

Setup & Operations

English edition · 26 September 2026

UIC Home 3.9.14-dev · xTMSAgent · .NET 8 · SQL Server

Contents

- 1. Scope and preparation / 2
- 2. Install the Windows server / 3
- 3. Ports, TLS and protected secrets / 4
- 4. Register a terminal and install UIC Home / 5
- 5. Development enrollment through USB / 6
- 6. Android permissions and acceptance / 7
- 7. Dashboard, fleet and remote operations / 8
- 8. Rotate the Device Secret / 9
- 9. IoT certificates: provision, rotate, revoke / 10
- 10. Application library and deployment / 11
- 11. Schemas, catalogs and device parameters / 12
- 12. Maintenance and troubleshooting / 13
- 13. Account, password recovery and MFA / 14
- 14. Device integration manuals and location history / 15

1. Scope and preparation

This guide describes the implemented system with UIC Home 3.9.14-dev (version code 27) and xTMSAgent. NEWPOS NEW9810, NEXGO N82 and UIC A10-T have been exercised against the portal; each production firmware still needs formal qualification. Development APKs and signing keys are not production release credentials.

The bank is the customer. SuperAdmin creates customers, registers models and devices, and rotates credentials. Customer administrators operate within their assigned customer scope. Select the correct bank before changing data.

Prepare Windows x64, .NET 8 and ASP.NET Core 8 runtimes, an existing SQL database, a reachable server DNS name, TLS certificates and protected backups. IIS, the .NET SDK and Windows Desktop runtime are not required. Keep runtimes patched through normal maintenance. The server package does not compile, configure or install Android apps.

APK inspection is included in the server package and runs in-process. No Android SDK or Java installation is required. Production agent enrollment, malware scanning and qualification of each terminal firmware remain rollout prerequisites.

Record server names, ports, installation paths, certificate expiry and administrator contacts. Keep secrets in an approved vault, never in screenshots or this manual.

2. Install the Windows server

1. Run `dotnet --list-runtimes`. Confirm `Microsoft.NETCore.App 8.x` and `Microsoft.AspNetCore.App 8.x`.
2. Create the SQL database and login separately. Defaults: `UICConnectTMS` and `app_uic_tms`. Setup needs migration permissions; services need application access. The installer does not create the database or SQL login.
3. Compare `Get-FileHash <package.zip> -Algorithm SHA256` with the supplied checksum. Extract the complete package into a staging folder.
4. Run `Configure.cmd` as Administrator. Keep its setup and payload folders together. Default configuration directory: `C:\ProgramData\UICConnectTMS`; installation directory: `C:\Program Files\UICConnectTMS`.
5. Enter SQL server, database, authentication and password. SQL encryption defaults to true; Trust server certificate defaults to false. Prefer a trusted SQL certificate.
6. Configure endpoints, certificates and artifact storage. APK inspection libraries are included; there are no Java or SDK paths to configure.
7. Create the initial administrator for a new database only, using at least 12 password characters. Existing users are preserved. Choose installation/start of Windows services.
8. Open the configured HTTPS/MQTT firewall ports yourself; setup does not create firewall rules. Sign in at `https://<server>:5102` or your configured port.

Confirm `UICConnectTMS.DataService`, `UICConnectTMS.Mqtt` and `UICConnectTMS.Portal` are Running using `Get-Service`. Service startup alone does not prove terminal connectivity.

Enter retains existing settings, including stored passwords. Passwords are masked and never displayed as defaults. On updates, select the same configuration directory to retain settings.

3. Ports, TLS and protected secrets

Component	Production default
Portal HTTPS and remote desktop	5102, external TLS
MQTT	5883, mutual TLS
Portal HTTP	15100, loopback only
Internal HTTP/2 gRPC	15101, loopback only
Remote session / admission / concurrency	10 minutes / 120 seconds / 100

The remote relay runs inside the portal. No separate legacy proxy service is installed. Session limits are configuration values, not measured capacity guarantees.

