OBS Tool

Version 0.1 – by **BloodyCactus** – 20230517

What it is

OBS Tool operates as a client/server to create timers + counters as application windows that can be seen by OBS designed to work in conjunction with a stream deck.

There are two components, the server part which draws the sections and listens for commands, and the client side which sends the commands to the server.

The idea is that you have a config file per game, but in each config the death counter is named the same between them all, or the timer is named the same, so its all controlled though the same calls, and then you dont have to re-setup OBS for each game, the window titles stay the same, so OBS does not "forget" them etc, and a single name means only 1 button on the stream deck to do the same thing across all your games!

How to Run it?

** IMPORTANT ** On Windows, the client and server apps have been setup with dll redirection, so you should be good to run the exe from anywhere and have it work (Ive tested it in Windows XP etc), without and needing to install anything extra.

Windows 11 (and probably 10?) firewall security pops up, because we sit on a port, so it only needs localhost access.

If your on Linux this should also not be a problem as its taken care of by the AppImage.

You can run it with just pointing to a config.ini file

```
obs_tools_server -g gamefile.ini
```

This is a client/server system, by default if you dont specifiy a server machine name, it will use localhost, which should on Windows, avoid any issues with the security manager (but you may have to give it permission to sit on a port, I dont know Windows enough).

The only real system configuration is that server needs to know your font directory, it assumes for

Windows $(SYSTEMROOT)/fonts => eg: windows/fonts usually.

Linux No default assumed, so manually set it to .fonts or wherever your local font directory is (and thus, your font names are case sensitive!).

Server Configuration

A configuration file is generated the first time you use it, here you can point to your font directory.

You can configure the port, font directory.. The Image directory option is a leftover, you can ignore it.

Config File Locations

Windows 64bit I think its in

```
AppData/Roaming/RaucousChicken/obs_tools/server.ini
```

Windows 32bit (aka XP)

```
Application Data/RaucousChicken/obs_tools/server.ini
```

Linux

```
~/.local/share/RaucousChicken/obs_tools/server.ini
```

Game Configuration

Counters and Timers, Gamebars, Oh My!

OBS Tool has a concept of a Counter, Timer, and collectively a GameBar. (With a side order of Text).

A GameBar is just a grouped collection of 1 or more Counters, Timers and Texts.

If you only need a timer, just have a timer. If you want to associate say a timer and a death counter, its probably nicer to combine them into a GameBar!

Each top level item is its own window, and because they are windows, unfortunatly they dont have a transparency when OBS captures the window.

This allows you to have a separate window just for a timer, or just for deaths, etc. (See the examples/example_separate_items.ini). Because the everything gets the same name each time, this allows OBS to not need to be reconfigured every time you change games!

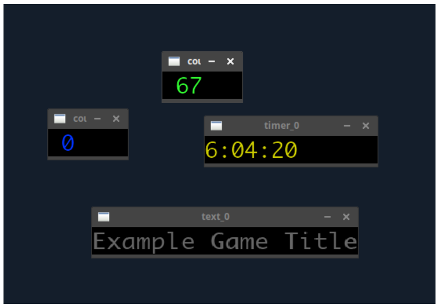


Figure 1: Example with just top level items

Stream Deck Command Examples

See below for example commands to add to the stream deck, here you can see that giving things a common name between all games (timer/deaths/hints) allows the commands to work across games regardless.

Start Timer

Start a timer named 'timer'

```
obs_tools_client.exec -c "timer_start|timer"
```

Stop Timer (Pause)

Stop the timer named 'timer'

```
obs_tools_client.exec -c "timer_stop|timer"
```

Increment Death Counter

Add 1 to the counter named 'deaths'

```
obs_tools_client.exec -c "counter_inc|deaths"
```

Increment Hint Counter

Add 1 to the counter named 'hints'

```
obs_tools_client.exec -c "counter_inc|hints"
```

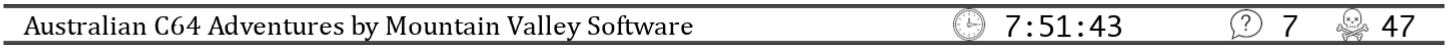
Shutdown obs_tools (save)

Close the current config

```
obs_tools_client.exec -c "shutdown"
```

Game Config Example

The below example configuration will produce the following;



Australian C64 Adventures by Mountain Valley Software 7:51:43 7 47

Figure 2: GameBar Example

We have a gamebar encapsulating a Text Title, a Timer, a Death Counter and a Hints Counter.

First we tell the tool we have 1 top level gamebar.

```
[obs_tools_game]
title = c64
timer_count = 0
counter_count = 0
gamebar_count = 1
```

The first gamebar item (counts from 0) contains a background image, 1 timer, 2 counters and 1 text.

```
[gamebar_0]
name = gamebar
background = game_bar_wide32.png
timer_count = 1
counter_count = 2
text_counter = 1
```

And thus the first timer (0) becomes gamebar_0_timer_0, and we have named it “timer”, its using the “monaco.ttf” font, point size 28, it has a foreground colour of black (0x000000). We are NOT displaying the days in front of our timer (so its just hours + minutes + seconds). the text_offset row/col says where to put it on the gamebar, and the last part, current_time is hex time in milliseconds.

```
[gamebar_0_timer_0]
name = timer
font = monaco.ttf
size = 28
colour = 0x000000
no_days = true
text_offset_row = 0
text_offset_col = 1000
current_time = 0x1AFDFB6
```

The first of our two counters is named “deaths” and the second one “hints”, the big change here is we are not displaying lead zeroes on the counter.

```
[gamebar_0_counter_0]
name = deaths
font = monaco.ttf
size = 28
colour = 0x000000
lead_zeroes = false
text_offset_row = 0
text_offset_col = 1360
current_counter = 0x2F

[gamebar_0_counter_1]
name = hints
font = monaco.ttf
```

```
size = 28
colour = 0x000000
lead_zeroes = false
text_offset_row = 0
text_offset_col = 1260
current_counter = 0x7
```

Lastly we have our textual title

```
[gamebar_0_text_0]
name = title
font = CAMBRIA.TTC
size = 28
text_offset_row = 0
text_offset_col = 20
colour = 0x000000
text = Australian C64 Adventures by Mountain Valley Software
```

Server Command API

These are the commands that you can send to the server. Parameters are separated with a pipe symbol (|)

Global API Commands

shutdown

Tells the server to flush the current game config to disk, and close down.

flush

Tells the server to save the current game config to disk (but keep running).

Timer Commands

stop_all_timers

Puts all running timers into pause

start_all_timers

Starts all timers… timing!

start_timer | {named_timer}

Starts a specific named timer. eg: "start_timer|how_long_i_played_this_game_for_timer"

stop_timer | {named_timer}

Stops a specific named timer. eg: "stop_timer|my_timer"

Counter Commands

counter_inc | {named_counter}

Adds 1 to a named counter. eg: "counter_inc|how_many_times_i_died"

counter_dec | {named_counter}

Subtracts 1 from a named counter. eg: "counter_dec|hearts"

counter_clear | {named_counter}

Reset a named counter to zero (0). eg "counter_clear|hits_taken"

counter_set | {named_counter} | {integer value}

Set a named counter to a specific number. eg: "counter_set|my_lives|3"

counter_set_random | {named_counter} | {lower_bound} | {upper_bound}

Randomises the named counter to a number >= the lower bound and <= upper bound. So calling with 0-10 would produce a number from 0 to 10… Hence, 1-10 would produce 1 to 10…