Skip to content

Command Line Arguments

Blender 4.4

Usage: `blender [args ...] [file] [args ...]`

Render Options

`-b, --background`

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

The audio device is disabled in background-mode by default and can be re-enabled by passing in `-setaudio Default` afterwards.

`-a, --render-anim`

Render frames from start to end (inclusive).

`-S, --scene <name>`

Set the active scene `<name>` for rendering.

`-f, --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, --frame-start <frame>`

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

`-e, --frame-end <frame>`

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

`-j, --frame-jump <frames>`

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

`-o, --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
```

`//render_` becomes `//render_####`, writing frames as `//render_0001.png`

`-E, --engine <engine>`

Specify the render engine. Use `-E help` to list available engines.

`-t, --threads <threads>`

Use amount of `<threads>` for rendering and other operations [1-1024], 0 to use the systems processor count.

Cycles Render Options

Cycles add-on options must be specified following a double dash.

`--cycles-device <device>`

Set the device used for rendering. Valid options are: `CPU CUDA OPTIX HIP ONEAPI METAL`.

Append +CPU to a GPU device to render on both CPU and GPU.

Example:

```
blender -b file.blend -f 20 -- --cycles-device OPTIX
```

`--cycles-print-stats`

Log statistics about render memory and time usage.

Format Options

`-F, --render-format <format>`

Set the render format. Valid options are: `TGA RAWTGA JPEG IRIS AVIRAW AVIJPEG PNG BMP HDR TIFF`.

Formats that can be compiled into Blender, not available on all systems: `OPEN_EXR OPEN_EXR_MULTILAYER FFMPEG CINEON DPX JP2 WEBP`.

`-x, --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, --window-border`

Force opening with borders.

`-W, --window-fullscreen`

Force opening in full-screen mode.

`-p, --window-geometry <sx> <sy> <w> <h>`

Open with lower left corner at `<sx>`, `<sy>` and width and height as `<w>`, `<h>`.

`-M, --window-maximized`

Force opening maximized.

`-con, --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, --enable-autoexec`

Enable automatic Python script execution.

`-Y, --disable-autoexec`

Disable automatic Python script execution (Python-drivers & startup scripts), (default).

`-P, --python <filepath>`

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.

Network Options

`--online-mode`

Allow internet access, overriding the preference.

`--offline-mode`

Disallow internet access, overriding the preference.

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 <filepath>`

Set a file to output the log to.

Debug Options

`-d, --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-uid

Verify validness of session-wide identifiers assigned to ID data-blocks.

--debug-ghost

Enable debug messages for Ghost (Linux only).

--debug-wintab

Enable debug messages for Wintab.

--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-gpu-compile-shaders

Compile all statically defined shaders to test platform compatibility.

--debug-gpu-scope-capture

Capture the GPU commands issued inside the give scope name.

--debug-gpu-renderdoc

Enable RenderDoc integration for GPU frame grabbing and debugging.

--debug-wm

Enable debug messages for the window manager, shows all operators in search, shows keymap errors.

--debug-xr

Enable debug messages for virtual reality contexts. Enables the OpenXR API validation layer, (OpenXR) debug messages and general informatio prints.

--debug-xr-time

Enable debug messages for virtual reality frame rendering times.

--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.

--debug-freestyle

Enable debug messages for Freestyle.

--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.

-q, --quiet

Suppress status printing (warnings & errors are still printed).

GPU Options

`--gpu-backend`

Force to use a specific GPU backend. Valid options: `vulkan` (experimental), `metal`, `opengl`.

`--gpu-compilation-subprocesses`

Override the Max Compilation Subprocesses setting (OpenGL only).

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 `startup.blend` 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-extensions`

Set the `BLENDER_SYSTEM_EXTENSIONS` 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. `None Default SDL OpenAL CoreAudio JACK PulseAudio WASAPI`.

`-c`, `--command <command>`

Run a command which consumes all remaining arguments. Use `-c help` to list all other commands. Pass `--help` after the command to see its help text.

This implies `--background` mode.

`-h`, `--help`

Print this help text and exit.

`/?`

Print this help text and exit (Windows only).

`-r`, `--register`

Register blend-file extension for current user, then exit (Windows & Linux only).

`--register-allusers`

Register blend-file extension for all users, then exit (Windows & Linux only).

`--unregister`

Unregister blend-file extension for current user, then exit (Windows & Linux only).

`--unregister-allusers`

Unregister blend-file extension for all users, then exit (Windows & Linux only).

`-v, --version`

Print Blender version and exit.

`--`

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

Other Options

`--disable-depsgraph-on-file-load`

Backround mode: Do not systematically build and evaluate ViewLayers' dependency graphs when loading a blendfile in background mode (*-b* or options).

Scripts requiring evaluated data then need to explicitly ensure that an evaluated depsgraph is available (e.g. by calling *depsgraph = context.evaluated_depsgraph_get()*).

NOTE: this is a temporary option, in the future depsgraph will never be automatically generated on file load in background mode.

`--disable-liboverride-auto-resync`

Do not perform library override automatic resync when loading a new blendfile.

NOTE: this is an alternative way to get the same effect as when setting the *No Override Auto Resync* User Preferences Debug option.

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_RESOURCES:

Replace default directory of all user files. Other `BLENDER_USER_*` variables override when set.

BLENDER_USER_CONFIG:

Directory for user configuration files.

BLENDER_USER_SCRIPTS:

Directory for user scripts.

BLENDER_USER_EXTENSIONS:

BLENDER_USER_EXTENSIONS:

Directory for user extensions.

BLENDER_USER_DATAFILES:

Directory for user data files (icons, translations, ..).

BLENDER_SYSTEM_RESOURCES:

Replace default directory of all bundled resource files.

BLENDER_SYSTEM_SCRIPTS:

Directories to add extra scripts.

BLENDER_SYSTEM_EXTENSIONS:

Directory for system extensions repository.

BLENDER_SYSTEM_DATAFILES:

Directory to replace bundled datafiles.

BLENDER_SYSTEM_PYTHON:

Directory to replace bundled Python libraries.

BLENDER_CUSTOM_SPLASH:

Full path to an image that replaces the splash screen.

BLENDER_CUSTOM_SPLASH_BANNER:

Full path to an image to overlay on the splash screen.

OCIO:

Path to override the OpenColorIO configuration file.

TEMP:

Store temporary files here (MS-Windows).

TMPDIR:

Store temporary files here (UNIX Systems). The path must reference an existing directory or it will be ignored.

Copyright © : This page is licensed under a CC-BY-SA 4.0 Int. License
Made with Furo
Last updated on 2025-05-10