Choose pfx, machine or user for the server certificate. PFX needs its path and password. Machine/User searches the Personal (My) certificate store by exact CN or friendly name. Select the numbered match and verify expiry, private key and hostname SAN. User means the account running elevated setup.

Store selection requires an exportable private key and creates an encrypted PFX copy for LocalService. Renewals are not automatic: rerun setup with the renewed certificate. A self-signed server certificate requires explicit client trust. Do not disable validation to resolve name or trust errors.

The server TLS certificate, IoT client CA and APK signing certificate have different purposes. Setup creates a separate IoT CA. Enter the 64-hex common bootstrap key in its masked prompt and deliver it only through protected terminal enrollment.

Database credentials and other secrets are encrypted in `runtime.secrets.dpapi` using machine-bound Windows DPAPI and restricted permissions. Moving this file alone to another server will not work: rerun setup and reenter secrets. Preserve the database, artifacts, TLS certificate and IoT CA together in protected backups. Never distribute development `.local` files.

4. Register a terminal and install UIC Home

1. As SuperAdmin, open Administration. Create/select the bank customer and its initial administrator.
2. Add the model with Brand, Model and OS (Android or Linux). Register the terminal under that customer and model.
3. Use the manufacturer label serial, not Android's internal serial. They differ on the verified NEW9810.
4. Securely save the one-time enrollment: Device ID, Device Secret, customer, assignment epoch and MQTT topic prefix. The enrollment screen may still label the secret Password. The server retains only its hash.
5. Add the reachable broker hostname, TLS port, HTTPS base URL, trusted server certificate and IoT bootstrap key to the enrollment configuration. Synchronize terminal time; bootstrap tolerates a two-minute difference.
6. Install the approved brand-specific UIC Home APK. Preserve its signing identity for updates. The NEWPOS development package is `com.uic.tms.home.uic.newpos.debug`; release/flavor package names may differ.

Current limitation: Android private-file enrollment import is debug-only. A release APK alone does not configure the terminal. Production deployment requires an approved provisioning channel before rollout. Follow the Android integration manual for APK installation from internal storage or USB/OTG media, enrollment, restrictions and acceptance checks.

For development USB installation, authorize the workstation on the terminal, run `adb devices -l`, select the exact serial and use `adb -s <serial> install -r <UIC-Home.apk>`. The `-r` update preserves existing app data. Do not uninstall or clear storage during routine updates.

5. Development enrollment through USB

This procedure is for authorized development terminals, not a production release enrollment method. ADB belongs on the provisioning workstation, not the TMS server.

Prepare protected enrollment JSON with these fields: `deviceId`, `password` (Device Secret), `customerId`, `serial`, `assignmentEpoch`, `topicPrefix`, `host`, `port`, `packageBaseUrl`, `serverCertificate` (PEM), and `iotBootstrapKey`. Use actual administrator-supplied values. The technical field name remains `password`.

Transfer the JSON through ADB run-as into `files/tms-enrollment.json` within the app sandbox. Send the JSON through standard input; do not place secrets in command arguments or shared storage. The exact receiver command is shown below; a provisioning script must supply the protected JSON bytes on its standard input.

```
adb -s <serial> shell run-as <package> sh -c 'cat > files/tms-enrollment.json'
```

Restart UIC Home after transfer. The debug importer encrypts enrollment using `AndroidKeyStore` and deletes the plaintext input. Verify `tms-enrollment.enc` exists and the input JSON is gone. Start `com.uic.home.LauncherActivity` within the installed package and check connection status.

Only for the current local USB lab, enrollment uses localhost, MQTT 18883 and HTTPS `https://localhost:15102`. It requires both reverse mappings:

```
adb -s <serial> reverse tcp:18883 tcp:18883
adb -s <serial> reverse tcp:15102 tcp:15102
```

Mappings disappear after USB disconnect. Production uses a real reachable server hostname and its configured ports, normally 5883/5102. USB success does not prove standalone connectivity.

Uninstall, Clear storage and Gradle connected-test cleanup can remove enrollment/private keys. Avoid that test runner on enrolled terminals; direct instrumentation can preserve the target app. Lost keys require re-enrollment and IoT certificate rotation.

