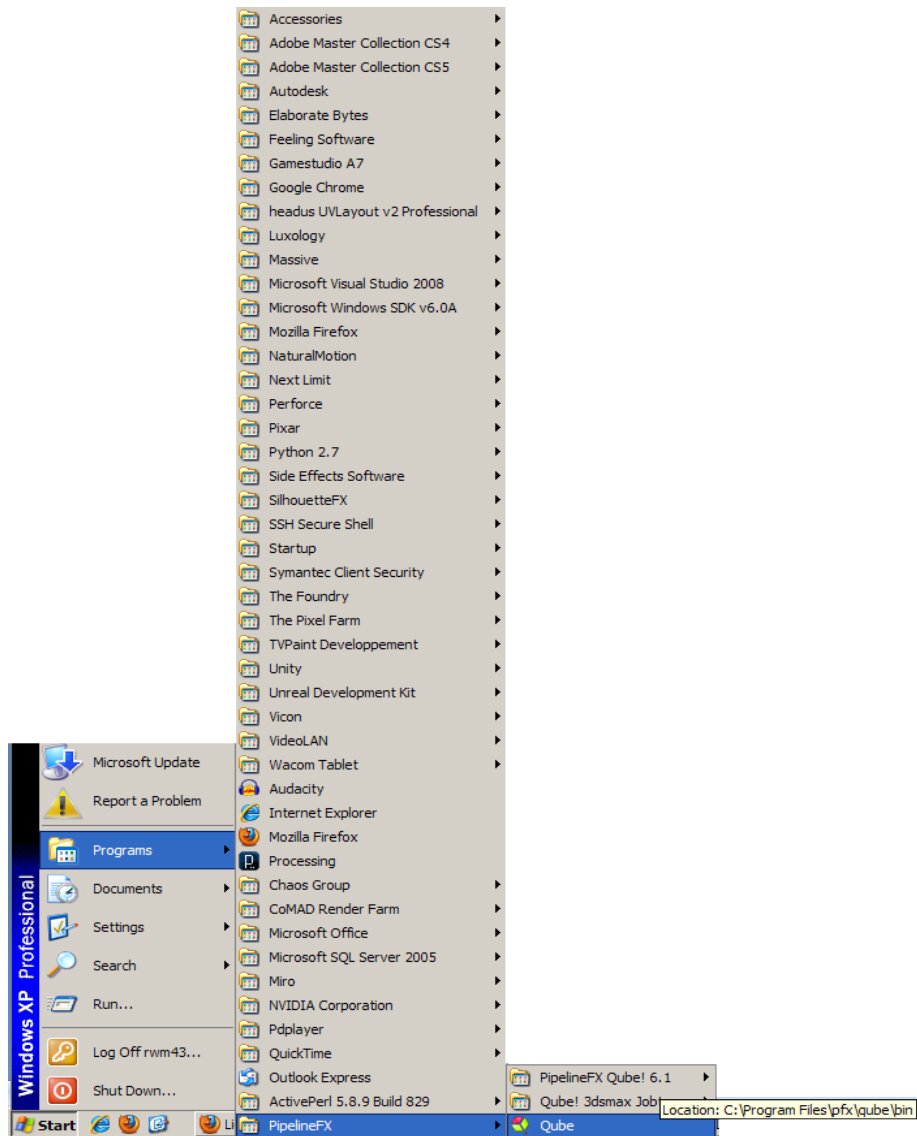


Renderfarm through the Qube! Interface – Ross Macaluso – June 2011



Qube Location

Opening Qube: For most of you Qube should be located as a shortcut on the desktop, however sometimes it's just hiding.



Qube Interface

Once Qube is opened you will be prompted with a screen that looks like this:

From the above image we can see that blue highlighted text are jobs that have been completed, red highlighted text are jobs that have failed, brown highlighted text are jobs that have been cancelled or “killed” and green highlighted text are jobs that are currently running.

First thing that we want to do once we open the program is check the box that I have highlighted in black that says “user” this will tell Qube to only display jobs that are linked to your user name, which will clean up all the completed, killed, failed, and running jobs currently started by other users. This isn’t necessary but I find that when adding jobs it gets messy if you are displaying both your jobs and the jobs currently in progress.

The other boxes that are highlighted in black will turn on or off the display of completed, killed, failed, running or currently pending “incomplete” jobs. These are very helpful when you don’t want to see your previously sent jobs and only want to focus on the ones you are currently running.

The buttons labeled “refresh” and “refresh selected” allow you to refresh all the jobs that are on the farm currently, displaying any sub jobs that are completed or updated newly added jobs. Refresh Selected allows you to select just one job that is currently on the farm, and refresh that one giving you updated information on its progress.

Subjobs – When you send a job you will designate how many computers to send the job to, we currently have 26 available computers. Each one of the computers that you send the job to is known as a sub processor and the job that it gets is the sub job. In basic terms it is the chunk of frames that each computer gets.

The last function that we need to discuss is the search bar that you can see in the top right corner. I don’t find this terribly useful since it searches based on the jobs name, and I just send my jobs through as “maya batch render” or “cmdrange” thus all my jobs are named the same thing. However, if you start naming your jobs you can use this search bar to find them.

