

The most common causes of Blender crashes:

- Running out of memory.
- Issues with graphics hardware or drivers.
- Bugs in Blender.

Firstly, you may be able to recover your work with File ▶ Recover ▶ Auto Save...

To prevent the problem from happening again, you can check that the graphics drivers are up to date ([Troubleshooting Graphics Hardware](#)), upgrade your machine's hardware (the RAM or graphics card), and disable some options that are more memory intensive:

- Reduce undo steps Preferences ▶ System ▶ Memory & Limits ▶ Undo Steps.
- Using multisample anti-aliasing also increases the memory usage and makes the display slower.
- On Linux, the Window Manager (KDE and Gnome for example) may be using hardware accelerated effects (e.g. window shadows and transparency) that are using up the memory that Blender needs. Try disabling the desktop effects or switch to a lightweight Window Manager.

To check memory usage by Blender:

- On Windows, use Task Manager and sort by Memory.
- On macOS, use Activity Monitor.app and open Memory tab. Alternatively, run `top -o MEM`.
- On Linux, run `top -o %MEM`.

In more extreme cases, you may want to consider reinstalling your operating system. Windows especially tends to build up latent issues due to successive updates and program leftovers.

Crash Log

When Blender crashes, it writes out a text file which contains information that may help identify the cause of the crash. Usually, this file is written in the [Temporary Directory](#) directory.

This file contains a log of tools used up until the crash as well as some other debug information. When reporting bugs about crashes it can be helpful to attach this file to your reports, especially when others are unable to reproduce the crash.

Windows

On a crash, a file is written based on the name of the currently loaded blend-file, so `test.blend` will create a file called `test.crash.txt`.

Batch scripts are provided in Blender installation directory which may be run to obtain the Blender debug log and system info text files:

- `blender_debug_log.cmd` is used in most cases.
- `blender_debug_gpu.cmd` and `blender_debug_gpu_workaround.cmd` log GPU-related errors.
- `blender_factory_startup.cmd` starts Blender with default settings which is recommended for debugging.

If the crash happens in Blender module, stack trace is also written to a file named `blender.crash.txt`. The path to that file can be found at the end of `blender_debug_log.txt` file.

macOS

After a crash, the macOS Crash Reporter shows a window with backtrace after some time, or when Blender is opened again. Copy the text in the crash report and save it in a text file. That file should be attached to the bug report while following other bug reporting guidelines.

Some `.crash` files can also be found in `~/Library/Logs/DiagnosticReports/` with the name of format: `Blender_YYYY-MM-DD-HHMMSS_MACNAME.crash`. If a report is present corresponding to the time of crash, that file can also provide hints about cause of the crash. Alternatively, Console.app can be used to navigate all "User Reports" (see sidebar in the app).

Linux

After a crash, a file named `blender.crash.txt` is written to the `/tmp` directory.

Note

More logs can be obtained by running Blender from Command Line and using `--factory-startup --debug-all` flags. See [Launching from the Command Line](#) and [Command Line Arguments](#).

[Previous](#)
[macOS – Other GPU](#)

Copyright © : This page is licensed under a CC-BY-SA 4.0 Int. License

[No](#)
[Python Err](#)

Made with [Furo](#)

Last updated on 2025-05-10

[View Source](#)
[View Translation](#)
[Report issue on this page](#)