

# Usage of batocera-settings

## Introduction



This command is a bit out of date to use now, most if not all things can be adjusted within Batocera EmulationStation itself or through manually editing the text file. Certain commands may no longer be functional. The following is still useful for scripting however.

**batocera-settings** is a command-line tool that can work with regular config files to read/write its content, useful for scripting (like say you need to change the config based on what machine you have the drive plugged into). The `/userdata/system/batocera.conf` contains most of the machine's settings and follows `batocera.conf`'s [regular syntax](#).

When manually editing the `batocera.conf` file:

- Use **## This is a text comment** for text comments.
- Use **#enable.godmode=hallelujah** for commenting values.
- Describe content and functions in comments when adding them for future reference.
- It's recommended to add content in their appropriate sections, but not strictly required.

Down here is a small excerpt of an example config file

```
# ----- B - Network ----- #
## Set system hostname
system.hostname=BATOCERA
## Activate wifi (0,1)
wifi.enabled=0
## Wifi SSID (string)
#wifi.ssid=new ssid
## Wifi KEY (string)
## after rebooting the batocera.linux, the "new key" is replace by a hidden
value "enc:xxxxx"
## you can edit the "enc:xxxxx" value to replace by a clear value, it will
be updated again at the following reboot
## Escape your special chars (# ; $) with a backslash : $ => \$
#wifi.key=new key
```

## Recommended commands and expressions

`batocera-setting-get` and `batocera-setting-set` is utilized by parameters parsed. The old `batocera-settings` itself was a relic of RecalBox times. FunFact: Because `batocera-settings` is based on C-code, it's much faster then the old `python-script`.



As `batocera-settings` is more modern and supports reading/writing values. For these examples,

replace the contents in the square brackets with your intended setting (and don't include the square brackets themselves).

- Synthax:
  - **batocera-settings-get -f [CONFIG\_FILE] [KEY]**
  - **batocera-settings-set -f [CONFIG\_FILE] [KEY] [VALUE]**
- BATOCERA usage
  - **batocera-settings-get [key]** read key from batocera.conf
  - **batocera-settings-set [key] [value]** write value key=value to batocera.conf
  - **batocera-settings-get -f [file] [key]** read value key from file
- BATOCERA board-specific usage (added in Batocera v36)
  - **batocera-settings-get-master [key]** read key from /usr/share/batocera/sysconfigs/batocera.conf.\${BOARD\_MODEL}

## Error code handling

Whenever batocera-settings-set or batocera-settings-get is called from a script you will receive an exit code number. This will help to identify errors (for debugging you can use the **status** command for more useful output).

File and Key/Values error		
Error Code	Error code explanation	Troubleshooting
EC 0	No Error, value found	 You made it!
EC 1	General error, e.g. command line error	 Check your command line for correct parameters

## Handling in scripts

I present here some short scripts, to show you how to make batocera-settings work in your script. As I'm more confident in shell scripting I give you just some small examples in shell script.

1. bash: Obtain value
2. bash: Activate UART in /boot/config.txt
3. bash: Set a new key
4. python: Obtain a value

**USE AT YOUR OWN RISK** 

### 1. bash: Obtain a value

[obtain\\_value.sh](#)

```
#!/bin/bash
#This is an example file how batocera-settings-get can be utilized
#to read a value out from /userdata/system/batocera.conf

value="$(batocera-settings-get power.switch.device)"
ret=$?
if [[ $ret -eq 0 ]]; then
    echo "Power Switch detected: '$value'"
else
    echo "No Power Switch detected!"
fi
```

## 2. bash: Activate UART in "/boot/config.txt"

[activate\\_uart.sh](#)

```
#!/bin/bash
#This is an example file how batocera-settings-set can be utilized
#to activate UART in /boot/config.txt

# Check status of file and make it writeable
[[ -w /boot/config.txt ]] && batocera-es-swissknife --remount

batocera-settings-set -f /boot/config.txt enable_uart 1
ret=$?
if [[ $ret -eq 0 ]]; then
    echo "UART activated, uncommented enable_uart"
    batocera-settings-set -f /boot/config.txt enable_uart 1
else
    echo "Key: enable_uart not found"
    echo "Not a Raspberry System?"
fi
```

## 3. bash: Set a new key

[set\\_new\\_key.sh](#)

```
#!/bin/bash
#This is an example file how batocera-settings-set can be utilized
#to set a new key in /userdata/system/batocera.conf

value=$(batocera-settings-set core.PS5.emulator SONY5EVER)
ret=$?
if [[ $ret -eq 0 ]]; then
    echo "PS5 core enabled!"
```

```
else
    echo "Another error occurred!"
fi
```

#### 4. python: Obtain a key

[obtain\\_value.py](#)

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#This is an example file how batocera-settings-get can be utilized
#to read a value out from /userdata/system/batocera.conf with python

import subprocess

value = (subprocess.check_output(['batocera-settings-get',
'power.switch.device']))
if value:
    print "Power Switch Detected: ", value
else:
    print "No power switch detected!"
```

## Older versions (Batocera 30, 5.27.2 and earlier)

Till B30 batocera-settings was used synthax was in general `batocera-settings -command <load> -key <value>` the command was under active development and more featres were added. To get the full commandset of your version just run batocera - settings from terminal or SSH. If you use BATOCERA versions 5.22 and earlier you will see the forked reminiscene - here the settings command was a python script and started with python /usr/lib/python2.7/site-packages/configgen/settings/batoceraSettings.py

From:

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

Permanent link:

[https://wiki.batocera.org/usage\\_of\\_batocera-settings?rev=1696593498](https://wiki.batocera.org/usage_of_batocera-settings?rev=1696593498)

Last update: **2023/10/06 11:58**