The screenshot displays the Qube 6.1.2 Supervisor rendermaster interface. The main window shows a table of jobs with columns for Name, Jobs, % Jobs Done, % Tasks, Status, Sta..., Priority, User, Jobs, Times, Timestart, Timecomplete, Timeelapsed, Ets, remaining, Ets, #P, #B, #R, #C, #F. Below the main table is an 'Agenda/Frames' tab showing a table of frames with columns for Order, Name, Status, Started, Elapsed, Completed, Host, SubjobID, and Output.

Job ID	Name	Jobs	% Jobs Done	% Tasks	Status	Sta...	Priority	User	Jobs	Times	Timestart	Timecomplete	Timeelapsed	Ets	remaining	Ets	#P	#B	#R	#C	#F
1425	maya batchrender	1	0% (0/1)	0% (0/2)	killed	P	9999	rmw43	1425	2011-05-31 05:46:38 AM	2011-05-31 05:46:38 AM	2011-05-31 05:46:38 AM	3:49:24				2	0	0	0	0
1424	maya batchrender	1	0% (0/1)	0% (0/2)	killed	P	9999	rmw43	1424	2011-05-31 05:45:50 AM	2011-05-31 05:45:53 AM	2011-05-31 05:46:28 AM	0:00:35				2	0	0	0	0
1423	maya batchrender	1	100% (1/1)	100% (3/3)	complete	C	9999	rmw43	1423	2011-05-31 05:44:29 AM	2011-05-31 05:44:29 AM	2011-05-31 06:58:56 AM	11:14:27				0	0	0	3	0
1422	maya batchrender	1	0% (0/1)	0% (0/3)	killed	P	9999	rmw43	1422	2011-05-31 05:44:02 AM	2011-05-31 05:44:03 AM	2011-05-31 05:44:18 AM	0:00:15				3	0	0	0	0
1421	maya batchrender	1	100% (1/1)	100% (3/3)	complete	C	9999	rmw43	1421	2011-05-31 05:43:13 AM	2011-05-31 05:43:13 AM	2011-05-31 08:35:59 AM	2:52:46				0	0	0	3	0
1420	maya batchrender	1	100% (1/1)	100% (4/4)	complete	C	9999	rmw43	1420	2011-05-31 03:27:07 AM	2011-05-31 04:02:27 AM	2011-05-31 08:09:37 AM	2:07:10				0	0	0	4	0
1419	maya batchrender	1	100% (1/1)	100% (4/4)	complete	C	9999	rmw43	1419	2011-05-31 03:26:59 AM	2011-05-31 03:26:59 AM	2011-05-31 05:03:32 AM	1:36:33				0	0	0	4	0
1418	maya batchrender	1	100% (1/1)	100% (4/4)	complete	C	9999	rmw43	1418	2011-05-31 03:23:49 AM	2011-05-31 03:23:49 AM	2011-05-31 05:42:59 AM	2:17:10				0	0	0	4	0

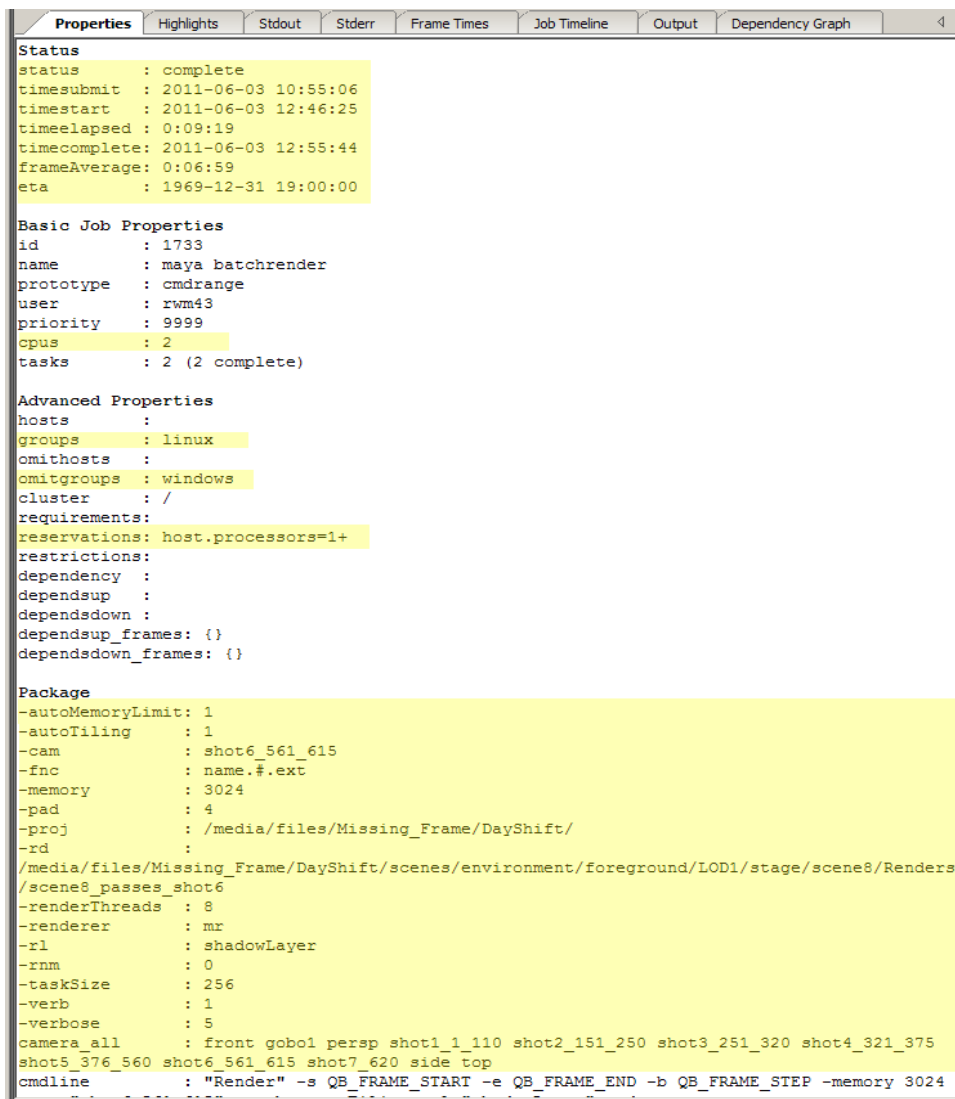
Order	Name	Status	Started	Elapsed	Completed	Host	SubjobID	Output
1	1-30	complete	2011-05-31 03:19:14 AM	3:17:46	2011-05-31 06:37:00 AM	render03	2	
2	31-60	complete	2011-05-31 03:19:14 AM	5:09:47	2011-05-31 08:29:01 AM	render14	1	
3	61-90	complete	2011-05-31 03:19:14 AM	3:14:08	2011-05-31 06:33:22 AM	render05	3	
4	91-110	complete	2011-05-31 03:19:14 AM	2:18:42	2011-05-31 05:37:56 AM	render10	0	

