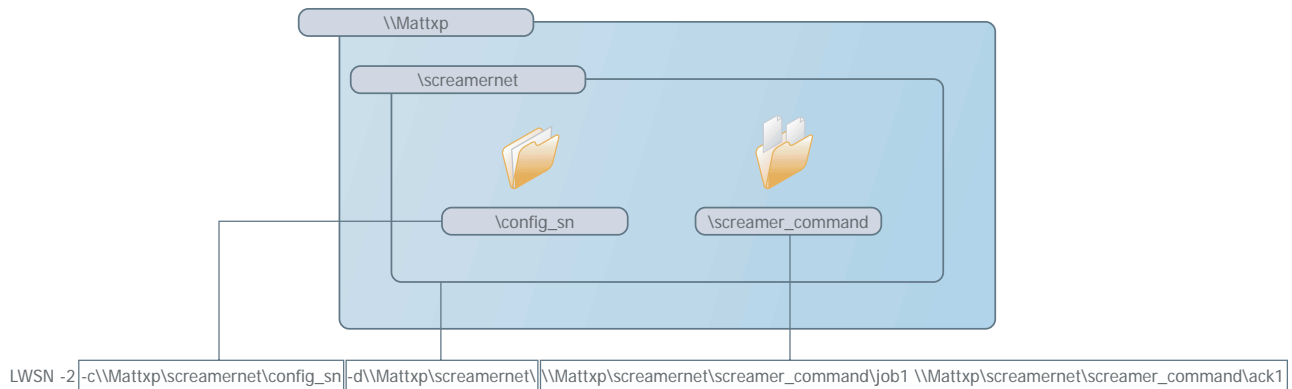
No 'switch' this time, but this is where you tell 'LWSN.exe' where to look for job commands, and which 'job number' it is to be allocated.

```
\\Mattxp\screamernet\screamer_command\ack1
```

Exactly the same as the above line but instead of being a job number it's the 'ACKnowledgement number', the number of this must be the same as the job number.



The diagram below shows how this all links up with our shared network folder we setup earlier.



That's node 1 setup. To setup the other nodes simply copy the 'sn_init_1.bat' file (or whatever you named it) to the 'LightWave' folder on all the other nodes.

Once copied you will need to rename the file to reflect the render node it is going to be, you also need to edit the 'job' and 'ack' numbers to reflect the node number. Depending on how many nodes you have, you should have something like this:

node1



sn_init_1.bat

```
@echo OFF
echo "LightWave ScreamerNet Node 1 Initialisation . . ."
```

```
cd c:\lightwave\programs\
```

```
LWSN -2 -c\\Mattp\screamernet\config_sn -d\\Mattp\screamernet\ \\Mattp\screamernet\screamer_command\job1 \\Mattp\screamernet\screamer_command\ack1
```

node2



sn_init_2.bat

```
@echo OFF
echo "LightWave ScreamerNet Node 2 Initialisation . . ."
```

```
cd c:\lightwave\programs\
```

```
LWSN -2 -c\\Mattp\screamernet\config_sn -d\\Mattp\screamernet\ \\Mattp\screamernet\screamer_command\job2 \\Mattp\screamernet\screamer_command\ack2
```