6. Android permissions and acceptance

Set UIC Home as the default Home application and press Home to verify. Wi-Fi and Cellular shortcuts intentionally need no passcode. Android Settings uses a six-digit UIC Home passcode; the development default is 896345. Change it before handover. This passcode is not the Device Secret.

Select Allow connection with screen off and accept Android's Allow prompt. The agent already uses a persistent foreground connection service, reconnect logic and a bounded partial wake lock. Battery exemption is additionally needed for reliable networking during Doze. Manufacturer policies still require testing.

Authorized USB provisioning can apply and verify the exemption:

```
adb -s <serial> shell dumpsys deviceidle whitelist +<package>
adb -s <serial> shell dumpsys deviceidle whitelist
```

Enable UIC remote input in Android Accessibility for remote taps/swipes and accept Android's confirmation. Screen capture still needs terminal approval for each session. Grant requested location permissions if location reporting is required; GPS support does not guarantee an indoor fix.

Acceptance checklist:

- IoT certificate is Provisioned; device is Online with fresh telemetry and correct manufacturer serial/model.
- Turn the display off, wait beyond the portal online interval, send a harmless message and verify wake-up and Succeeded. Success means displayed, not read.
- Request status; verify a new report. Test Home, Wi-Fi and Cellular access.
- Request remote assistance, approve sharing locally, verify an actual remote tap and end the session.
- After configuring a real endpoint, disconnect USB and verify standalone connectivity.

7. Dashboard, fleet and remote operations

Dashboard is the home page and includes fleet counts and Recent commands. Devices provides combined filters for serial, group/subgroup, tags, model and online status. Group filters include descendants. Use Groups and tags to define the bank hierarchy and assign devices; define hardware models in Administration.

Device Overview shows terminal/network identifiers, organization, free and total storage, available and total memory, and location. Missing IMEI, ICCID, MAC or coordinates may reflect hardware or OS restrictions. Reported coordinates create a location-history point only after a movement of at least 100 metres. When a device has never reported a valid position, an authorized operator can set one by address search, map pin or coordinates; a later device report replaces that manual current position. Diagnostics shows technical telemetry for troubleshooting.

Operations provides Send message, Reboot, Request status and Remote assistance. Buttons depend on Online and agent capability. Send message opens a dialog; Reboot requires confirmation. Inspect Recent commands: Queued or Dispatched does not mean completed. Failed and Expired require investigation rather than repeated submission.

Messages wake the display but do not bypass a secure Android lock. Reboot requires supported firmware. The remote viewer requires local screen-sharing consent. A View only notice means remote input is unavailable; enable UIC remote input on the device. Use maximize/full-screen and End session. Sessions are not recorded and end on portal restart; a new session requires another request.

8. Rotate the Device Secret

This is the per-terminal authentication credential, separate from the Home passcode, portal login, SQL password and IoT certificate.

1. Prepare an authorized method to update enrollment on the terminal before rotation. The new secret is not automatically delivered over the old connection.
2. As SuperAdmin, open Devices > terminal > Operations > Credential management. Expand Rotate Device Secret and read the notice.
3. Select Rotate Device Secret. This changes the credential and cancels pending commands. Do not use it as a harmless connection test.
4. Save the new one-time enrollment securely. The enrollment page may still call the value Password. Leaving without saving it may require another rotation.
5. Update the terminal's private enrollment with the new secret and returned binding values, preserving endpoints, server trust and bootstrap settings. Restart/reconnect through the approved provisioning process.
6. Confirm Online and fresh telemetry. Review cancelled work and recreate only tasks that are still required.

Changing only the portal value leaves the terminal with an invalid credential. If the terminal also lost its IoT private key, rotate the certificate too. A secret cannot be recovered from its server-side hash.

9. IoT certificates: provision, rotate, revoke

UIC Home generates its private key in AndroidKeyStore. It submits only the public key over authenticated HTTPS, downloads a signed client certificate and proves possession. Confirmation changes the status to Provisioned and locks further downloads. MQTT requires both the certificate and the correct device credential.

