

# Oload Administrator Manual

Installer setup, first-run admin bootstrap, updates, uninstall, and GitHub self-service reference.

Last updated: 2026-05-13

## Oload Administrator Manual

Installer setup, first-run admin bootstrap, updates, uninstall, and GitHub self-service reference.

Last updated: 2026-05-13

This manual covers what the Windows and Linux installers ask for, what they write under the install root, how first-run admin setup works, what uninstall removes, and where operators can fetch packages and scripts directly from GitHub.

### Quick reference

#### Installer prompts are operator-facing settings

The packaged installers do not just drop files. They ask for install location, port, network bind mode, Ollama base URL, live-update settings, default language, optional bootstrap admin password, and session secret behavior.

#### Install footprint is recorded on disk

Both installers write `INSTALL-MANIFEST.txt`, `UNINSTALL-NOTES.txt`, `.oload-install-state`, `.oload-install-binding`, and `.env.runtime` so admins can see exactly what was installed and what uninstall may remove.

#### Uninstall is provenance-aware

The uninstall flows read the saved install state before touching Node.js, Ollama, or model directories. Shared dependencies are not removed silently, and Ollama model removal always requires explicit confirmation.

#### GitHub is the self-service source of truth

Operators can fetch release artifacts, update bundles, packaging docs, and installer scripts directly from the snowolfex/olaod repository and its GitHub Releases page without depending on the mirrored site copy alone.

### Windows installer

- Package: `OloadSetup.exe`
- Local mirror: `/downloads/OloadSetup.exe`
- GitHub source: `https://github.com/snowolfex/olaod/releases`
- Default install root: `%LOCALAPPDATA%\Oload`
- Launch later with: `start-oload.cmd` or `start-oload.ps1`

- Uninstall later with: Native Oload uninstaller, `uninstall-oload.cmd`, or `uninstall-oload.ps1`
- The native wrapper is built with Inno Setup 6 and collects installer choices in a GUI before calling `install-oload.ps1`.
- The package copies the app payload, broker payload, start launchers, uninstall launchers, `README.md`, `EULA.txt`, and `SOURCE-AVAILABLE-NOTICE.txt` into the install root.
- Machine identity, install identity, manifest files, and runtime environment values are written during install so later repair, update, and uninstall flows know what this machine owns.

## Linux installer

- Package: `OloadInstaller-linux-x64.run`
- Local mirror: `/downloads/OloadInstaller-linux-x64.run`
- GitHub source: <https://github.com/snowolfex/olaod/releases>
- Default install root: `~/.local/share/oload`
- Launch later with: `start-oload.sh`
- Uninstall later with: `uninstall-oload.sh`
- The Linux `.run` installer shows the branded install splash, requires EULA acceptance, and then prompts in-terminal for install settings.
- The package writes the app payload, broker payload, runtime files, `.env.runtime`, `INSTALL-MANIFEST.txt`, `UNINSTALL-NOTES.txt`, and optionally a desktop launcher under `~/.local/share/applications/oload.desktop`.
- The installer can use an existing shared Node.js or Ollama path, or install Oload-managed runtimes under the install root when isolation is preferred.

## Installer prompts and defaults

### Install location and service port

Windows defaults to `%LOCALAPPDATA%\Oload` and Linux defaults to `~/.local/share/oload`. Both installers also ask what port Oload should serve on; the shipped default is 3000.

### Host binding and Ollama target

Linux asks whether to expose Oload on the local network and then prompts for the Ollama base URL. Windows asks for the same effective values through the GUI wrapper that feeds `install-oload.ps1`.

### Node.js, Ollama, and language selection

Both installers verify existing Node.js and Ollama locations, offer managed-versus-shared runtime choices, and persist the selected default language into `.env.runtime` so first-run and signed-out flows start with that language.

### Live updates, admin password, and session secret

The installers ask for optional update manifest URL, update channel, update public key, optional bootstrap admin password, and an optional session secret. If the secret is left blank, Oload generates one automatically.

## **What the installers write**

### **Application and broker payload**

Installed builds place the standalone Next app in `app/` and the control broker in `broker/` so admin actions can later start, stop, and restart the server without relying on the original installer bundle.

### **Runtime and environment files**

Each install writes `runtime/`, `.env.runtime`, and `.oload-install-state` so Oload knows which Node.js and Ollama paths were detected, chosen, or installed under the current machine-bound install.

### **Ownership and uninstall files**

`INSTALL-MANIFEST.txt` records the owned file footprint. `UNINSTALL-NOTES.txt` explains what uninstall will ask before removing runtimes, shared dependencies, or Ollama model stores. `.oload-install-binding` records the machine-bound install identity.

