Q5000 - CDR Reporter

Administrator Guide

Version 2.8.3

IntelliVoss

Q5000 - CDR Reporter

Administrator Guide

Version 2.7.5

Deployment, Configuration & Maintenance

1. Overview

CDR Reporter is a self-contained Windows desktop application for analyzing PBX call detail records with integrated Gemini AI for intelligent phone number identification. This guide covers deployment, configuration, and maintenance for system administrators.

Architecture:

• **Desktop Application Layer:** WPF user interface (.NET 8)

• **Business Logic Layer:** Call analysis, cost calculation, reporting, billing, compliance tracking, Gemini AI integration (via secure proxy), N-level department hierarchy, multi-PBX server support

• **Data Access Layer:** Per-server SSH tunnels with trusted SSH host pinning, MySQL via tunnel, local SQLite database with AES-256-GCM encrypted credentials per server

• **Advanced Features:** Billing, call policies, Inbound Dashboard with SLA metrics + AI Insights, Outbound Dashboard with AI Budget Forecast + AI Security Alert, AI Suggest Departments in Extension Management, AI pattern detection in Personal vs Business, automated backup/restore, full PDF/Excel parity

Deployment Model:

• Self-contained .NET 8 Desktop Runtime (~250 MB)

• Single MSI installer (~90-180 MB)

• No separate runtime installation required

• In-place upgrades supported

• First-run wizard for guided setup

2. System Requirements

Client Workstations:

• Windows 10 (64-bit) version 1809+ or Windows 11

• Dual-core 2.0 GHz or faster

• 4 GB RAM minimum, 8 GB recommended

• 500 MB disk space for installation

• 1366x768 minimum resolution

• Network access to PBX

• Outbound HTTPS (port 443) for Gemini AI (optional)

PBX Server:

• FreePBX, Issabel, VitalPBX, or compatible Asterisk system

• SSH access enabled (port 22)

• MySQL/MariaDB 5.7+ with CDR database

• Standard Asterisk CDR schema (table 'cdr')

• CDR table columns: calldate, src, dst, duration, billsec, disposition, accountcode

• **PBX Timezone** configuration recommended (v1.9.0+)

Database:

• MySQL/MariaDB 5.7+ or later

• 100+ MB free space for CDR data

• Read-only access sufficient for CDR Reporter user

Firewall Requirements:

• **Outbound SSH** (port 22) from clients to PBX

• **Outbound MySQL** (port 3306) via SSH tunnel

• **Outbound HTTPS** (443) to **`q5000.co.za`** for licensing, AI lookup, and email delivery (clients no longer make direct SMTP connections - email is relayed by the Q5000 service)

3. Deployment

3.1 Manual Installation

Per-User Installation:

1. Download **`CDRReporter-Setup.msi`**

2. Double-click to run

3. Follow wizard

4. On first launch, First Run Setup Wizard appears (see Section 3.5)

3.2 Silent Installation

For batch deployments:

```
msiexec /i CDRReporter-Setup.msi /quiet /norestart
```

With log file:

```
msiexec /i CDRReporter-Setup.msi /quiet /norestart /l*v install.log
```

After silent install, First Run Setup Wizard appears on first launch.

3.3 Group Policy Deployment

Enterprise deployment:

1. Copy MSI to network share: **`\\server\share\CDRReporter-Setup.msi`**

2. Open Group Policy Management

3. Navigate: Computer Configuration -> Software Settings -> Software installation

4. Right-click -> New -> Package

5. Select MSI from network share

6. Choose "Assigned" deployment

7. Link to target OU

8. Apply and let GPUPDATE /FORCE run on clients

3.4 Installation Locations

Program Files:

• Default: `C:\Program Files\Intellivoss\Q5000\`

• Includes executable, libraries, configuration

User Data:

• Location: `%LocalAppData%\CDRReporter\`

• Windows: `C:\Users\{username}\AppData\Local\CDRReporter\`

• Per-user isolated data

Local Database:

• Location: `%LocalAppData%\CDRReporter\cdrreporter.db`

• SQLite database with encrypted credentials

• Size: ~1-5 MB (depends on configuration)

Logs:

• Location: `%LocalAppData%\CDRReporter\logs\`

• Format: `cdr-reporter-YYYYMMDD.log`

• Retention: 7 days rolling

3.5 First Run Wizard (v1.7.0+)

The **First Run Setup Wizard** appears on first launch, providing 4 configuration paths:

Wizard Paths:

Path 1: SSH Connection (Real PBX System)

• Administrator enters SSH host, port, username, password

• Enters MySQL CDR database details

• Wizard tests both connections

• On success, dashboard loads with live CDR data

Path 2: Demo Mode (No PBX Needed)

• Creates sample data: about 900 calls across 4 months, 7 departments, 12 staff, 2 call policies

• User explores features without real PBX

• Demo data can be cleared anytime from Settings

Path 3: CSV Import (Bulk CDR Data)

• User uploads CDR records as CSV

• Wizard maps CSV columns to CDR fields

• On completion, imported data loads in dashboard

Path 4: Restore from Backup

• User selects previous backup file

• Wizard restores configuration, PIN mappings, departments, settings

• On completion, restored dashboard appears

Reinstall Behavior:

• Registry flag **`NewInstall`** resets wizard on reinstall

• Uninstall -> Reinstall triggers fresh wizard

• Useful for reconfiguring PBX or resetting to demo

Administrative Notes:

• Wizard runs per-user (each Windows user sees their own wizard)

• Configuration stored in user's LocalAppData

• Multiple users can configure different PBX servers on same machine

• No centralized configuration needed

• Activating a license on first run flows directly into the wizard - no restart needed between activation and configuration (v2.7.1+)

3.6 Start with Windows (v2.7.0+)

Q5000 can launch automatically each time the user signs in to Windows. This is turned off by default and is controlled per user from **Settings -> Sync -> Startup** with the "Start Q5000 automatically when Windows starts" checkbox (click **Save Startup Setting** to apply).

