

Bloch Node — Reachability / GHOSTDAG Coloring Upgrade

How to apply the exact interval-reachability fix that resolves the ~blue_score 395,897 stall.

Operator guide · dated 2026-07-18 · branch `feat/reachability-durable` · UNAUDITED reference software, pre-1.0

READ THIS FIRST. This is a **COORDINATED soft-fork, not a silent drop-in**. Upgrading the binary alone changes **nothing** on mainnet. The fix only takes effect at a coordinated activation height **H** that the project publishes in its release notes, which every mining/validating node must set before the tip reaches it. See Step 6.

1 Who needs this & what it fixes

If your node froze and reports `syncing: true` without advancing past roughly **blue_score 395,897**, this is for you. The old GHOSTDAG classifier answered DAG-ancestry questions with two **depth-bounded** walks (`past_blue_set` capped at $K \cdot 100 = 1000$ hops; `is_ancestor` capped at 1024). On a deep chain those caps silently truncate and under-count a block's blue ancestors. Two honest nodes that hit the cap differently then pick a different selected parent, derive a different ASERT difficulty target, reject each other's blocks as "invalid difficulty" (around height 394,630), and the chain freezes at blue_score 395,897.

The fix replaces both bounded walks with an **exact interval-reachability index** ($O(1)$ chain-ancestry, $O(\log n)$ DAG-ancestry, validated against a brute-force oracle) and seeds the blue set unbounded, so no silent-approximation path remains. **This is UNAUDITED, pre-1.0 reference software** — review it before you trust it. Already-mined blocks and balances are loaded verbatim and never recomputed.

2 Back up first

Stop the node cleanly, then copy your data directory and any wallet files.

```
# 1. Stop the node cleanly (systemd example, or send SIGINT to the process)
sudo systemctl stop bloch-node          # or: kill -INT <pid> and wait for exit

# 2. Back up the data-dir (default is ./bloch-data; whatever you pass to --data-dir)
cp -a ~/bloch/bloch-data ~/bloch-data.backup-$(date +%Y%m%d)

# 3. Back up wallet files separately (these are your keys — guard them)
cp -a ~/wallet-hd.json ~/founder.json ~/wallet-backups/ 2>/dev/null
```

The reachability index is a rebuildable cache, but back up the whole data-dir anyway. Never share wallet files, seed phrases, or passwords.

3 Get the fixed build

Pull the public branch `feat/reachability-durable` (or, once published, the tagged coordinated release — prefer that, since it carries the announced activation height baked in), then build.

```
cd ~/bloch
git fetch origin
git checkout feat/reachability-durable    # or: git checkout <tagged-release>
cargo build --release
```

Binaries land at:

- node: `target/release/bloch`
- CLI: `target/release/bloch-cli`

4 First start = one-time index rebuild (no resync)

The durable reachability index is only built when the activation gate is **armed** — i.e. on the first boot of the coordinated release (the build carrying the published **H**), or under the dev-only throwaway override. On that first boot the node **loads or rebuilds the index from your existing DB** in topological order — a few minutes, **no resync from peers**, balances untouched. A random-sample self-check against the brute-force oracle gates acceptance before the index colors any block.

```
# Start the node exactly as you normally do (your flags/data-dir), e.g.:  
./target/release/bloch --data-dir ~/bloch/bloch-data
```

Expected log lines on that first armed boot (any one of these families):

```
# fresh build of the index from your DB:  
reachability index absent/version-mismatch (have None, want N); rebuilding from CF_DAG (one-  
time)  
reachability index rebuilt from CF_DAG: <NNNNN> blocks  
reachability self-check OK (sampled pairs match brute-force oracle)  
  
# or, on a later boot, a fast load of the cached index:  
reachability index loaded from disk: <NNNNN> records  
reachability self-check OK (sampled pairs match brute-force oracle)
```

Honesty note

If you built the plain branch with the gate still OFF (the default), the node boots byte-for-byte unchanged and

no index is built

— that is expected. The rebuild only appears once you run the activation-armed build. This is the same fact as "the binary alone changes nothing on mainnet" (Step 6).

