

Satori Circulation 0.763 Beta

Config Files Documentation

last_update_20_may
© SATORI, s.r.o. 2002-2007

```
*****  
*  circulation.ini  *  
*****
```

Command line parameters for Circulation:

circulation.exe [config_file\$]

Hint: By default circulation.ini config is loaded, however you can enter a different file.ini filename as the command-line parameter.

General syntax for Circulation config tags is:

tag\$ = [value# | value\$]

To put some line into comment use:

; this is comment

use can also use this

-\ Circulation general tags

Enable interactive mode, this command must be used in order to have switched on interactive inputs to software.

interactive

Note: Useful if you create non-interactive scripted installations with Circulation.

Allow logging of loading sequence and errors to circulation.log.

log

If you want to select your custom log file stored in executing folder:

log = file\$

You can get rid of Y/N question if you quit app.

fastquit

Disables loader splash with monitoring of loading sequence.

noloader

Set lower priority to loader splash.

loader_priority = val#

val# = <0..20> {1}

Note: Use 0 for very high priority of loader, 10 for low and 20 for very low.

Recording of Circulation actions in timeline script (.ctl) stored in folder dat\.

record = file\$.ctl

Playback of recorded script or every other kind of script stored in folder dat\.

replay = file\$.ctl

This tag disables recording of Circulation actions so 'record' tag has no effect but 'replay' can be used.

norecord

To hold a specific maximum fps use command.

fps_max = val#

50fps us used as default.

fps_max

No real-time mode, but off-line 25fps (the same one happening while saving to AVI).

Fps25

Setup AVI saving frame-rate (default 25fps).

```
fps_avi = 25 //TO-DO
```

No limit in real-time frame-rate (all depends on power of your computer).

```
fps_nolimit
```

Your recorded AVI videos will be stored into this folder.

```
avi_dir = [relative_path$ | absolute_path$] {\avi}
```

TGA screenshots going to be stored in this folder.

```
snap_dir = [relative_path$ | absolute_path$] {\snap}
```

Your composition presets (.crs) are stored here.

```
state_dir = [relative_path$ | absolute_path$] {\state}
```

Configure the usage of screens on your system.

```
display_primary = [gui | output | <none>] {gui}
```

```
display_secondary = [gui | output | <none>] {<none>}
```

```
display_tertiary = [gui | output | <none>] //TO-DO
```

gui works in 1024x768x32bits mode or higher (if not redefined, see below)

output works by default in 640x480x32bits mode (if not redefined, see below)

<none> string defines no usage of this display

Generally for output is 640x480x32bits mode is initialised but you can select another mode and your output will be interpolated to this resolution.

Hint: This is usefull if you use for example Sony Anycast which accepts only 1024x768 and higher input modes for VGA (very stupid but anyway you have this option to go around).

```
output_mode_width = width#
```

```
output_mode_height = height#
```

Note: If new selected mode is not a proper 32bits video mode of your graphics card 640x480x32 default is applied.

Automatically starts AVI recording from the beginning of app start.

```
saveavi
```

Don't split AVI file each 1 minute (respectively every 1500 frames).

```
saveavi_nosplit
```

Saves video in 720x576 format but without stretching.

```
saveavi_pal
```

Select x position of 640x480 output in PAL frame.

```
saveavi_pal_x = pos_x#
```

```
pos_x# = <0..79> {40}
```

Select y position of 640x480 output in PAL frame.

```
saveavi_pal_y = pos_y#
```

```
pos_y# = <0..95> {48}
```

```
saveavi_pal_stretch //TO-DO
```

Before final output you can also apply the following image operations: image shader and image shift.

First the image is shifted and then shaded via RGB image. This is usefull if you need to simulate widescreen mode etc.

Note: shademask must be stored in folder dat\ - however "dat\" is not used in file path!

```
output_shader = [path$\file$ | file$]
```

(640x480xRGB24) image (tga/jpg) as output shader mask stored in dat\path\$\file\$

Shifts final output pos_x# pixels horizontally.

```
output_shift_x = pos_x#
```

```
pos_x# = <0..639>
```

Shifts final output pos_y# pixels vertically.

```
output_shift_y = pos_y#
```

```
pos_y# = <0..479>
```

Enables to operating of GUI in other resolutions than 1024x768 and interface is centered in the middle of higher res.

```
gui_mode_width = width#
```

```
gui_mode_height = height#
```

This RGB bitmap (tga/jpg) placed in dat\ folder must be in resolution of selected GUI mode.

```
gui_wallpaper = [path$\file$ | file$]
```

(width# x height# x RGB24) image (tga/jpg) as wallpaper stored in dat\path\$\file\$

Note: Wallpaper must be stored in folder dat\ - however "dat\" is not used in file path!

