

NAME

blender – a full-featured 3D application

SYNOPSIS

blender [**args ...**] [**file**] [**args ...**]

DESCRIPTION

blender is a full-featured 3D application. It supports the entirety of the 3D pipeline - modeling, rigging, animation, simulation, rendering, compositing, motion tracking, and video editing.

Use Blender to create 3D images and animations, films and commercials, content for games, architectural and industrial visualizations, and scientific visualizations.

<https://www.blender.org>

OPTIONS

Color management: using fallback mode for management Color management: Error could not find role data role. Blender 3.0.1 Usage: blender [**args ...**] [**file**] [**args ...**]

Render Options:

-b or --background

Run in background (often used for UI-less rendering).

-a or --render-anim

Render frames from start to end (inclusive).

-S or --scene <name>

Set the active scene <name> for rendering.

-f or --render-frame <frame>

Render frame <frame> and save it.

* +<frame> start frame relative, -<frame> end frame relative.

* A comma separated list of frames can also be used (no spaces).

* A range of frames can be expressed using '..' separator between the first and last frames (inclusive).

-s or --frame-start <frame>

Set start to frame <frame>, supports +/- for relative frames too.

-e or --frame-end <frame>

Set end to frame <frame>, supports +/- for relative frames too.

-j or --frame-jump <frames>

Set number of frames to step forward after each rendered frame.

-o or --render-output <path>

Set the render path and file name.

Use '/' at the start of the path to render relative to the blend-file.

The '#' characters are replaced by the frame number, and used to define zero padding.

* 'animation_##_test.png' becomes 'animation_01_test.png'
 * 'test-#####.png' becomes 'test-000001.png'

When the filename does not contain '#', The suffix '####' is added to the filename.

The frame number will be added at the end of the filename, eg:

```
# blender -b animation.blend -o //render_ -F PNG -x 1 -a
```

-E or --engine <engine>

Specify the render engine.

Use '-E help' to list available engines.

-t or --threads <threads>

Use amount of <threads> for rendering and other operations

[1-1024], 0 for systems processor count.

Format Options:

-F or --render-format <format>

Set the render format.

Valid options are:

Formats that can be compiled into Blender, not available on all systems:

-x or --use-extension <bool>

Set option to add the file extension to the end of the file.

Animation Playback Options:

-a <options> <file(s)>

Instead of showing Blender's user interface, this runs Blender as an animation player, to view movies and image sequences rendered in Blender (ignored if '-b' is set).

Playback Arguments:

-p <sx> <sy>

Open with lower left corner at <sx>, <sy>.

-m

Read from disk (Do not buffer).

-f <fps> <fps-base>

Specify FPS to start with.

-j <frame>

Set frame step to <frame>.

-s <frame>

Play from <frame>.

-e <frame>

Play until <frame>.

-c <cache_memory>

Amount of memory in megabytes to allow for caching images during playback.

Zero disables (clamping to a fixed number of frames instead).

Window Options:

- w or --window-border**
Force opening with borders.
- W or --window-fullscreen**
Force opening in fullscreen mode.
- p or --window-geometry <sx> <sy> <w> <h>**
Open with lower left corner at <sx>, <sy> and width and height as <w>, <h>.
- M or --window-maximized**
Force opening maximized.
- con or --start-console**
Start with the console window open (ignored if '-b' is set), (Windows only).
- no-native-pixels**
Do not use native pixel size, for high resolution displays (MacBook 'Retina').
- no-window-focus**
Open behind other windows and without taking focus.

Python Options:

- y or --enable-autoexec**
Enable automatic Python script execution.
- Y or --disable-autoexec**
Disable automatic Python script execution (pydrivers & startup scripts), (compiled as non-standard default).
- P or --python <filename>**
Run the given Python script file.
- python-text <name>**
Run the given Python script text block.
- python-expr <expression>**
Run the given expression as a Python script.
- python-console**
Run Blender with an interactive console.
- python-exit-code <code>**
Set the exit-code in [0..255] to exit if a Python exception is raised (only for scripts executed from the command line), zero disables.
- python-use-system-env**
Allow Python to use system environment variables such as 'PYTHONPATH' and the user site-packages directory.

--addons <addon(s)>

Comma separated list (no spaces) of add-ons to enable in addition to any default add-ons.

Logging Options:

--log <match>

Enable logging categories, taking a single comma separated argument.

Multiple categories can be matched using a `'.*'` suffix,

so `'--log "wm.*"'` logs every kind of window-manager message.

Sub-string can be matched using a `'*'` prefix and suffix,

so `'--log "*undo*"'` logs every kind of undo-related message.

Use `"^"` prefix to ignore, so `'--log "*,^wm.operator.*"'` logs all except for `'wm.operators.*'`

Use `"*"` to log everything.

--log-level <level>

Set the logging verbosity level (higher for more details) defaults to 1,

use `-1` to log all levels.

--log-show-basename

Only show file name in output (not the leading path).

