SquidNet

Network Render Manager

SquidNet Step-by-step Instructions

(DRAFT)

Version 1.0
November 27, 2011

Table of Contents

About this document:

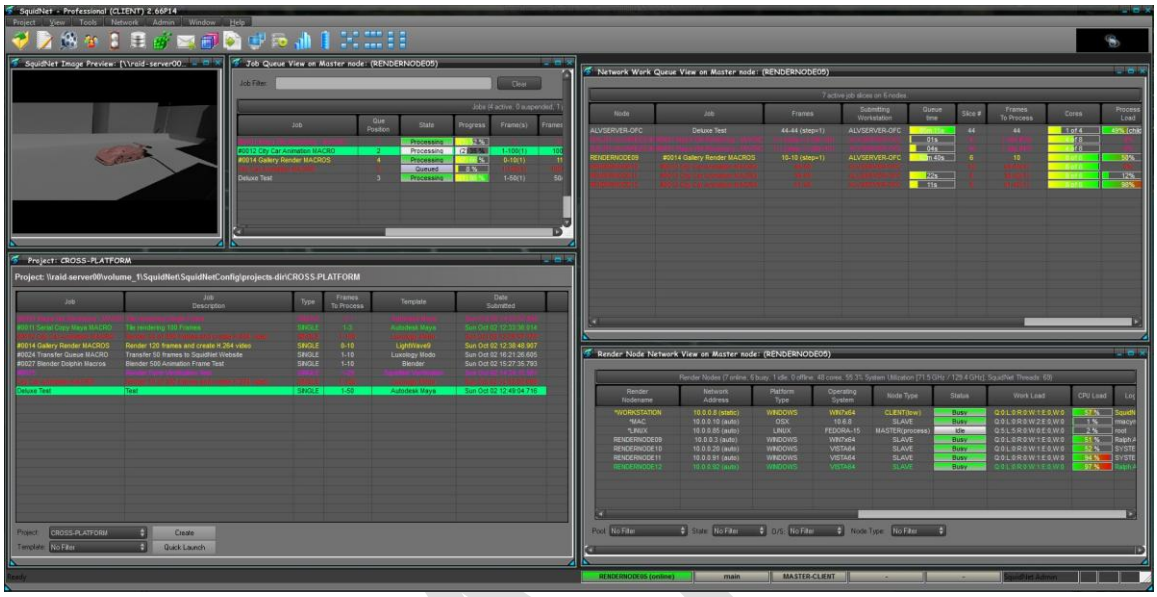
This document will provide step-by-step instructions on how to setup SquidNet for a typical render farm configuration.

It's assumed that the reader is familiar with the rendering and compositing applications that are installed on their system and, as such, this manual will not attempt to cover any material related to those applications.

For any suggestions or comments, feel free to contact us; contact information can be found in the Help menu in the SMC console and on our website at http://www.squidnetsoftware.com/.

Always check website http://www.squidnetsoftware.com for the latest documentation updates.

1 Introduction



This document will provide instructions for setting up your render farm with SquidNet using a few rendering applications as examples. This document assumes the following:

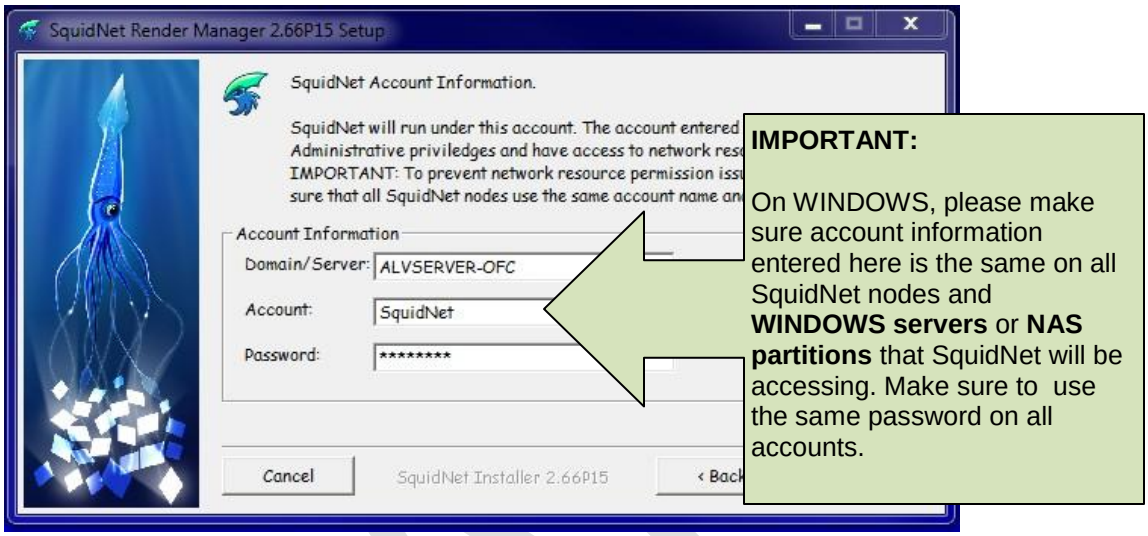
- Rendering applications are already installed and configured on each computer.
- SquidNet is installed on each computer with the appropriate licenses installed.
- The SquidNet license server has been properly configured on one of the local nodes. Preferably one the MASTER node. See Admin->Preferences dialog.
- The local SquidNet configuration path has been set to a path outside the SquidNet installation folder. Preferably on a network storage folder. This is helpful so that system settings are not lost if you need to uninstall/re-install SquidNet. See Admin->Preferences dialog.
- For WINDOWS machines, all SquidNet ADMIN accounts will use the same user name and password: (i.e. username: SquidNet password: SquidNet)
- All network destination paths (UNC, SAMBA, mapped drives, etc…) have been properly configured.
- On LINUX and OS X machines, access to all network shares have been properly configured (/etc/fstab, mount points, etc...)