Enables posteffects and effects pannels LOCK switch by default after startup.

gui_plock_on

Disables interpolated rescaling in Quadra view mode and uses previous nonaliased but a bit faster scaling.

quadra_raw

This option disables audio usage. Otherwise default initialization of default audio device.

nosound

Register source 'name' to be loaded (without this you cannot use this source in Circulation).

source_name\$

name\$ = <aviplay, picblur, shifter, black, noise, capture, capture4x, typer, slider, render, swfplay>

You can assign to F1..F10 keys source plugins and activate source by pressing to selected focus (source1, 2, 3).

F# = source_name\$

= <1..10>

name\$ = <aviplay, picblur, shifter, black, noise, capture, capture4x, typer, slider, render, swfplay>

You can assign to CTRL-F1..F10 keys source plugins and activate source by pressing to selected focus (source1, 2, 3).

CF# = source_name\$

= <1..10>

name\$ = <aviplay, picblur, shifter, black, noise, capture, capture4x, typer, slider, render, swfplay>

By stating this tag you enable Open Sound Control network listening mode on this port. Please use supplied utility

oscmd.exe to send commands from other computer and test scriping control of Circulation.

osc_port = port#

This is OSC address on which script commands are accepted and without using this tag it is by default set to *"/circulation"*.

osc_address = address\$ {/circulation}

-\ source plugins tags

You can apply buffer effects on live capture stream and you can select how big buffer you gonna use. Be sure if you work in bigger

capture resolution (e.g. 640x480) and have insufficient RAM, you select a reasonable value (16-32).

capture_buffer_size = size#

size# = <1..128> {8}

If you want to use previously selected capture options (source, image quality and resolution) on startup in capture config

dialogs you can supress this action with the following tag.

capture_nodialogs

Select which subfolder in dat\pictures folder is used for input of images. Configure list.txt file inside as list of used images.

picblur_dir = dir\$

Selects which list file in dat\aviplay will be used for config.

aviplay_list = file\$.txt

Selects another as default config list in dat\swfplay folder.

swfplay_list = file\$.txt

You can select which truetype will be used (currently only 1 true-type is possible in use) located in dat\typer folder.

typer_font = file\$.ttf

This file definies current list of noise images to be loaded in dat\noise folder.

noise_list = list\$.txt

-\ midi mapping tags

All tags are in the format:

```
midi_tag = controller#  
controller# = <0..127>
```

You can also define contasts:

```
midi_slider1 = controller#  
midi_slider2 = controller#  
midi_button1 = controller#
```

etc. so first you can define all inputing midi controllers (sliders/knobs/buttons and define controller address for each of them)

...

and later assign for tags

```
midi_tag = midi_slider1  
midi_tag2 = midi_button1
```

There are 2 types of midi tags, marked with prefix **ANA** or **BUT**:

- * **ANA** - first type is accepting sliders and knobs as input
- * **BUT** - second type are triggers which are connected to buttons

Note: Now midi support is designed for mixer with absolute midi controlers and control of sources and posteffects with relative midi controllers (see below).

Circulation general:

```
BUT midi_snap_screen  
BUT midi_snap_state  
BUT midi_save_avi  
BUT midi_monoscope
```

Basic controls of mixer panel:

```
ANA midi_mixer_xfade  
ANA midi_mixer_xfade2  
ANA midi_mixer_contrast  
ANA midi_mixer_brightness  
ANA midi_mixer_colorize  
ANA midi_mixer_colorize_color  
ANA midi_mixer_blur  
ANA midi_mixer_tdistort  
ANA midi_mixer_tdistort_map
```

Midi controls for mixer 1:

```
ANA midi_mixer_fader_fmap  
ANA midi_mixer_fader_key_level  
BUT midi_mixer_fader_fmap_3rd  
BUT midi_mixer_fader_outline  
ANA midi_mixer_fader_outline_r  
ANA midi_mixer_fader_outline_g  
ANA midi_mixer_fader_outline_b  
BUT midi_mixer_fader_rotate  
ANA midi_mixer_fader_rotate_add  
BUT midi_mixer_fader_rotate_dir  
BUT midi_mixer_fader_seq  
ANA midi_mixer_fader_seq_delay  
ANA midi_mixer_fader_gen_rnd  
ANA midi_mixer_fader_gen_sin  
ANA midi_mixer_fader_gen_frq  
ANA midi_mixer_fader_gen_dly
```

