

Installation Guide for Managed Windows Environments — LocalTranscript

LocalTranscript · Documentation for IT and data protection

Status: version 0.6.65, September 2026

This document is aimed at IT security officers and administrators who roll out LocalTranscript to managed Windows workstations (AppLocker/WDAC, Defender/EDR, corporate proxy, Intune/GPO). For fine-grained control of app behaviour after installation (enforcing Hard Offline Mode, locking cloud providers, disabling features), see our document "Administrative policy".

1. Package contents and signing

Stable Windows releases are **Authenticode-signed**: the app executable, every bundled .exe/.dll (llama-server.exe, ffmpeg.exe, the ONNX Runtime and DirectML libraries, the meeting-mcp helper process), and the MSI itself — with an OV code-signing certificate issued to **Jenzer Innovations** (SSL.com), timestamped.

Signing is a per-build switch: **beta-channel builds and internal test builds can be shipped unsigned**. A managed environment should therefore consistently use the **stable channel**. For an AppLocker publisher rule you need the certificate subject and the thumbprint — you can request these at hello@localtranscript.com or read them yourself from a downloaded MSI file:

```
Get-AuthenticodeSignature "Local.Transcript_<Version>_directml_x64_en-US.msi" |
Format-List *
```

The MSI is Python-free (package size approx. 224 MB): the entire transcription, diarisation, and embedding stack runs natively in the same process — ONNX Runtime + DirectML, whisper.cpp, and the MiniLM text model. No python.exe and no OpenVINO library are bundled; EDR or AppLocker rules that target a "bundled Python subprocess" therefore find nothing to match. Windows receives exclusively the directml variant (the earlier openvino variant has been discontinued; existing installations migrate automatically).

Child processes: the only executable programs launched by the app are llama-server.exe (local AI inference for Gemma) and ffmpeg.exe (audio processing). The meeting-mcp helper process is only launched if an MCP-capable program explicitly addresses it via standard input/output, and only if local MCP integration has been enabled in settings.

2. Paths table

Path	Purpose	Permissions
C:\Program Files\Local Transcript\	App binary, ONNX Runtime/DirectML libraries, whisper.cpp (in-process), llama-server.exe, ffmpeg.exe	Write access for administrators only (set by the MSI)

%LOCALAPPDATA%\com.localtranscript\	Models (Whisper, Pyannote, Gemma), recordings, SQLite library database	Write access for the user account
%LOCALAPPDATA%\com.localtranscript\runtime\	Runtime binaries possibly relocated there by the first-run wizard — see the AppLocker note in section 4	Write access for the user account
%LOCALAPPDATA%\com.localtranscript\tmp\	Cached recording/mixed audio during an ongoing meeting; cleared on every app start (since version 0.6.65; previously %TEMP%)	Write access for the user account
%LOCALAPPDATA%\com.localtranscript\network-audit.jsonl	Network audit log (host, purpose, allowed/blocked)	Write access for the user account
%LOCALAPPDATA%\com.localtranscript\app-settings.json	App settings (API keys and the calendar URL are DPAPI-protected)	Write access for the user account
%LOCALAPPDATA%\com.localtranscript\logs\	Rotating log files (local-transcript.log); since 0.6.65 without transcript or contact content, 3-day retention	Write access for the user account

The MSI installs **per machine** (`installMode: perMachine`), not per user account. The installation itself requires administrator rights; ongoing operation afterwards does not.

3. Silent installation and deployment

Direct MSI installation. For an unattended deployment, use the usual silent MSI procedure:

```
msiexec /i "Local.Transcript_<Version>_directml_x64_en-US.msi" /qn /norestart
```

Note: this is the standard silent syntax for WiX-based MSI packages; we test it together against your software deployment during the pilot phase.

Intune (Win32 app): the MSI can be deployed as a classic Win32 app (`.intunewin`) with the installation line above. For the **administrative policy** (`policy.json`) itself, a separate, payload-less Win32 app with a PowerShell script that runs as SYSTEM and writes the file is recommended:

```
$dir = "$env:ProgramData\com.localtranscript"
New-Item -ItemType Directory -Force -Path $dir | Out-Null
Set-Content -Path "$dir\policy.json" -Value $PolicyJson -Encoding utf8
# %ProgramData% ACLs already deny write access to non-administrators.
```

GPO: for distributing the administrative policy, a Group Policy Preferences file item that copies

policy.json to %ProgramData%\com.localtranscript\policy.json is suitable — applied to the **machine**, not the user, so that it lands in the machine-wide %ProgramData%, to which ordinary user accounts have no write access.