When you select a job you will be presented with new information under the agenda/frames tab which is located on the bottom. If you don’t see this panel, you can enable it by pressing *shift + ctrl + f* or going to view --- > toggle job frames panel.

The information displayed here are the current computers on the render farm that are running your job, the frames that have been allocated to them, if they have failed, been killed, completed, or are still running. If you click the sub job processes tab within this panel you will see similar information, here it just lists which computers are running your job.

On the right hand side you will be presented with a bunch of tabs ranging from properties, to callbacks. Properties, StdOut, StdErr are the three important tabs that require farther explanation. Frame Times displays a bar graph with frame number along the bottom, and time it took to complete along the left. Job Timeline gives you another bar graph with time on the bottom and processor/computer on the right. This will tell you how long it took to complete the chunk of frames assigned to the specified computer, the last tabs I wouldn't worry about.

Properties –



```
Properties Highlights Stdout Stderr Frame Times Job Timeline Output Dependency Graph
Status
status      : complete
timesubmit  : 2011-06-03 10:55:06
timestart   : 2011-06-03 12:46:25
timeelapsed : 0:09:19
timecomplete: 2011-06-03 12:55:44
frameAverage: 0:06:59
eta         : 1969-12-31 19:00:00

Basic Job Properties
id          : 1733
name       : maya batchrender
prototype  : cmdrange
user       : rwm43
priority   : 9999
cpus       : 2
tasks      : 2 (2 complete)

Advanced Properties
hosts      :
groups     : linux
omithosts  :
omitgroups : windows
cluster    : /
requirements:
reservations: host.processors=1+
restrictions:
dependency :
dependsup   :
dependdown  :
dependsup_frames: {}
dependdown_frames: {}

Package
-autoMemoryLimit: 1
-autoTiling      : 1
-cam             : shot6_561_615
-fnc            : name.#.ext
-memory         : 3024
-pad            : 4
-proj           : /media/files/Missing_Frame/DayShift/
-rd             :
/media/files/Missing_Frame/DayShift/scenes/environment/foreground/LOD1/stage/scene8/Renders
/scene8_passes_shot6
-renderThreads  : 8
-renderer      : mr
-rl            : shadowLayer
-rnm           : 0
-taskSize      : 256
-verb         : 1
-verbose       : 5
camera_all    : front gobol persp shot1_1_110 shot2_151_250 shot3_251_320 shot4_321_375
shot5_376_560 shot6_561_615 shot7_620 side top
cmdline       : "Render" -s QB_FRAME_START -e QB_FRAME_END -b QB_FRAME_STEP -memory 3024
```

The screenshot above is of the properties tab that will appear on the right hand side of the Qube interface, within this tab (when you have a job selected) you will be given the information that you have sent Qube through the job dialog box. This information read-out might be a bit daunting so we will go through some of it step-by-step. The parts highlighted with yellow are the sections that I will discuss below.

Starting from the top:

Status is either completed/killed/failed/running

Below status are read-outs of when you submitted the job, the expected time it will be done, the time that the job has been running, or the time it ran for, and the time that the job actually started rendering.

Frame Average Time: this isn't the time that it takes to complete 1 frame, instead it is the time that it takes to complete 1 sub job, or the time it took for 1 computer to render however many frames you sent to it (you can specify this number in the job dialogue box, which will be covered later).

Basic Job Properties:

The only information that you need to worry about here is the (cpus) which shows you how many computers you sent the job to, however, this information is also available under the agenda/frames panel.

Advanced Properties:

Groups: The included operating systems that are going to be used with your job, for the purpose of our render farm 22 of the computers on the render farm are within the Linux operating system, 4 are on windows.

