

f00tils

Freestanding assembly GNU userland * <https://f00.sh>

STANDARD OPERATING PROCEDURE

Document: SOP-F00-OPS-001
Date: 2026-07-27
Version: v0.16.10
Standard: NASA NAI 1450.3 memorandum layout

Reply to Attn of: OPS/F00

TO: Maintainers and Release Operators
FROM: Project Engineering
SUBJECT: f00tils operator standard operating procedure (clone, build, gates, install, release)
REF: house docs pack + package-manifest sync; packaging bare names v0.16.10

Bottom line: Distro packages install bare names into /usr/bin and conflict with coreutils/findutils/grep/diffutils. After `paru -S f00`, `find/grep/diff/ls` are f00 in every session including TTY - no PATH setup. Every ship or ops/docs change keeps README + NASA SOP PDF + site Documents + NASA release memo (on tag) in step. `--core` is still never auto for scripts.

1. Purpose

This memorandum is the living operator handbook for f00tils (binary f00 / tools f00-*). It states how to clone the repository, build the freestanding multicall, run quality gates, install, and ship a release.

2. Scope

This SOP covers full project operations. It does not replace man pages, README narrative, or the scoreboard on <https://f00.sh>. Those remain the public ship surfaces.

- repository layout and product laws
- host prerequisites
- x86-64 build and boring-solid gates
- hot-path and speed law
- aarch64 freestanding port
- local and end-user install
- release, packaging, and site install sync

- user-facing documentation rules

3. Product identity

Item	Value
Product name	f00tils
Binary / multcall	f00
Tool names	f00-* (bare names when replace is on)
Site	https://f00.sh
Repository	github.com/theesfeld/f00
License	MIT only
Primary target	Linux x86-64 freestanding (NASM, no libc)
Port path	Linux aarch64 freestanding (GNU as, asm/port/aarch64)
Surface	115 tools (coreutils + grep + findutils + diffutils)

4. Product laws (non-negotiable)

Language purity: no Rust, C application code, libc, or polyglot product dependencies in first-party product code under asm/. Shell is allowed only for bootstrap, install, packaging, and benches.

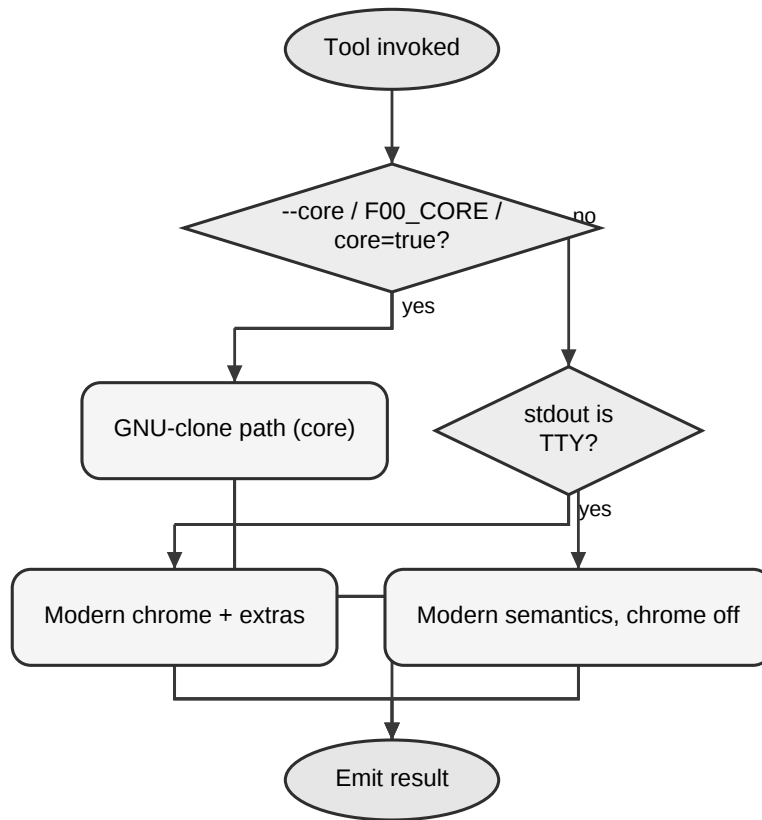
4.1 Clone first (--core). Every covered GNU tool has a f00-* counterpart. Under --core, match the GNU tool for scripts: flags, exit codes, and byte-identical stdout/stderr for the same inputs.