node3



sn_init_3.bat

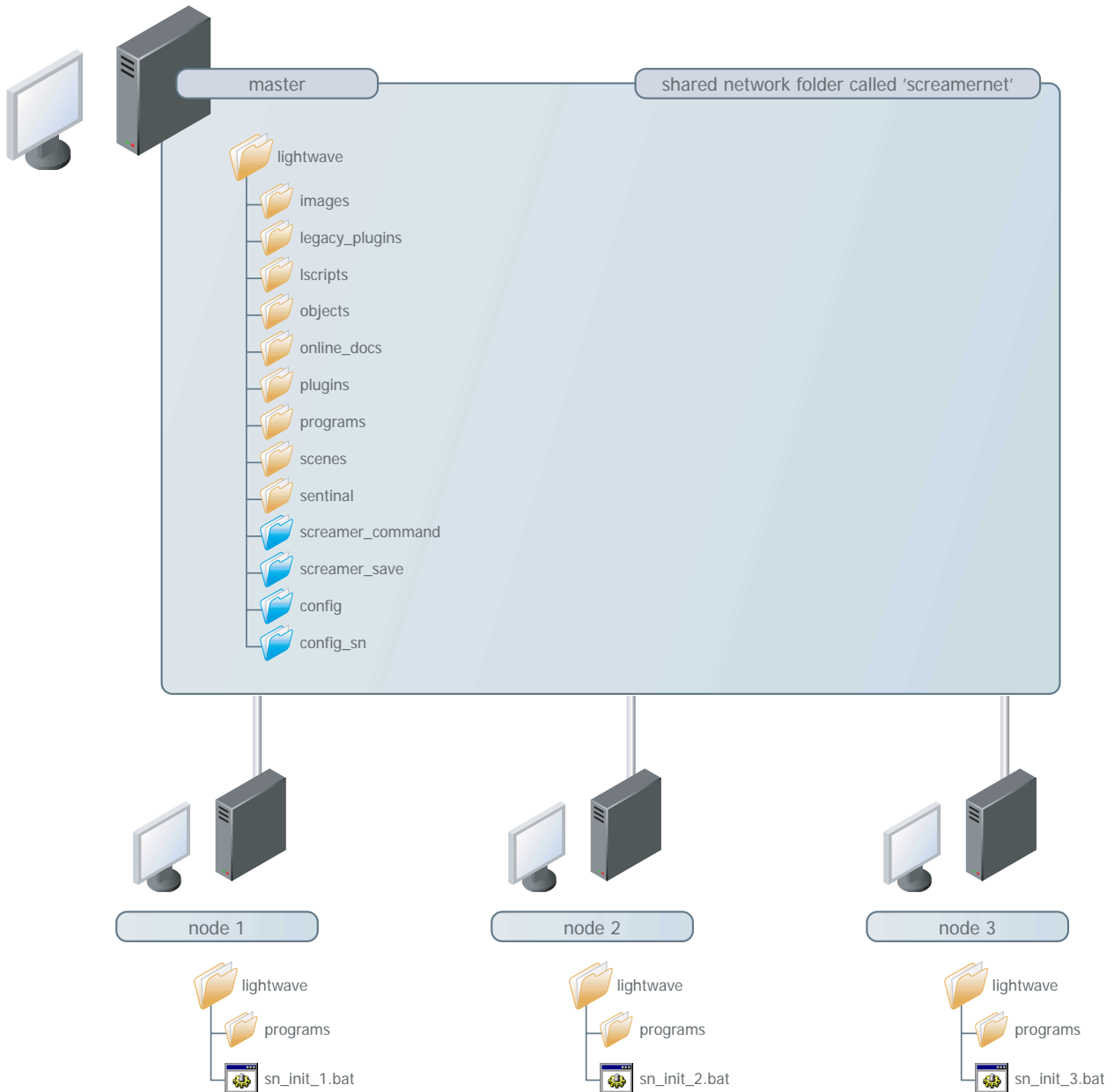
```
@echo OFF
echo "LightWave ScreamerNet Node 3 Initialisation . . ."
```

```
cd c:\lightwave\programs\
```

```
LWSN -2 -c\\Mattp\screamernet\config_sn -d\\Mattp\screamernet\ \\Mattp\screamernet\screamer_command\job3 \\Mattp\screamernet\screamer_command\ack3
```

