



Show Mode for Windows user guide

Show Mode is a notification-area app that locks a Windows PC down for a show, then puts every setting back how you had it. One click (or **Ctrl+Alt+Shift+F11**) stops the PC sleeping, stops notifications and system sounds landing on an output, blacks out the wallpaper so your desktop never appears on the projector, keeps the projector from becoming the main display or a mirror, keeps a stray Windows key and the mouse off the audience's screens, and can mute every app that isn't the show. Before doors, its pre-show checks point out what it can't switch off, a waiting Windows restart above all.

Status (2026-10-24): a preview, released at v0.2.0 and unsigned. Bench-tested on two Windows 11 25H2 machines, not on a show.

Checked on winlab (Windows 11 Pro 25H2, build 26200, x64 VM, one display, run with the token a standard user gets): the app starting in the signed-in session, the tray icon in both states on a light taskbar, the menu (opened through the icon), one copy only, and `showmode://on` and `off` handed over from a second copy. Starting a show took the keep-awake request, switched to a copy of the power plan, turned Snap off, turned Do Not Disturb on (priority only, and Windows' own reading agreed), hid crash dialogs, and left the wallpaper alone because it was Windows Spotlight, saying so in the note. Ending it put every one back exactly as found: the original plan active and the copy deleted, Snap on, Do Not Disturb off, the crash setting gone. Killing the app outright mid-show left everything changed, as it should, and the next launch found the journal, put it all back and said so. Ctrl+Alt+Shift+F11 registered (the menu shows it), but nobody pressed it.

Checked on a Parallels guest (Windows 11 25H2, ARM64), recording the output with test tones at -40 dBFS: with muting on, the allowed app's 1 kHz tone stayed at -40 dB while the 440 Hz app dropped below -120 dB, and so did a second 440 Hz app opened mid-show. Ending the show brought both back and the watchdog went with it. Killing Show Mode mid-show unmuted the running app at once. An app that had quit while muted came back muted when relaunched (Windows had kept the mute), and was unmuted the moment Show Mode was relaunched.

Fixed during that bench: a command handed over by a `showmode:` URL couldn't reach Do Not Disturb or the wallpaper; Snap didn't come back on at the end (it did once that was fixed, and the restore now reads the setting back); Spotlight wasn't recognised on 25H2; and Show Mode took over a `showmode:` handler that belonged to something else.

The display lock, checked on winlab (x64 build, standard-user token) with the Virtual Display Driver adding two 800×600 screens beside its own 1280×800 one. Mid-show, a virtual screen made main was moved back about a second later. The two virtual screens mirrored onto one desktop were split back out a second later, and so was Windows' own Duplicate of all three. Both virtual screens dropped out, then came back in Duplicate, which Windows had been set to remember for them; they were extended again two seconds after they reappeared. A mirror that was there at the start was left alone through the show and after it. Both screens dropped out and came back for half a second each: nothing needed correcting, and nothing was done. Something moving the main display every 2.5 seconds got four corrections, then the lock stood down and said so in a note and the menu. Ending that show made the original display main again, and so did the next launch after the app was killed mid-show and a virtual screen made main. After every correction, the layout Windows remembers for those screens read back unchanged.

The pre-show checks, checked on winlab: every check read in under 40 ms with a standard-user token. Its Public network and "minimise windows when a monitor is disconnected" were reported as they were. A RebootRequired key, added for the test and then removed, put the

restart warning first in the note (which stayed up 20 seconds) and at the top of the menu, and both went with the key. The taskbar on all displays (with no setting, which showed a taskbar on all three screens, and with it set) and the minimise setting switched off each changed the list. **Not checked:** a restart that the Windows Update Agent or the servicing stack itself reports (winlab had none, and both read "no"); battery and Energy saver (winlab is a VM on mains, and the Mac under the Parallels guest was on mains too); a Remote Desktop session. winlab's display adapter can't do Night light, so that was checked on the guest.

Fixed during that bench: a correction that couldn't work was retried for the whole show, because the five-second check spaced the retries exactly far enough apart for the fight limit never to trip, so two failures in a row now stand the lock down too. Windows' Duplicate across two graphics adapters couldn't be split one screen at a time, so Show Mode now uses Windows' extended layout for that. The taskbar check missed the Windows default.

