

OBS NINJA AUTOMATION CHEAT SHEET



MAC - CHROME FROM COMMAND LINE

- run any macOS app from the command line or from a script using the **open** command
- helpful parameters for the open command are -a (the app), -n (open a new instance) and --args (any other arguments that are passed to the app)
- example command line to open Chrome with a URL:
`open -n -a "Google\ Chrome" --args "https://obs.ninja/?push=s001&q=1"`
- the URL passed as an argument can contain the entire OBSN stream configuration

MAC - CHROME FROM APPLICATION

- use Automator to create an app that starts Chrome with the parameters of your choice
- in Automator, create a new Application
- add the "Run Shell Script" action
- add your command line to the Run Shell Script box
- example: `open -n -a Google\ Chrome --args "https://obs.ninja/?push=s001&q=1"`

ELECTRON APP FROM COMMAND LINE

- the electron app handles the following parameters:
`-t TITLE`
`--width WIDTH`
`--height HEIGHT`
`--url URL`
- example macOS command line:
`open -n -a OBSN --args -t GUEST1 --width=1280 --height=720 --url "https://obs.ninja/?view=s001"`
- example Windows command line:
`OBSN -t GUEST1 --width=1280 --height=720 -u https://obs.ninja/?view...`

ELECTRON APP FROM MAC APPLICATION

- use Automator to create an app that starts the electron app with the parameters of your choice
- in Automator, create a new Application
- add the "Run Shell Script" action
- add your command line to the Run Shell Script box
- example macOS command line:
`open -n -a OBSN --args -t GUEST1 --width=1280 --height=720 --url "https://obs.ninja/?view=s001"`

START HERE (GENERAL INFORMATION)

- in OBS Ninja (OBSN) you normally select video/audio input/output devices in the GUI
- alternatively you can use URL parameters to automate this selection and build a hands-free solution
- this is especially helpful if you repeatedly need to open the same devices and for scripting
- you can start Chrome or the electron app from the command line with according parameters
- you can use scripts to start multiple instances of OBSN simultaneously
- you can even pre-select audio and video input devices and an audio output device for quicker setup
- on Mac you can even create an Application to start OBSN with no further manual interaction

AUDIO OUT - SINK METHOD

- after you selected an audio output device inside the # ELECTRON APP, you will notice that the URL got a `&sink=` parameter added to it
- this parameter is the internal ID of the exact audio device that you selected
- you can reuse that sink parameter to circumvent future output device selection
- the `&sink` parameter is browser and version specific. After an update to a new version of the electron app or a new version of Chrome, the sink will be different
- also `obs.ninja/devices` will show you all IDs

Parameter reference: `params.obs.ninja`
Github: `github.obs.ninja`
Discord: `discord.obs.ninja`
Reddit: `reddit.com/r/obsninja`

NO-CLICK AUTO JOIN

- to make the process of joining a stream even quicker (e.g. hands-free), OBSN offers the `&autojoin` parameter
- when added to the sending URL, and when combined with `&ad`, `&vd` and/or `&od`, it can make joining with the right devices pre-selected a one-click process
- auto join requires the `&wc` parameter to also be present

AUTO DEVICE SELECTION

- Instead of manually selecting devices, you can ask OBSN to select based on name:
`&ad=<audio input device>`
`&vd=<video input device>`
`&od=<audio output device>`
- `<... device>` will be pattern matched against all devices of the according type on your system
- the match is case independent and will find all respective devices that contain that string
- if the string matches multiple devices, the first matching device will be used by OBSN
- example:
`...&ad=rode&vd=face&od=rode` will pre-select "RODE NT-USB" and "FaceTime HD Camera (built-in)" for audio/video input and "RODE NT-USB" as headphone output

WIN - CHROME AUTOMATION

- [I need help here. How do I create simple shortcuts or some form of pseudo app on Windows to simplify starting complex command lines?]