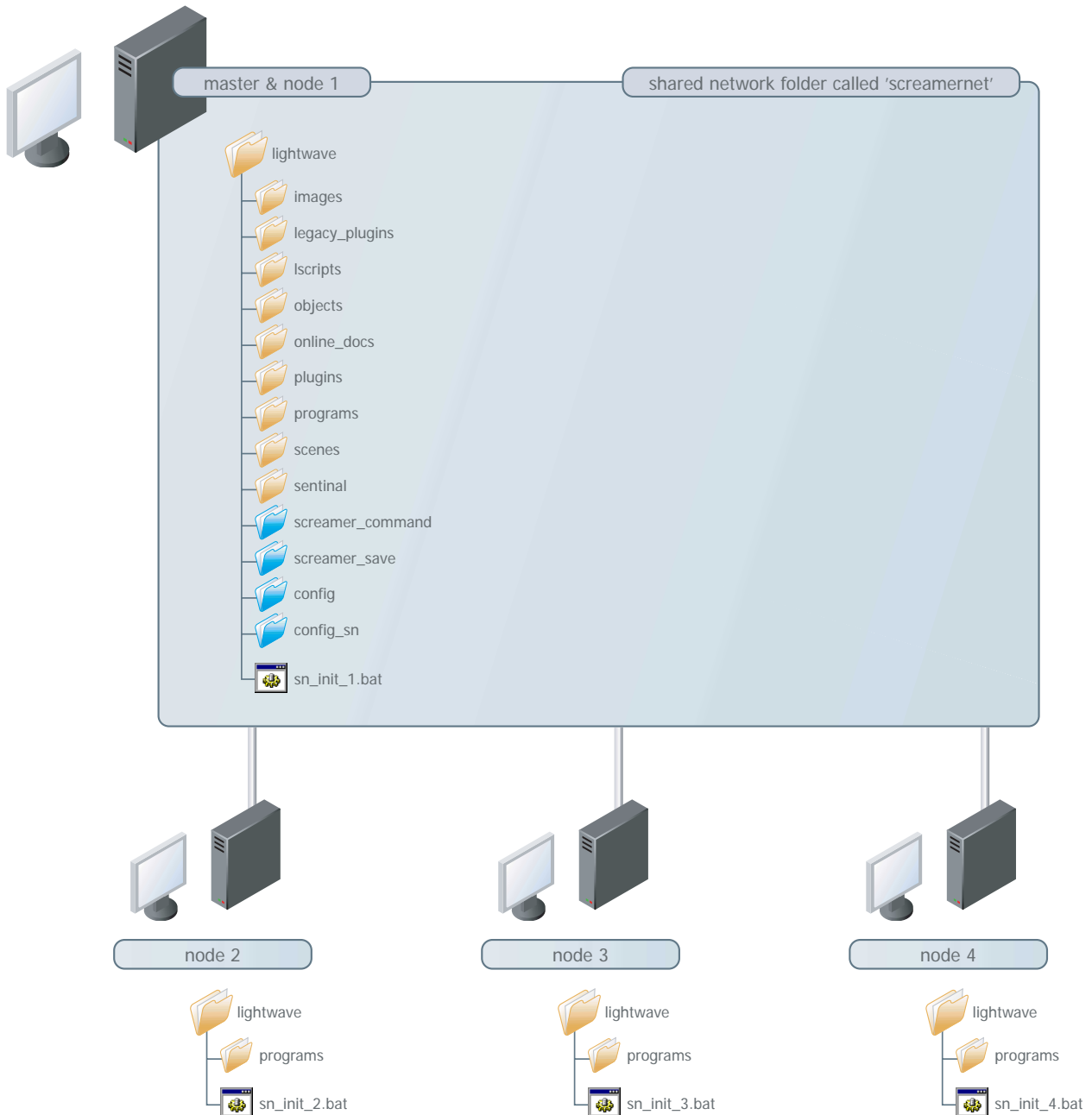


Before we run ScreamerNet through it's paces here's an overview of what we've setup:





Or if you decided to setup the master computer as a node, it will look like this:





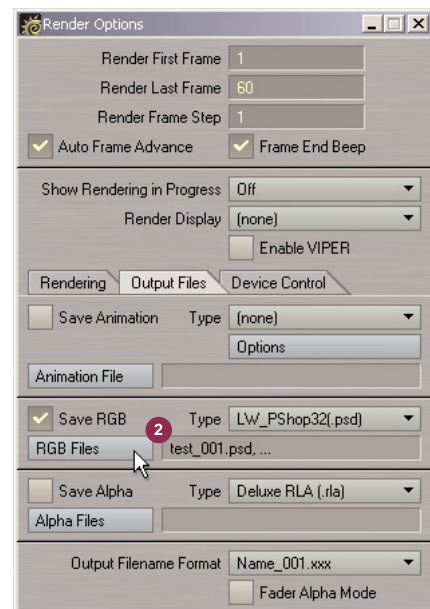
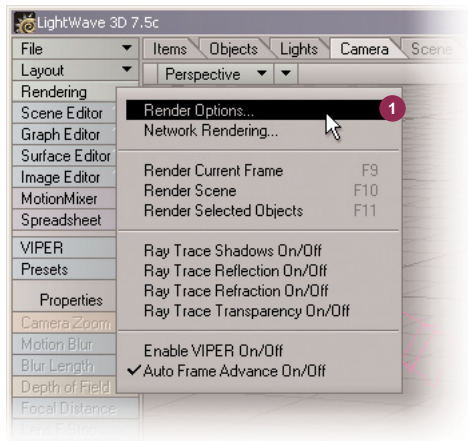
Setting the save location:

There is one last thing you need to do before we run ScreamerNet, and that is set the save directory for the rendered files. You could have all the render nodes save their frames locally, the only downside to this is that you have to go hunting for the frames across the network before you can build them into the final animation.

As we have a shared folder that all the machines can see, it makes sense to use that, then all the frames are stored in one place making them nice and easy to find. As I'm sure you guessed already this is why we created the 'screamer_save' folder earlier.

The only place you can set the save location for the rendered frames is within the scene file itself. So run the version of LightWave using the normal config files, open up the scene you want to render and bring up the render options panel (1) (found under the 'Rendering -> Render Options . . .' popup menu).

In the render options panel under the 'Output Files' tab there is an option to 'Save RGB' (2). Clicking the check box will invoke a file save window, it's here where you need to set the location for the rendered frames, this should ideally point to the 'screamer_save' folder in the shared network folder, although this can be anywhere you like as long as ALL render nodes can see it on the network. With that done save the scene and we're ready to render it using ScreamerNet.





Rendering using ScreamerNet:

Almost there, first a quick check list:

Master Computer:

- Created new folders; 'config', 'config_sn', 'screamer_command', 'screamer_save' in main LightWave folder
- Copied existing config files to new 'config' and 'config_sn' folders
- Set main LightWave folder to share across network using the name 'screamernet'
- Created two shortcuts to LightWave Layout
- Point one shortcut to the config files in the folder 'config', the other shortcut to 'config_sn'
- Run the LightWave Layout using the 'config_sn' config files
- Set the 'command directory' to the folder 'screamer_command' in the shared network folder
- Set the 'content directory' to the shared network folder
- Re-scanned the plugin list from the shared network folder location

Render Nodes:

- Created folder called 'LightWave' on main hard drive
- Copied 'Programs' folder from LightWave folder on main computer to LightWave folder on render node
- Set up render 'batch file' - using 'config_sn' as config folder, shared network folder as content directory and each with unique job/ack numbers

Scene Files:

- Set save folder to 'screamer_save' in the shared network folder

Okay? let's go!

If you can remember that far back, I mentioned that ScreamerNet had two 'parts' to it, the first is the **controller** part that is the network rendering panel in LightWave, the second part is the 'LWSN.exe' program sitting on each node.

When a node is activated it's looking for it's first job file, however there won't be any there until you initialise ScreamerNet in the network rendering panel, but it's pointless initialising it until you've activated all the nodes!

When you run a node is doing two things, looking for job files and sticking it's hand in the air saying "I'm here, I'm over here!" Well not quite but it's waiting to be found by the controller part of ScreamerNet. This is where some confusion lies with ScreamerNet, nodes complain they can't find job files when they're first run, this is perfectly normal. Bearing this in mind double- click each render batch file to activate them.



```
C:\WINDOWS\System32\cmd.exe
"LightWave ScreamerNet Node 1 Initialisation . . ."
LightWave x86 ScreamerNet Module <Build 583>
CPU number: 1

Current directory is now "\\Mattxp\screamernet\".
```

This first thing the node should tell you is where it's looking for the content files, if this is wrong or it comes up with an error close it down and check the script.

