

IT Deployment Guide

DOCUMENT	VERSION	DATE	LANGUAGE	PAGES
IT Deployment Guide	1.0	2026-10-20	English (United States, Singapore)	4

For hospital and clinic IT teams deploying MIA Toolkit. MIA Toolkit is a free, open-source (MIT), **offline** desktop application that consolidates a patient's imaging media (CDs, USB folders, ZIP downloads) into one DICOMDIR archive (the standard DICOM media file-set) on a USB drive. Its only network call is the user-initiated update check described below; it makes **no network calls** otherwise.

† Derived from the installer configuration and the Inno Setup documentation; not yet tested by us on a Windows machine. Validate these items in a pilot, or contact support@miatools.tech before a per-machine (all-users) deployment.

System requirements

	Minimum
macOS	macOS 12 or newer (LSMinimumSystemVersion 12.0)
Windows	Windows 10 or 11, 64-bit (x64; also installs on Windows 11 on ARM, running under x64 emulation)
Disk	Free space sufficient for the imaging media being consolidated
Network	None required (download only)

Install model

Windows (Inno Setup installer MIA-Toolkit-Setup-<ver>.exe)

The installer is configured with `PrivilegesRequired=lowest` and `PrivilegesRequiredOverridesAllowed=dialog`. That gives two install modes:

Mode	How it is chosen	Default folder	Start-menu shortcut	Uninstall registration
Per-user (default)	default; or <code>/CURRENTUSER</code>	%LOCALAPPDATA%\Programs\MIA Toolkit	current user's Start menu	HKCU
All users (machine-wide)	"Install for all users" in the install-mode dialog (needs admin / UAC); or <code>/ALLUSERS</code>	see caveat below	all-users Start menu	HKLM

- Per-user mode needs **no administrator rights** and makes no machine-wide changes.
- Per the Inno Setup documentation, allowing `dialog` automatically allows the `/ALLUSERS` and `/CURRENTUSER` command-line switches, and using either one skips the dialog.†
- **Caveat for machine-wide installs:** the default folder is defined with `{localappdata}`, which is a per-user location even in all-users mode. An `/ALLUSERS` install therefore defaults into the *installing admin's* %LOCALAPPDATA%\Programs unless you pass `/DIR`. For a machine-wide install, always pass `/DIR="C:\Program Files\MIA Toolkit"`.†
- Upgrades install in place: the installer has a fixed AppId, so a newer version replaces the old one in the same folder and mode.
- There is no MSI. For Intune, wrap the `.exe` as a Win32 app (`.intunewin`) and use the command lines below; choose the **User** install context for per-user mode or **System** for `/ALLUSERS`.†

macOS (MIA-Toolkit-<ver>.dmg)

The app ships as a signed and notarized `.app` inside a `.dmg`. Copy `MIA Toolkit.app` into `/Applications` (machine-wide) or `~/Applications` (per-user). There is no installer, licensing agent, server enrolment or activation step.

Silent / unattended install**Windows**

```
# Per-user, fully silent (run in the user's context):
MIA-Toolkit-Setup-<ver>.exe /VERYSILENT /SUPPRESSMSGBOXES /NORESTART /CURRENTUSER

# Machine-wide, fully silent (run elevated / as SYSTEM):
MIA-Toolkit-Setup-<ver>.exe /VERYSILENT /SUPPRESSMSGBOXES /NORESTART /ALLUSERS /DIR="C:\Program Files\MIA Toolkit"

# Useful extras:
# /SILENT progress window only, instead of /VERYSILENT
# /LOG="C:\path.log" write an install log
# /TASKS="desktopicon" also create a desktop shortcut (off by default)
```

- A silent install does **not** launch the app afterwards (the post-install launch is skipped in silent mode).
- `/CURRENTUSER + /VERYSILENT + /DIR=` is exactly what our release pipeline runs on a Windows build machine before each release is published (see "Post-install check"). The `/ALLUSERS` line is not yet tested by us on Windows[†].

macOS

```
# Mount, copy to /Applications, detach:
hdiutil attach MIA-Toolkit-<ver>.dmg -nobrowse -readonly -mountpoint /tmp/mia-dmg
ditto "/tmp/mia-dmg/MIA Toolkit.app" "/Applications/MIA Toolkit.app"
hdiutil detach /tmp/mia-dmg
```

MDM (Jamf, Intune, Kandji, Mosyle):

- Deploy the `.dmg` through your MDM's DMG/app workflow, or build a component package from the notarized app and push that: `pkgbuild --component "/tmp/mia-dmg/MIA Toolkit.app" --install-location /Applications MIA-Toolkit-<ver>.pkg`
- That `.pkg` is **not signed** by us (we have no Developer ID *Installer* certificate). Some MDM workflows require a signed package (Intune's macOS LOB app type, for example). Sign it with your organisation's installer certificate, or use your MDM's DMG / unmanaged-PKG route. (*verify with your MDM*)
- The app bundle itself stays signed by our Developer ID and notarized by Apple whichever way you wrap it.
- **Air-gapped Macs:** the notarization ticket is stapled to the `.dmg`, not to the `.app` inside it (checked on v0.1.13 with `xcrun stapler validate`). On a Mac with internet access Gatekeeper confirms notarization online. Apps placed by an MDM normally carry no quarantine flag, so Gatekeeper does not assess them on first launch. If a fully offline Mac shows a Gatekeeper prompt for an app copied by hand from the DMG, deploy through the MDM or contact us.

