Legacy Batocera scripts

This is a historical page for how scripts in Batocera behaved **v37** and lower. For the current page, click here.

You can opt to have scripts launched at various points in Batocera. This will change the location you save the script to and how it opens.

Once during boot and once during shutdown

Sometimes you just want to fire up one script after successfully booting, for example when you want to start up a VPN or other after-boot tasks. In order to do this, create a script text file at `/userdata/system/custom.sh`. Be aware that the script will be executed independently of the executable (x) attribute being set to the script file or not!

This script is the very last one that will be invoked by Batocera at boot time after EmulationStation has launched (check `/etc/init.d/` to see all the modules that are loaded before then, `S99custom` is when the user script is launched). The filename must be `custom.sh` otherwise it will not be checked.

Make sure your script ends with Unix line terminators (LF), not Windows-style line terminators (CR/LF) otherwise the script will not launch. Use a real text editor to edit your scripts, especially if you edit them under Windows.

Events

To distinguish between different events, the first argument parsed in the script will change:

- `start` Only when Batocera is booting.
- `stop` Only when Batocera is shutting down.
- `restart` 🔧Fix Me!
- `reload` 🔧Fix Me!

Putting code outside of cases which check these arguments will always be executed on both boot and shutdown.

A simple custom script as an example

custom.sh

```
#!/bin/bash
# Code here will be executed on every boot and shutdown.
```

```
# Check if security is enabled and store that setting to a variable.
securityenabled="$(/usr/bin/batocera-settings-get
system.security.enabled)"

case "$1" in
    start)
        # Code in here will only be executed on boot.
        enabled="$(/usr/bin/batocera-settings-get
system.samba.enabled)"
        if [ "$enabled" = "0" ]; then
            echo "SMB services: disabled"
            exit 0
        fi
        ;;
    stop)
        # Code in here will only be executed on shutdown.
        echo -n "Shutting down SMB services: "
        kill -9 `pidof smbd`
        RETVAL=$?
        rm -f /var/run/samba/smbd.pid
        echo "done"
        ;;
    restart|reload)
        # Code in here will executed (when?).
        echo "SMB services: reloaded"
        ;;
    *)
        # Code in here will be executed in all other conditions.
        echo "Usage: $0 {start|stop|restart}"
        ;;
esac

exit $?
```

Once early in the boot process and once late in the shutdown process

In Batocera **v32** and higher, scripts can also be launched early in the boot process, before much of Batocera has even begun loading. This is done by storing the script in the boot partition at `/boot/boot-custom.sh`. Keep in mind that this script will be limited to more basic commands/modules, as most things have not loaded yet. The filename must be `boot-custom.sh` otherwise it will not be checked.

Precisely, this is the first thing executed once *init.d* has begun. This is before the *network/share* population/IR remote daemon have even loaded so capabilities are limited. Check `/etc/init.d` to see the exact order of all the modules have been loaded in (`S00bootcustom` is when the `/boot/boot-custom.sh` is executed).

This can be used to effectively patch and/or override the modules loaded by Batocera. Here are some examples that are only possible with this method:

- Customize or disable `S33disablealtfn` to allow a certain text mode console on a particular TTY session.
- Tinker with hwclock to save local time instead of universal time to RTC.
- Run `fsck -a` or `e2fsck -p` to harden the file system against data corruption in the case of a power cut/disconnection.

This script is triggered by the same events and follows the same rules as it.

Watch for a game start/stop event

This still functions the same in current version of Batocera. Refer to the current page for its info.

EmulationStation scripting

EmulationStation scripting has not changed yet, so it is still on the current page.

Real use cases

Only the older script examples are here, for more examples check the current page.

Batocera event watcher scripts

- Method to not lose SRM and other meta-data

custom.sh

```
#!/bin/bash
# custom.sh - place to /userdata/system
# by cyperghost 23/11/19
#

if [[ $1 == stop ]]; then
    batocera-es-swissknife --emukill
fi
```

- Splash video during game load (x86/x86_64) (credit to @algernon)

videolaunchx86.sh

```
#!/bin/bash
```

```
# Adds system specific loading screen videos to x86 builds of batocera
via VLC
# Add your .mp4 videos to /userdata/loadingscreens naming them each
with the same as in your roms folder i.e named snes.mp4 to play a video
before for each snes game.
# Add this script in /userdata/system/scripts then chmod+x
/userdata/system/scripts/videolaunchx86.sh in terminal to give this
script permissions to run

do_videostart ()
{
    video="$1"
    vlc play --fullscreen --no-video-title-show --play-and-exit $video
}

# Comment out the above phrase and uncomment this one to enable playing
loading videos concurrently with emulator launch instead of before
(experimental)
#do_videostart ()
#{
# video="$1"
# vlc_opt="play --fullscreen --no-video-title-show --play-and-exit"
# vlc $vlc_opt $video &
# PID=$!
#}

videopath="/userdata/loadingscreens"

if [[ "$1" == "gameStart" && -n "$2" ]]; then
    video="${videopath}/$2.mp4"
    [[ -f "$video" ]] || exit 1
else
    exit 1
fi

do_videostart "$video"
#wait $PID
exit 0


# Replace or comment out all the other code above except for the first
bash line then uncomment the following phrase to enable a single
loading splash for all systems. It should be named default.mp4 in the
same folder as above.
#default="/userdata/loadingscreens/default.mp4"
#case $1 in
# gameStart)
#     vlc play --fullscreen --no-video-title-show --play-and-exit
$default
# ;;
```

```
#
#esac
#
#exit 0
```