• **Scope:** the setting applies only to the current Windows user account. It does

not require administrator rights and does not affect other users on the machine.

• **How it works:** enabling it adds a **`Q5000`** value under the per-user registry key

`HKCU\Software\Microsoft\Windows\CurrentVersion\Run` pointing at the installed executable; disabling it removes that value. On login the application opens as it would when launched manually (any configured startup sync then runs as usual).

• **Verifying:** the checkbox always reflects the actual registry state, so it stays

accurate even if the entry is changed outside the app. If a locked-down environment prevents the change from taking effect, Q5000 reports a warning rather than a false success and leaves the setting unchanged (details are written to the application log).

• **Uninstall (v2.7.1+):** uninstalling Q5000 removes the **`Q5000`** Run entry as part of

the uninstall, so no orphaned startup entry is left behind.

4. PBX Configuration

4.1 Create Dedicated SSH User

On PBX (Linux shell):

```
# Create user
useradd -m -s /bin/bash cdrreader

# Set password
passwd cdrreader
```

> ***Q5000 authenticates to the PBX with a username and password.*** *SSH key-based authentication is not supported in this version, so the* **`cdrreader`** *account must keep password authentication enabled. Do* ***not*** *set* **`PasswordAuthentication no`** *in* **`/etc/ssh/sshd_config`** *for this account - Q5000 will no longer be able to connect.*

>

> ***Best Practice:*** *give* **`cdrreader`** *a long, unique, randomly generated password used nowhere else, and restrict the account as described in section 6.3.*

4.2 Create MySQL User

You have two options: let CDR Reporter create the read-only user for you (recommended), or create it manually.

Option A: Automatic (recommended, v2.3.3+)

If you have SSH access to the PBX but do not know the MySQL password, CDR Reporter can create a dedicated read-only database user for you. In the **Setup Wizard** (CDR Database step) or in **Settings -> Add/Edit PBX Server**, after testing the SSH connection click **"Create read-only user automatically"**.

CDR Reporter will, over your existing SSH session:

1. Discover a working privileged database credential from the PBX's own configuration. It looks (in priority order) at the Issabel **`/etc/issabel.conf`** and Elastix **`/etc/elastix.conf`** **`mysqlrootpwd`**, **`/root/.my.cnf`**, the FreePBX **`/etc/freepbx.conf`** **`freepbxuser`**, **`/etc/amportal.conf`**, the live **`res_odbc`** ODBC credentials, and finally root via the local MySQL unix socket (as used on the FreePBX distro).

2. Create a least-privilege user named **`cdrreporter_ro`** with a strong generated password, granting **SELECT only** on both the CDR database (e.g. **`asteriskcdrdb`**) and the **`asterisk`** configuration database (so PBX extension import works).

3. Verify the new user and save its credentials, AES-256-GCM encrypted, in the server settings.

No MySQL root password is ever typed by you. The privileged password is written to a temporary, permission-restricted file on the PBX and deleted immediately; it is never passed on the command line. The operation is idempotent (re-running it rotates/repairs the user) and rolls back the half-created user on any failure.

If no usable credential can be discovered (or the stored one is outdated), the wizard reveals a field to enter a privileged MySQL user and password once, then creates the user with that.

Option B: Manual

On PBX MySQL:

```
mysql -u root -p

CREATE USER 'cdrreader'@'localhost' IDENTIFIED BY 'securepassword';
GRANT SELECT ON asteriskcdrdb.* TO 'cdrreader'@'localhost';
-- also grant the config database if you use PBX extension import:
GRANT SELECT ON asterisk.* TO 'cdrreader'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Security Notes:

• Grant SELECT only (read-only)

• Use strong password (16+ characters)

• Restrict to localhost if possible

• Rotate credentials quarterly

4.3 PBX Timezone Configuration (v1.9.0+)

Purpose: Convert PBX CDR calldate timestamps from PBX timezone to local workstation timezone during sync.

Setup:

On PBX, determine the timezone identifier (e.g., **`Africa/Johannesburg`**, **`America/New_York`**):

```
# List available timezones
timedatectl list-timezones

# Check current PBX timezone
timedatectl
```

In CDR Reporter, **Settings -> PBX Connection -> Timezone**:

1. Select the PBX timezone from dropdown

2. Click **Test Connection** to verify

3. Save

Effect:

• When CDR records are synced, calldate times are converted to your local timezone

• Dashboard, reports, and analytics display times in local timezone

• Useful for multi-site deployments with different timezones

4.4 Firewall Configuration

On PBX:

```
# Check SSH service
systemctl status sshd

# Allow SSH
firewall-cmd --permanent --add-service=ssh
firewall-cmd --reload

# Verify
firewall-cmd --list-all
```

4.5 Test Connection

From Windows client:

```
# Test SSH connection
ssh cdrreader@pbx.example.com

# Test database access (via SSH tunnel)
mysql -u cdrreader -p asteriskcdrdb
SELECT COUNT(*) FROM cdr;
EXIT;
```

5. Database Setup

5.1 CDR Database Schema

CDR Reporter expects standard Asterisk CDR schema:

Table: `cdr`

Required Columns:

• **`calldate`** - Call timestamp (indexed for performance)

• **`src`** - Calling number

• **`dst`** - Called number

• **`duration`** - Total call duration (seconds)

• **`billsec`** - Billable duration (seconds)

• **`disposition`** - Call result (ANSWERED, NO ANSWER, BUSY, FAILED)

• **`accountcode`** - Account code (PIN)

Optional Columns:

• **`uniqueid`** - Unique call identifier

• **`channel`** - Channel name

• **`dcontext`** - Dialed context

• **`userfield`** - Custom field

5.2 Local Database

SQLite database stored locally on each client:

Location: `%LocalAppData%\CDRReporter\cdrreporter.db`

Stores:

• PBX connection credentials (encrypted)

• PIN mappings (PIN -> User -> Department)

• Departments and budgets with N-level hierarchy (v1.9.1+)

• Phonebook entries with categories

• Cost rules and rate cards

• Report templates

• Call policies and violation history

• Billing configurations

• User preferences and settings

• Alert history and acknowledgements

• Multi-PBX server configurations (v1.8.4+)

• Call Identifier Mode per server (v1.8.8+)

• N-level department hierarchy with materialized paths (v1.9.1+)

Size: Typically 1-10 MB depending on configuration

Budget review state (v2.8.0+): the quarterly budget-review reminder is driven by two rows in the **`settings`** table - **`BudgetReviewLastRun`** (stamped when a scan completes) and **`BudgetReviewSnoozedUntil`** (set three months out when the user chooses **Ignore**). Both hold ISO 8601 timestamps. Clearing them makes the dashboard reminder reappear on the next load; no schema change was needed for the feature. The scan itself is arithmetic over the local CDR table - it needs no license token and makes no AI or network calls, but it does read up to three months of call records, so on a very large database expect it to take a few seconds.

5.3 Encryption

Credential Security:

• All passwords encrypted with **AES-256-GCM**

• Windows DPAPI for key management

• User-specific encryption (non-transferable)

• No plaintext passwords in logs or memory

Security Implications:

• Backup database is encrypted

• If database restored to different user, passwords must be re-entered

• Migration between machines requires re-entry of credentials

5.4 Startup Schema Verification (v1.7.0+)

On every startup:

• Application verifies all required tables exist

• Auto-creates missing tables from SQL templates

• Detects schema version and applies migrations if needed

• Logs any schema changes to application log

• Prevents crashes from missing tables after restore

• **Orphaned reference repair** - nulls out **`department_id`** values in **`extension_mappings`** and **`pin_mappings`** that point to deleted departments (v1.8.8+)

What it Checks:

• All 8 core tables present (server, pin_mappings, departments, etc.)

• All required columns exist with correct types

• Indexes present and current

• Schema version matches current version

5.5 Schema Version Tracking

Current: Schema version **v50** (v2.7.9+; unchanged through v2.8.2)

Migrations v32-v34 (v1.9.1+) added N-level department hierarchy:

• **`hierarchy_path`** materialized path column for efficient ancestor/descendant queries

• Database triggers to maintain hierarchy_path on INSERT/UPDATE

• Support for unlimited depth parent/child nesting

Migration v42 (v2.0.7+) adds the **`pending_cost_recalc`** flag for automatic historic cost recalculation on upgrade. When the flag is set, the next startup runs **`ReclassifyAllCallsAsync`** once (which also recomputes cost per row) and then clears the flag. This path runs even when auto-sync-on-startup is disabled, so cost recalibrations following a rate-rule change take effect on every install without user intervention. The splash screen shows **"Recalculating call costs under current rates..."** during this pass.

Migration v44 (v2.3.5) adds a unique dedup index on **`local_cdr(server_id, calldate, src, dst, billsec)`** so re-uploading the same CSV can no longer create duplicate call rows; existing duplicates are cleaned up during the upgrade (an automatic database backup is taken before the migration).

Migration v45 (v2.4.0) deactivates scheduled billing jobs created before v2.4.0. Those jobs were saved by a dialog that could not configure a billing filter, and the background scheduler that would run them was never active - so none of them ever fired. Now that the scheduler works, they are switched off on upgrade so they cannot surprise-fire with an empty filter; recreate the schedules you want from **Billing -> Scheduled Jobs**.