Pre-staging models / offline model package. For environments without internet access to the public model hosts, there are three routes: (1) the administrative policy allows additional internal hosts (`allowedHosts`), so that an internal model mirror remains reachable even in Hard Offline Mode; (2) the app automatically switches to a GitHub fallback mirror on a network error against the primary Hugging Face host; (3) an **offline model package**: the model files are placed into the model folder under %LOCALAPPDATA%\com.localtranscript\ before the first start (via reference image or script), and the app uses the existing files without downloading them again. We provide the package with the exact target folders together with the setup package, and verify it together with you during the pilot phase.

Skipping the first-run wizard. The administrative policy recognises the key `skipWizard: true`: this marks the first-run wizard as already completed, and end users land directly in the application.

4. AppLocker / WDAC

Recommended: a publisher rule, once signed stable builds are in use. A publisher rule checks the certificate subject and applies independent of the file path — it therefore covers both C:\Program Files\Local Transcript\ and a copy of the same signed file relocated by the first-run wizard to %LOCALAPPDATA%\com.localtranscript\runtime\. As long as all relevant programs (app, llama-server.exe, ffmpeg.exe) are shipped signed, **no separate path rule for %LOCALAPPDATA%** is therefore needed.

For unsigned beta/test builds this does not apply — here only the classic combination of path rules remains:

- a path rule allowing %PROGRAMFILES%\Local Transcript**, and
- a second path rule for %LOCALAPPDATA%\com.localtranscript\runtime**, in case the first-run wizard has relocated a runtime binary there.

The reason: standard AppLocker rule sets trust %PROGRAMFILES% but not the user-writable %LOCALAPPDATA%. If the second rule is missing for an unsigned build, this shows up as "Could not start helper subprocess" with Windows error 1260 in the log.

WDAC: for Windows Defender Application Control we do not have separately documented, product-specific guidance; the publisher principle described above (allowlisting via the code-signing certificate instead of paths) is the generally recommended approach for WDAC as well, and should work functionally equivalently — please verify with hello@localtranscript.com before a WDAC rollout.

5. Notes on antivirus/EDR products

The following symptoms and remedies are documented for unsigned builds; they occur less often, or not at all, with signed stable builds (see the end of this section).

Microsoft Defender (default configuration). *Symptom:* installation completes, first

summary/transcription fails or hangs. *Cause:* Defender briefly quarantines newly unpacked, unsigned binaries while a reputation check is running. *Remedy:* wait 5–10 minutes after installation before opening the app for the first time — or add `C:\Program Files\Local Transcript\` under "Windows Security → Virus & threat protection → Exclusions."

CrowdStrike Falcon. *Symptom:* app starts, summarisation (Gemma) hangs, or transcription/audio mixing fails silently. *Cause:* Falcon's behavioural monitoring flags the unsigned child processes `llama-server.exe` and `ffmpeg.exe` (launched by the unsigned parent process) and blocks or terminates them. *Remedy:* IT adds `localtranscript.exe` **and its child processes** (`llama-server.exe`, `ffmpeg.exe`) to the Falcon exception list (machine-learning prevention exception).

SentinelOne. *Symptom:* installation aborts partway through ("There is a problem with this Windows Installer package"). *Cause:* SentinelOne flags the unsigned, WiX-generated MSI and blocks the installation. *Remedy:* IT-side allowlisting of the SHA-256 hash of the MSI file — this is published with every release in the `SHA256SUMS` file (signed) alongside the installation package.

Effect of signing. On signed stable builds, Defender's content inspection is replaced by a fast signature check, AppLocker only needs the publisher rule from section 4, CrowdStrike/SentinelOne trust the signed program via vendor reputation, and the SentinelOne MSI rollback problem is eliminated entirely.