Omitted Groups: The operating system/'s that you are not included within this job, for this particular job windows machines were omitted. This is because windows and linux require a different file path to your scene file, and project, which will be discussed later.

Reservations IS EXTREMELY IMPORTANT, this is basically telling the computers that you send your job to, to use all available processors, if you don't specify this then it will treat each processor as another subjob, thus you will be able to send X jobs to the same computer, which will not only slow your render down but it will also be like running X command line jobs on 1 computer. SET IT TO "**host.processors = 1+**"

Package:

Package is the main settings that you send along with your job. Within this you have your camera that you will be rendering from, your scene file location, your project location, and the render directory.

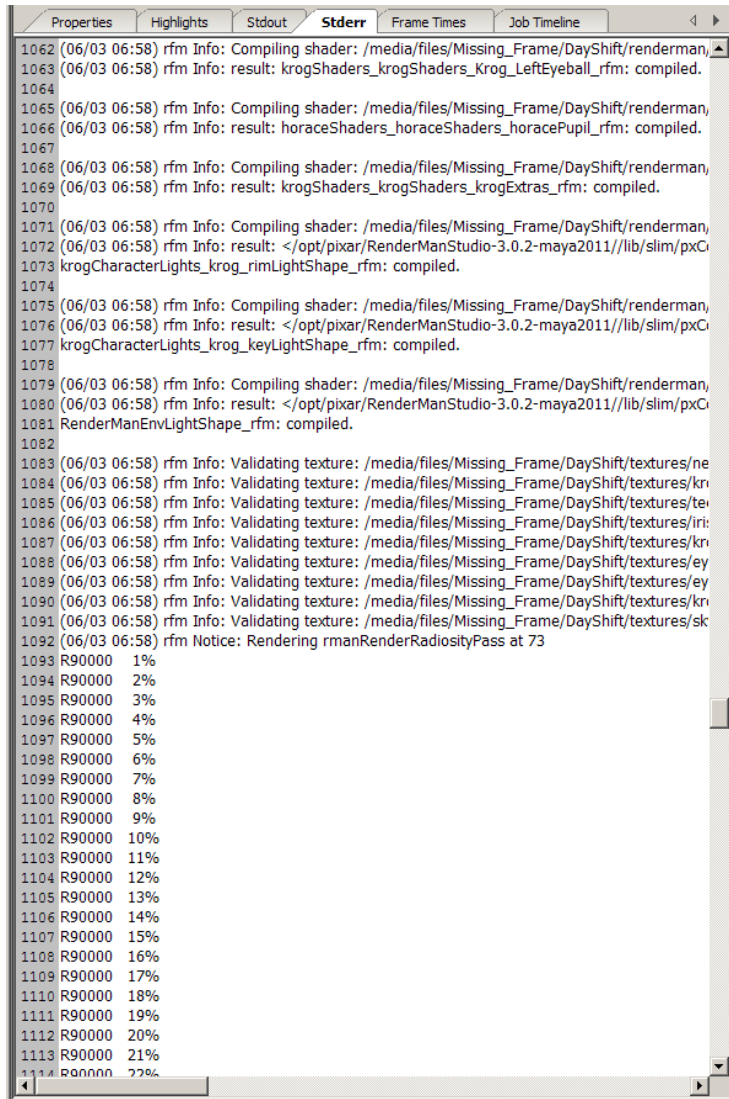
In addition to these project specific settings you will need to tweak the following settings for an optimal render:

I like to check on auto tiling and auto memory this will make sure the computer uses all available memory, and it takes the tiling size from Maya.

renderThreads is very important since without this mentalRay will not find more than 2 processors, so even though the render farm will show that your job was sent to all 4, or 8 processors your job will actually only be running on 2 processors per computer. So set this value to 8, if you are sending a render man job, set "n" to 8.

Verb tells the renderfarm to output the job information, such as the percentage that a frame is at, or if there are any errors. So checking the verb box turns it on for renderman, but for mentalRay you also need to specify your verb level, I like 5 which gives you a very detailed read out. However, if you want less information verb 3 is just the percentages for the current frame; all this information will be accessible in the stderr tab.