4.2 --core must win on resources. On the core path, f00 must beat the GNU tool on wall time and CPU (user+sys). Correct-but-slower is not done.

4.3 Modern is amazing (default). Non--core is never a pale GNU subset: themed chrome, better layout, icons where relevant, --json/--csv, and extra power.

4.4 One binary. Multicall by argv0 (f00-grep, grep, ...).

Figure. Script vs interactive path



5. Repository layout

Path	Role
asm/	Product source, Makefile, man pages, benches
asm/port/aarch64/	aarch64 freestanding multcall
site/	https://f00.sh (GitHub Pages) and site install.sh
install.sh	Root installer (must match site/ and scripts/)
packaging/	AUR, nfpdm (deb/rpm/arch)
Formula/	Homebrew formula
docs/	Compliance, UX, progress, this SOP
file_id.diz	Release scene card (GitHub Release asset only)
scripts/	Release, package, and bench generators

6. Prerequisites

6.1 x86-64 host (primary): Linux x86-64; nasm; ld (binutils); make; python3; GNU coreutils/grep/findutils/diffutils on PATH for parity and speed comparison.

6.2 aarch64 port (optional): aarch64-linux-gnu-as and aarch64-linux-gnu-ld; qemu-aarch64-static (or equivalent) for smoke on x86-64 hosts.

7. Procedure - clone and build (x86-64)

Do this on every new worktree before you edit product code. Root convenience targets: make (asm), make check, make aarch64, make install.

7.1 Clone or update the repository.

```
git clone git@github.com:theesfeld/f00.git
cd f00
git checkout main
git pull --ff-only
```

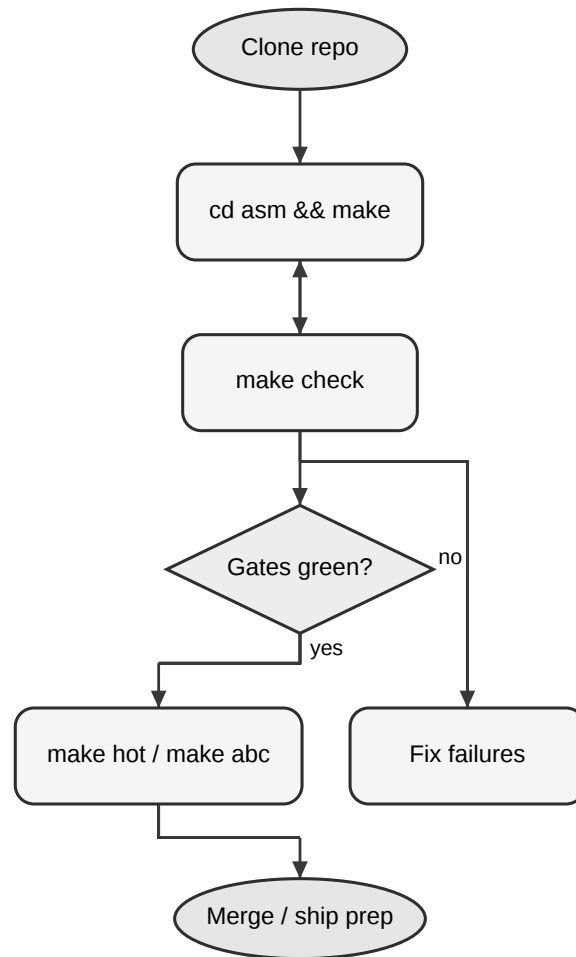
7.2 Enter the product tree and build the multicall.

```
cd asm
make
```