Also, the instructions provided here are shown for specific applications. However, the same guidelines apply to all applications.

2 Network Permission Settings

2.1 Network Accounts

On WINDOWS based farms, the one thing that drives render farm administrators crazy is network permission problems. In order for WINDOWS machines to access each other's network content the correct security settings must in place beforehand. In most cases, just setting up the correct account information is good enough. For SquidNet on WINDOWS platforms it's important to install SquidNet using on the same account (and password) information on **ALL** nodes. Also, if accessing network share content on a WINDOWS server machine (WIN2003, WIN2008, etc...) **those server nodes MUST have the same account that was used during the SquidNet installation procedure**.



Before setting up SquidNet, we need to make sure that SquidNet has read/write access to network folders where scene output content will be written to. To do this we'll use the Access Check tool under the Tool->Diagnostics menu.

PLACE PICS HER

If test fails on WINDOWS, make sure that the same account (and password) used to install SquidNet exists on the server where the shared folder exists.

3 Application and Folder paths

All render farm managers need to perform one basic function: call specific applications to process job requests and let those applications know how to access the content of those job requests. In a render farm environment in which all the computer operating systems are the same, this is generally not an issue because each node can access content when provided a familiar application name or folder path. In mixed operating system environments, things aren't as easy. For example, consider how Maya is used to render a scene on each operating system: Maya uses "render.exe" on WINDOWS, "Render" on OS X and "render" on Linux. So you can see, there's a need to be able to "map" the name of each application so that only one application

name is used when submitting a job request. In a SquidNet job request, this is accomplished by using the $APP() macro.

The same holds true for folder paths. On WINDOWS, paths like S:\MyModoProjects and \\MYSERVER\MyModoProjects are common. On Linux paths like /mnt/MyModoProjects are used and on OS X, /Volumes/MyModoProjects are the norm. All of these paths point to the same physical folder but their syntax is obviously different. So again, when submitting a job to a mixed system render farm, a common name must be used to access each folder accurately. In a SquidNet job request, the $XPATH() macro is used.

3.1 $APP() and $XPATH() macros

So why the need for $APP() and $PATH() macros? In a mixed operating system environment, any jobs submitted from one operating system that must be processed on a different operating system must have a way to translate the correct paths on each node.

The following items need to be configured before your SquidNet render farm is ready for use:

1. The path to your applications on each node must be registered using the **Application Path Manager**. Registering an application's path let's SquidNet know how to submit jobs to an application on a specific machine. When registered, use the **$APP(<profile-name>)** macro in the application field of the job submission form.
2. In the mixed render farm environments (WINDOWS, LINUX, OS X), path translations must be registered using the **Path Translation Manager.** The Path Translation Manager tells SquidNet how to "map" different physical network paths to the same logical path.