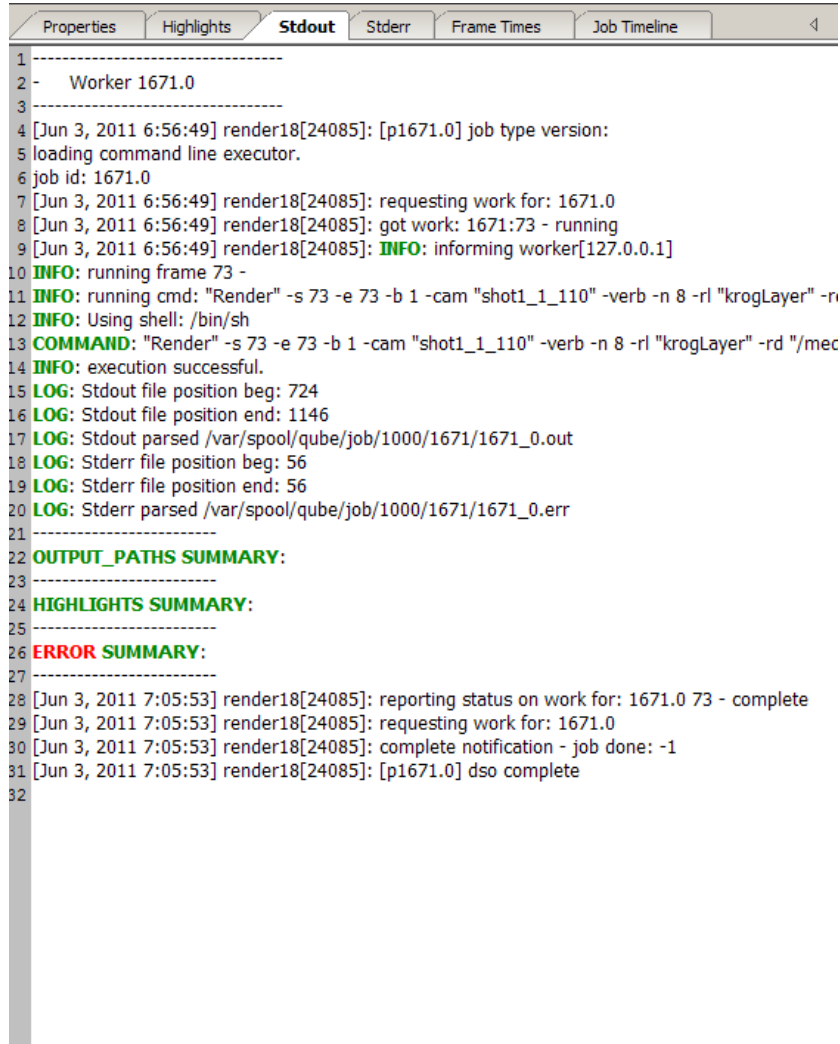
StdErr –



```
1062 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1063 (06/03 06:58) rfm Info: result: krogShaders_krogShaders_Krog_LeftEyeball_rfm: compiled.
1064
1065 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1066 (06/03 06:58) rfm Info: result: horaceShaders_horaceShaders_horacePupil_rfm: compiled.
1067
1068 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1069 (06/03 06:58) rfm Info: result: krogShaders_krogShaders_krogExtras_rfm: compiled.
1070
1071 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1072 (06/03 06:58) rfm Info: result: </opt/pixar/RenderManStudio-3.0.2-maya2011/lib/slim/pxC
1073 krogCharacterLights_krog_rimLightShape_rfm: compiled.
1074
1075 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1076 (06/03 06:58) rfm Info: result: </opt/pixar/RenderManStudio-3.0.2-maya2011/lib/slim/pxC
1077 krogCharacterLights_krog_keyLightShape_rfm: compiled.
1078
1079 (06/03 06:58) rfm Info: Compiling shader: /media/files/Missing_Frame/DayShift/renderman,
1080 (06/03 06:58) rfm Info: result: </opt/pixar/RenderManStudio-3.0.2-maya2011/lib/slim/pxC
1081 RenderManEnvLightShape_rfm: compiled.
1082
1083 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/ne
1084 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/kr
1085 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/te
1086 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/iri
1087 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/kr
1088 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/ey
1089 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/ey
1090 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/kr
1091 (06/03 06:58) rfm Info: Validating texture: /media/files/Missing_Frame/DayShift/textures/sk
1092 (06/03 06:58) rfm Notice: Rendering rmanRenderRadiosityPass at 73
1093 R90000 1%
1094 R90000 2%
1095 R90000 3%
1096 R90000 4%
1097 R90000 5%
1098 R90000 6%
1099 R90000 7%
1100 R90000 8%
1101 R90000 9%
1102 R90000 10%
1103 R90000 11%
1104 R90000 12%
1105 R90000 13%
1106 R90000 14%
1107 R90000 15%
1108 R90000 16%
1109 R90000 17%
1110 R90000 18%
1111 R90000 19%
1112 R90000 20%
1113 R90000 21%
1114 R90000 22%
```

Within the StdErr tab you will find the current frame and percent that the job is at. Any errors that the job has run into, what files it is loading, (which will display any missing textures, caches, shaders). Most of this information is only visible if you turn on verbose (verb) under the settings option which is described above. Mainly this tab is used to see the percentage a frame is on, and if there are any missing textures, shaders, files, or if there are any errors.