7.3 Confirm the binary runs.

```
./f00 --version
./f00-ls --version
```

Figure. Build and gate flow



8. Procedure - quality gates

8.1 Boring-solid (required before merge).

```
cd asm  
make check
```

8.2 Hot path (real-work wall + CPU). Use when you change sort, ls, or other hot tools.

```
cd asm  
make hot
```

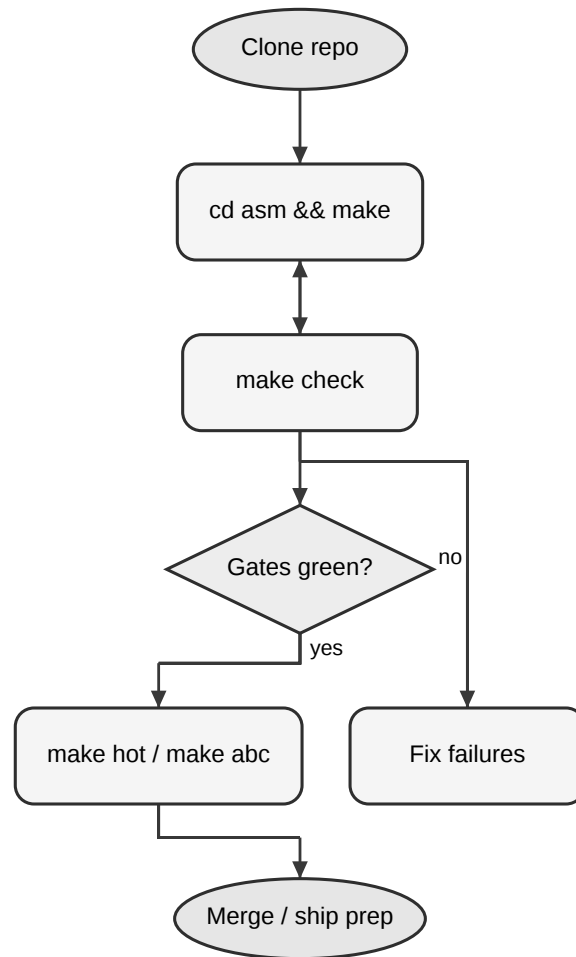
8.3 Speed law (wall + CPU). Optional each commit; required for law-2 claims. Includes hot.

```
cd asm  
make speed
```

8.4 Structural speed proof (stress).

```
cd asm  
make speed-stress
```

Figure. Build and gate flow



Target	Meaning
make / make all	Build f00 + tool links
make smoke	Minimal runtime smoke
make parity	GNU --core parity battery
make check	smoke + parity (default bar)
make hot	Real-work wall+CPU + stdout assert
make speed	wall+CPU law (includes hot)
make speed-stress	Repeated full-speed-gate proof
make aarch64	Build aarch64 multcall
make aarch64-smoke	qemu smoke of aarch64 binary

9. Procedure - aarch64 port

The aarch64 path is shipped and grows toward full surface. It is not the default CI/release install path today.

9.1 Install the cross assembler and linker, and qemu-user if you build on x86-64.

9.2 Build and smoke.

```
cd asm
make aarch64
make aarch64-smoke
```

9.3 On failure, fix the port under asm/port/aarch64. Do not weaken x86-64 gates to paper over port gaps.

10. Procedure - local and end-user install

Default installer places multcall f00, all f00-* names, and bare tool names in INSTALL_DIR so replace wins on PATH.

10.1 Install from a local build.

```
cd asm
make install
# or
F00_LOCAL=/path/to/f00/asm bash /path/to/f00/install.sh
```

10.2 End-user install (primary ship story).

```
curl -fsSL https://f00.sh/install.sh | bash
```

10.3 Knobs: F00_VERSION=vX.Y.Z (pin tag); F00_SUPERSEDE=0 (f00-* only);
INSTALL_DIR=~/.local/bin; F00_LOCAL=...; F00_MAN=1 (man pages, default on).

10.4 Config.

```
f00
f00-config
f00-config replace off
```

11. Procedure - keep installers identical

Root install.sh is the source of truth. Site and scripts copies must match byte-for-byte. Run after any edit to install.sh and before release.

11.1 Sync from repository root.

```
make sync-install
```

11A. Procedure - regenerate screenshots (every release)

Public screenshots must match the shipped binary and explain features (multi-command sessions), not decoration one-liners. Capture uses a real PTY and forced color environment.

11A.1 Build: `cd asm && make` (binary version string must match the release).

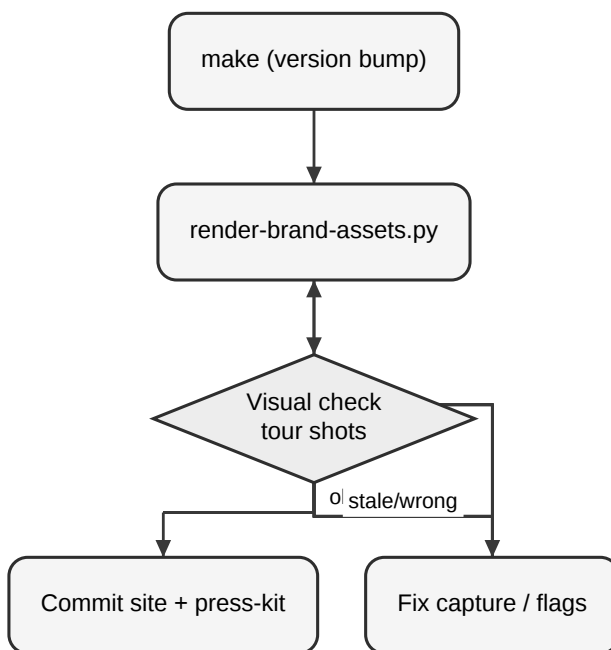
11A.2 Regenerate: `python3 scripts/render-brand-assets.py`