- Splash video during game load (Raspberry Pi)

videolaunchpi.sh

```
#!/bin/bash
do_videostart ()
{
    video="$1"
    # Launch the video
    omx_fnt="--font=/usr/share/fonts/dejavu/DejaVuSans-BoldOblique.ttf"
    omx_opt="--no-keys --layer=10000 --aspect-mode=fill"
    omx_srt="--no-ghost-box --lines=1 --align=left $omx_fnt --font-
size=20 --subtitles=/usr/share/batocera/splash/splash.srt"

    # Disable sound
    omx_nosound="-n -1"
    # -911 = Volume set to 35% (On omxplayer)
    # 1/(10^(911/2000)) = 0.35034828830157

    /usr/bin/omxplayer -o both --vol -2500 $omx_opt $omx_srt
$omx_nosound $video &
    PID=$!
}


videopath="/userdata/videoloadingscreens"

if [[ "$1" == "gameStart" && -n "$2" ]]; then
    video="${videopath}/$2.mp4"
    # Filecheck
    [[ -f "$video" ]] || exit 1
else
    exit 1
fi

do_videostart "$video"
#wait $PID
exit 0
```

- OGST Screen Support (original post)

OGST_screen.sh

```
#!/bin/bash
```

```
# Install videos of ZIP file to /userdata/OGST/
# Install scripts to /userdata/system/custom.sh and
/userdata/system/scripts/OGST_screen.sh to use this script.

logo_folder='/userdata/OGST'
default_logo='start.mp4'
logo_file=""
loop=false
default_logo_loop=false
cache_file='/userdata/system/.cache/ffmpeg_PID'

clean_cache () {
    if [ -f $cache_file ]; then
        rm -f $cache_file
    fi
}

case "$1" in

    start)
        # init TFT screen
        [ `/sbin/lsmod | grep -c spi_s3c64xx` -ge 1 ] && rmmod
spi_s3c64xx
        modprobe spi_s3c64xx force32b=1
        modprobe fbtft_device name=hktft9340 busnum=1 rotate=270
force32b=1

        # wait fb1 loaded
        N=0
        while ! test -e /dev/fb1 -o $N -gt 15; do
                sleep 1
                let N++
        done

        # rename fb1 to fb_ (trick to ovoid conflics with some
emulators like reicast...)
        mv /dev/fb1 /dev/fb_

        clean_cache
        logo_file=$default_logo && loop=default_logo_loop
    ;;

    stop)
        clean_cache
        logo_file='stop.mp4' && loop=false
    ;;

    gameStop)
        logo_file=$default_logo && loop=default_logo_loop
    ;;
```

```
    gameStart)
        logo_file=$default_logo && loop=default_logo_loop
        case "$2" in
            3do)                logo_file='3do/video.mp4' && loop=true
;;
            amiga)                logo_file='amiga/video.mp4' &&
loop=true ;;
            amigacd32)            logo_file='amigacd32/video.mp4' &&
loop=true ;;
            amstradcpc)            logo_file='amstradcpc/video.mp4' &&
loop=true ;;
            apple2)                logo_file='apple2/video.mp4' &&
loop=true ;;
            atari800)            logo_file='atari800/video.mp4' &&
loop=true ;;
            atari2600)            logo_file='atari2600/video.mp4' &&
loop=true ;;
            atari5200)            logo_file='atari5200/video.mp4' &&
loop=true ;;
            atari7800)            logo_file='atari7800/video.mp4' &&
loop=true ;;
            atarist)            logo_file='atarist/video.mp4' &&
loop=true ;;
            atomiswave)            logo_file='atomiswave/video.mp4' &&
loop=true ;;
            c64)                logo_file='c64/video.mp4' && loop=true
;;
            cavestory)            logo_file='cavestory/video.mp4' &&
loop=true ;;
            colecovision)        logo_file='colecovision/video.mp4' &&
loop=true ;;
            dreamcast)            logo_file='dreamcast/video.mp4' &&
loop=true ;;
            fba)                logo_file='fba/video.mp4' && loop=true
;;
            fds)                logo_file='fds/video.mp4' && loop=true
;;
            gameandwatch)        logo_file='gameandwatch/video.mp4' &&
loop=true ;;
            gamegear)            logo_file='gamegear/video.mp4' &&
loop=true ;;
            gb)                    logo_file='gb/video.mp4' &&
loop=true ;;
            gba)                logo_file='gba/video.mp4' && loop=true
;;
            gbc)                logo_file='gbc/video.mp4' && loop=true
;;