5 Confirm health via RPC

All commands talk to your local node (default 127.0.0.1:16210).

```
./target/release/bloch-cli info      # peers, blocks, blue_score, syncing  
./target/release/bloch-cli peers    # peer_count, syncing  
./target/release/bloch-cli dag      # tip_blue_score, tip_height, chain_length, k
```

You want to see, over a few minutes:

- `peers > 0` in `info / peers`;
- `blue_score` and `tip_height` climbing between calls;
- `syncing` flips to `false` once you catch up;
- after activation, the tip advances **past 395,897** instead of freezing.

Cross-check your `blue_score` / height against the project's public reference node so you know you're on the network's chain:

```
curl -s https://posternlabs.com/rpc -H 'Content-Type: application/json' \  
-d '{"jsonrpc":"2.0","id":1,"method":"getdaginfo","params":null}'
```

The public reference endpoint URL above is the project-published one (external to the node repo). Your local numbers should track it; a persistent gap means you're behind, pre-activation, or on a mismatched H (Step 8).

6 Coordinated activation — the load-bearing step

This is a coordinated soft-fork. Upgrading the binary alone fixes nothing on mainnet.

The gate ships

OFF

: `CORRECTED_COLORING_ACTIVATION_HEIGHT = u64::MAX` (`src/consensus/mod.rs`). Until a height

H

is set, your node stays byte-for-byte on the OLD (Legacy) classifier.

The activation height **H** is **chosen and published by the project**, in the release notes — you do **not** pick it yourself. Your job is to deploy the same published H in time:

- Below **H**: every node uses the OLD (Legacy) classifier.
- At / above **H**: every node uses the EXACT (Fast) classifier.
- All mining and validating nodes must be upgraded and carry the same H **before** the tip reaches it.

How to apply the published H:

- **Preferred:** run the tagged coordinated release — it already carries the announced H. Nothing to edit.
- **From source:** set `CORRECTED_COLORING_ACTIVATION_HEIGHT = <published H>` in `src/consensus/mod.rs`, then `cargo build --release` and restart. Confirm your value matches the release notes exactly.

Running a different H than the network forks you off.

A different (or missing) H means you color new blocks differently from everyone else. Deploy the exact published value, before the tip crosses it. H at or below the current tip is forbidden (it would recolor already-mined blocks).

7 Rollback

- **While the gate is OFF (no H set):** fully safe. Revert to your previous binary and restart — the reachability index is an inert, rebuildable cache and is never part of the integrity chain, so no balance is affected.
- **After activation (H reached):** rollback is a **coordinated downgrade** — all nodes must revert H together, subject to the same fork caveat as any soft-fork reversal. Do not unilaterally downgrade a live, activated node.

8 Troubleshooting

- **Index self-check failed** (`loaded reachability failed self-check ... rebuilding from CF_DAG`): the node rebuilds the index once automatically and re-checks. If it still fails, the process exits by design rather than color with a bad index — restart from your backed-up data-dir and re-report.
- **Still stuck at 395,897** after upgrading: you are almost certainly **pre-activation** (no H set / running the gate-OFF build) or on a **mismatched H**. Confirm you're on the tagged release or that your `CORRECTED_COLORING_ACTIVATION_HEIGHT` equals the published value, and that the network tip has actually crossed H. Compare against the public reference RPC (Step 5).
- **Rebuild is slow / disk pressure:** the one-time rebuild replays your existing DB in topological order (minutes, proportional to chain length, no peer download). Ensure free disk for the new `CF_REACHABILITY` column family before starting; the index is a modest per-block record, not a second copy of the chain.
- **Node won't connect / cannot connect ... is the node running?** : the CLI defaults to `127.0.0.1:16210`; pass `--rpc-host` / `--rpc-port` if you moved it.

passwords, or private host details. The activation height H is authoritative only as published in the project's official release notes. · Guide dated 2026-07-18.