Midi controls for mixer 2:

```
ANA midi_mixer_fader2_fmap  
ANA midi_mixer_fader2_key_level  
BUT midi_mixer_fader2_outline  
ANA midi_mixer_fader2_outline_r
```

```
ANA midi_mixer_fader2_outline_g
ANA midi_mixer_fader2_outline_b
BUT midi_mixer_fader2_rotate
ANA midi_mixer_fader2_rotate_add
BUT midi_mixer_fader2_rotate_dir
BUT midi_mixer_fader2_seq
ANA midi_mixer_fader2_seq_delay
ANA midi_mixer_fader2_gen_rnd
ANA midi_mixer_fader2_gen_sin
ANA midi_mixer_fader2_gen_frq
ANA midi_mixer_fader2_gen_dly
```

Mouse generator:

```
ANA midi_mixer_mouse_gen_rnd
ANA midi_mixer_mouse_gen_sin
ANA midi_mixer_mouse_gen_frq
ANA midi_mixer_mouse_gen_dly
```

Additional master posteffects:

```
ANA midi_mixer_key_down
ANA midi_mixer_key_up
ANA midi_mixer_key_color
ANA midi_mixer_row_size
ANA midi_mixer_row_pos
ANA midi_mixer_row_start
```

Sound

```
ANA midi_mixer_master_volume
```

-\ midi touchpad tags

(e.g. for Korg's KaosPad1/2 - please use servise menu of KP to set your desired midi controllers for 6 buttons, X/Y axis and hold state).

With this tag you activate usage of touchpad.

```
midi_touchpad_on
```

```
ANA midi_touchpad_x
ANA midi_touchpad_y
```

Expecting positive value if touchpad is pressed and zero if not used.

```
ANA midi_touchpad_onoff
```

```
BUT midi_touchpad_but1
BUT midi_touchpad_but2
BUT midi_touchpad_but3
BUT midi_touchpad_but4
BUT midi_touchpad_but5
BUT midi_touchpad_but6
```

With 6 buttons you select which section of Circulation to control with touchpad.

buttons sequence X axis controls Y axis controls		
=====		
1	mixer2 crossfade	mixer1 crossfade
1, 4	brightness	contrast
1, 5	colorize color	colorize level
1, 6	mouse x	mouse y
2	TPX in MPE	TPY in MPE
2, 4	TPX in PE1	TPY in PE1
2, 5	TPY in PE2	TPY in PE2
2, 6	TPX in PE3	TPY in PE3
3	blur level	time distort level
3, 1	source1 X	source1 Y
3, 2	source2 X	source2 Y
3, 3	source3 X	source3 Y
=====		

If you use touchpad and activate it with midi_kaospad_on tag you will obtain in posteffects, the possibility to choose not just GN1 and GN2 for certain values but also **TPX** and **TPY**.

In the appropriate touchpad focus you can control those values where **TPX** or **TPY** is turned **ON** as 'generator'.

Support for sources is always specific and depends on selected source (e.g. video scratching for aviplay).

Note: if you press for example button 1 on touchpad you control crossfade1/2 and your button focus is (1).

If you press now button 4, you control directly brightness/contrast and don't need to press 1, 4 sequence (but you can) and if you then press 5 you control colorizer directly.

```
*****
*  aviplay\list.txt  *
*****
```

A configuration file is located in `dat\aviplay` folder. By default `dat\aviplay\list.txt` is searched and loaded or, alternatively, you can set a different configuration list in `circulation.ini` by using the **aviplay_list=<list_file>** tag.

This file includes a list of videos assigned for use by AVIPLAY source and you can setup default settings for each clip.

Every line of the configuration file describes one video.

In this document you can use for commenting:

```
; comment
# comment
/*
start of commenting
all comments // c-style
commenting ends here
*/
```

No comments possible end the end of valid configuration line. So this is not allowed:

```
1 Q helloworld.avi // my first video
```

Compulsory part of the configuration are 3 parameters separated with SPACE and describing:

- 1) number of BANK,
- 2) KEY letter for the video
- 3) followed by it's relative or absolute path.

In case of spaces inside file-path the whole string must be in " ... ".

syntax:

```
b# k$ [""]<file$.avi | relative_path$\file$.avi | absolute_path$\file$.avi>[""] [additional
parameters]
b# = <0..99>
k$ = <A..Z>
```

What is altogether 2600 combinations and available positions for your videos.