### **Launchers and legal copies**

Windows writes `start-oload.cmd`, `start-oload.ps1`, `uninstall-oload.cmd`, `uninstall-oload.ps1`, `README.md`, `EULA.txt`, and `SOURCE-AVAILABLE-NOTICE.txt`. Linux writes `start-oload.sh`, `uninstall-oload.sh`, `README.md`, `EULA.txt`, `SOURCE-AVAILABLE-NOTICE.txt`, and can add a desktop launcher plus icon.

## **First-run admin setup**

### **Finish installation and launch the packaged app**

Windows can start Oload automatically after install. Linux offers the same choice and prints the later launcher path so the app can be started again without rerunning the installer.

### **Create the first user account**

Fresh installed builds keep the same local-first auth model as development builds. The first user created in a clean install becomes the admin account.

### **Verify runtime ownership before changing anything**

Check `INSTALL-MANIFEST.txt`, `UNINSTALL-NOTES.txt`, and `.oload-install-state` before moving files, reusing a shared runtime, or changing uninstall expectations. Those files tell you what the current install owns.

### **Confirm update settings and support paths**

If live updates are enabled, verify `OLOAD_UPDATE_MANIFEST_URL`, `OLOAD_UPDATE_CHANNEL`, and `OLOAD_UPDATE_MANIFEST_PUBLIC_KEY` in `.env.runtime`. Operators can also fetch packages manually from GitHub Releases if needed.

## Uninstall and cleanup

### Use the shipped uninstall entry point

Windows installs can be removed with the native Oload uninstaller or the shipped `uninstall-oload` scripts. Linux installs use `uninstall-oload.sh` from the install root.

### Shared dependency removal is never assumed

If Node.js or Ollama existed before Oload, uninstall warns and asks again before removing them. If Oload installed an isolated runtime under the install root, that managed runtime can be removed with less ambiguity.

### Ollama models always get extra confirmation

Because model directories can be large and expensive to recreate, both uninstall flows require an additional confirmation before removing Ollama model stores.

### Install identity files are part of cleanup

The uninstall flow removes `INSTALL-MANIFEST.txt`, `UNINSTALL-NOTES.txt`, `.oload-install-state`, `.oload-install-binding`, `.env.runtime`, log files, and the recorded install root after cleanup decisions are complete.

## GitHub self-service links

- [GitHub repository](<https://github.com/snowolfex/olaod>) - Main source repository for oload, including README, release history, and packaging scripts.
- [GitHub releases](<https://github.com/snowolfex/olaod/releases>) - Direct self-service source for tagged release artifacts, update bundles, and release notes.
- [Installer docs in repo](<https://github.com/snowolfex/olaod/blob/main/installer/README.md>)
- Repository-side install, update, and packaging reference mirrored from the shipped installer behavior.
- [Windows installer script](<https://github.com/snowolfex/olaod/blob/main/installer/windows/install-oload.ps1>) - Interactive Windows installer implementation and prompt/default reference.
- [Linux installer script](<https://github.com/snowolfex/olaod/blob/main/installer/linux/install-oload.sh>) - Interactive Linux installer implementation and prompt/default reference.
- [Windows uninstall script](<https://github.com/snowolfex/olaod/blob/main/installer/windows/uninstall-oload.ps1>)
- Scripted uninstall path for Windows installs, including dependency cleanup prompts.
- [Linux uninstall script](<https://github.com/snowolfex/olaod/blob/main/installer/linux/uninstall-oload.sh>) -

Scripted uninstall path for Linux installs, including managed runtime cleanup prompts.

## **Repo mirror requirements**

### **Installer bundles and native outputs**

The repository should tell operators where the packaging outputs live:

dist/installers/windows/install-oload.ps1, dist/installers/linux/install-oload.sh,  
dist/native/OloadSetup.exe, and dist/native/OloadInstaller-linux-x64.run.

### **Install manifest and uninstall notes**

The repository docs should explicitly say that packaged installs now write

INSTALL-MANIFEST.txt and UNINSTALL-NOTES.txt so operators can see what Oload owns and what  
uninstall may remove.

### **Self-service GitHub path**

The repository docs should point people at GitHub Releases for packages and at  
installer/README.md for the detailed install/update/uninstall behavior.

### **No art direction note**

This manual intentionally ships without llamas, mascot art, or decorative SVG illustrations.

The document is text-first so install and uninstall details stay readable and accurate.