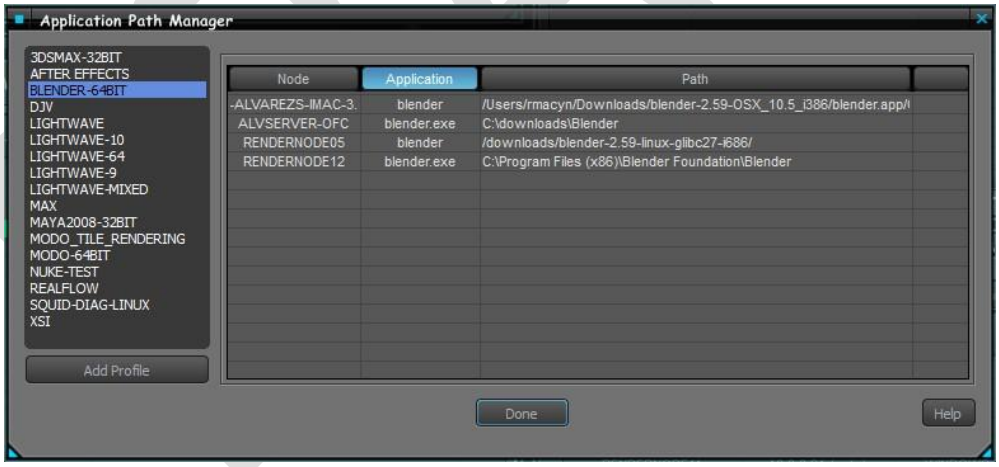
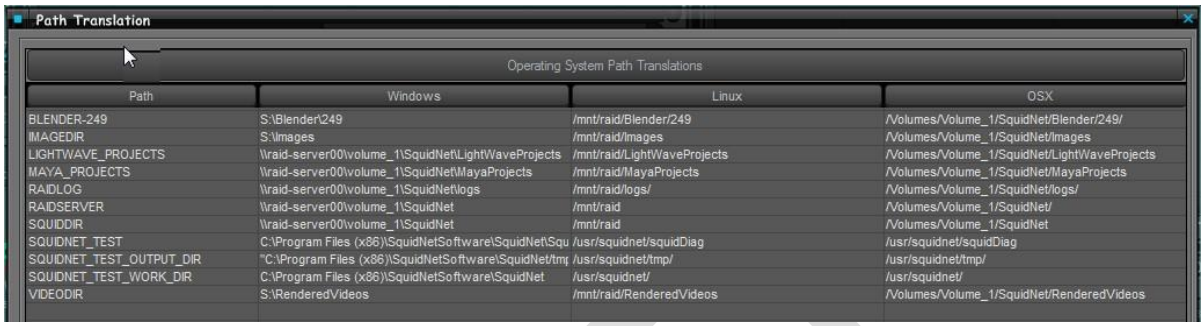


Figure 1. Application Path Manager

Figure 1 above shows the **Application Path Manager** dialog that that contains profiles to each type of application that SquidNet will send requests to. Each profile contains a list of entries that have the following information:

- Name of the computer that the application is installed on
- The name of the processing application (modo_cl.exe, render.exe, etc...)
- The physical path to that application on that machine.

Typically, you want to create a single profile for each of your applications. In your job request, just use the $APP() macro to refer to a specific profile (example: $APP(LIGHTWAVE), $APP(MAYA-32BIT), etc...). You can create as many profiles as you'd like and the same application can be references in different profiles.



Path	Windows	Linux	OSX
BLENDER-249	S:\Blender\249	/mnt/raid/Blender/249	/Volumes/Volume_1/SquidNet/Blender/249/
IMAGEDIR	S:\Images	/mnt/raid/Images	/Volumes/Volume_1/SquidNet/Images
LIGHTWAVE_PROJECTS	\\raid-server00\volume_1\SquidNet\LightWaveProjects	/mnt/raid/LightWaveProjects	/Volumes/Volume_1/SquidNet/LightWaveProjects
MAYA_PROJECTS	\\raid-server00\volume_1\SquidNet\MayaProjects	/mnt/raid/MayaProjects	/Volumes/Volume_1/SquidNet/MayaProjects
RAIDLOG	\\raid-server00\volume_1\SquidNet\logs	/mnt/raid/logs/	/Volumes/Volume_1/SquidNet/logs/
RAIDSERVER	\\raid-server00\volume_1\SquidNet	/mnt/raid	/Volumes/Volume_1/SquidNet/
SQUIDDIR	\\raid-server00\volume_1\SquidNet	/mnt/raid	/Volumes/Volume_1/SquidNet
SQUIDNET_TEST	C:\Program Files (x86)\SquidNetSoftware\SquidNet\Squ	/usr/squidnet/squidDiag	/usr/squidnet/squidDiag
SQUIDNET_TEST_OUTPUT_DIR	"C:\Program Files (x86)\SquidNetSoftware\SquidNet\tmp	/usr/squidnet/tmp/	/usr/squidnet/tmp/
SQUIDNET_TEST_WORK_DIR	C:\Program Files (x86)\SquidNetSoftware\SquidNet	/usr/squidnet/	/usr/squidnet/
VIDEODIR	S:\RenderedVideos	/mnt/raid/RenderedVideos	/Volumes/Volume_1/SquidNet/RenderedVideos

Figure 2. Path Translation Manager

Figure 2 above shows the Path Translation Manager. As you can see, there's a few path translation profiles that contain the physical paths on WINDOWS, LINUX and OS X computers that point to the same logical drive on the network. For example:

- Translation profile **MAYA_PROJECTS** maps the following paths on each machine to the same network path:
  - **WINDOWS**: \\raid-server00\volume_1\SquidNet\MayaProjects
  - **LINUX**: /mnt/raid/MayaProjects
  - **OS X**: /Volumes/Volume_1/SquidNet/MayaProjects

In a mixed operating system environment, use the path translations to allow SquidNet to properly map job requests between operating system paths. In your job request, just use the $XPATH() macro to refer to a specific translation profile (example: $XPATH(MAYA_PROJECTS), $XPATH(LIGHTWAVE_PROJECTS), etc...)