StdOut –



The screenshot shows a software interface with several tabs: Properties, Highlights, Stdout (selected), Stderr, Frame Times, and Job Timeline. The Stdout tab displays a log of a rendering job. The log starts with a separator line, followed by the worker ID 'Worker 1671.0'. It then shows the job type version, the loading of the command line executor, and the job ID '1671.0'. The log continues with the worker requesting work for frame 1671.0, receiving work for frame 73, and informing the worker of the frame number. It then shows the command being executed: 'Render -s 73 -e 73 -b 1 -cam "shot1_1_110" -verb -n 8 -rl "krogLayer" -r'. The log indicates that the execution was successful. It then shows the position of the stdout and stderr files. The log ends with a reporting status on work for frame 73, which is complete. The log also includes sections for 'OUTPUT_PATHS SUMMARY', 'HIGHLIGHTS SUMMARY', and 'ERROR SUMMARY', which are currently empty.

```
1 -----
2 - Worker 1671.0
3 -----
4 [Jun 3, 2011 6:56:49] render18[24085]: [p1671.0] job type version:
5 loading command line executor.
6 job id: 1671.0
7 [Jun 3, 2011 6:56:49] render18[24085]: requesting work for: 1671.0
8 [Jun 3, 2011 6:56:49] render18[24085]: got work: 1671:73 - running
9 [Jun 3, 2011 6:56:49] render18[24085]: INFO: informing worker[127.0.0.1]
10 INFO: running frame 73 -
11 INFO: running cmd: "Render" -s 73 -e 73 -b 1 -cam "shot1_1_110" -verb -n 8 -rl "krogLayer" -r
12 INFO: Using shell: /bin/sh
13 COMMAND: "Render" -s 73 -e 73 -b 1 -cam "shot1_1_110" -verb -n 8 -rl "krogLayer" -rd "/mec
14 INFO: execution successful.
15 LOG: Stdout file position beg: 724
16 LOG: Stdout file position end: 1146
17 LOG: Stdout parsed /var/spool/qube/job/1000/1671/1671_0.out
18 LOG: Stderr file position beg: 56
19 LOG: Stderr file position end: 56
20 LOG: Stderr parsed /var/spool/qube/job/1000/1671/1671_0.err
21 -----
22 OUTPUT_PATHS SUMMARY:
23 -----
24 HIGHLIGHTS SUMMARY:
25 -----
26 ERROR SUMMARY:
27 -----
28 [Jun 3, 2011 7:05:53] render18[24085]: reporting status on work for: 1671.0 73 - complete
29 [Jun 3, 2011 7:05:53] render18[24085]: requesting work for: 1671.0
30 [Jun 3, 2011 7:05:53] render18[24085]: complete notification - job done: -1
31 [Jun 3, 2011 7:05:53] render18[24085]: [p1671.0] dso complete
32
```

StdOut tab will display the information that each computer you sent a job to on the renderfarm is currently doing. As you can see denoted by worker 1671 this is the 1671 job sent to the renderfarm, only 1 cpu is being used since there is only 1 worker, denoted by the 0. It is currently rendering frame 73, this number is static and displays the first frame of your range.

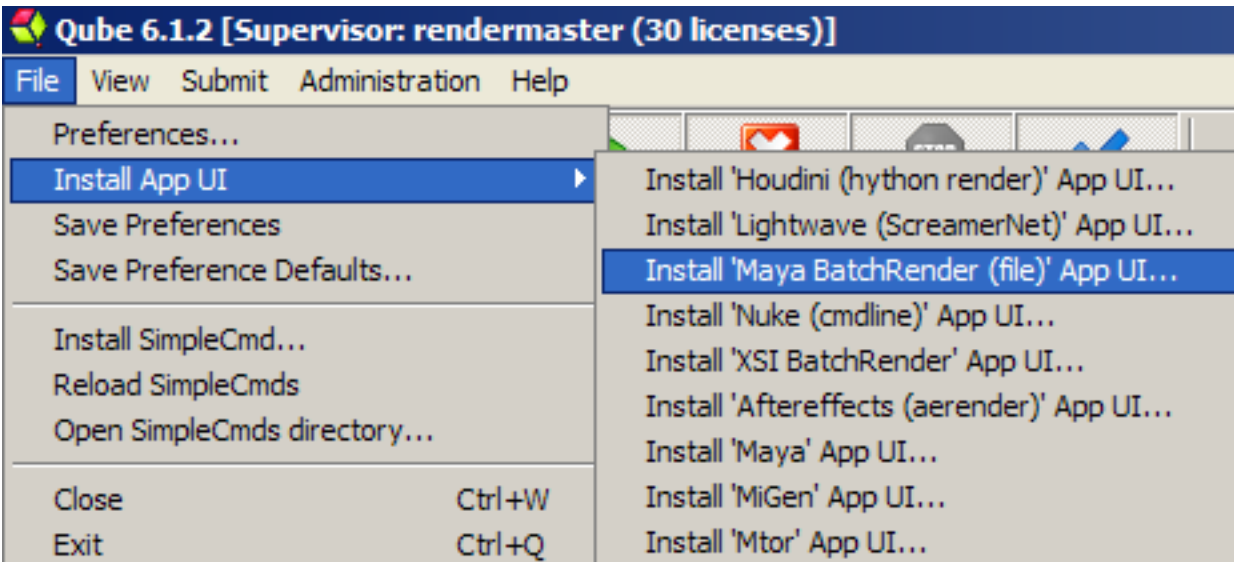
The important information found here is the error summary, and if the job was executed successfully or not, generally I use StdErr for all other updates on my job.

View Tab –

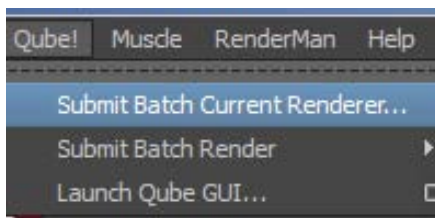
View	Submit	Administration	Help
Refresh			F5
Refresh Selected			Shift-F5
Refresh (Clear Cache)			Alt+F5
Find			Ctrl-F
SearchFilter: Dependent Jobs			Ctrl+d
SearchFilter: Clear			Ctrl+.
Toggle MetaJobs Panel			Shift+Ctrl+m
Toggle Log Panel			Shift+Ctrl+l
Toggle Job Details Panel			Shift+Ctrl+j
Toggle Job Frames Panel			Shift+Ctrl+f
Toggle Farm Worker Details Panel			Shift+Ctrl+w

The view tab contains the ability to turn on/off many panels and tabs within Qube, you can toggle on and off all the panels that I listed above along with refresh jobs that are currently running on the farm. However the most important option under the view tab is the refresh (clear cache) *Alt + F5*, this allows you to clear the Renderfarm's cache file which will show any jobs that you know you correctly added but for some reason aren't displaying. This problem generally only occurs when you are adding multiple jobs with little time between each submission.

