# 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This command is a bit out of date to use now, most if not all things can be adjusted within Batocera EmulationStation itself or through manually editing the text file.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. On older versions of Batocera, the list of available functions can be found by entering in `batocera-settings` and pressing Enter.

**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

The commands are called with `batocera-setting-get` and `batocera-setting-set`. For the syntax below, replace the contents in the square brackets with your intended setting (and don't include the square brackets themselves).

Syntax for reading setting values:

```
batocera-settings-get -f [CONFIG_FILE] [KEY]
batocera-settings-get [KEY]
```

This will read the KEY setting from the specified config file and output its current file. If no `-f` flag specified, it will search through `userdata/system/batocera.conf` by default.

Syntax for setting those values:

```
batocera-settings-set -f [CONFIG_FILE] [KEY] [VALUE]
batocera-settings-set [KEY] [VALUE]
```

This will search for the key in the config file and then replace its value with the new specified one. If the key doesn't already exist in the config file, a new line with that key will be created.

In Batocera **v36** and above, syntax for reading the current board's default setting:

```
batocera-settings-get-master [key]
```

This will read the 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 explaination** | **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
#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
command=(["batocera-settings-get", "system.power.switch"])
rc = subprocess.call(command, stdout=subprocess.DEVNULL)
if rc == 0:
    value = subprocess.check_output(command).decode("utf-8")
    print ("Power Switch Detected: ", value)
else:
    print ("No power switch detected!")
```