Six composite indexes added automatically on first launch after upgrade (v1.8.6+):

• **`call_policy_violations(server_id, detected_at)`**

• **`billing_runs(created_at)`**

• **`pin_mappings(department_id)`**

• **`pin_mappings(server_id, is_active)`**

• **`extension_mappings(department_id)`**

• **`departments(parent_id)`**

These indexes improve query performance for billing, policy violations, and department filtering - no manual action required.

Migration v50 (v2.7.9) adds **`contact_phone`** / **`contact_name`** to the billing tables for individual-contact scope on billing runs and rate cards. In v2.8.0 the migration's error handling was tightened: previously any failure of the four **`ADD COLUMN`** statements was swallowed and the database was stamped v50 anyway, leaving a half-migrated database that never re-ran and made billing fail permanently on "no such column: contact_phone". Only the harmless "duplicate column name" case is now tolerated; anything else leaves the schema version below 50 so the next startup retries. Note this prevents the broken state rather than repairing it - a database already stamped v50 with missing columns is past the migration and needs a restore from a pre-upgrade backup.

All prior schema changes from v1-v50 are applied automatically (calldate index, CDR schema tracking, multi-PBX support, hierarchy support, cost recalc flag, CDR dedup index, legacy billing-job deactivation, billing contact scope, etc.).

Multi-PBX schema (v1.7+): **`servers`** table holds per-server encrypted SSH and CDR credentials (each with their own IV and auth tag for AES-256-GCM). **`cdr_schema_multi`** tracks per-server CDR column variations. **`sync_status_multi`** tracks per-server last-sync state. **`trusted_ssh_hosts`** stores SSH host key fingerprints for pinning (warns on key change).

6. Security

6.1 Credential Storage

• **AES-256-GCM encryption** for all passwords

• Windows DPAPI key management (per-user)

• Credentials never plaintext in logs

• No password caching in memory

6.2 Network Security

• **All PBX communication over SSH** - encrypted tunnel

• **MySQL tunneled through SSH** - no direct database connection

• **No listening ports** on client machines

• **HTTPS (TLS) to q5000.co.za** for licensing, AI lookup, and email relay - no SMTP credentials on the client

• **Update downloads restricted to q5000.co.za** (v2.7.2+) - the auto-updater only accepts installer downloads from the official q5000.co.za server; any other download source is rejected

6.3 Best Practices

PBX Security:

• Use read-only MySQL accounts

• Give the Q5000 SSH account a long, unique, randomly generated password (see section 4.1 -

key-based authentication is not supported in this version, so password authentication must stay enabled for that account)

• Restrict the SSH account: no sudo rights, and **`/sbin/nologin`** is not usable (Q5000 needs a

shell to open the tunnel), so limit it with **`AllowUsers`**/**`Match User`** rules instead

• IP whitelist SSH access if possible - restricting **`cdrreader`** to the reporting machine's

address is the strongest control available here

• Change default SSH port if desired

• Rotate the SSH and MySQL passwords on your normal schedule and update them in

Settings -> PBX Servers

Client Security:

• Strong Windows passwords

• Regular password rotation (90 days)

• Limit application access to trusted users

• Use Windows DPAPI user-specific encryption

• Do not share logins between users

Database Security:

• Regular backups (daily or weekly)

• Verify backup integrity monthly

• Audit database permissions quarterly

• Monitor failed login attempts

• Archive old CDR data

AI & Gemini (v1.8.0+):

• Phone numbers sent to Gemini API (not full CDR records)

• No employee names, departments, or call durations transmitted

• API key managed server-side (GEMINI.md configuration)

• HTTPS encryption for all API calls

• See "AI & Gemini Integration" section for data privacy details

6.4 Audit Trail

What's Logged:

• Connection attempts (success/failure)

• Database queries and results

• Report generation

• Configuration changes

• PIN/department/phonebook changes

• Policy violations detected

• Backup/restore operations

• Application errors and warnings

• Gemini AI lookups and limits reached

Log Location:

• **`%LocalAppData%\CDRReporter\logs\`**

Log Format:

• Filename: **`cdr-reporter-YYYYMMDD.log`**

• Rolling: 7 days (configurable)

• Separate debug and error levels

• Timestamps in UTC

Retention:

• Default: 7 days

• Recommended: Archive logs monthly

• Compliance: Retain per policy (1-7 years)

7. AI & Gemini Integration (v1.8.0+)

CDR Reporter integrates Gemini AI for intelligent phone number identification, anomaly detection, pattern-based personal number classification, and investigation.

7.1 Gemini API Setup

Server-Side Configuration (GEMINI.md):

• Gemini API key stored server-side (never on client)

