

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This command is a bit out of date to use now, most if not all things can be adjusted within Batocera EmulationStation itself or through manually editing the text file.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 uses the **key[game-if-applicable].setting=value** 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 is utilized by parameters parsed. These parameters can be used in the long and in the short format. It's a relic of RecalBox times thus its syntax should seem familiar.

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).

- BATOCERA Basic usage
 - **batocera-settings-get** [key] read key from batocera.conf
 - **batocera-settings-set** [key] -v [value] write value key=value to batocera.conf
 - **batocera-settings-get -f** [file] [key] read value key from file
- BATOCERA Extended usage  **Fix Me!** This flag no longer exists, was it moved or outright deleted?
 - **batocera-settings -e -r** [key] -s [system] iterate keys system.key and if not available use global.key
 - **batocera-settings -e -r** [key] -s [system] -g [game] iterate keys like above, but start with system["game"].key

Basic usage: `batocera-settings-set -f [CONFIG_FILE] [KEY] [VALUE]`

Extended usage: `batocera-settings-set -e -g [game] -s [system] -r [key]`



This flag no longer exists, was it moved or outright deleted?

-f - Specifies the config file to read from. Defaults to /userdata/system/batocera.conf.


-v - Set value to selected 'key', any alphanumeric value


-e - Activate extended mode, needed for parsing game/system specific keys



This flag no longer exists, was it moved or outright deleted?

-g - Filename of the specified game, enclose in double-quotes ("") to avoid

globbing  **Fix Me!** This flag no longer exists, was it moved or outright deleted?

-s - Shortname of specified system  **Fix Me!** This flag no longer exists, was it moved or outright deleted?



This flag no longer exists, was it moved or outright deleted?







If -e is not set the parameters -g and -s are ignored!

Use the errorlevels for scripts!

Always use quotes if you use arguments containing blanks!

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
EC 2	File error, e.g. the config file is not available	 Check file path and r/w access to it
EC 10	Value error, key found but value is empty	 Unusual setup but no error at all
EC 11	Value error, key found but it is commented out	 Activate the key entry by uncomment command
EC 12	Key not found	 Add the key by manual or check your command line for typos

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

batocera-settings-set -f /boot/config.txt enable_uart 1
ret=$?
if [[ $ret -eq 0 ]]; then
    echo "UART activated, uncommented enable_uart"
elif [[ $ret -eq 2 ]]; then
    echo "File is write protected!"
    echo "Remounting boot-partition with write permissions..."
    mount -o remount, rw /boot
    batocera-settings-set -f /boot/config.txt enable_uart 1
    if [[ $ret -eq 0 ]]; then
        echo "UART activated, uncommented enable_uart"
    fi
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=$(batocera-settings-set core.PS4.emulator SONY4EVER)
ret=$?
if [[ $ret -eq 0 ]]; then
    echo "PS4 core enabled!"
fi
```

```
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!"
```

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

Permanent link:
https://www.wiki.batocera.org/usage_of_batocera-settings?rev=1629979676

Last update: **2021/08/26 12:07**