Open Devices > terminal > Operations > Credential management > IoT certificates. Customer scope controls visibility; SuperAdmin rotates/revokes. Issued means acknowledgement is pending. Provisioned permits MQTT. Revoked remains in history and cannot reconnect.

To rotate: select Rotate certificate and confirm the portal dialog. The previous certificate is revoked and replacement issuance is enabled. A correctly enrolled agent retries, generates a new key/certificate and acknowledges it. Verify a new Provisioned fingerprint, the old Revoked entry, Online and recent telemetry. Restore enrollment too if app data was erased.

To revoke: select Revoke certificate and confirm. This invalidates existing certificates and blocks further downloads. Use it for lost or retired terminals. Recovery is not automatic: explicit rotation reopens issuance. The broker rechecks the database revocation registry for new and active connections.

Track expiration and rotate before expiry. Automatic renewal, CA rollover and CRL/OCSP publication are not implemented. Keep the IoT CA private key, bootstrap configuration and database in coordinated protected backups. Restoring stale revocation data can restore obsolete authorization state.

10. Application library and deployment

Select the package file first. The portal shows upload progress and parses APK metadata: package, name, version and available icon. APKs allow Android models only. Associate every compatible model with that exact binary. Another APK can target other models for the same application.

Save the validated build, inspect version details/integrity and explicitly release it before deploying. Compare signing-certificate SHA-256 with previous versions; the file hash normally changes while signing identity must remain compatible. Signature verification is not malware scanning. Production scanning/approval remains required.

Create a deployment by choosing the exact released build and terminals. A blank UTC schedule starts immediately; otherwise select a future UTC time. Set the delivery window, review eligible/excluded targets and confirm. Offline terminals can receive eligible work within that window. Cancellation affects queued work, not already dispatched installation.

Delivery progress lists each deployment as one row with its application, creation date, overall status, target count, successes, failures and pending count. Select Details to inspect individual devices and search by serial number or model. Refresh updates the task states and counts without reloading the page.

On a device's Applications tab, Push application selects app/version and optional Start app after install. Inventory reports user apps with names and package IDs, excluding system apps. The uninstall button requires confirmation and is unavailable for UIC Home. After an install or removal is confirmed, the page shows live task status, requests a fresh terminal report when the task succeeds, and reloads the application inventory automatically.

Dispatched is not installed. The agent downloads through authenticated HTTPS and verifies the installation before success. xTMSAgent installs and removes approved native A10 application packages, reports inventory from user app¹, and reboots after a confirmed install or removal. UIC Home installs Android APKs only. See the Linux integration manual; formal A10-T field-install instructions are intentionally reserved for a later release.

11. Schemas, catalogs and device parameters

Open Parameters, select the bank and create/import a named schema. Define tables and fields with a single Label in the bank's chosen language. Portal controls support English/Spanish; custom labels are not translated. Use the visible Terminal > Acquirer > Issuer hierarchy and shared catalogs for hosts, card ranges and EMV data.

Save schema and catalogs. There is one current saved configuration, without Publish. Application package and Signing certificate SHA-256 are optional caller restrictions; each blank field skips only that restriction. Device authentication and tenant isolation remain required. Ordinary fields are not a vault; payment keys and nonempty encrypted/secret values are unsupported.

From a device's Parameters link, choose a schema, create empty parameters or start from a template, edit records and Save. Choose valid catalog entries for references. Reference-only rows have a compact selector/delete action; normal rows can be duplicated. Delete requires confirmation. Resolve required-field and unknown-reference errors before saving.

Save as template and Copy to device use saved values. Templates are independent afterward. Save first, then review terminal/merchant identifiers before copying to a replacement and confirm any overwrite.

When the payment app requests a schema through Home's read-only provider, the service compiles current schema, catalogs and device rows on demand. Save makes data available; it does not prove the payment app applied it. Scheduled parameter push, offline fallback and application-applied receipts are not yet available.