• Requests routed through the Q5000 proxy at **q5000.co.za** over HTTPS

• The proxy was updated from the Gemini preview endpoint to the full release endpoint in May 2026 after intermittent 502 errors on the preview channel - current behaviour is stable

• Rate limiting per license tier applied at the proxy

Firewall:

• Clients need **outbound HTTPS (port 443)** to the q5000.co.za proxy

• Direct connections to api.gemini.google.com are NOT used - all AI traffic goes through the proxy

• If blocked, AI features hide gracefully (cards do not appear, snackbar shows "Feature unavailable")

7.2 AI Data Privacy

What data leaves the client machine:

• Phone number (digits only, no formatting)

• Approximate location (inferred from number prefix)

• Lookup timestamp

What does NOT leave:

• Call duration, cost, or time

• CDR details or call metadata

• Employee names or department information

• Your phonebook or PIN mappings

• Any company-specific information

• User configuration or settings

Example:

• User clicks "AI Brain" on call from +1-512-555-0123

• Sent to Gemini: **+1512** (area code only, for location)

• Returned: "Austin, TX - Business number"

• Not sent: employee name, call cost, or any CDR details

7.3 License Tier AI Limits

• **Starter License:** 50 AI lookups per day (user limit)

• **Business License:** 100 AI lookups per day (user limit)

• **Professional License:** Unlimited AI lookups

Enforcement:

• Daily counter resets at midnight UTC

• Snackbar notification when limit reached

• AI buttons become disabled when exhausted

• User can still use all non-AI features

• No data loss if limit exceeded

7.4 Disabling AI Features

If organization policy requires disabling Gemini AI:

1. Contact admin to disable API key in GEMINI.md

2. All AI buttons disappear from UI

3. Users see "Feature not available" if they try

4. Non-AI features unaffected

7.5 Admin Monitoring

AI Usage Logging:

• Each AI lookup logged with timestamp, user, number, result

• Check logs: **`%LocalAppData%\CDRReporter\logs\`**

• Pattern analysis available for suspicious bulk lookups

7.6 AI Cost Optimization Model (v2.1.x)

The v2.1 AI features are designed to keep query costs near zero in normal operation. Administrators should understand the cache and gate strategy:

Weekly-cached cards (Outbound Dashboard AI Insights, Inbound Dashboard AI Insights - v2.1.1):

• One AI call per dashboard per week per user

• Refreshes automatically on the 8th day, or manually via the Refresh button

• Hidden for license tiers without AI

Daily-cached card (Outbound Dashboard AI Budget Forecast - v2.1.2):

• One AI call per day, refreshing once burn rates have shifted overnight

• Only invoked when at least one department has a monthly budget set AND the current day of month is >= 3 (needs minimum data)

• Hidden if no department has a budget configured

Anomaly-gated card (Outbound Dashboard AI Security Alert - v2.1.2):

• A free SQL anomaly check runs every 6 hours - NO AI is called at this stage

• The check compares each extension's last 7 days against a 30-day baseline

• Gemini is only called when an anomaly is detected:

• Volume spike: > 3x baseline AND >= 15 calls

• International ratio spike: >= 20% recent (was < 5% baseline) AND >= 5 international calls

• In normal operation (no anomaly), AI calls for this feature are zero per day

• Card is completely hidden until an anomaly fires

On-demand button (Suggest Departments in Extension and PIN Management - v2.1.2, reworked v2.4.0):

• User-triggered only; never runs in the background

• Since v2.4.0 a **free deterministic name cross-match runs first** (user names vs department names, no AI call at all); AI analysis is only offered afterwards, for extensions that are still unmapped but have recent call activity

• Single AI call per use; the payload bundles all candidate extensions to amortise the cost

• Filtered to extensions with >= 3 calls in last 30 days (excludes truly dormant extensions)

Practical impact: Even on a Starter tier (50 lookups/day), normal usage of the v2.1.x AI features consumes approximately 2-3 lookups per day per user (1 weekly Insights cache refresh 0.3/day per dashboard, 1 daily Budget Forecast, occasional Suggest Departments invocations). The daily-limit budget remains primarily for manual phone-lookup operations.

8. Multi-PBX Extensions (v1.8.4+)

Extensions are now linked per-PBX server, enabling multi-PBX support.

8.1 Extension-to-Server Assignment

v1.8.4+ Change:

• Each extension belongs to a specific server (not global)

• Extension Management shows **Server column**

• Department dialog filters extensions and PINs by server

Example:

• Extension 101 on Server A (New York PBX)

• Extension 101 on Server B (London PBX)

• Both can coexist without conflicts

• Department "Sales" -> Extensions from Server A only

8.2 Extension Import with Server Assignment

When importing extensions:

1. Administrator specifies target PBX server

2. Imported extensions assigned to that server