Post-install check (headless)

Releases after v0.1.13 have headless flags (v0.1.13's `--selftest` performs an import check only) that open no window and make no network call:

- `--version` prints the installed version.
- `--selftest` imports the whole bundle, builds a DICOMDIR from a tiny synthetic DICOM study in a temporary folder, deletes it, prints one result line and exits 0 (pass) or 1 (fail).

```
# macOS
"/Applications/MIA Toolkit.app/Contents/MacOS/mia" --selftest
```

```
# Windows (a windowed app, so read the result from the report file + exit code)
$env:MIA_SELFTEST_REPORT = "$env:TEMP\mia-selftest.txt"
(Start-Process "$env:LOCALAPPDATA\Programs\MIA Toolkit\MIA Toolkit.exe" -ArgumentList --selftest -Wait
-PassThru).ExitCode
Get-Content "$env:TEMP\mia-selftest.txt"
```

Network, proxy, and firewall

MIA Toolkit makes **no background network calls**. There is no telemetry, no analytics, no account, no cloud sync, and no automatic update check. The app's single network call is the **user-initiated "Check for Updates"** (one HTTPS GET of `https://miatools.tech/version.json`, see Update policy). Other outbound connections are **standard links the user explicitly clicks** (e.g. the website or support email), which open in the user's default browser or mail client — not from the app.

- **Nothing to allowlist.** No domains, ports, or endpoints need to be opened for the app to function. Allow `miatools.tech` (HTTPS/443) only if you want users to be able to run the update check.
- The app works fully on an **air-gapped / fully isolated** workstation.

Data locations

Data	Location
Working project (copied disc contents, inventory, archive build, session logs)	<code>~/Documents/MedicalArchive</code> by default (Windows: <code>%USERPROFILE%\Documents\MedicalArchive</code>); the user can choose another folder
Delivered archive	the USB drive or folder the user picks at the end
App settings (only the interface language)	Windows: <code>%APPDATA%\mia-toolkit\config.json</code> · macOS: <code>~/Library/Application Support/mia-toolkit/config.json</code>
Program files	Windows: the install folder above · macOS: the <code>.app</code> bundle

- The app does not write to the registry beyond the installer's own uninstall entry, and it has no machine-wide settings or policy file.
- The default project path is built from the user's home folder (`~/Documents`), not from the Windows "Documents" known-folder API. If your policy redirects Documents (e.g. OneDrive Known Folder Move), the default may not follow the redirect. Users can pick a different folder.[†]
- Imaging media (PHI) is read from the source discs or folders and written only to local disk and the destination USB drive. It never leaves the workstation.

Update policy

- **No auto-update.** The app never checks for or downloads updates on its own.
- **IT controls versions.** New versions are published on GitHub Releases. IT decides when to test and roll out a version, using the same silent-install commands above (in-place upgrade). You can pin a validated version indefinitely.

Update check (user-initiated only): the Help/app menu's "Check for Updates" performs a single HTTPS GET of a static version file (`https://miatools.tech/version.json`) **only when a user clicks it**. It uses a generic User-Agent with no version, no query parameters, no identifiers and no background checks, ever. It only reports whether a newer version exists; it never downloads or installs anything. Blocking that domain disables the check cleanly with no other effect on the app.

Code signing and verification

- **macOS:** signed (Apple Developer ID) and notarized since v0.1.5. Verify with `spctl --assess --type execute -vv "/Applications/MIA Toolkit.app"` (expect source=Notarized Developer ID).
- **Windows:** the installer is **not yet code-signed**. SmartScreen may show "Unknown publisher": choose More info → Run anyway, or allow it via your endpoint policy (e.g. by file hash). An Authenticode certificate is planned; this section will be updated when signed installers ship.
- **Checksums:** the SHA-256 of each installer is published in <https://miatools.tech/version.json>. Releases after v0.1.13 also attach a SHA256SUMS file (`shasum -a 256 -c SHA256SUMS --ignore-missing`, or `Get-FileHash <file> -Algorithm SHA256` on Windows).
- **Provenance + SBOM (releases after v0.1.13):** GitHub build-provenance attestations (`gh attestation verify <file> --repo Fritanga-Collective/mia-toolkit`) and one CycloneDX SBOM per platform (*.cdx.json).

Uninstall

- **Windows:** Settings → Apps → MIA Toolkit → Uninstall, or run the uninstaller in the install folder silently: "`<install folder>\unins000.exe`" `/VERYSILENT /SUPPRESSMSGBOXES /NORESTART` (per-user default: `%LOCALAPPDATA%\Programs\MIA Toolkit\unins000.exe`). The uninstall entry lives under `HKCU\...\Uninstall` (per-user) or `HKLM\...\Uninstall` (all users), key `{8F3A1C2E-7B4D-4E9A-9C1F-2D5E6A7B8C90}_is1`.[†]
- **macOS:** delete MIA Toolkit.app from `/Applications` (or `~/Applications`).
- **Residual data:** the uninstaller removes only program files. The settings file and any project under `~/Documents/MedicalArchive` stay until removed by hand. Delete them if a full wipe is required (the project folder can contain patient imaging).

Support

We never need patient data. Send de-identified screenshots or logs only — never studies, DICOM files or PHI. Support is best-effort by email at support@miatools.tech, with a target first response within **2 business days**.