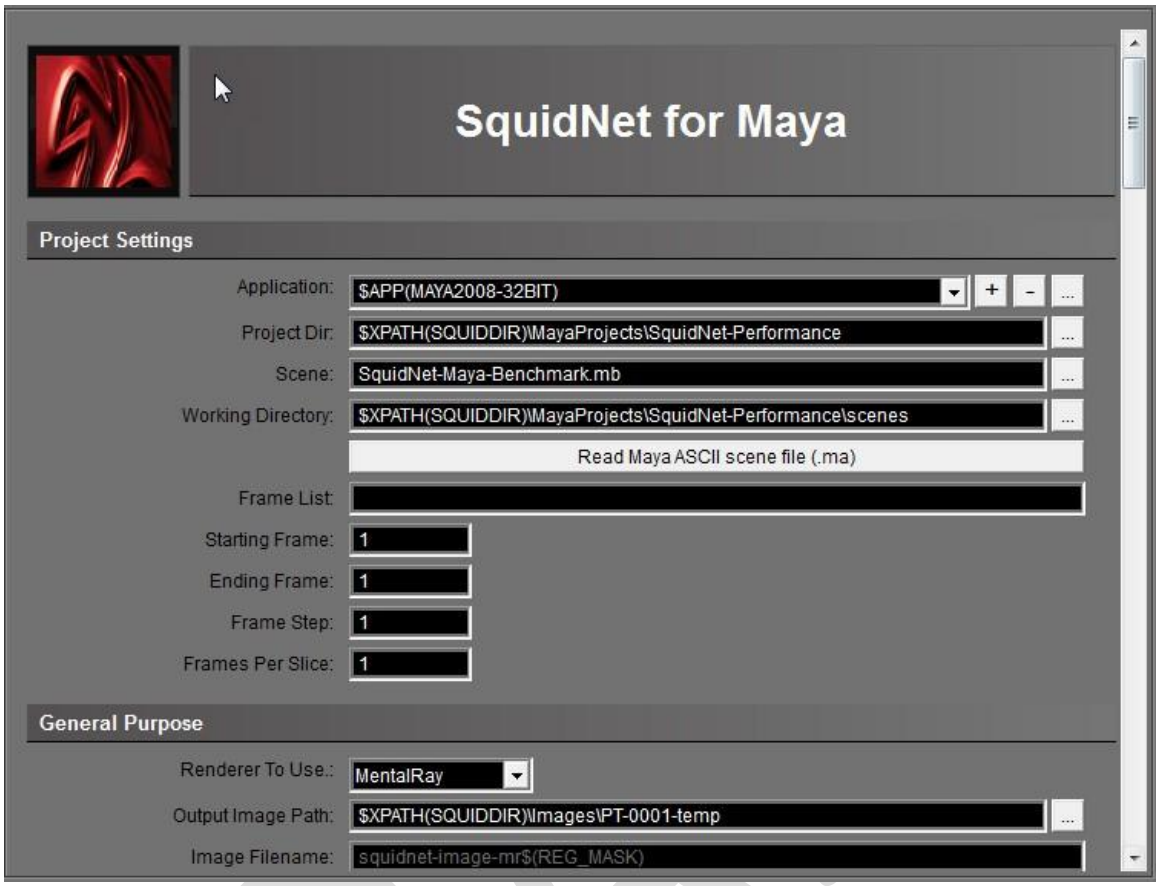


Figure 3. Maya job template using application ($APP) and path ($XPATH) translation macros.

Figure 3 above shows what a typical cross-platform job with $APP() and $XPATH() macros looks like. To allow SquidNet to seamlessly process mixed environment jobs requests, $APP() and $XPATH() macros must be used. In the example shown, the macros translate as follows:

- **$APP(MAYA2008-32BIT)** maps to the following applications:
  - OS X Maya command line renderer application (Render) installed on **/Applications/Autodesk/maya2011/Maya.app/Contents/MacOS** folder on a specific OS X based node.
  - WINDOWS command line renderer application (render.exe) installed on **C:\Program Files\Autodesk\Maya2011\bin** on a specific WINDOWS based computer.
- **$XPATH(SQUIDDIR)** maps to the follow paths (all point to same location on the network):
  - **\\raid-server00\volume_1\SquidNet** on a WINDOWS-based machine.
  - **/mnt/raid/SquidNet** on a LINUX-based machine.
  - **/Volumes/Volume_1/SquidNet** on an OS X based machine.