Do Not Disturb and Windows' Duplicate, checked on winlab (x64 build, standard-user token, the same two virtual screens). While the screens were duplicated, Windows' own "when duplicating your display" rule put an alarms-only Do Not Disturb over Show Mode's priority-only one; it went when the display lock split the screens 1.5 seconds later. With the lock off and the Duplicate still up, ending the show turned Do Not Disturb off, Windows' alarms-only one with it, but the build before this couldn't read it back as off straight away: it kept it in the journal, and the next launch found it already off. That happened once in the display lock bench and once here. The read-back now tries again for up to a second: of five ends of a show during a Duplicate, four read back at once and one after 50 ms, and one start of a show needed 50 ms too.

The ARM64 build, checked on the Parallels guest (standard-user token): `--status` read the one Parallels screen and every pre-show check, in 13 to 29 ms. The Mac's battery shows through to the guest, and on mains it reported nothing. Night light, switched through Settings: turned on, it read as on and the checks listed it; with its schedule on, it read as scheduled and the checks listed that instead; both were put back off. A show with only the display lock on kept the one screen main, had nothing to correct, and ended with everything put back. Killed mid-show, the next launch found the journal and put it back. **Not checked on ARM64:** the display lock with a second screen. Parallels can only give the guest the Mac's own screens (it has one), and Windows wouldn't import the Virtual Display Driver there: with Code Integrity enforced it refused the driver's signature (0x800B0109), even with its root added to the machine's trusted roots for the attempt.

Fixed during that bench: Night light switched off by its own schedule (Windows stores that state differently) read as unknown instead of off. The checks were right either way, since unknown reports nothing, but it reads as off now.

The v0.1.0 downloads, checked on the Parallels guest: the `ShowMode.exe` from each zip ran `--status` and read every guard and pre-show check, the ARM64 one natively and the x64 one under Windows' emulation. They were built from the same code as the bench runs above, but that's the only run the release files themselves have had.

Added in v0.2.0, checked 2026-10-24 on winlab (x64, standard-user token) and the Parallels guest (ARM64).

System sounds, recorded on the guest: with the error sound pointed at a 1 kHz test tone at -40 dBFS, it recorded at -40 dB before the show, nothing at all (-120 dB) during it, whether asked for by name or with MessageBeep, and -40 dB again after. An app's own tone carried on at the same level throughout. On winlab all 49 event sounds went silent, stayed silent through the show and came back as they were stored.

The keys, on winlab: Win, Win+Tab, Win+D, Ctrl+Esc, Print Screen and the Copilot key (Win+Shift+F23), sent as input mid-show, opened nothing. With no show running the same keys opened Search, Task View and the Snipping Tool. Win was blocked on the guest's ARM64 build too, and opened Search again after End and after a kill. The accessibility shortcuts were switched off and back on, but I didn't see five Shifts stay quiet: five Shifts sent as input don't bring up the box even with the shortcut on, so that needs a real keyboard.

The taskbar, Energy saver and AutoPlay, on winlab: the Show Desktop corner left the taskbar for the show and came back after it (read through UI Automation). With Always use energy saver on, Windows reported Energy saver on before the show, off during it, and on again after End and after a kill and the next launch. Window shake and AutoPlay were switched off and back on, but I had nothing to shake or plug in to see either working.

The cursor fence, on winlab, with two 800×600 virtual screens to the right of its 1280×800 one and the middle one ticked: a cursor gliding at it 10 pixels at a time stopped at its edge (x=1279), and a jump onto it landed on that edge. A jump to a point nearer the right-hand screen went there instead. A cursor put on it directly, which no hook sees, was pulled back within 50 ms, and Ctrl+Alt+Shift+F10 paused the fence and set it going again. Ticking the second virtual screen under Keep Cursor Off mid-show fenced it straight away and greyed out the last unticked screen. With every screen ticked the fence stood down and the cursor went anywhere, as it did after End. The menu was opened by its own message and checked by screenshot, since UI Automation couldn't open it from the overflow.

Other apps, on winlab: f.lux, installed for the test, quit a second after being asked and came back with /noshow at the end. OneDrive quit through its own /shutdown and came back. A stand-in that ignores being asked was left running and named in the note; nothing was killed. After a kill mid-show, the next launch reopened them. Syncthing 2.1.7 had its two syncing folders paused and only those resumed, leaving one I'd paused by hand alone, over HTTPS too. After a kill the next launch resumed them, and with Syncthing gone at the end of the show they stayed in the journal until a launch found it running again. A stand-in AnyDesk showed up in the pre-show checks.

