

List of console commands supported by RandomControl FRYRENDER v1.5

Latest revision: 2009.12.14

by: Chema Guerra

FRYRENDER's technology is provided in the form of two different executables:

1- The standalone UI application (*fryrender.exe*)

2- The console application (*frycmd.exe*)

Render-wise both applications provide the exact same functionality, and mostly the same command-line control arguments. The main difference is that while the standalone provides a fully-fledged User Interface, the console application is a lean and mean, light-weight command-line renderer.

Being so, and despite both apps can be given the same use, the console application (*frycmd.exe*) is more suitable for automated tasks, such as scripted rendering, custom render farming, and similar. In fact, the *NetWarrior* protocol uses *frycmd.exe* to deploy rendering tasks across the network slaves, even across different Operating Systems.

STANDALONE APP (*fryrender.exe*)

fryrender.exe

This command starts an empty instance of the standalone.

```
fryrender.exe  myscene.fry  
fryrender.exe  my scene.fry  
fryrender.exe  "my scene.fry"
```

This command starts an instance of the standalone and loads the scene provided. This is what happens when you double-click on a *RandomControl scene file* in *Windows Explorer*.

```
fryrender.exe  -scene:<path>  
                [-dsi:<path>]  
                [-rgb:<path>]  
                [-w:<int>]  
                [-h:<int>]  
                [-threads:<int>]  
                [-camera:<int>]  
                [-seed:<int>]
```

```
[-passes:<int>]  
[-minutes:<int>]  
[-seq:<string>]  
[-channels:<string>]  
[-xml:<path>]  
[-xmlid:<int>]  
[-prio:{lo|hi}]  
[-start]  
[-quit]  
[-minimized]
```

Some modifiers need a parameter. Parameters must follow these rules:

<path> can be any string representing a local or network file path. Quotes are required in case the path has any spaces. For example, `-scene:file.fry`, `-scene:"file.fry"` and `-scene:"my file.fry"` are all correct parameters, while `-scene:my file.fry` is not, due to the space in between. For safety, always use quotes when specifying path parameters.

<int> is any integer number.

<string> is any text string, *delimited by quotes*.

Some examples:

```
fryrender.exe -scene:scene.fry -dsi:output.dsi -seq:"5,10"
```

This command will start the standalone, open the scene `'scene.fry'`, enable the `.DSI` output path, and initialize the custom frame sequence for animations to the range `[5..10]`.

```
fryrender.exe -scene:scene.fry -dsi:output.dsi -seq:"5,10" -start
```

This command will do the same as the previous one, but will also start the render automatically as soon as the scene is loaded.

```
fryrender.exe -scene:"\\node1\scenes\helicopter\scene.fry"  
-w:800 -h:600 -dsi:"\\node1\output\output.dsi" -minutes:60  
-rgb:"\\node1\output\output.tga" -start -quit
```

This command will start the standalone, open a scene located in a network folder, and set the output `.DSI` and `.TGA` files. It will also set the resolution of the render to `800x600` and make the render start automatically, setting the app to stop and quit after 60 minutes.

```
fryrender.exe -scene:"\\node1\scenes\helicopter\scene.fry"  
-w:800 -h:600 -dsi:"\\node1\output\output.dsi" -minutes:60  
-rgb:"\\node1\output\output.tga" -threads:8 -seed:12345 -start
```

This command will do the same as the previous one, and will also set the number of CPU threads to 8, and the QRN Seed to 12345.

```
fryrender.exe -scene:"z:\scene.fry" -w:800 -h:600 -channels:"C D A M"
```

This command will open a scene, set the resolution to `800x600`, and enable the *Color*, *Depth*, *Alpha* and *MtIID* compositing channels, disabling all other channels.

The string passed to the *-channels* modifier can be any case-sensitive combination of the following characters:

C	<i>Color channel (the render)</i>
A	<i>Alpha channel</i>
N	<i>Normals</i>
O	<i>Object ID</i>
M	<i>Material ID</i>
T	<i>Matte / Shadow</i>
D	<i>Depth channel</i>
K	<i>Mask channel</i>
V	<i>Velocity buffer</i>
B	<i>Ambient Occlusion</i>

Some remarks about the parameters supported and their arguments:

- The *RGB* format used by the *-rgb* parameter is selected implicitly through the extension used (*.TGA*, *.JPG*, ...)
- In case both *-minutes* and *-passes* are used, the render will stop as soon as the first of the two conditions is met.
- In case *-threads* is not specified, the maximum number of threads will be detected and used automatically.
- In case *-seed* is not specified, the QRN Seed will be randomized automatically (so in general it is not necessary to set the seed manually).
- The *-camera* modifier expects ID 0 for the first camera, 1 for the second, etc...
- In general, for those parameters not set, the corresponding original parameter present in the scene file will be used instead.

