

dlvr.it Automation — Summary of Problems and Rationale for Cancellation

Prepared: 31 August 2026

Scope: dlvr.it RSS-to-social automation for Art of FACELESS / The Hollow Circuit (Threads, Bluesky, LinkedIn routes)

What it was supposed to do

dlvr.it was set up to auto-post new content from three RSS feeds (the Art of FACELESS blog, thehollowcircuit.com's lore feed, and the Research feed on artoffaceless.org) out to Threads, Bluesky, and LinkedIn, each tagged with a curated set of hashtags per platform, so new posts would surface automatically without manual re-posting on each network.

Problems encountered

1. **Hashtag field capped at 3 tags on the current plan.** The curated tag sets proposed for each route (8 tags for the main brand/lore routes, 5 for the research route) were truncated by dlvr.it's Fixed Hashtags field to only the first 3, regardless of which tags were actually the most useful for discovery. There was no way to see this cap in advance — it only became visible after the tags were entered and saved.
2. **The 3 tags that survived truncation were the wrong ones.** Because the field kept tags in entry order rather than by relevance, what ended up live on the Hollow Circuit lore feed route was a set of self-referential brand tags (#HollowCircuit #ArtOfFACELESS #Hyperstition) rather than broader, genre/topic tags that a Bluesky user actually searches or follows. Brand-only tags don't help strangers discover a post — they only make sense to people who already know the brand, which defeats the purpose of tagging at all.
3. **No real reach resulted.** Posts going out through this route weren't reaching new audiences. You deleted 5 posts from the feed directly because of this — clear evidence the automated tagging wasn't doing its job.
4. **Legacy plan debt compounded the problem.** A leftover "Dynamic Hashtag Pool" on the Research feed route (a Plus-plan feature) couldn't be deleted on the current plan — it had to be defused by setting it to pull 0 tags rather than removed outright, adding clutter and confusion to the route configuration.

5. **Broken routes elsewhere in the account.** Two Instagram routes and a Pinterest-RSS route were already non-functional and had to be removed entirely as dead weight before the remaining 3 routes could even be properly audited.
6. **High setup-to-output ratio.** Fixing the hashtag caps, auditing and removing dead routes, and adjusting privacy settings took multiple rounds of investigation and manual reconfiguration — a disproportionate amount of time and back-and-forth for a tool whose entire value proposition is saving time.

Root cause

The core issue isn't a one-off misconfiguration — it's a structural mismatch between what the current dlvr.it plan allows (a 3-tag cap, no per-post tag curation, legacy features stuck half-on) and what effective social distribution actually needs (platform-appropriate, discoverable tags chosen per post, not a single fixed set blindly reused across every future post from a feed). No amount of reconfiguration within the current plan fixes the underlying cap — it would take a plan upgrade to even test whether more tags would help, with no guarantee that would fix the reach problem either, since the deeper issue is that generic automated tagging can't replace tags chosen for a specific post's actual content.

Why manual posting wins here

One hand-posted update with tags picked for what that specific post is actually about will consistently outreach several automated posts carrying a fixed, generic tag set. The automation was meant to remove manual effort, but the effort it actually removed (posting) was smaller than the effort it added (auditing, reconfiguring, and troubleshooting a platform whose constraints kept working against the goal) — while also costing a recurring subscription fee for the privilege.

Recommendation

Cancel the dlvr.it subscription and revert to manual posting on Bluesky (and Threads/LinkedIn as needed), where each post's tags can be chosen for that specific piece of content rather than inherited from a fixed, plan-capped list.