12. Maintenance and troubleshooting

Before server updates, back up SQL, artifacts, configuration, TLS certificate and IoT CA together. Stop all three services, extract the new package separately, run `Configure.cmd` with existing paths and preserve certificates. Verify portal, agent connection and an operation afterward. There is no automatic rollback; migrations may require a coordinated database restore.

Symptom	Checks and recovery
Offline with screen off	Network, foreground service, battery exemption, vendor policy; USB reverse mappings in the lab
IoT provisioning fails	Time, HTTPS trust/hostname, Device Secret, serial, epoch and bootstrap key; rotate if a previously provisioned private key was lost
Certificate revoked	Explicitly rotate to reopen issuance after approving return to service
Remote view only / no image	Accessibility for input; terminal capture approval and secure WebSocket access for images
APK upload/install fails	Keep bundled payload libraries intact; check signature, platform/model and firmware support
Parameters cannot save	Required fields, parent relations, current catalog references and concurrent edits

Do not disable TLS validation or extend Online timeouts to hide failures. Record device/command IDs, timestamps, version and exact errors for support; exclude secrets/private keys. Test backup restoration in a separate environment.

Technical references in this documentation set: architecture chapters 02, 17, 27, 28 and 29. Android background behavior: <https://developer.android.com/training/monitoring-device-state/doze-standby>

13. Account, password recovery and MFA

The Help link in the portal header opens this manual before or after sign-in. Select your account name to open My profile, update your name or display name, change your password, and manage multi-factor authentication (MFA).

1. On the sign-in page, select **Forgot password?**, enter your sign-in email and submit.
2. The portal always displays the same response so it does not reveal whether an account exists. When email recovery is enabled and the account is active, a bilingual recovery message is sent.
3. Open the single-use HTTPS link within 30 minutes and set a 14–128 character password. A successful reset revokes existing portal sessions. The link cannot be reused.

If no message arrives, check spam and ask the administrator whether email recovery is enabled. Customer administrators can set a new password for users in their customer; SuperAdmins manage platform administrators. Administrators cannot read an existing password.

1. Open **My profile > Multi-factor authentication > Set up MFA**.
2. Add the displayed setup key to a standards-compatible TOTP authenticator. The code changes every 30 seconds.
3. Enter the current six-digit code and your portal password. Setup expires after 10 minutes.
4. Store the ten recovery codes in an approved password manager. They are shown once and each works once.

Future sign-ins require the password and then an authenticator or unused recovery code. Five failed codes or a five-minute delay invalidate the sign-in challenge. To disable MFA, enter the current password and a valid factor in My profile. Password recovery does not disable MFA.

Windows setup records the public HTTPS portal URL and supports either standard SMTP or the native AWS SES v2 API. SMTP requires a host, port, TLS selection, sender and optional credentials. AWS SES requires a verified sender identity and region, then uses either the AWS default credential chain, such as an EC2 instance role, or a dedicated access key with `ses:SendEmail` permission. SES sandbox accounts can send only to verified recipients.

SMTP passwords, explicitly configured AWS SES credentials and the MFA encryption key are stored only in the machine-protected DPAPI secrets file. Keep the same protected configuration during upgrades. Audit records include recovery requests, completed resets, MFA enablement and disablement without storing passwords, tokens, setup keys or recovery codes.

14. Device integration manuals and location history

Use the Android integration manual for UIC Home package selection, APK installation from USB/OTG or device storage, enrollment, MQTT/HTTPS interaction, permissions, restrictions and acceptance. Use the Linux integration manual for xTMSAgent onboarding, communication setup, enrollment, telemetry, application inventory and native A10 package behavior. The Linux manual deliberately does not prescribe the future A10-T field-install procedure.

Both agents require the SuperAdmin to create the exact device serial under a compatible portal model before enrollment. A terminal cannot create its own record. The first authenticated MQTT connection closes credential download and marks the device IoT-provisioned. Re-enrollment then requires an explicit administrative rotation or reset.

Location history records reported or manual changes only when the distance from the last stored point is at least 100 metres. Manual location is available only before a valid device-reported fix exists. Every manual change is audited; reported telemetry remains authoritative.