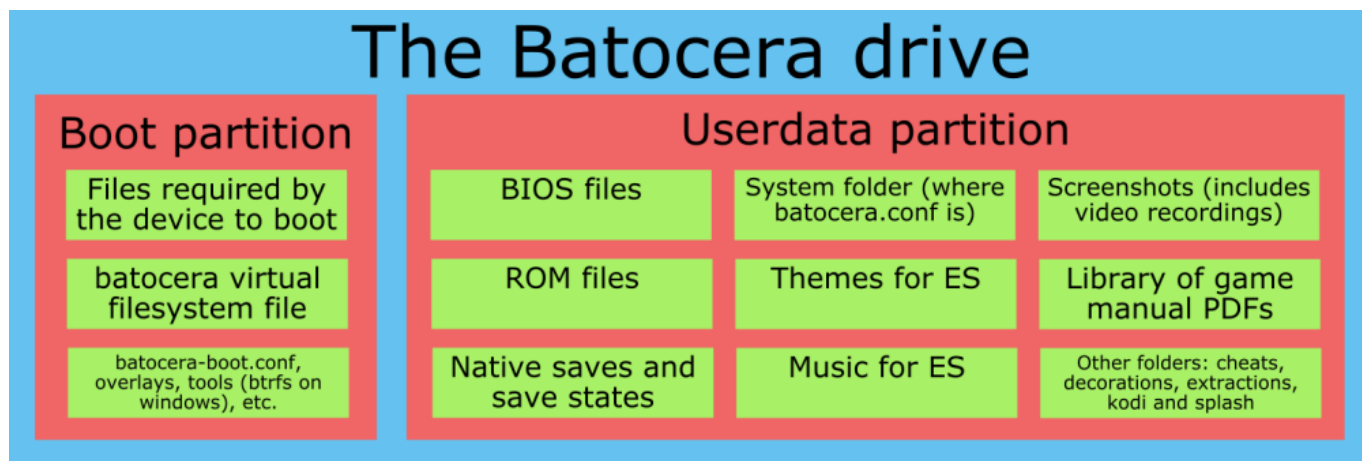


# Overlays (modify the live system)


Batocera is like a firmware. It's a huge filesystem image at /boot/boot/batocera (it has no extension in its filename) which is virtually mounted as the root of the drive in RAM at boot. You cannot modify the batocera file directly, and for most use cases, you shouldn't *need* to modify it. All folders except for userdata and boot in the / directory stem from this image.




Under normal circumstances, this file cannot be written to, and any changes made to the filesystem mounted in RAM will be lost upon shutdown. That said, you can utilize overlays to “patch” certain files over the batocera filesystem image at boot.

## Creating a Batocera filesystem overlay

For modifications to the regular Batocera filesystem, you can directly edit the files in their directory using [the built-in file manager](#) while booted (for example if you want to play with the configgen scripts in /usr/lib/pythonX.X/site-packages/configgen/generators/ or change the default platform parameters in /usr/share/batocera/configgen/configgen-defaults-arch.yml) or sign in [via SSH](#) or [SCP/FTP](#) to make those alterations. Once the modifications are done, you can save them with [the command](#) batocera-save-overlay. This command script creates a patch file at /boot/boot/overlay which stores the differences between the current mounted filesystem and the one stored in /boot/batocera.

 Refer to [the notable files/folders page](#) for some more interesting files!

By default, Batocera overlay files are limited to 100 MB. If your modifications are over 100 MB, you can specify the size of the overlay with the command. Keep always a bit of margin, the overlay will embed a few more files than those you were expecting (some system files created on-the-fly at boot time). For example, if you need to save 120 MB of data on the system, you can type batocera-save-overlay 125.

 This overlay file is automatically removed at the next upgrade.



To view the contents of the overlay independently from the underlying filesystem, it can be mounted in a separate location:

```
mount /boot/boot/overlay /mnt
find /mnt
umount /mnt
```

## Removing the overlay

Even if you don't use the `batocera-save-overlay` script you can still edit the files in any of the root folders, but any modification made outside of `/userdata` or `/boot` will be lost on reboot. If you want to remove all your overlay modifications made after running `batocera-save-overlay` previously, simply [remove](#) the `/boot/boot/overlay` file.

## Cases when you don't need to modify the system

Some best practices here:

- You don't need to edit `/etc/fstab` to mount external drives, use [configure your external storage](#) or [configure your NAS](#) instead.
- You should not modify `/usr/share/emulationstation/es_systems.cfg` for altering the systems listed in EmulationStation. This is the base file updated with every release of Batocera. Instead, follow the instructions at [the customize systems page](#) to create a “patch” that's loaded on top of the base file, allowing for updates to the base file.

## I want to modify something in the boot folder though

The boot partition is mounted with read-only permissions at `/boot/`, but can be [remounted with write permissions post-boot](#). You do not need to save an overlay for file edits made in the boot partition, as this is a physical partition where your edits are permanent (and because of that, any edits you make here are more dangerous and can result in a non-booting machine, always make backups before attempting this). All the user data is stored in a separate `userdata` partition, which is mounted with write permissions at the path `/userdata/` (like the boot partition, this is just your regular `userdata` storage partition, no need to save overlays for this either).

From:  
<https://wiki.batocera.org/> - **Batocera.linux** - Wiki

Permanent link:  
[https://wiki.batocera.org/modify\\_the\\_system\\_while\\_it\\_s\\_running](https://wiki.batocera.org/modify_the_system_while_it_s_running)

Last update: **2023/04/09 14:10**



