

Rendering NSI file

`renderdl` reads a file containing scene description commands and "executes" them. Such files are commonly referred to **NSI files** (NSI stands for *Nodal Scene Interface*). There are two kinds of NSI files: ASCII encoded and binary encoded. A binary NSI file is smaller than its ASCII encoded equivalent, but an ASCII NSI file has the advantage of being human readable and editable in a text editor.

To render a file named `file.nsi`, just type:

```
renderdl file.nsi
```

Command Line Options

Option	Description
-t n	Specifies the number of threads to use for the rendering. "n" can take any of the following values: n > 0 : Use "n" threads. n = 0 : Use as many threads as there are available cores. n < 0 : Use all but "n" threads. By default, renderdl use as many threads as there are available cores.
-stats	Embed statistics in rendered images. This is supported for EXR and TIFF files only. Statistics are explained in more detail in Detailed Statistics .
-progress	Prints a progress status after each rendered bucket.
-cat	Print NSI commands instead of sending them into the renderer. An example usage is converting a binary (or compressed) NSI into an human readable ASCII version: <pre>renderdl -cat binary.nsi.gz > ascii.nsi</pre>
-cat -binary	Outputs the NSI commands in binary format. For example: <pre>renderdl -cat -binary ascii.nsi > binary.nsi</pre>
-cat -gzip	Outputs the NSI file in compressed form. For example: <pre>renderdl -cat -gzip ascii.nsi > ascii.nsi.gz</pre>
-v	Prints the current version of the renderer.
-h	Prints the following help: <pre>Usage: renderdl [options] [file1 ... fileN] -v : output version to console -h : output this help -q : don't print the name of files rendered -d : add a framebuffer display -lua : input file is a NSI scene in LUA -id : add an i-display interactive framebuffer -stats : save statistics in rendered images -progress : print rendering progress at each bucket -test : render a test image -t n : launch the render using 'n' threads -cat : output NSI to stdout -binary : encode NSI stream in binary format -gzip : compress NSI stream using gzip format</pre>

No File Name Specified

If no file name is specified, `renderd1` reads scene description commands from the standard in. This feature enables piping commands directly in `renderd1`. For example, to enter scene description commands interactively (which is not really practical), do the following:

```
renderd1
Reading (stdin)
<enter commands here>
```

If you wish to pipe the content of `file.nsi` in `renderd1`, type:

```
cat file.nsi | renderd1
```

Shell Return values

The `renderd1` executable will return one of the following values:

Return Value	Description
0	No error.
1	Bad combination of parameters. An error message will explain why.
199	Option "licensing" "waitforlicense" 0 was used and no license was available.
255	The NSI file specified on the command line could not be read.