--log-show-backtrace

Show a back trace for each log message (debug builds only).

--log-show-timestamp

Show a timestamp for each log message in seconds since start.

--log-file <filename>

Set a file to output the log to.

Debug Options:

-d or --debug

Turn debugging on.

* Enables memory error detection

* Disables mouse grab (to interact with a debugger in some cases)

* Keeps Python's `'sys.stdin'` rather than setting it to `None`

--debug-value <value>

Set debug value of `<value>` on startup.

--debug-events

Enable debug messages for the event system.

--debug-ffmpeg

Enable debug messages from FFmpeg library.

--debug-handlers

Enable debug messages for event handling.

- debug-libmv**
Enable debug messages from libmv library.
- debug-cycles**
Enable debug messages from Cycles.
- debug-memory**
Enable fully guarded memory allocation and debugging.
- debug-jobs**
Enable time profiling for background jobs.
- debug-python**
Enable debug messages for Python.
- debug-depsgraph**
Enable all debug messages from dependency graph.
- debug-depsgraph-eval**
Enable debug messages from dependency graph related on evaluation.
- debug-depsgraph-build**
Enable debug messages from dependency graph related on graph construction.
- debug-depsgraph-tag**
Enable debug messages from dependency graph related on tagging.
- debug-depsgraph-no-threads**
Switch dependency graph to a single threaded evaluation.
- debug-depsgraph-time**
Enable debug messages from dependency graph related on timing.
- debug-depsgraph-pretty**
Enable colors for dependency graph debug messages.
- debug-depsgraph-uuid**
Verify validness of session-wide identifiers assigned to ID datablocks.
- debug-ghost**
Enable debug messages for event handling.
- debug-gpu**
Enable GPU debug context and information for OpenGL 4.3+.
- debug-gpu-force-workarounds**
Enable workarounds for typical GPU issues and disable all GPU extensions.

- debug-wm**
Enable debug messages for the window manager, shows all operators in search, shows keymap errors.
- debug-all**
Enable all debug messages.
- debug-io**
Enable debug messages for I/O (Collada, ...).
- debug-fpe**
Enable floating-point exceptions.
- debug-exit-on-error**
Immediately exit when internal errors are detected.
- disable-crash-handler**
Disable the crash handler.
- disable-abort-handler**
Disable the abort handler.
- verbose <verbose>**
Set the logging verbosity level for debug messages that support it.

Misc Options:

- open-last**
Open the most recently opened blend file, instead of the default startup file.
- app-template <template>**
Set the application template (matching the directory name), use 'default' for none.
- factory-startup**
Skip reading the BLENDER_STARTUP_FILE in the users home directory.
- enable-event-simulate**
Enable event simulation testing feature 'bpy.types.Window.event_simulate'.
- env-system-datafiles**
Set the BLENDER_SYSTEM_DATAFILES environment variable.
- env-system-scripts**
Set the BLENDER_SYSTEM_SCRIPTS environment variable.
- env-system-python**
Set the BLENDER_SYSTEM_PYTHON environment variable.

-noaudio

Force sound system to None.

-setaudio

Force sound system to a specific device.

-h or --help

Print this help text and exit.

/?

Print this help text and exit (Windows only).

-R

Register blend-file extension, then exit (Windows only).

-r

Silently register blend-file extension, then exit (Windows only).

-v or --version

Print Blender version and exit.

--

End option processing, following arguments passed unchanged. Access via Python's 'sys.argv'.

Other Options:**--debug-freestyle**

Enable debug messages for Freestyle.

Argument Parsing:

Arguments must be separated by white space, eg:

```
# blender -ba test.blend
```

...will exit since '-ba' is an unknown argument.

Argument Order:

Arguments are executed in the order they are given. eg:

```
# blender --background test.blend --render-frame 1 --render-output '/tmp'
```

...will not render to '/tmp' because '--render-frame 1' renders before the output path is set.

```
# blender --background --render-output /tmp test.blend --render-frame 1
```

...will not render to '/tmp' because loading the blend-file overwrites the render output that was set.

```
# blender --background test.blend --render-output /tmp --render-frame 1
```

...works as expected.

ENVIRONMENT VARIABLES

\$BLENDER_USER_CONFIG Directory for user configuration files.

\$BLENDER_USER_SCRIPTS Directory for user scripts.

\$BLENDER_SYSTEM_SCRIPTS Directory for system wide scripts.

\$BLENDER_USER_DATAFILES Directory for user data files (icons, translations, ..).

\$BLENDER_SYSTEM_DATAFILES Directory for system wide data files.

\$BLENDER_SYSTEM_PYTHON Directory for system Python libraries.

\$TMP or \$TMPDIR Store temporary files here.

SEE ALSO

luxrender(1)

AUTHORS

This manpage was written for a Debian GNU/Linux system by Daniel Mester <mester@uni-bremen.de> and updated by Cyril Brulebois <cyril.brulebois@enst-bretagne.fr> and Dan Eicher <dan@trollwerks.org>.