

Companion — WebLinked user guide

This module drives [WebLinked](#) over its HTTP control API from a Stream Deck or any other Bitfocus Companion surface — graphics, outputs, formats, and the health numbers that tell you whether a page is keeping up.

The [README](#) covers installing the module. This is how to build a surface with it.

Before you rely on this: this module polls. WebLinked has no push channel for state, so every feedback and variable lags reality by up to one poll interval. Nothing here is a live wire.

This module was built with AI assistance, directed and reviewed by a human author.

Connecting

`127.0.0.1:7654` by default. If WebLinked was started with `--bind 0.0.0.0` it should also have `--token` — put that token in the config. **There is no TLS**, so keep it on a trusted show network.

Finding an instance

WebLinked 0.8.0 and later advertise themselves over mDNS, so the config panel offers a **list to pick from** instead of an address to type. Choosing one hides the host and port fields, because they are then ignored.

"Manual" is always available, and is the answer whenever discovery comes up empty. It comes up empty for four reasons, in rough order of how often they are the cause:

- **The instance is on the default loopback bind.** It refuses to advertise on purpose — nothing off its machine could reach the address it would publish — and says so in its log. Start it with `--bind 0.0.0.0`.
- **Companion is on another subnet.** mDNS does not cross a router.
- **Multicast is filtered**, which is common on managed venue networks.
- **The instance is older than 0.8.0**, or was started with `--no-mdns`.

None of these stop the module working. They only stop it filling the address in for you.

Run JavaScript is the high-leverage action

A graphic that defines its own functions needs no integration work at all:

```
lowerThird.show('Anna Kowalski', 'Head of Sound')
setScore('home', 3)
```

That is the action to build a show around. Anything the page can do, a button can do, without anyone writing a protocol for it.

Enabled is not running

- **Enabled** is the configured intent.
- **Running** means the device actually opened and is putting frames out.

An output can be enabled and not running — a DeckLink claimed by another application, or a format the card will not take. The generated output presets light on **running** for exactly that reason, and you should keep it that way.

Disabling an output stops the device and frees it for another application. That is the point of it, not a cosmetic toggle — it is how you hand a card to something else without quitting WebLinked.

The health numbers, and how to read each one

Feedback	Read it as
Dropped ticks	Should be 0 . Anything else means the clock fell behind.
Repeated frames	High is normal for a static graphic. Rising on an animated one means the page is too slow — set the threshold for what <i>this</i> graphic should be doing.
Console errors	The page is throwing. The graphic is probably wrong.
Popup attempts	The page tried to open a window; it is usually about to misbehave.
Audio underruns	A few at start-up are normal. A rising count is not.

Repeated frames is the one that catches people: the same number means "fine" on a static lower third and "this page cannot keep up" on an animation, so the threshold is per graphic.

Background is per output

Each output composites the page over either **its own transparency** — a keyed SDI fill, or NDI with alpha — or **a flat colour**, for a switcher that only has a chroma keyer.

One browser paint serves both, so the same graphic can leave as a key down one path and over green down another, with no second render.

Changing it applies on the next tick **without restarting the output**, so it will not drop frames on air.

The source selector

WebLinked can run several pipelines in one process. Every per-source action and feedback has a **Source** dropdown defaulting to *Primary*, which sends no source parameter at all — WebLinked's own convention.

Leave it alone unless this rig runs more than one pipeline.

When Format returns 409

A 409 means the format did change, but an output could not reopen at it.

Fire **Diagnostics: log the current state** and read `outputs[].error`. That is the only place the real reason appears.

Building a surface that fails safe

1. **WebLinked is connected** on any page carrying output colour — every other feedback keeps evaluating against the *last known* state while WebLinked is unreachable, so a brief blip does not blank the page, and a stale green looks exactly like a live one.
 2. **Output tiles lighting on running**, not enabled.
 3. **Dropped ticks somewhere visible**. It should be zero, so any colour at all is a signal.
 4. **Repeated-frames thresholds set per graphic**, not globally.
-

Troubleshooting

Symptom	Cause
Discovery list is empty	One of the four reasons above. Use Manual; the module still works.
Output shows enabled but nothing is on air	Enabled is intent; check running. The device may be claimed elsewhere.
Format action returned 409	The format changed but an output could not reopen. Log the state and read <code>outputs[].error</code> .
Repeated frames alarming on a static graphic	Normal. Set the threshold for what this graphic does.
Everything on the page is stale	The module polls. Also check <i>WebLinked is connected</i> .
Run JavaScript does nothing	The function does not exist on the page, or the page is throwing — check the console-errors feedback.

See also

- [README](#) — installing, and the full action/feedback/variable list
- [companion/HELP.md](#) — the same material, in Companion's help panel

Companion — WebLinked · v1.1.0 · guide updated 2026-08-22 · stoatworks-labs.com/software/companion-weblinked/

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

<https://github.com/stoatworks-labs/companion-module-weblinked>