

iOS app · Runbooks · pinned from gurbani-soul-ios

Runbook: iOS hotfix

🕒 2 min read · Updated 27 October 2026

<https://docs.gurbanisoul.com/ios/process/runbooks/ios-hotfix/> · a pinned copy of Algorythmos-AI/gurbani-soul-ios at a386422

Pinned copy. The canonical file is [gurbani-soul-ios/docs/process/runbooks/ios-hotfix.md](https://docs.gurbanisoul.com/ios/process/runbooks/ios-hotfix.md)¹ at commit a386422 — edit it there. This wiki publishes it by pin ([ADR-0012](#)); a newer version arrives as a reviewed bump of `docs-site/sources.lock.json`.

iOS has no rollback. The levers, mildest first:

1. **Pause the phased release** — stops new automatic updates; users who already updated keep the build. A pause has a time limit (check the current one in App Store Connect). Updates only: a first version has no phased release, so for it this lever does not exist — go to step 2.
2. **Ship a fixed build** (below), optionally asking for an **expedited review** through App Store Connect → Contact Us → App Review. Use it rarely and say plainly what is broken; a scripture-fidelity defect is a legitimate reason.
3. **Remove from sale** — last resort; existing installs keep working (the app is fully offline).

Procedure

1. Reproduce and write the failing test first. Scripture text is never edited to “fix” anything — see the prime directive in the platform’s [docs/engineering/invariants.md](#)².
2. Branch `hotfix/<slug>` **from this app’s release tag** (not from a later main, which may already hold unreleased work): `git worktree add <scratch>/wt-hotfix -b hotfix/<slug> vX.Y.Z`
3. One number, both surfaces: the platform releases `X.Y. (Z+1)` too (never a one-sided hotfix — see its release runbook). Once that tag exists, `make vendor-sync-platform REF=vX.Y. (Z+1)` and set `MARKETING_VERSION` to `X.Y. (Z+1)`. Build numbers restart at the ledger’s next for the new version.
4. PR `hotfix/<slug>` → `main` (squash). CI must be green: gates, parity, app.
5. Tag the app release `vX.Y. (Z+1)` on that `main` commit once its CI is green (the tag cannot be moved afterwards). Then, from the tag — a worktree, or the iOS TestFlight upload workflow run at the tag: `make testflight TEAM_ID=... BUILD=N CHANNEL=appstore UPLOAD=1` The script refuses a dirty tree, an off-trunk or untagged commit and red CI, exactly as for a normal release.
6. Commit the ledger row via PR. `make appstore-preflight` must pass. Re-sign Charter S on the new build number; re-run only the hardware checks the fix could affect, and say which.
7. Submit. After approval, release and resume (or restart) the phased rollout.

8. Write the incident note: what users saw, how it got past the gates, which gate now catches it. A hotfix without a new test or gate is not finished.

Links

1. <https://github.com/Algorythmos-AI/gurbani-soul-ios/blob/a386422e90c2c681f9ecc4f7167b4deb6a35fb92/docs/process/runbooks/ios-hotfix.md>
2. <https://github.com/Algorythmos-AI/siggs-platform/blob/main/docs/engineering/invariants.md>

Sri Guru Granth Sahib Ji — Knowledge Base wiki. Generated from commit c3d9360. Updated 27 October 2026. This copy may be out of date; the page is at <https://docs.gurbanisoul.com/ios/process/runbooks/ios-hotfix/>