The v0.2.0 downloads, checked the same way on the guest: the ShowMode.exe from each zip read all 19 guards and every pre-show check with --status, the ARM64 one natively and the x64 one under emulation. They're built from the code benched above, and that's the only run the release files themselves have had.

Fixed during that bench: the accessibility guard also switched off High Contrast's shortcut, and writing High Contrast's settings, even unchanged, re-applies the Windows theme: 27 of the 49 system sounds came back mid-show, and the start took up to six seconds longer. High Contrast is left alone now. And with Syncthing set up but not running, each call to it cost two seconds while Windows gave up on the connection; Show Mode checks it's running first now.

Not checked yet: SmartScreen and Smart App Control meeting a downloaded copy; turning off a screen saver that is actually on (winlab has none); the wallpaper blackout itself and putting a picture or slideshow back; pressing the hotkey; Launch at Login; Finish Restoring; a restart or sign-out mid-show; the display lock with a real projector, a laptop's built-in screen, or a laptop with two GPUs; the display lock on ARM64 with a second screen; the keys and the cursor fence with a real keyboard and mouse, a touchscreen, or an app running as administrator in front; ticking the main display under Keep Cursor Off (it asks first); f.lux's tint going with it (winlab's display adapter has none); Dropbox, Google Drive, Box, MEGA, Resilio Sync and LightBulb; Windows 11 24H2; and a real show.

This codebase was created with AI assistance, directed and reviewed by a human author.

Install

Download the zip for your PC from the [releases page](#): x64 for nearly every PC, ARM64 for a Snapdragon laptop or Windows on a Mac. Unzip it somewhere it can stay, such as `%LOCALAPPDATA%\Programs\ShowMode`, and run `ShowMode.exe`. There's no installer. Settings and the journal live in `%LOCALAPPDATA%\ShowMode`, and the `showmode:` handler follows the program wherever it's started from. Launch at Login remembers where it was when you ticked it, so tick it again after moving it. To update, quit Show Mode, replace `ShowMode.exe` and start it again.

The builds aren't code-signed yet, so the first time you run one, SmartScreen says "Windows protected your PC". Click **More info**, then **Run anyway**, and it won't ask again for that copy. With Smart App Control switched on, Windows blocks an unsigned app it doesn't know and offers no Run anyway, so Show Mode won't start on that PC until it's signed. I haven't had a PC with it on to try.

Show Mode needs Windows 11. It lives in the notification area as a pair of theatre masks: outlined while show mode is off, filled while it's on. Windows may tuck it under the ^ arrow; drag it onto the taskbar to keep it in view.

Starting and ending a show

Click the masks and choose **Start Show Mode**, or press **Ctrl+Alt+Shift+F11** anywhere. **End Show Mode** (or the same keys again) puts everything back.

If something couldn't be applied, a short note appears by the notification area for a few seconds and stays listed in the menu, marked ▲, until the show ends. It isn't a Windows notification, since show mode may just have switched those off, and it appears on the main display, not the projector.

While a show is running, an ordinary restart or sign-out stops at Windows' list of apps holding it up, with Show Mode's reason. That's a courtesy for a slipped click, not a guard: a restart forced by Windows Update doesn't ask. So Show Mode checks for a waiting restart: if there is one, it leads the note when the show starts and sits at the top of the menu (see [Pre-show checks](#)). Restart before doors.

What it turns off

Untick any guard in **What Show Mode Disables**. A change there takes effect at the next start.

