Ports (native to Linux)

Ports are great. You can just straight up run Linux games on Batocera, if the game itself carries all of its dependencies. Fun fact: most don't. You typically need to refer to the game's documentation or scour around the internet (the Arch Wiki and the PCGamingWiki are pretty good resources for this) to find out how to get it working in Batocera. Chuck the data in the roms/ports/.data/ folder and chuck the respective .sh file required to launch the game in the roms/ports/ folder. Don't forget to mark them as executable!

If the game is available as an AppImage, try that as it has the highest chances of working out of the box.

In Batocera v32 and lower, you would have to create the roms/ports/ subfolder yourself first in order to start using it.

This system scrapes metadata for the “ports” group(s) and loads the ports set from the currently selected theme, if available.

Grouped with the “ports” group of systems.

Quick reference

- **Accepted ROM formats:** .sh
- **Folder:** /userdata/roms/ports

Program files

Note that in many cases you may need a physical (USB) keyboard attached to your Batocera system to configure your native Linux game for the first time, most times to set up your physical controllers for the first time.

If you'd like to see some examples, you can install some native Linux ports in the content downloader. For testing purposes, you can also SSH in and try running the commands to launch the game there, this might help you see if there are any error messages being printed. Don't forget to run /etc/init.d/S31emulationstation stop to kill EmulationStation and batocera-es-swissknife --restart to bring it back to life! You may also need to pre-emptively run export
DISPLAY=:0.0 to get the game to be able to “see” your screen.

**Example**

As already mentioned above, the /userdata/roms/ports folder is where you want to place your game's main folder which contains the according native Linux game. The following example will show how you can make the popular game Celeste running on Batocera.

First of all, you will have to buy/download the native Linux version of Celeste, which will give you a single package file, e.g. celeste-linux.zip. After successfully downloading, copy the file over from your local computer to your Batocera system (for example via WinSCP) to the following path: /tmp/celeste-linux.zip. Now SSH into your Batocera system and execute the following command to extract the package file's content to the correct path:

```
unzip -d /userdata/roms/ports/Celeste /tmp/celeste-linux.zip
```

Sidenote, for better understanding of the following steps: In most cases, depending on where you've downloaded your Celeste game sources, Celeste is kind of well “prepared” for running it as a native Linux game. In other words: The downloaded game package should contain all files/dependencies without the hassle of having to search/add manually any external files/dependencies. Unfortunately, as already mentioned above, this definitely is not the case for all native Linux games!

Now check the content of the game's directory...

```
ls -l /userdata/roms/ports/Celeste
```

...which should give you the following similar output:
For Celeste, in his case, the correct executable file to start the game is
/userdata/roms/ports/Celeste/Celeste as marked in the screenshot above. Now to make the
game show up as Celeste in your Batocera GUI's Ports section, you would have to have a file called
/userdata/roms/ports/Celeste/Celeste.sh. You could indeed rename the game's main executable file from /userdata/roms/ports/Celeste/Celeste to
/userdata/roms/ports/Celeste/Celeste.sh and adjust it afterwards, but a better, more flexible and more reliable way would be to leave the source files untouched and create a separate executable file which will call the main executable file. To do so, just run the following command:

```
nano /userdata/roms/ports/Celeste/Celeste.sh
```

Now paste the following content:

```
#!/bin/bash
cd /userdata/roms/ports/Celeste && export DISPLAY=:0.0; ./Celeste
```

Save the file and quit the editor.

Note: You can use the script content above for most any native Linux game to create an according executable start file. Just use:

```
#!/bin/bash
cd /userdata/roms/ports/<yourNativeLinuxGame> && export DISPLAY=:0.0; ./<yourNativeLinuxGameExecutable>.sh
```
As you may have mentioned on the screenshot above, none of the executable files are marked as executable yet, which will be required to run Celeste successfully. So let's do this by executing:

```
chmod +x /userdata/roms/ports/Celeste/Celeste.sh
/userdata/roms/ports/Celeste/Celeste
/userdata/roms/ports/Celeste/Celeste.bin*
```

Beware: The required executables are different for every native Linux game. There's no general rule you can follow, you have to find out by yourself by searching through the internet or by just trying to execute the executables via command line and analyze the errors that may occur (see “Troubleshooting” section below).

Now update the gamelist via the Batocera GUI and the game should show up in the Ports section. That's it, start the game from there and have fun!

**Troubleshooting**

Native Linux games not running on Batocera can have many different reasons. If you are sure your game can run on Batocera but it just won't start (for example, if it puts you back to the main Ports section screen after starting the game via the Batocera GUI), you can check some potential error messages by running the game via command line, for example:

```
/userdata/roms/ports/Celeste/Celeste.sh
```

Sometimes there are multiple executable files involved in the process of running a game, so in many cases there may be only permission issues causing the game not to start, which would give you similar output(s) to this:

```
[root@htpc01 /userdata/roms/ports/Celeste]# ./Celeste.sh
./Celeste: line 31: ./Celeste.bin.x86_64: Permission denied
```

In such a case, just set the required permission(s) (chmod +x <YourExecutableFile(s)>) to the according executable file(s) and you should be ready to go.

**A note about integrated ports**

Batocera-integrated ports (recognized as their own unique system, but are grouped under the “Ports” system along with native Linux Ports by default) usually contain an _info.txt file in their folder explaining how to install them and sometimes have their own page in system overview page under the “Port” header.
Troubleshooting

Further troubleshooting

For further troubleshooting, refer to the **generic support pages**.

If you need further help, feel free to ask on our **Discord server** or **forum**.

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

Permanent link: https://wiki.batocera.org/systems:ports

Last update: **2022/03/13 17:08**