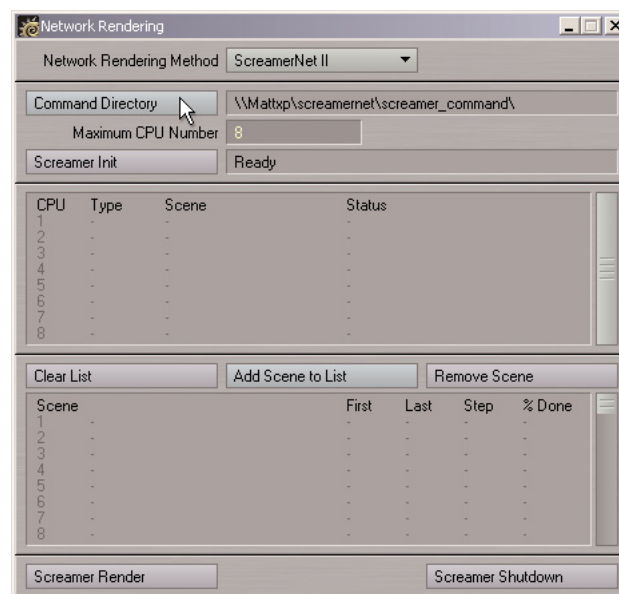


```
C:\WINDOWS\System32\cmd.exe
"LightWave ScreamerNet Node 1 Initialisation . . ."
LightWave x86 ScreamerNet Module (Build 583)
CPU number: 1

Current directory is now "\\Mattxp\screamernet\".
Can't open job file "\\Mattxp\screamernet\screamer_command\job1".
Can't open job file "\\Mattxp\screamernet\screamer_command\job1".
Can't open job file "\\Mattxp\screamernet\screamer_command\job1".
```

the next thing you should see is the node complaining it can't open a job file, that's fine there isn't one to open, but the important thing to note is that the node is running. If the render nodes still repeat the line "Can't open job file" after you have initialised ScreamerNet then they can't find the command folder, close then down and check the path in the script to the command folder, make sure it's visible on the network.

Back to LightWave, run the version of LightWave that uses the configuration files in 'config_sn' and bring up the network rendering panel.



The box called 'Maximum CPU Number' is how many nodes you have setup on your network, so if you have 8 like me, put the number in here. Nodes can be numbered from 1-1000, they don't even have to be in sequence, you could have nodes 1,7 and 68 running but you would HAVE to put 68 as your maximum CPU number NOT 3 as you might think.

Now click the 'Screamer Init' button, ScreamerNet will now check for nodes that are sitting there complaining about job files! If successful you will be told how many CPUs (or render nodes) it has found, if the number isn't correct then check the job/ack numbering in the render script files and try again. If no CPUs are found check that the nodes and network rendering panel are both looking for the same command folder, which should be visible on the network, also check that all machines have access rights to read/write to that folder.

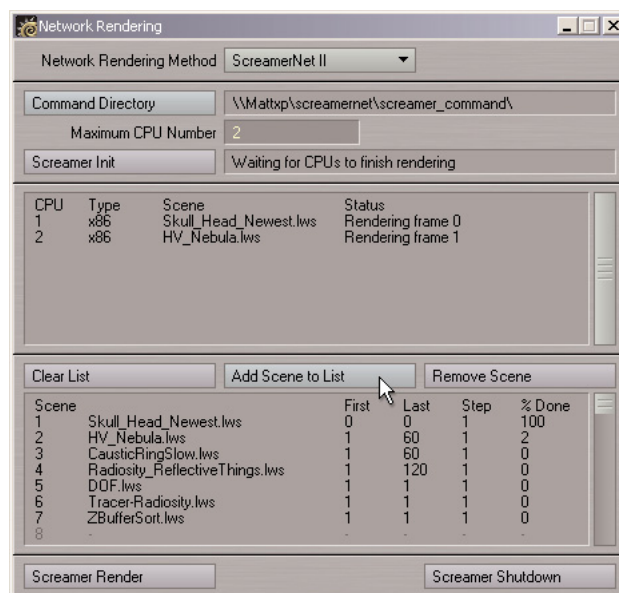


If everything has gone well the nodes should now be happily repeating this:

```
C:\WINDOWS\System32\cmd.exe
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
LightWave command: wait.
```

This means they have successfully found a job file with a command inside saying “wait there and do nothing”!

