

# Customize ES Systems

EmulationStation displays systems based on a file called `es_systems.cfg`, located at `/usr/share/emulationstation`. In the past, you had to edit this file and use the command `batocera-save-overlay`. This had issues when updating as all overlays have to be removed, meaning you'd have to redo these steps every update.



Batocera introduced the ability to copy the file to `/userdata/system/configs/emulationstation/es_system.cfg` and edit it there, however this meant you would have to compare and copy the file every update, as well as appending your changes and working out if something had broken (as Batocera adds new systems nearly every major release and changes the launch methods for other systems too).



Batocera now treats extra `/userdata/system/configs/emulationstation/es_systems_<custom_name>.cfg` as overlays to the original `/usr/share/emulationstation/es_systems.cfg` file. This has the benefit of only requiring you to maintain the list of *your* personally added system entries, as opposed to needing to scour through the entire document every update.



If you upgrade from a previous Batocera version to **v31** or higher, you **need to update** your `es_systems*.cfg` files to reflect the change of Python version. As you can see below, starting with Batocera **v31**, the location for the emulator launcher is `/usr/lib/python3.9/site-packages/configgen/emulatorlauncher.py` and not `/usr/lib/python2.7/site-packages/configgen/emulatorlauncher.py` as it used to be in previous versions. There is no Python 2.7 on Batocera **v31** or higher.

## Create a new system-specific ES systems CFG file

You can create a file named `es_systems_<custom_name>.cfg` where `<custom_name>` is a name you wish to use for it. This file needs to follow the same conventions as the original `es_systems.cfg`, but does not need to include the entirety of the file. For example:

[es\\_systems\\_3do.cfg](#)

```
<?xml version="1.0"?>
<systemList>
  <system>
    <fullname>3D0 Interactive Multiplayer</fullname>
    <name>3do</name>
    <manufacturer>Panasonic - Sanyo - Goldstar</manufacturer>
    <release>1993</release>
    <hardware>console</hardware>
```

```
<path>/userdata/roms/3do</path>
<extension>.iso .chd .cue</extension>
<command>python /usr/lib/python3.9/site-
packages/configgen/emulatorlauncher.py %CONTROLLERCONFIG% -system
%SYSTEM% -rom %ROM%</command>
<platform>3do</platform>
<theme>3do</theme>
<emulators>
  <emulator name="libretro">
    <cores>
      <core default="true">opera</core>
    </cores>
  </emulator>
</emulators>
</system>
</systemList>
```

would only affect the 3DO system on your Batocera, with the rest of the systems still referring to `/usr/share/emulationstation/es_systems.cfg` for their system CFG. When both files contain the same systems, `es_systems_<custom_name>.cfg` will take priority.

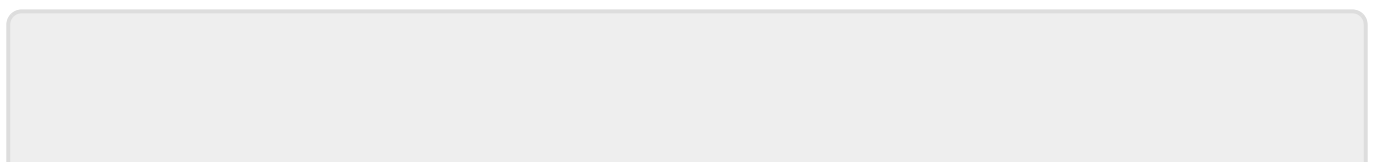
In case you'd like to have complete control over ES systems CFG, you can still use the two previous methods to completely override it (ie. using overlays or copying the entirety of `es_systems.cfg` to `/userdata/system/configs/emulationstation/es_systems.cfg` without appending a custom system name to its filename).

Batocera is smart enough to *only* apply the changes you make. Such as, if the only thing you need to change for a system is its ROM path, you can use the following in your `/userdata/system/configs/emulationstation/es_systems_pico8.cfg`:

#### [es\\_systems\\_pico8.cfg](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<systemList>
  <system>
    <name>pico8</name>
    <path>/userdata/roms/pico8real</path>
  </system>
</systemList>
```

Here, it'll use the folder `/userdata/roms/pico8real/` to look for Pico-8 games instead of the regular one.



From:

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

Permanent link:

[https://wiki.batocera.org/emulationstation:customize\\_systems?rev=1632960444](https://wiki.batocera.org/emulationstation:customize_systems?rev=1632960444)

Last update: **2021/09/30 00:07**