11A.3 Confirm outputs under `site/assets/screenshots/`, `press-kit/screenshots/`, `docs/images/`.

11A.4 Confirm site Feature tour (`#tour`) captions still match the shot set.

11A.5 Commit screenshots with the release; never leave stale version strings in `hero.png`.

Figure. Screenshot regen on ship



12. Procedure - release and ship

Primary release story: GitHub Release tarball + `install.sh`. `deb/rpm/AUR/brew` are secondary. Release assets must include `file_id.diz`, `sop-f00tils-ops.pdf`, and `memo-release-VERSION.pdf` (enforced by `.github/workflows/release.yml`).

12.1 Ensure main is green: build, make check, and speed gates as required for the change set.

12.2 Update version metadata so the binary, man pages, README pin examples, and CHANGELOG agree on `vMAJOR.MINOR.PATCH`.

12.3 Move Unreleased notes in `CHANGELOG.md` into a dated version section.

12.4 Refresh root `file_id.diz` for this version. Commit it. Show a fixed-width Scene card preview in README. Attach the same file as a GitHub Release asset. Do not spotlight the card on the public website (project rule).

12.5 Sync installers (`make sync-install`).

12.6 Confirm man page(s), README.md, and site Documents links match behavior and point at the current SOP PDF and release memo PDF.

12.7 Render NASA SOP and NASA release memo PDFs via memo-pdf (nasa-sop / nasa-memo). Commit JSON sources and PDFs under docs/. Link both from README and site.

12.8 Create an annotated tag and push it.

```
git tag -a vX.Y.Z -m "vX.Y.Z"
git push origin vX.Y.Z
```

12.9 Release workflow (tag v*) builds Linux x86-64 ASM, runs smoke/parity/speed-gate, verifies binary version, packages tarball, publishes GitHub Release with SOP PDF, memo PDF, and file_id.diz.

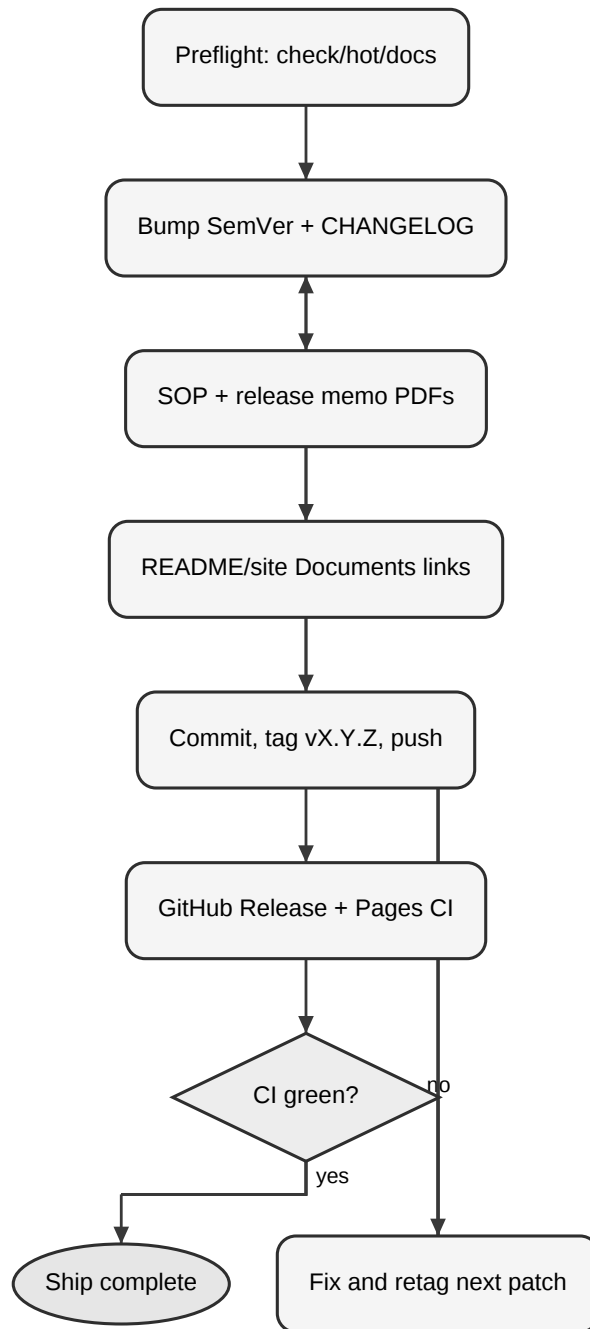
12.10 After SHA256SUMS exist for the release, sync package manifests so Formula and AUR match the tag.

```
scripts/sync-package-manifests.sh X.Y.Z path/to/SHA256SUMS
git add Formula packaging/aur && git commit -m "chore(release): sync package manifests to vX.Y.Z"
```