            gx4000)                logo_file='gx4000/video.mp4' &&
loop=true ;;
            intellivision)        logo_file='intellivision/video.mp4'
&& loop=true ;;
```

```
            jaguar)                 logo_file='jaguar/video.mp4' &&
loop=true ;;
            lynx)                  logo_file='lynx/video.mp4' &&
loop=true ;;
            mame)                  logo_file='mame/video.mp4' &&
loop=true ;;
            mastersystem)          logo_file='mastersystem/video.mp4' &&
loop=true ;;
            megadrive)              logo_file='megadrive/video.mp4' &&
loop=true ;;
            msx)                  logo_file='msx/video.mp4' && loop=true
;;
            n64)                  logo_file='n64/video.mp4' && loop=true
;;
            naomi)                  logo_file='naomi/video.mp4' &&
loop=true ;;
            neogeo)                 logo_file='neogeo/video.mp4' &&
loop=true ;;
            neogeocd)              logo_file='neogeocd/video.mp4' &&
loop=true ;;
            nes)                  logo_file='nes/video.mp4' && loop=true
;;
            ngp)                  logo_file='ngp/video.mp4' && loop=true
;;
            ngpc)                  logo_file='ngpc/video.mp4' &&
loop=true ;;
            o2em)                  logo_file='o2em/video.mp4' &&
loop=true ;;
            pcenginecd)             logo_file='pcenginecd/video.mp4' &&
loop=true ;;
            pcengine)              logo_file='pcengine/video.mp4' &&
loop=true ;;
            pcfx)                  logo_file='pcfx/video.mp4' &&
loop=true ;;
            pokemini)              logo_file='pokemini/video.mp4' &&
loop=true ;;
            prboom)                 logo_file='prboom/video.mp4' &&
loop=true ;;
            psp)                  logo_file='psp/video.mp4' && loop=true
;;
            psx)                  logo_file='psx/video.mp4' && loop=true
;;
            satellaview)          logo_file='satellaview/video.mp4' &&
loop=true ;;
            saturn)                 logo_file='saturn/video.mp4' &&
loop=true ;;
            scummvm)              logo_file='scummvm/video.mp4' &&
loop=true ;;
            sega32x)              logo_file='sega32x/video.mp4' &&
loop=true ;;
            segacd)                 logo_file='segacd/video.mp4' &&
```

```
loop=true ;;
            sg1000)                 logo_file='sg1000/video.mp4' &&
loop=true ;;
            snes)                  logo_file='snes/video.mp4' &&
loop=true ;;
            sufami)                 logo_file='sufami/video.mp4' &&
loop=true ;;
            supergrafx)             logo_file='supergrafx/video.mp4' &&
loop=true ;;
            thomson)               logo_file='thomson/video.mp4' &&
loop=true ;;
            vectrex)               logo_file='vectrex/video.mp4' &&
loop=true ;;
            virtualboy)             logo_file='virtualboy/video.mp4' &&
loop=true ;;
            wswan)                  logo_file='wswan/video.mp4' &&
loop=true ;;
            wswanc)                 logo_file='atari800/video.mp4' &&
loop=true ;;
            x68000)                 logo_file='x68000/video.mp4' &&
loop=true ;;
            zx81)                  logo_file='zx81/video.mp4' &&
loop=true ;;
            zxspectrum)             logo_file='zxspectrum/video.mp4' &&
loop=true ;;
            *)
        esac
    ;;
esac

draw_logo () {
    loop=0
    if [ $3 = true ]; then loop=-1; fi
    dd if=/dev/zero of=/dev/fb_ 2>/dev/null >/dev/null   #clear before
draw
    ffmpeg -hide_banner -re -stream_loop $loop -i $1/$2 -c:v rawvideo -
pix_fmt rgb565le -f fbdev /dev/fb_ 2>/dev/null >/dev/null &
    echo $! > $cache_file
}

# draw logo
if [ "$logo_file" != "" ]; then
    # kill last played video if running
    if [ -s $cache_file ] ; then
        kill -9 $(cat $cache_file)
    fi
    # draw
    draw_logo $logo_folder $logo_file $loop
fi
```

```
exit 1
```

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

Permanent link:
https://www.wiki.batocera.org/launch_a_script_legacy

Last update: **2023/10/21 04:43**