Sending Jobs Through Maya



The easiest way to send a Maya job to the farm is to first install in the 'maya batchRender' app, this will allow us to access Qube through Maya.



Within Maya (after you install the app) there will now be a new drop down menu across the top called "Qube!" If this doesn't appear either you need to restart Maya, or install the 'Maya BatchRender (file)' app UI again.

Click on the Qube! Drop down menu and select "Submit Batch Current Renderer" or "Submit Batch Render → mr/rman" the current Renderer is taken from either your layer or the Render Settings.

After you select "Submit Batch Current Renderer" or "Submit Batch Render → mr/rman" the following dialogue box will be displayed:

Submit Maya BatchRender (mr)

Qube Job Basics

Name: maya batchrender
 Priority: 9999
 Subjob Processes: 3

Qube Frame Range

Range: 140-360
 Execution: Chunks with n frames (35)
 rangeOrdering: ascending

Parameters

mayaExe: Render
 scenefile: /media/files/Missing_Frame/DayShift/scenes/environ
 osxMayaEnv: automatic

Common options

verb:
 keepMel:
 renderer: mr
 proj: /media/files/Missing_Frame/DayShift/
 log:
 otherArgs:

General purpose flags

rd: /media/files/Missing_Frame/DayShift/scenes/environ
 im:

Presets Set Presets Clear Expert Cancel Submit

Submit Maya BatchRender (mr)

output format (of):

Frame numbering options

pad: 4
 rfs: 0
 rfb: 0
 naming (fnc): name.#.ext

Camera options

cam: cam_shot4_140_360
 rgb:
 alpha:
 depth:
 iip:

Resolution options

x: 0
 y: 0
 percentRes:
 ard:
 par:

Render Layers and Passes

rl: ambientFogLayer
 rp:
 sel:
 l:

Presets Set Presets Clear Expert Cancel Submit

Submit Maya BatchRender (mr)

Mel callbacks

preRender:
 postRender:
 preLayer:
 postLayer:
 preFrame:
 postFrame:

Other

verbose: 5 : for progress messages
 renderThreads: 8
 autoRenderThreads:
 memory: 0
 autoMemoryLimit:
 taskSize: 0
 autoTiling:
 rnm:
 lic:
 reg:

Qube SimpleCmd and Shell Parameters

Cmd Template

```
%(mayaEnvCommand)s"%(mayaExe)s" -s QB_FRAME_START -e QB_FRAME_END -b QB_FRAME_STEP %(argv)s %(otherArgs)s "%(scenefile)s"
```

Shell (Linux/OSX):

Presets Set Presets Clear Expert Cancel Submit

Submit Maya BatchRender (mr)

Qube Worker Selection

Hosts:
 Groups: linux
 Omit Hosts:
 Omit Groups: windows
 Priority Cluster:
 Host Order:
 Requirements:
 Reservations: host.processors=1+
 Restrictions:

Qube Advanced Job Control

Flags:
 Dependency:
 Email (job complete): rwm43
 Email (failed frames): rwm43
 Blocked:
 Stderr -> Stdout: Send email when any Frame or Subjob Process fails. Send to list of recip
 Job Label:
 Job Kind:
 Process Group:
 Retry Subjob Process: -1
 Retry Frame/Work: -1
 Subjob Timeout: -1

Presets Set Presets Clear Expert Cancel Submit

From this dialogue box you should delete the file name of your scene from the 'Name' field so that the 'Name' says "maya batchrender" you can name the job whatever you want, but leaving it as the default "maya batchrender" is perfectly acceptable.

Don't worry about Priority, leave that field alone.

Change the Subjob Processes to the number of computers on the farm that you want your job to be sent to, keep in mind we only have 22 Linux computers, and 4 windows computers, so be courteous to your fellow classmates and don't use up all the computers available, unless no one else is using the farm.

Range: this is where you will put the frame range that your scene goes to, as you can see my scene for this camera is from 140-360.

Execution: You want to change this to '**Chunks with n Frames**' this will tell the farm to send each computer on the farm N frames, which will "chunk" your range up into parts consisting of N frames. I generally pick 15-50 depending on how long each of my frames takes, however, if you are sharing the farm and doing a large job you would want to pick a much higher number.

Scene File: This is where you want to browse to where your scene file is located, keep in mind though the computers on the farm are **Linux** based so your file paths **NEED** to be in Linux; this will not be an issue if you set a project. An example scene file with a Linux path would be:

```
"/media/files/Missing_Frame/DayShift/scenes/environment/foreground/LOD1/stage/scene8/scene8_v9.mb"
```

make sure your slashes are facing the correct way, and that you have a slash in from of media. Media is equivalent to 'Y:/' and 'files' is digmfiles, thus 'files' will change depending on where your scene file is located.

