

# General information

This page provides information on the Corona Standalone.

## Overview

Corona Standalone is a renderer-only application that uses the Corona core to produce images from file inputs. The input format is described here, and Corona for 3ds Max can export it as well. Except for geometry description (which uses binary format and compression for smaller size), the format is textual and human-readable, based on XML.

While currently one has to create the inputs elsewhere and can only use Corona for rendering, we may add some limited editing capabilities to the application in the future.

The application requires activation to work (be it a free 45-day demo, or a full license)

## 3ds Max export limitations

- Cannot export 3ds Max native (nonphysical) lights
- Cannot export 3ds Max native atmospheric effects
- Cannot export Forest Pack, Multiscatter maps
- Export of some maps is not supported
- Some exports slightly differ (for example the noise texture will have slightly different seed)
- Bokeh custom shape is not supported

## Command line parameters reference

There are several modes in which you can run Corona, determined by the first command line parameter:

Parameter	Meaning
-?	Prints usage help
-confHelp	Prints a detailed list of settings for the .conf files (names, types, ranges)
-mtlPreview	Starts the material editor preview mode. This is useful for deep integration into a 3rd party application. In this mode Corona reads settings and material config from the standard input and renders a preview image for the material. This happens in a loop until stdin is closed. In each cycle, <b>1 (ASCII) float, 2 ints</b> , and a string are read from the standard input before the material definition: <b>[quality] [resolutionX] [resolutionY] [filename]</b> Material definition in the XML format ( <b>&lt;material class="..."&gt;...&lt;material&gt;</b> - see <a href="#">..mtl file description</a> ) is then read from stdin until <b>\0</b> is encountered. The file is rendered and saved to the specified filenames. End of rendering is signaled by writing '.' on the standard output. This can be repeated multiple times.
sceneFilename.scn	Starts the regular rendering. This is the most common mode - see details below.

## Regular rendering command line parameters

The first parameter for regular rendering is always a path to a .scn scene definition file. Any number of optional parameters can follow:

Parameter	Meaning
-silent	Runs with no frame buffer displayed and does not wait for user to close the window after rendering. Output is only written to files specified with the -o parameter
-c configuration File.conf	Parses an additional configuration (.conf) file on top of configuration files included from inside the .scn file. Any setting values found overwrite the values from previous .conf files.
-o outputFile.jpg	Saves the result into a file with given name at the end of the rendering. The file format is automatically determined from the file extension. Currently supported formats are: BMP, PNG, JPG, EXR, GIF, TGA, TIFF, HDR.
-oA outputFile.png	Same as -o, but saves the image with Alpha channel. Only some formats support alpha channel.

<b>-oR outputFile.exr</b>	Same as -o, but no tonemapping will be applied to the saved image (which is useful when saving high dynamic range images). Can be combined with <b>-oA</b> : <b>-oAR</b> will save the image with alpha, but without tone mapping.
-------------------------------	--

## XML input format description

- All input XML documents must have single root element and must be well-formed.
- Most names of elements, attributes, materials, ... are case-insensitive.

## Documentation conventions

Symbol	Meaning	Syntax
[float]	Single floating point value	Single number, possibly with a decimal point
[int]	Single integer value	Single number, decimal point is forbidden
[string]	A string value	Any sequence of characters (properly XML-escaped if necessary)
[filename]	A string holding a file name	[string]
[RGB]	RGB color value	[float] [float] [float]
[XY]	2D geometry vector	[float] [float]
[XYZ]	3D geometry vector	[float] [float] [float]
[animXYZ]	Animated 3D geometry vector (for motion blur)	<animatedXYZ samples="[int]"><sample>[XYZ]</sample><sample>[XYZ]</sample>...</animatedXYZ>
[TM]	Affine transformation matrix (4 columns, 3 rows) written row-wise	12 × [float]
[animTM]		dTm samples="[int]"><sample>[TM]</sample><sample>[TM]</sample>...</animatedTm>
[map]	XML sub-tree defining a map	<map class="[string]">...</map>
[material]	XML sub-tree defining a material	<material class="[string]">...</material>
[includeList]	list of XML nodes defining a set of included or excluded scene nodes	Either 1-n <included>[string]</included> or 1-n <excluded>[string]</excluded>
X Y	Either X or Y	

## Element rules

<tag>	Means that the element of type tag must be present exactly once
?<tag>	Means that the element of type tag can be present 0-1 times. There is no "?" in the actual file, we just us it as convenient notation.
*<tag>	Means that the element of type tag can be present 0-n times. There is no "*" in the actual file, we just us it as convenient notation.
+<tag>	Means that the element of type tag can be present 1-n times. There is no "+" in the actual file, we just us it as convenient notation.

## License Agreement

The Software uses "Qt" libraries, licensed under the terms of the GNU Lesser General Public License Version 3 ("LGPL v3"). The LGPL v3 incorporates terms of the Version 3 of the GNU General Public License ("GNU GPL v3").  
On each supported platform the Software is dynamically linked to the Qt Libraries. The Qt Libraries are Copyright (c) The Qt Company Ltd. and other contributors.  
Contact: <https://www.qt.io/licensing/>.  
Full text of the LGPL v3 and the GNU GPL v3 is included in the .txt file in the main installation folder of the Software on your device.  
The source code of the utilized Qt Libraries can be downloaded from <https://corona-renderer.com/link/qt-sources>.