Guard	What it does
Prevent display & system sleep	Holds a power request, the way a video player does. It goes with the app, so a crash can't leave a PC that never sleeps. Windows still ends it on a lid close, the power button, Start > Sleep and the lock screen, which is what the next guard is for.
Show power plan	Switches to a copy of your active power plan with the lid set to do nothing and no display, sleep or hibernate timeouts, so a laptop driving a projector keeps going with its lid shut. Your own plan is never edited; at the end it's switched back and the copy deleted. If you pick another plan mid-show, your choice stands.
Turn off Energy saver	Turns Energy saver off in that same copy of your plan, on battery and on mains, so it can't switch itself on as the battery runs down and slow everything with it. Always use energy saver goes off too. Your own plan isn't touched. If it's still on once the show plan is in, Show Mode says so.
Turn off the screen saver	Only if it's on.
Do Not Disturb	Turns on Do Not Disturb, so no notification banner lands on an output. Whatever it lets through (alarms, reminders and the apps you've set as priority in Settings > System > Notifications) still gets through, so check that list on the show machine. Ending a show while the screens are duplicated also ends the Do Not Disturb that Windows' own "when duplicating your display" rule turned on, as switching it off by hand would.
Turn off system sounds	Switches the sound scheme to No Sounds, as the Sound control panel does, so the error ding, a device plugging in or out and the Recycle Bin can't go out through the show's audio. Sound from apps, the show included, plays as normal, and nothing is turned down. Each event's own sound comes back at the end, and one you change mid-show stays as you set it.
Hide app crash dialogs	When an app crashes, Windows can put a "stopped working" window over everything. This keeps it away; the crash report is still saved.
Turn off AutoPlay	Stops the box asking what to do with a USB stick, a card reader or a phone the moment it's plugged in.
Turn off Snap	Windows stop snapping to the screen edges, the Snap layouts box stops appearing over the maximise button, and dragging a window to the top no longer maximises it.
Turn off the Show Desktop corner & window shake	The far right of the taskbar sweeps every window aside when clicked, the show included. This takes it away, along with title bar shake, which minimises everything else.
Block the Windows key, its shortcuts & Print Screen	See Keys .
Turn off accessibility shortcuts	Shift pressed five times, right Shift held down, Num Lock held down, and left Alt + left Shift + Num Lock each put a box over the show asking whether to turn on Sticky Keys or one of its friends. This switches off the shortcuts, never the features, and

Guard	What it does
	puts them back afterwards. High Contrast's shortcut is left alone (see the status above).
Fence cursor off show screens	See Keeping the cursor off the show screens .
Black out desktop wallpaper	Paints the desktop black on every screen, including one plugged in mid-show, so an extended desktop never shows your wallpaper on the projector. Your pictures, their position and a running slideshow come back at the end. With Windows Spotlight as the background it's left alone, because Spotlight can't be put back exactly.
Lock main display & keep new screens extended	Keeps the display that was main at the start as the main one, and splits a screen that starts mirroring mid-show back out to extended. See below.
Quit f.lux & LightBulb, and reopen them after	Both tint every screen with the time of day, the projector included. See Other apps in the background .
Mute apps not allowed to play	Off by default. See below.
Pause Synching	See Other apps in the background .
Quit sync apps	Off by default. OneDrive, Dropbox, Google Drive, Box Drive, MEGA and Resilio Sync. See Other apps in the background .

Keeping the screens where they are

Windows often puts a laptop meeting a projector it hasn't seen before into Duplicate, and a projector that drops out and comes back gets whatever layout Windows remembers for it. **Lock main display & keep new screens extended** (on by default) holds the layout you started with.

The main display stays put. The display that was main when the show started stays main. If something makes another screen main during the show, Show Mode moves it back about a second later. The arrangement itself is kept: the screens stay where they are relative to each other.

New mirrors become extended. Whatever's mirrored when the show starts is taken as intentional (a confidence monitor showing the output, say) and left alone. A screen that starts mirroring during the show, whether Windows decided to duplicate onto a projector or someone pressed Win+P, is split back out to an extended display, to the right of the others, a second or two later. When Windows won't split one screen off on its own (a mirror across two graphics adapters, as on a laptop with two GPUs), Show Mode asks Windows for its own extended layout for those screens instead, unless something was already mirrored when the show started.

Every correction is temporary. Show Mode never saves a layout, so the layouts Windows remembers for each set of screens stay yours, and the next time these screens meet, Windows

brings back its own.

Each correction makes the screens blink for a moment, the way they do in Settings. If something keeps undoing the layout (four corrections inside twenty seconds), or a correction doesn't work twice running, Show Mode stops, says so in a note and in its menu, and leaves the screens alone for the rest of the show.

When the show ends, the display that was main before it becomes main again. Screens split out of a mirror stay extended: the mirror was Windows' guess, not yours.

Keys

Block the Windows key, its shortcuts & Print Screen (on by default) makes the Windows key do nothing while the show runs, on its own or with anything else. So a slipped hand can't open Start, Task View, Show Desktop (Win+D), another virtual desktop, the Project panel (Win+P), Game Bar or the Magnifier over the show, and the Copilot key goes too, since underneath it's Win+Shift+F23. A key pressed with the Windows key goes with it, so nothing reaches the show app as a stray letter. Ctrl+Esc (Start again) and Print Screen (Windows 11's Snipping Tool, which dims every screen, the projector included) are blocked as well.