COMMAND-LINE RENDERER (*frycmd.exe*)

frycmd.exe

This command will do nothing but display the expected syntax for *frycmd.exe*.

```
frycmd.exe      -scene:<path>  
                  [-dsi:<path>]  
                  [-rgb:<path>]  
                  [-w:<int>]  
                  [-h:<int>]  
                  [-threads:<int>]  
                  [-camera:<int>]  
                  [-seed:<int>]  
                  [-passes:<int>]  
                  [-minutes:<int>]
```

```
[-seq:<string>]  
[-channels:<string>]  
[-xml:<path>]  
[-xmlid:<int>]  
[-prio:{lo|hi}]
```

fryrender.exe and *frycmd.exe* share the exact same command-line management routines, so everything explained previously in this document for *fryrender.exe* applies to *frycmd.exe*.

There are a few differences though:

- *frycmd.exe* is a console application and will render silently, without any UI.
- Since there's no UI, the modifier *-minimized* does not apply to *frycmd.exe*.
- Renders always start automatically in *frycmd.exe*, hence the modifier *-start* is not necessary.
- *frycmd.exe* always quits itself once the render is finished, hence the modifier *-quit* is not necessary.
- Besides rendering, *frycmd.exe* also provides some extended functionality to manage *.DSI* files (formerly provided by the now defunct *dsitool.exe*).

frycmd.exe is intended to be used for automated tasks where the user is not expected to interact with the application, making the UI is unnecessary or undesirable. In such situations it may be interesting to be able to *take a look* at what the application is doing. The *-xml* and *-xmlid* modifiers can be used to make the renderer output a *.XML* file with some progress information that can be used to monitor the process from the outside. In fact, these modifiers are used by the *NetWarrior* protocol to keep track of what each node is doing.

The *.XML* information file will be rewritten every minute or so, under the file name set in the *-xml* modifier. The file will contain a field with the value set in the *-xmlid* modifier (which can be used to identify which station produced the file).

The *.XML* file output will look similar to this:

```
<?xml version="1.0" encoding="ISO-8859-15"?>  
<info>  
  <xid>9999</xid>  
  <off>0</off>  
  <app>fryrender</app>  
  <clk>60560</clk>  
  <pas>2277</pas>  
  <eff>519859</eff>  
  <frm>0</frm>
```

```
<end>00:01:00</end>
<prg>75</prg>
<res>720 x 480</res>
</info>
```

- <xid>** is the value specified in the `-xmlid` modifier (or 0 if no value is set).
- <off>** tells whether the process finished or not.
- <clk>** is the number of milliseconds since the process started.
- <pas>** is the total number of passes $\times 100$.
- <eff>** is the efficiency $\times 100$.
- <frm>** is the animation frame being rendered.
- <end>** is the termination condition, formatted properly.
- <prg>** is the percentage completed so far.
- <res>** is the render resolution, formatted properly.

`frycmd.exe -merge:<path> <path> <path>...`

This command will take as many input *.DSI* files as specified (at least 2) and will merge them sequentially into a single, cleaner one.

Example:

```
frycmd.exe -merge:"c:\output\merged.dsi" "c:\output\output1.dsi"
"c:\output\output2.dsi" "c:\output\output3.dsi"
```

This command will merge the *.DSI* files *output1*, *output2* and *output3* into the *.DSI* file *merged.dsi*. Note that for *merged.dsi* to be cleaner than any of the source ones, these must have been rendered with different (random) QRN Seeds. All the *.DSI* files must have been generated from the same scene, at the exact same resolution, etc...

As an important remark, it is highly recommended to invoke *frycmd.exe* locally for merge operations. So you may want to copy all the source *.DSI* files involved in the merge operation to your local hard drive for a much faster merge speed in case the *.DSI* files are hosted in a remote network location.

Also note that a valid *license.ini* file must be supplied for the `-merge` command to merge any *.DSI* files. This *license.ini* file must be reachable by *frycmd.exe* (ie. in the same folder).

`frycmd.exe -dsiinfo:<path>`

This command will display some relevant information about a *.DSI* file such as its resolution and number of samples / passes.

```
frycmd.exe -tonemap:<path> -rgb:<path> -scene:<path>
```

This command will tonemap a *.DSI* file using the tonemapping settings in the scene provided, producing an *RGB* output. Note that a valid *license.ini* file must be provided for the *RGB* output to be unwatermarked.

Example:

```
frycmd.exe -tonemap:"c:\input.dsi" -rgb:"c:\output.tga"  
-scene:"c:\scenes\input.fry"
```

This command will produce only one image regardless of how many *LightMixer* layers the input *.DSI* has set. Also, note that the tonemapping values (exposure, gamma, RGB levels, bloom, glare, etc...) are stored in the scene file, and not in the *.DSI*. Hence, the scene provided must be the same one the *.DSI* file was generated with.