3. Auto-detection from CDR assigns to source server

4. Bulk operations respect server boundaries

8.3 PIN Import with Server Assignment (v1.9.0+)

When importing PINs via CSV:

1. **Server Assignment dialog** appears after CSV upload

2. Administrator selects target server from dropdown (or "All Servers")

3. Optional **Skip Duplicates** checkbox to skip existing PINs

4. Click Import

5. All imported PINs assigned to the selected server

This allows controlled PIN management across multiple PBX systems.

8.4 Department -> Extensions Filtering

When configuring a department:

1. **Server filter** dropdown shows available PBX servers

2. **Extensions list** shows only extensions from selected server

3. PINs list filtered similarly (multi-server PIN support)

4. Each department can use extensions from one or multiple servers

8.5 Reassigning Extensions

To move an extension between servers:

1. Go to Extension Management

2. Select extension

3. Change **Server** dropdown

4. Save

5. Extension now associated with new server

8.5a Server Filter on Extension / PIN Management (v2.6.0+)

Both the Extension Management and PIN Management screens have a **server filter** dropdown next to the search box. It uses "applies-to" semantics:

• **All Servers** (default) - shows every mapping.

• A **specific server** - shows the mappings scoped to that server **plus** the "All Servers" mappings, i.e. everything that actually takes effect when reporting on that server.

The filter composes with the search box, and your selection is kept when you press Refresh.

> ***Scope and imports (v2.6.0):*** *A mapping saved as "All Servers" (server_id NULL) is considered to already exist for* ***any*** *server. PBX and CSV imports skip a code that already exists in any scope, so re-running an import - or importing after a backup restore - no longer creates server-specific duplicates. Backups now record each mapping's server scope and restore it (matching servers by name across machines); backups taken before v2.6.0 restore all mappings as "All Servers".*

8.6 Call Identifier Mode (v1.8.8+)

After each CDR sync, the application automatically detects whether the PBX uses PINs, extensions, or a combination of both and stores the result per server. This detection drives correct user grouping across all reports - no manual configuration is required on a standard PBX.

Detection logic:

• Examines **`accountcode`** (PIN) and **`src`**/**`dst`** (extension) fields across recent CDR records

• If most calls have a populated accountcode, mode is set to **PIN Only**

• If most calls use numeric short extensions without accountcodes, mode is set to **Extension Only**

• If both patterns are present, mode is set to **Mixed**

Admin override:

1. Open **Settings**

2. Locate **Call Identifier Mode**

3. Select the correct mode from the dropdown:

• **Auto** - re-detect on next sync (recommended)

• **Extension Only** - always group by extension

• **PIN Only** - always group by account code

• **Mixed** - use PIN when available, fall back to extension

4. Save

The override persists until manually changed or until an administrator resets it to Auto.

9. Email Configuration (v2.2.0+)

CDR Reporter sends email through the **Q5000 secure email relay** - no SMTP setup required, and no email credentials are stored on the client.

9.1 Secure Email Relay (v2.2.0+)

How it works:

• Budget alerts, scheduled reports (v2.4.0+), and scheduled billing runs (v2.4.0+) are POSTed to the Q5000 email service (**`https://q5000.co.za/api/email/send`**), which sends them on the client's behalf.

• The request is authenticated with the installation's license activation token. The SMTP credentials live only on the Q5000 server and never ship with the app.

• **Email delivery requires an activated license.** Trial installs cannot send email (the features are visible but gated).

• A per-license daily send cap applies (100/day on standard tiers; unlimited on professional).

Change from earlier versions: prior releases bundled an encrypted **`smtp_credentials.db`** next to the executable. That file (and its fixed-key **`SmtpCredentialStore`**) has been removed - the shared SMTP credential is no longer distributed with the installer.

What This Means:

• Budget alerts automatically emailed to department managers (when activated)

• Scheduled reports automatically sent to recipients with PDF attachments

• No SMTP host/port/username/password to configure anywhere on the client

9.2 Email Feature Verification

To verify email is working:

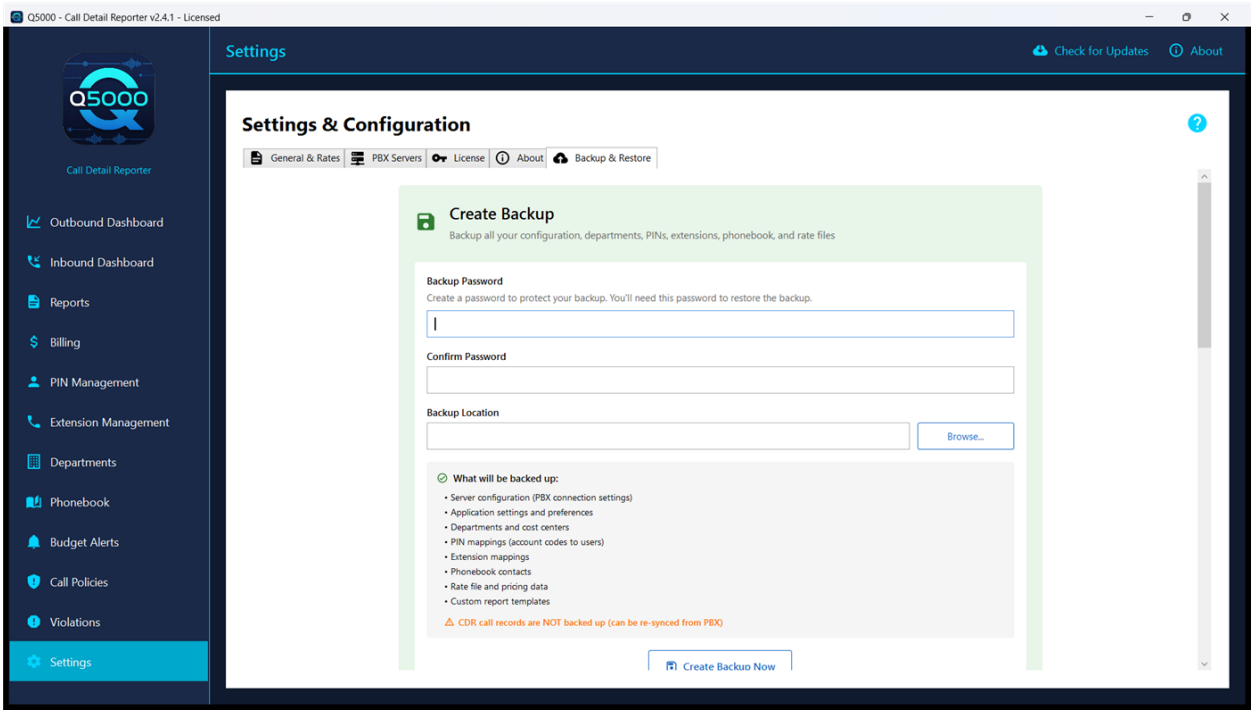
1. Open the **Email** screen - confirm it shows "Email delivery is active for this license."

2. Enter a destination address and click **Send Test**; confirm the test email arrives.

3. For alerts end-to-end: set a department budget low enough to trip a threshold, generate/sync a call, and confirm the manager receives the alert.

4. If sending fails, check logs: **`%LocalAppData%\CDRReporter\logs\`** (a 401 means the license is not activated).