All that's left to do now is to fill the network rendering panel with scenes, click render and watch it fly! So click the ‘Add Scenes’ button and browse to the projects you want to render. Once you’ve added them all you can click the button you been waiting to click for a while now - ‘Screamer Render’!



The panel will tell you what's going on as it goes, which node is rendering which frame and how far through the rendering it is.

When all the frames have rendered the render nodes will sit there repeating the “LightWave command: wait” line again, at this point you can click the ‘Screamer Shutdown’ button, this will close all the node windows down. You should now have lots of frames in the save folder you specified earlier ready for building into an animation.

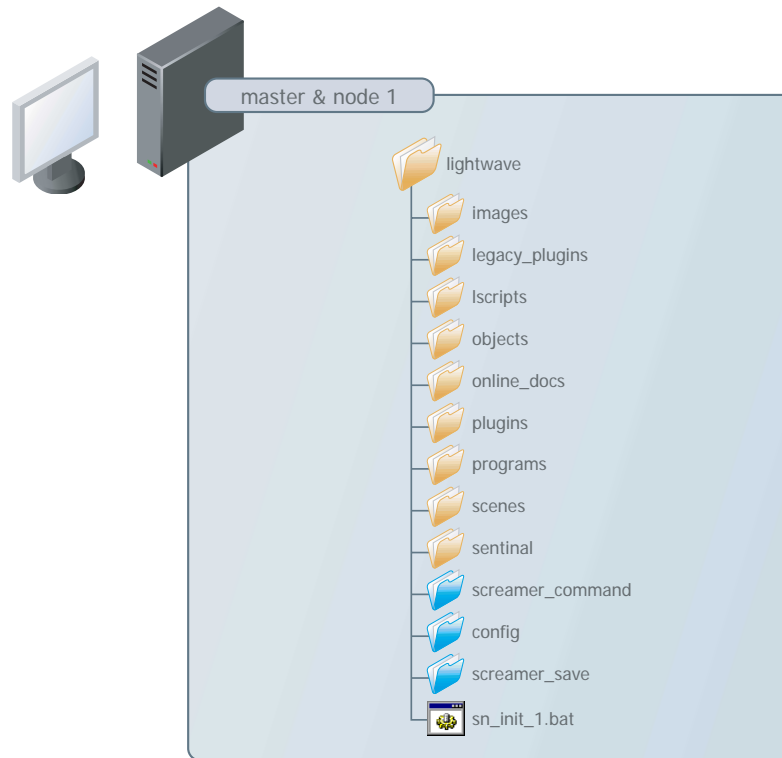
Batch rendering on one machine:

As mentioned in the early pages of this tutorial ScreamerNet is also useful for batch rendering on one machine, this is for users who don't have a network but need to render multiple scenes.

The good news is the process is no different than if you had 1000 computers, there is no bad news!



The overview of the folder structure for a batch render would look like this:



This setup assumes all of your content files are located in the main LightWave folder, and that you have copied all your config files to the new folder 'config', then set the LightWave shortcut to 'point' to that folder as described on pages 8-9.

The only difference with batch rendering on one machine and network rendering is that all of the pathnames to the config, content and command folders can be local, you won't need to re-scan your plugins either since they are already setup for normal LightWave use.

Setting up batch rendering on one machine:

It's still worth reading the whole tutorial so you are familiar with the terminology and concepts, just in case you've nipped straight to this section!

If you haven't already created the folders 'screamer_command' and 'screamer_save'. Now create a batch file (node1) with the following red text (replacing 'c:\lightwave\programs\' with the path to your LightWave install):

```
@echo OFF
echo "LightWave ScreamerNet Node 1 Initialisation . . ."
```

```
cd c:\lightwave\programs\
```

```
LWSN -2 -cc:\lightwave\programs\config -dc:\lightwave c:\lightwave\screamer_command\job1 c:\lightwave\screamer_command\ack1
```

Load the scene files you want to render and set the 'Save RGB' path to 'screamer_save' (assuming you want to save them there) then save the scene.