6. Proxy and firewall

If updates or model downloads are wanted, allow the following hosts, to bypass SSL inspection (Zscaler, Bluecoat, etc.) that occasionally aborts large downloads (> 100 MB):

- `huggingface.co` (primary model source)
- `github.com/localtranscript/lt_models/releases/*` (fallback mirror for models, used automatically on a network error)
- `release-assets.githubusercontent.com` and `objects.githubusercontent.com` (delivery of GitHub release files)
- `github.com` (the update check deliberately does **not** run via `localtranscript.com`, because newly registered domains are often blocked as "uncategorised" by corporate DNS filters for 30–90 days, but directly against the public GitHub release)

If operation should be fully offline, no allowlisting is needed — instead enforce Hard Offline Mode centrally via the administrative policy (`offlineMode: true`) and, if needed, allow an internal model mirror via `allowedHosts`.

Mobile companion (if enabled): the first bind attempt on all network interfaces triggers the Windows firewall dialog; on a restrictively managed machine, this dialog can appear under a different user context and be silently denied. An organisation-wide GPO firewall policy can permanently block the inbound rule without any indication appearing — the phone then never finds the desktop. If the feature is to be used, `localtranscript.exe` needs an inbound firewall allowance for the private/domain profile.

7. Terminal server / multi-user operation

All user-related data (models, recordings, library database, settings, network audit log) reside under %LOCALAPPDATA% or %APPDATA% and are therefore separated **per user account** — on a terminal server with multiple simultaneously logged-in user accounts, each account has its own, independent installation environment within the shared program installation.

The two local loopback services (MCP HTTP endpoint and llama-server inference, see our network disclosure, section 5) have been **token-protected** since version 0.6.65; before this version, on a shared machine, another logged-in user account could reach the same port, because the loopback binding alone does not protect against other local accounts.

We recommend version 0.6.65 or newer for terminal server or other multi-user scenarios, since only this version enforces the token/key requirement for both loopback services.

8. Update operation

Automatic update (default): the built-in updater checks against GitHub Releases, verifies the Ed25519 signature of every installation package before installing it, and refuses anything signed with a different key.

Centrally managed distribution (recommended for institutional environments): the automatic updater can be fully disabled via the administrative policy (`autoUpdate: false`) — the channel is thereby fixed to stable, and every check as well as every download is refused. Institutional deployments then patch on their own schedule via the new stable MSI, distributed through the tooling already in place (Intune, GPO, SCCM, etc.).

Portable fallback: if the MSI cannot be installed on a machine at all (e.g. no administrator access available), an installation image can be extracted via admin extraction into %LOCALAPPDATA% and run from there. In this mode, the automatic update does **not** work (the updater requires the package registered via `msiexec`); updates then happen by re-running the admin extraction.

9. Rollout checklist

- [] Use a stable-channel build (signed); beta builds for testing only
- [] Obtain the certificate subject and thumbprint (`hello@localtranscript.com` or `Get-AuthenticodeSignature` on the downloaded MSI)
- [] Set up the AppLocker publisher rule (or the two path rules, if an unsigned build is needed for now)
- [] Set EDR/AV exceptions depending on the product in use (Defender folder exclusion, CrowdStrike allowlist for `localtranscript.exe + children`, SentinelOne SHA-256 allowlist)
- [] Set proxy/firewall allowances if updates/model downloads are wanted (`huggingface.co`, `github.com/localtranscript/*`, `release-assets.githubusercontent.com`, `objects.githubusercontent.com`) — otherwise allow nothing and set `offlineMode: true` via policy
- [] Place `policy.json` under `%ProgramData%\com.localtranscript\` (root/SYSTEM context; the default ACL of %ProgramData% is sufficient)
- [] Test silent installation (`msiexec /qn`, Intune Win32 app, or GPO)
- [] For multi-user operation/terminal server: ensure version 0.6.65 or newer

- [] Clarify open questions with hello@localtranscript.com
-

Notes

- The `msiexec` silent syntax (`/qn /norestart`) is the standard form for WiX MSI packages; we confirm it with you together with the package and test it during the pilot phase against your deployment.
 - No product-specific guidance exists for WDAC; allowlisting via the code-signing certificate (publisher principle) is the recommended approach there as well. We are happy to verify a WDAC rollout together with you.
-

Jenzer Innovations · Markgräflerstrasse · 4057 Basel · Switzerland UID CHE-334.396.708 · hello@localtranscript.com