10. Backup & Recovery (v1.6.5+, enhanced v1.7.0+)



Settings - Backup & Restore tab: password-protected backup of all configuration

10.1 What to Backup

Essential:

• Local database: **`%LocalAppData%\CDRReporter\cdrreporter.db`**

• This contains all configuration and settings

Included in Backup:

• cdr_schema.json (database schema definitions) - v1.7.5+

• All PIN mappings with department links

• All departments with parent/child hierarchy and N-level nesting (v1.9.1+)

• All phonebook entries and categories

• All cost rules and rate cards

• All report templates

• All policies and violation history

• All billing configurations

• All preferences and settings

• Encrypted credentials (AES-256-GCM)

Optional but Recommended:

• Application logs: `%LocalAppData%\CDRReporter\logs\`

• For troubleshooting and audit trail

NOT backed up:

• PBX CDR database (backup separately from PBX)

• Installed application files (can reinstall from MSI)

10.2 Automated Backup (v1.6.5+)

Machine-Derived Password (v1.6.5+):

• Backup password automatically derived from machine hardware

• No manual password needed for scheduled backups

• Backup can restore without password prompt

• User-specific (tied to Windows user and machine)

Setup:

1. Go to Settings -> Backup section

2. Enable **Scheduled Backup**

3. Set frequency (daily recommended)

4. Set location (network share or local)

5. Backups created automatically with no password

6. Old backups auto-retained (configurable)

Backup hardening (v2.7.2+):

• Manual (in-app) backups now require a password of **at least 12 characters**

• The **entire backup archive is encrypted** - all configuration and data, not just the stored server credentials

• Automatic backups remain passwordless for the user: they are protected with a machine-stored secret that is managed automatically

• Backups created by earlier versions still restore normally

10.3 Manual Backup

1. Close CDR Reporter application

2. Navigate to: **`%LocalAppData%\CDRReporter\`**

3. Copy **`cdrreporter.db`** to backup location

4. Name with date: **`cdrreporter-backup-YYYYMMDD.db`**

5. Verify file copied successfully

10.4 Transactional Restore (v1.7.0+)

Safe restore with rollback:

• All 8 table groups wrapped in transaction

• If restore fails mid-way, all changes rolled back

• No partial/corrupted state possible

• Safe to abort restore (Ctrl+C or close dialog)

Restore Procedure:

1. Close CDR Reporter application

2. Locate backup: **`cdrreporter-backup-YYYYMMDD.db`**

3. Rename current: **`cdrreporter.db`** -> **`cdrreporter.db.old`**

4. Copy backup to: **`%LocalAppData%\CDRReporter\cdrreporter.db`**

5. Launch CDR Reporter

6. Startup Schema Verification runs (auto-repairs if needed)

7. Verify data restored correctly

8. **Important:** Re-enter SSH/MySQL passwords (DPAPI user-specific)

10.5 Restore & Launch (v1.8.3+)

When restoring from backup:

1. Splash screen appears with "Syncing CDR data..." message

2. SSH connection to PBX established automatically

3. Latest CDR records synced before dashboard loads

4. Ensures dashboard shows current data post-restore

10.6 Department & Extension Hierarchy Restore (v1.7.4)

Fixed in v1.7.4:

• PIN department links correct after restore

• Extension department links correct after restore

• Parent/child department hierarchy preserved

• N-level hierarchy fully restored with all ancestor links (v1.9.1+)

• Two-pass restore approach ensures consistency

• All relationships re-established correctly

10.7 Expanded Scope & Hardened Restore (v2.7.3)

Backups now also include:

• Billing rate cards, billing run history and scheduled billing jobs

• Invoice settings and the invoice number sequence

• Call policies and budget-alert history

• Each server's PBX timezone setting

Restore hardening:

• The invoice sequence merges by **highest number per year** on restore, regardless of

conflict mode - a restored machine can never re-issue invoice numbers already sent to customers

• Overwrite restore is fully transactional: a failure mid-restore rolls everything

back (previously the server list could be wiped before the failure)

• Restored servers keep their original IDs and any surviving cached CDR history is

re-pointed at them by server name - per-server dashboards stay populated and re-syncing cannot duplicate call data

• Departments restore matched by **name within the same parent**, so two departments

with the same name under different branches survive intact

• Encrypted (v2.7.2+) backup files whose contents have been swapped for unencrypted

data are rejected as tampered

• Servers configured without a CDR database password restore cleanly

Clearing a server's CDR cache (Server Management -> Clear Cache): as of v2.7.3 this also resets the server's sync watermark, so the next sync re-downloads the full history from the PBX. Previously the cleared history was unrecoverable in-app. As of v2.7.5 the button refuses to run while a sync is in progress for that server - checked again after you confirm, in case a scheduled sync started while the confirmation dialog was open.