Verb: Check this box, it stands for Verbose and will "voice" or output the information about the job to the stderr panel that we discussed above. Without this box checked only Qube specific information will be displayed.

Proj: This is where you put your project path, Linux file path again, unless you're sending to a windows computer. An example:

```
"/media/files/Missing_Frame/DayShift"
```

Rd: This is where your render directory will go, or where you want your rendered frames to be placed. An example:

```
"/media/files/Missing_Frame/DayShift/scenes/environment/foreground/LOD1/stage/scene8/Renders/scene8_passes_shot6"
```

Im: you can set this within Maya, but it is renaming your frames on a frame by frame basis, for example, if you wanted all your rendered frames to be called scene1_shot1_ball.XXXX.tga this is where you would put "scene1_shot1_ball"

Output Format (of): This refers to the output file type, so .jpg, .tga, .exr. This can be set within Maya.

Pad: This is where you would set your frame padding, generally this is 1 more digit than your max frames, for example, if you have 2 digits of frames, 20, 10, then you would set your padding to 3 so that your frames will be called name.010, name.011. You can make it just the same amount of digits if you wish, but the purpose of padding is so your frames are read in order by programs such as Nuke, or AfterEffects, without padding sometimes the frames are out of order.

Naming (fnc): This stands for naming (file name convention), this is where you would specify how your frames are handled when multiple frames are required. Generally "name.#.ext" is fine, this will output files called name.framenumber.extension. This can be done within Maya as well.

Cam: Input your camera here; if you created this dialogue box through Maya all your scene cameras should be available within the drop down menu, so just select the camera you wish to render from.

RL: RL stands for RenderLayer, this is where you would select the layer you wish to render from, if you don't use layers, or wish to render from the masterLayer you don't need to input anything into this field. Maya automatically renders from the masterLayer if no layers were created. Likewise, if you created this dialogue box through Maya all your renderLayers should be available through the "choices" box, just check the layer which you wish to render from. **Warning:** don't select more than one layer per job, as it won't render correctly.

L: Stands for layer, I believe this is to specify just one layer, but I just use renderLayer.

MayaCallbacks: This is where you would input any mel scripts that you want to run alongside your job, I don't use these and they can also be specified within Maya.

Verbose: For renderman this input field won't exist, as the checkbox "verb" takes care of it. However, for MentalRay you need to specify a "level" of verbose I usually use 5, but if you wish to have less detailed progress then you can pick a lower value, 3 is just the percents for the frames with no information about what shaders are being loaded, and I believe way less information regarding any errors.

RenderThreads: This is where you need to specify how many threads you want your job to be run on, if you don't put anything here mentalRay will only find 2 processors no matter how many exist on the subJob computer. For our current renderfarm you should input 8. This number should reflect the max amount of processors on the target computer, so if you send to a 16 core computer, increase this number to 16.

Memory/Task Size: I always check automemory and autotasksize just to make sure the job uses all available memory, and it finds the best tasksize for the target computer, Maya's default mentalRay tasksize is 64, this number is similar to the renderMan "gridsize" which for renderMan is calculated by squaring the "bucketSize"

Hosts: This is where you can select only certain computers to send your job to.

Groups: here you will need to select Linux, or Windows depending on which group of computers you are sending your job to. Currently only the Linux computers accept renderMan jobs.

Omit Hosts: This is where you can omit certain computers from sending a job to.

Omit Groups: Set this to Windows if you are sending to the Linux machines so your job doesn't accidentally go to the windows computers, or set this to Linux if you are sending to the Windows computers.

Reservations: This is **VERY IMPORTANT** set this to **host.processors=+1** this will tell the target computer that your sending your job to, to use all available processors. If you leave this at the default of **host.processors=1**, then only 1 processor per specified computer will be used, greatly diminishing your rendering speed. This

can be easily set by clicking browse, and checking the box that says “all” then clicking okay.

All other settings are fine to leave at the default, but feel free to read the Qube help files, and read what the settings do. Don't just change fields unless you are aware of what they do, as some of the fields could drastically alter your render job and potentially harm the farm.

Happy Rendering

Advanced Features

For all the advanced users out there that want a bit more out of their rendering experience, here's a few things that I learned along the way:

If you right click a job you can click "resubmit" instead of adding a job through the Qube interface, or through Maya. Here you can change any of the settings and "resubmit" the job.

Killing a job is done by right clicking and selecting "kill" removing a job is done by doing the same but by selecting "remove" (don't remove jobs unless you have a specific reason, as they are used for diagnostic purposes)

"modify" will allow you to change which computers your job is sent to, and other renderfarm specific options even after you have submitted the job.

If you click the little blue "disk" button on the job dialogue box you can save out a job and use it later. Jobs can be loaded into Qube by going to Submit → Job from File.

Once a job is saved out, it will be saved as an .XJA file. These files can be opened in Notepad or any texteditor and modified. The code is XML, and the tags they use are very easy to understand, if you CTRL + F (find) "range" you will be taken to where most of the information regarding your scene is stored. Such as your cameras, layers, renderlayers, scenefile name. What I do is delete any unused cameras and layers so they don't appear in my choices when I'm creating a job. Likewise you can do quicker editing here than within the dialogue box; when finished you can save out the .XJA for future reference, and just load it back into Qube by doing "job from file."