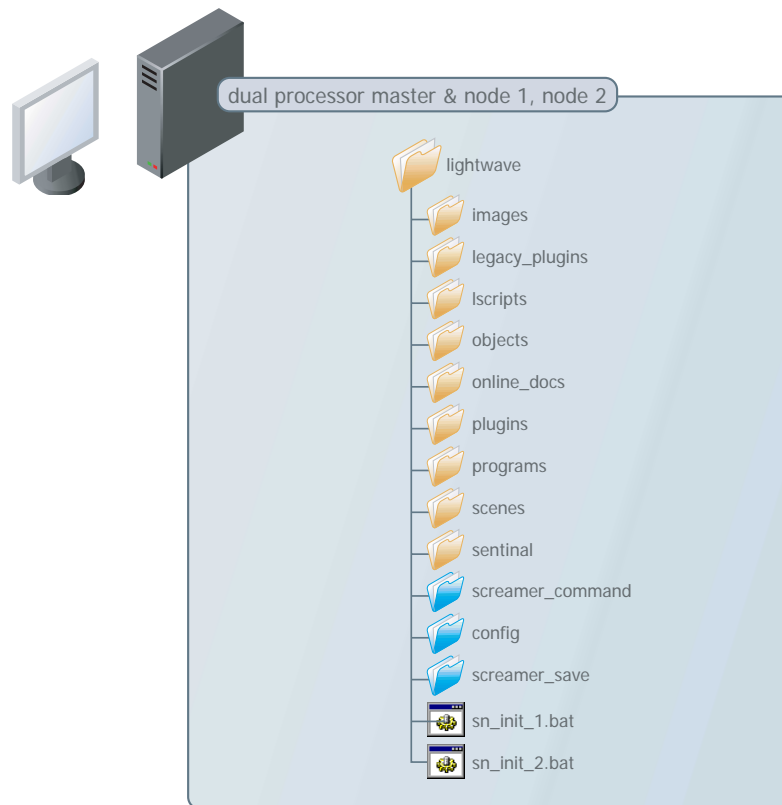
Now go through the process starting on page 19, which is essentially starting the render batch file, then in the network rendering panel in LightWave Layout - set the 'Max CPU Number' to 1, press the 'Screamer Init' button, add the scenes for render then click the 'Screamer Render' button.



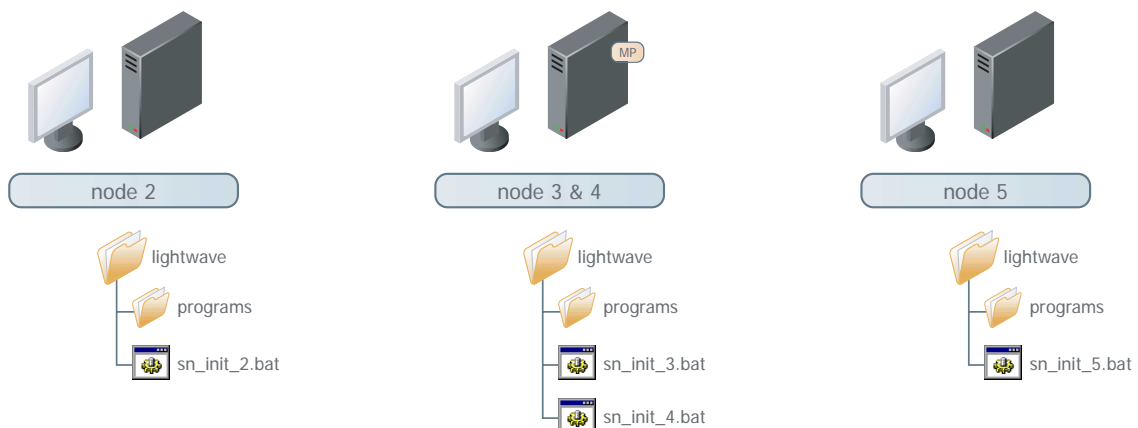
Setting up multi-processor machines:

Whether you have multi-processor machines available on your network, or you want to batch render on one multi-processor machine the process is the same and very simple!