12.11 Smoke-test the public install path. Do not move or reuse a published tag for different bits.

```
curl -fsSL https://f00.sh/install.sh | F00_VERSION=vX.Y.Z bash
f00 --version
```

Figure. Release and deploy flow



13. Procedure - secondary packaging

Use packaging scripts under scripts/ and configs under packaging/ and Formula/ when you intentionally ship secondary channels. They must not replace tarball + install.sh as the default install story.

- scripts/build-linux-packages.sh
- scripts/gen-aur-pkgbuild.sh / scripts/publish-aur.sh
- scripts/gen-homebrew-formula.sh / scripts/publish-homebrew-tap.sh
- scripts/sync-package-manifests.sh

14. Procedure - GitHub Actions constraints

- Use only Node 24 JS action majors (checkout@v6+, upload-artifact@v6+, download-artifact@v7+, softprops/action-gh-release@v3+, Pages @v5/@v6 as house rules specify).
- Do not reintroduce Node 20-era pins (checkout@v4, upload-artifact@v4, action-gh-release@v2). Deprecation annotations are a defect.

15. Procedure - user-facing text

- Name the project f00tils in narrative. Keep command names f00 / f00-.*.
- Procedures, man pages, and install how-to: ASD-STE100 (Simplified Technical English).
- Public story (README lead, website): NASA Stylebook + AP habits.
- Ship narrative surfaces: README + <https://f00.sh>. Docs under docs/ are optional depth.

16. Daily operator checklist

- Branch from main; keep PRs small and Conventional Commits-friendly.
- Edit only assembly under asm/ for product logic.
- cd asm && make && make check before you open or update a PR.
- Run make hot / make speed when the change touches core-path performance or hot tools.
- Update man pages and site/README when user-visible behavior changes.
- Never claim --core parity without parity evidence.
- Never claim a speed win without gate evidence.

17. Failure responses

Symptom	Action
make fails	Fix asm build. Confirm nasm/ld. Clean with make clean.
make check fails	Read smoke/parity logs. Restore --core GNU match.
make hot/speed fails	Profile the tool. Do not ship a slower core path.
install.sh mismatch	Run make sync-install. Re-diff site and scripts.
Release CI red	Match tag to binary version. Re-run gates locally.
aarch64 smoke fail	Fix port. Keep x86-64 gates independent.
Wrong action majors	Upgrade to Node 24-era pins per house rules.

18. References

- AGENTS.md - product laws and layout
- README.md - public install and modes
- CONTRIBUTING.md - contributor build bar
- asm/Makefile - build and gate targets
- install.sh - primary installer
- .github/workflows/release.yml - tag release pipeline
- <https://f00.sh> - site, scoreboard, install entry
- site/bench/suite.json - suite bench data
- ~/.grok/skills/memo-pdf - house memo PDF renderer used for this document

19. Approval and control

This SOP is controlled under docs/. Update it when gates, install paths, or release mechanics change. Prefer short procedural edits over narrative expansion. Regenerate the PDF with the memo-pdf skill (nasa-sop type); do not freestyle layout.

Project Engineering

f00tils / OPS

Document: SOP-F00-OPS-001

Date: 2026-07-27

Enclosure: None. Companion sources are in-repository paths listed in section 18.

cc: Maintainers, Release Operators