What still gets through: Win+L and Ctrl+Alt+Del, which Windows deals with before any app sees them, and the keys while an app running as administrator has the focus, since Windows doesn't let an ordinary app see those. Alt+Tab, Ctrl+Shift+Esc and everything else are left alone. Nothing is written anywhere for this one: if Show Mode stops, the keys work again.

Keeping the cursor off the show screens

Under **Keep Cursor Off**, tick each screen the audience sees. Screens are remembered by Windows' name for the monitor and the socket it's in, so plug the projector into the same port each time. Two screens with the same name show their size and position, so you can tell them apart, and the main display is marked, since the taskbar, the notification area and Show Mode's own icon live there. A change here takes effect straight away, mid-show included.

The cursor always keeps at least one screen: the last unticked screen can't be ticked. Ticking the main display asks first, because during the show you couldn't reach the taskbar with the mouse. Usually the better fix is to make the operator's screen the main display in Settings > System > Display before the show.

Every mouse movement that would land on a ticked screen is caught on its way, and the cursor stops at the nearest edge it's allowed. Unlike the Mac, Windows wants no permission for this. Anything that puts the cursor somewhere without moving it (another program, say) is caught by a check every 10 ms. A screen mirroring one you haven't ticked isn't fenced: it shows the same desktop.

Ctrl+Alt+Shift+F10 pauses the fence (press it again to resume) when you need a show screen on purpose, to drag a window onto it, say. **Pause Cursor Fence** under Keep Cursor Off does the same, and the tip on the masks says when it's paused.

The fence stands down rather than trap you: if every screen is ticked, or the only unticked one gets unplugged, it does nothing until that changes. The lock screen, Ctrl+Alt+Del and a UAC prompt free the cursor whatever any app does, and the fence takes hold again when they've gone. I haven't tried it with an app running as administrator in front, which an ordinary app's hooks don't hear from.

Other apps in the background

Quit f.lux & LightBulb (on by default) asks them to quit for the show and opens them again afterwards, so the projector's colour doesn't drift with the time of day.

Pause Syncthing (on by default) pauses every Syncthing folder that's syncing, through Syncthing's own controls on this PC, and resumes only those afterwards: a folder you'd already paused stays paused. If Syncthing isn't running when the show ends, its folders stay paused until it is, and the next launch of Show Mode resumes them.

Quit sync apps (off by default) asks OneDrive, Dropbox, Google Drive, Box Drive, MEGA and Resilio Sync to quit for the show, and opens them again afterwards. None of them can be paused from outside. **Leave it off if your show media lives in one of their folders**, which on Windows is likelier than you'd think: OneDrive often holds the Desktop and Documents folders, and a file kept online only can't be opened while OneDrive isn't running. With it off, the pre-show checks name any that are running, so you can pause them from their own icons instead.

Each app is asked to quit the way closing it would ask (OneDrive through its own /shutdown), and given five seconds. One that won't go is left running and named in the note. Nothing is ever killed: f.lux killed outright would leave its tint on the screens. They're opened again the way their own startup entries open them, in the background, and after a crash mid-show that happens at the next launch.

Keeping other apps off the PA

When the PC's output feeds a PA, Outlook's new-mail chime or a browser tab is one stray click away from the audience. With **Mute apps not allowed to play** ticked, every app is silent during the show except the ones ticked under **Allowed to Play** in the menu: your show app, and anything else that should be heard. The menu lists the apps with sound now, marks the ones playing, and remembers your choices.

- **It catches sound that's already playing**, as well as anything that starts later. That's one thing Windows does better than the Mac here.

- **Apps that play through ASIO or exclusive mode** (a lot of show and audio software) bypass Windows' mixer, so they can't be muted, and they're never muted by accident either.
- **Apps that record or stream the output are muted too** unless they're ticked. A recording of the output (OBS's desktop audio, say) is an audio session like any other, and a muted one records silence. Tick it under **Allowed to Play** if it should hear the show.
- **Nothing is turned down.** The output's volume and mute are never touched.
- **Windows remembers a mute.** An app muted by Show Mode would stay muted, even after a crash and even the next time it opens. So a small watchdog runs alongside Show Mode while apps are muted and unmutes them the moment Show Mode goes. An app that was muted and then quit mid-show is unmuted the next time it plays, as long as Show Mode is running then (turn on **Launch at Login**).