All you have to do is treat each processor as a separate render node. We know that each render node has a batch script file that is run to identify that it is render node 'X'. So if a machine has more than one processor it simply has **more than one** batch script, but with different job/ack numbers. So an overview for a batch render on a multi-processor machine would look like this:



and a network render would be something like:





Troubleshooting and limitations:

Hopefully everything is running smoothly (if not re-read sections that don't click until they do, it's the only way!) but problems can occur, usually because something is setup wrong. Some of the main problems that arise are listed here along with possible remedies.

The biggest limitation to ScreamerNet are plugins that are not written to take advantage of it. I'm not aware of a definitive list of all the ones which do and don't work, if in doubt read the documentation that came with the plugin, it should say whether it has problems or not. If it doesn't say but the plugin still won't seem to be working re-scan your plugins to update the 'LWEXT3.cfg' file, if it still doesn't work then chances are it's not compatible with ScreamerNet.

If your scene uses procedural textures (fractal noise etc.) you may experience differences in the pattern if you render on machines with different processors. Mixing old and new processors, AMD and Intel could all lead to problems. This is really only a problem with animations as textures could suddenly change from frame to frame, if in doubt run a test or use all computers with the same processor type.

Another known problem with ScreamerNet (correct as of v7.5c of LightWave) is it's dislike for scenes that have had Spreadsheet used on them. Plugins like spreadsheet save data to the scene file which causes ScreamerNet to hang. The only way around this is to remove the entry in the scene file made by Spreadsheet.

Scene files are just text files, if you force-load a scene file into a plain text editor you can read it!

Near the top you will find an entry that says:

Plugin MasterHandler 1 .SpreadsheetStandardBanks

followed by:

EndPlugin

There will be another right after that entry that starts with:

Plugin MasterHandler 2 SpreadsheetSceneManager

and if you scroll down there will be another:

EndPlugin

Highlight all the text between the first '**Plugin MasterHandler**' and the last '**EndPlugin**', delete it, then save the file, it should now render okay.

Problem:

"My nodes can't find the job files"

Possible solution:

They can't find the job files because they can't see the command folder. The command folder holds the job files so open up the batch file and check the script lines that end in 'job' and 'ack'. Make sure that pathname before these words point to your command folder. Also check that all the render node computers have access to read and write to the folder.

Problem:

"My nodes seem fine but when I press 'Screamer Init' it can't find any CPUs"

Possible solution:

Again this is a command folder problem, check that the network rendering panel has the same 'command directory' path set that the nodes are pointing to. Also check that the master computer has access to read and write to the folder.

Problem:

"Rendering seems to be working but no files are saved"



Possible solution:

First make sure your scene file is saving **RGB FILES** and **not** an animation (.MOV / .AVI etc.) in the 'Output' tab of the 'render options' panel. ScreamerNet can **ONLY** save files.

Next check that your plugins ('LWEXT3.cfg') file is up to date by re-scanning them using the pathname all the nodes will be using to find them on the network.

Finally check that the render batch files are looking in the correct place for the config files, and that they can access the folder on the network.

Problem:

"Rendering seems to work okay but all my saved files are in .FLX format"

Possible solution:

Two possible reasons why this could be, the first is that ScreamerNet can't find the plugin to save in the format you've specified, so check that your plugins ('LWEXT3.cfg') file is up to date by re-scanning them using the pathname all the nodes will be using to find them on the network. Also check that the render batch files are looking in the correct place for the config files, and that they can access the folder on the network.

The second reason is that the render node that saved the file ran out of memory to load the saver plugin so used the last-resort built in FLX saver.

To convert them you can use LightWave as a converter; load the FLX images into LightWave using the 'Images' panel, highlight the image in question and double-click the preview to open it up in the image view, now you can save in the format you need them in.

Problem:

"My particles aren't working"

Possible solution:

ScreamerNet can have problems with particle FX, best way is to save the particle FX calculation to disk (as a .PFX file) in the folder with the scene file.

Problem:

"Motion designer isn't working"

Possible solution:

ScreamerNet can have problems with motion designer, best way is to save the motion designer calculation to disk (as a .MDD file) in the folder with the scene file.



I hope it made some sense, if not drop me an email!

Take it easy
Matt
(matt@creative-design.co.uk)