Note: Circulation creates thumbnail image in the same location of the video (videofilename\$.tga)

[additional parameters] syntax:

Other parameters are voluntary and set various attributes and they are separated by SPACE:

```
[name=name$]
[group=group_name$]
[prev=prev_name$]
[next=next_name$]
```

Note: Name strings can NOT contain the following characters 'SPACE' and '=' otherwise the commands are not parsed and recognized incorrectly.

```
[nolooop | pingpong]
[start_frame=fr#]
[end_frame=fr#]
[deinterlace] [odd_field | even_field]
[zoom_all]
{sound=audio_file$} //TO-DO
{midi=controler#} //TO-DO
```

example:

```
; the very basic and compulsory setup: <b#> <k$> <file.avi>
1 Q hello.avi
; usage of absolute path and extra attributes
1 W "c:\Program Files\lol.avi" noloop deinterlace even_field
; looping 3 video in a row
1 E vid1.avi name=one next=two start_frame=10 end_frame=30
1 R vid2.avi name=two next=three
1 T vid3.avi name=three next=one
```

If you wish to automatically assign key letters to videos there is simplified form. You define which bank you assign videos in with

into_bank = bank#

bank# = <0..99>

and then every video config line starts instead of "<b#> <k\$>" with "__" (2 times underscore)

example:

```
into_bank=3
__ circ1.avi
__ circ2.avi
__ circ3.avi
; will be assigned 3 Q, 3 W, 3 E
```

If you absolutely don't care about destination bank you can use:

bank_free

tag to declare assigning videos to first available free slot in bank starting from 0 to 99.

*** swfplay\list.txt ***

A configuration file is located in dat\swfplay folder. By default dat\swfplay\list.txt is searched. You can set a different configuration list in circulation.ini by using the **swfplay_list**=<list_file> tag. This file configures a list of flash files assigned for use by SWFPLAY source.

Every line of the configuration file describes one flash file.

Note: In this document you can use the same commenting rules as for aviplay\list.txt configuration file.

Compulsory part of the configuration are 3 parameters separated with SPACE and describing:

- 1) number of BANK,
- 2) KEY letter shortcut for the flash file
- 3) followed by it's relative or absolute path.

In case of spaces inside file-path the whole string must be in " ... ".

syntax:

```
b# k$ ["]<file$.swf | relative_path$\file$.swf | absolute_path$\file$.swf>["] [additional
parameters]
b# = <0..9>
k$ = <A..Z>
```

What is altogether 260 available positions for your flash files.

~~Note: Circulation creates thumbnail image in the same location of the flash file (flashfilename\$.tga) [TO-DO]~~

[additional parameters] syntax:

Other parameters are voluntary and set various attributes and they are separated by SPACE:

```
[bgcolor=color#]
color# = <RR#GG#BB#>
RR# = <00..FF>, GG# = <00..FF>, BB# = <00..FF>
```

Note: Background color is represented by it's hexadecimal number value:

white = FFFFFFFF, black = 000000, red = FF0000, yellow = FFFF00 etc.

example:

```
1 Q spiska.swf bgcolor=000000
1 W balloonhead.swf bgcolor=FFFFFF
```

* slider\list.txt *

This is a configuration file located in [dat\slider](#) folder. In this file you configure your slider show set.

Every line of the configuration file describes one slide show.

Note: In this document you can use simple commenting rules:

; comment

comment2

Compulsory part of the configuration are 4 parameters separated with SPACE and describing:

1) KEY letter shortcut for the slide show,

2) followed by it's subdirectory name where actual images are stored,

Note: directory name can't hold any space characters otherwise the config line is misinterpreted.

3) still show time in milliseconds

4) crossfade time in milliseconds

syntax:

k\$ subdir\$ still_time# cross_time# [additional parameters]

k\$ = <0..9 | A..Z>

You have an option of 36 different slide shows to be configured and used (26 for A-Z + 10 for 0-9).

[additional parameters] syntax:

[**rand**]

With this parameter you allow random selection of images inside slide show instead of displaying in specified order.

example:

Q bank1 500 1000

W bank2 1000 2000 rand

Inside specified subdirectories (in our example it's *bank1* and *bank2*) all images for certain slider set are stored.

For each folder a **list.txt** must be created holding the sequence of filenames to be used.

Each image file is on one line. Only jpg and tga images are loaded in exact resolution of 640x480, RGB or gray scale mode.

day\slider\subdir\$\list.txt syntax:

filename1\$.tga | jpg]

filename2\$.tga | jpg]

filename3\$.tga | jpg]

...

example:

; example of configuration list.txt located in dat\slider\bank1 holding 3-image-sequence

; please note these images are stored in the same folder

image1.jpg

image2.jpg

image3.jpg