Pre-show checks

Some things no app can safely switch off. **Pre-show Checks** in the menu lists the ones Show Mode finds on this PC, each with the one thing to do about it. It only looks: nothing in the checks changes anything.

- **A restart is waiting.** Windows Update restarts a PC on its own deadline, and no app can refuse a forced restart, Show Mode included. So when Windows has a restart waiting (Windows Update says so, or the servicing stack has one pending), it leads the note when a show starts, the note stays up longer, and it sits at the top of the menu whether a show is running or not. Choosing it there opens Windows Update. Restart before doors.
- **Running on battery, or Energy saver** on, which slows the PC down.
- **Night light on, or scheduled.** It tints every screen, the projector too, and a schedule can switch it on mid-show. Show Mode reads Night light's settings but never writes them: they're kept in a form only Windows can safely change.
- **The taskbar on every display**, the projector included. Windows 11 does this unless it's switched off in Settings > Personalisation > Taskbar > Taskbar behaviours.
- **Minimise windows when a monitor is disconnected.** If the projector drops out for a moment, Windows minimises whatever was on it, and it doesn't come back by itself. It's on unless it's unticked in Settings > System > Display > Multiple displays.
- **A Public network.** Windows blocks discovery on a Public network, so NDI and Dante may find nothing. Make the show network Private in Settings > Network & internet.
- **A Remote Desktop session.** While someone's connected by Remote Desktop, the PC's own screens show the lock screen.
- **A remote-access tool running:** TeamViewer, AnyDesk, RustDesk, Parsec, Chrome Remote Desktop, Splashtop, TightVNC, UltraVNC or RealVNC. Show Mode doesn't block remote access, since whoever runs the show may need it, but a session can change the resolution or put the wallpaper back when someone connects. Set it not to, and rehearse with it.

- **A backup tool running:** Backblaze, CrashPlan, Carbonite or Duplicati can start a backup mid-show. Pause it for the show.
- **A sync app running** (OneDrive, Dropbox, Google Drive, Box Drive, MEGA, Resilio Sync) while **Quit sync apps** is off. Pause it from its own icon.

The ones that can change during the show (a waiting restart, battery, Energy saver, Night light coming on, the network, a remote session) also stay at the top of the menu, marked ▲, while it runs.

The menu also reminds you to open every show app once before doors, so a Windows Firewall prompt comes then and not mid-show, and to plug in every screen you'll use.

Driving it from somewhere else

Show Mode answers a URL scheme:

```
ShowMode.exe showmode://on
```

The others are `showmode://off` and `showmode://toggle`. Companion's shell action can run that on the same PC. A copy of Show Mode started with a command hands it to the running one and leaves.

If another app already handles `showmode:` (a Parallels guest passes it to the Mac's Show Mode, for one), Show Mode leaves that alone, and `ShowMode.exe showmode://on` still works.

Putting things back

Before Show Mode changes anything it writes down what was there, in `%LOCALAPPDATA%\ShowMode\journal.json`. Ending the show restores those values and nothing else, so a setting you'd already switched off yourself stays off. The same happens when you quit, and if the app crashes or the PC restarts mid-show the next launch finds the journal, restores it and tells you it did. **Launch at Login** in the menu is worth turning on for a show machine, so a restart mid-show is followed by a clean restore.

If something can't be put back, the menu says so and offers **Finish Restoring**, and the next launch tries again.

What it did

Show Mode keeps a plain log of what it changes in `%LOCALAPPDATA%\ShowMode\ShowMode.Log`. **Open Log** in the menu opens it.

`ShowMode.exe --status` prints what it can see (the active power plan, the screen saver, Snap, Do Not Disturb, the system sounds, the taskbar and AutoPlay settings, the accessibility shortcuts, the

wallpaper, the displays and the layout Windows remembers for them, the screens ticked for the cursor fence, apps with sound, the apps it would quit, Syncthing, the journal, and what each pre-show check read) without changing anything. `--status --out <file>` writes it to a file instead.

Show Mode for Windows · v0.2.0 · guide updated 2026-10-24 · stoatworks-labs.com/software/showmode-windows/

Latest version of this guide: stoatworks-labs.com/software/showmode-windows/guide/ · Source and issues:

<https://github.com/stoatworks-labs/showmode-windows>