Restore improvements (v2.7.5):

• Restoring **without** the departments section now matches PIN mappings, extension

mappings, department-scoped call policies and budget alerts to this machine's departments **by name**. Previously they kept the backup's internal department numbers, which could silently attach them to the wrong department when the two machines' department lists differed. Anything that cannot be matched by name is restored **unassigned** (mappings) or dropped with a logged count (alerts), never mis-assigned.

• An **Overwrite** restore permanently deletes the cached call history of any local

server that is not in the backup (the restored configuration no longer contains that server). The restore now lists those servers - with their cached record counts - in an explicit confirmation before anything is deleted. If the PBX itself no longer holds those calls, declining is the only way to keep them.

10.8 cdr_schema.json in Backup (v1.7.5+)

Database Schema Tracking:

• CDR MySQL schema exported and included in backup

• Used for validation and migration during restore

• Helps detect schema changes on PBX between backups

• Enables better error messages if schema incompatible

10.9 Factory Reset (v1.7.0+)

Clear all application data:

1. Go to Settings

2. Click **Factory Reset** button (red, at bottom)

3. Confirm dialog: "All configuration will be deleted"

4. Application clears database and exits

5. On next launch, First Run Wizard appears again

When to Use:

• Complete reset needed (moving to different PBX)

• Troubleshooting persistent issues

• Prepare machine for repurposing

• Clean exit before uninstalling

11. Troubleshooting

SSH Connection Failed

Symptoms: "Cannot connect to PBX" error

Checklist:

• Verify PBX is reachable: **`ping pbx.example.com`**

• Check SSH service running: **`systemctl status sshd`**

• Verify port 22 not blocked: **`netstat -tlnp | grep :22`**

• Confirm credentials correct (test manually: **`ssh cdrreader@pbx`**)

• Check PBX logs: **`/var/log/secure`** or **`/var/log/auth.log`**

• Verify firewall rules allow outbound SSH

Advanced:

• Verify SSH config allows password auth: **`PasswordAuthentication yes`** in

`/etc/ssh/sshd_config` (Q5000 does not support key-based authentication - if the PBX was hardened to keys-only, the connection will fail here even though the password is correct)

• Check the account is not blocked by an **`AllowUsers`**, **`DenyUsers`**, or **`Match`** rule in

`/etc/ssh/sshd_config`

• Test SSH with verbose: **`ssh -vvv cdrreader@pbx`**

Database Connection Failed

Symptoms: "Cannot connect to CDR database" error

Checklist:

• Verify MySQL running: **`systemctl status mysqld`**

• Check database exists: **`mysql -u root -p -e "SHOW DATABASES;"`**

• Confirm user privileges: **`mysql -u root -p -e "SELECT user FROM mysql.user;"`**

• Test user credentials: **`mysql -u cdrreader -p asteriskcdrdb`**

• Verify database name correct (usually **`asteriskcdrdb`**)

• Check port 3306 accessible via SSH tunnel

Advanced:

• Check MySQL logs: **`/var/log/mysqld.log`**

• Test tunneling: **`ssh -L 3306:localhost:3306 cdrreader@pbx`**

• Verify SSH tunnel working for database

Application Won't Start

Symptoms: Application crashes on launch or won't open

Checklist:

• Check Windows Event Viewer for errors

• Verify Windows 10/11 64-bit OS: **`winver`**

• Check .NET 8 runtime installed: bundled in MSI

• Verify disk space: 500 MB minimum free

• Check %LocalAppData% folder exists and writable

• Try reinstalling: Uninstall -> Reboot -> Install

Advanced:

• Check application logs: **`%LocalAppData%\CDRReporter\logs\`**

• Run Event Viewer: **`eventvwr.msc`** -> Windows Logs -> Application

• Clear application cache: Delete **`%LocalAppData%\CDRReporter\*`**

• Verify temp directory writable

No Data Displayed

Symptoms: Application runs but no CDR data shown

Checklist:

• Verify connections configured in Settings

• Test connections click "Test" buttons

• Check CDR table has records: **`SELECT COUNT(*) FROM cdr;`**

• Verify date filter includes data

• Check PBX can access database

• Review logs: **`%LocalAppData%\CDRReporter\logs\`**

Advanced:

• Check CDR data age: **`SELECT MAX(calldate) FROM cdr;`**

• Verify CDR table schema matches expectations

• Check MySQL user has SELECT permissions

• Test CDR retrieval manually from PBX

• Check **Schema Verification** in logs for auto-repair details

Inbound Calls Show 0 with Ring Groups

Symptoms: Inbound analytics show zero calls when using ring groups

Root Cause (Fixed v1.8.9):

• Ring groups use multi-stage channel names for transfers

• Previous versions could not parse complex DstChannel properly

• Affected: Inbound counts, missed call counts (inflated via non-distinct counts)

Solution:

• Upgrade to v1.8.9 or later

• Two-stage DstChannel parsing now detects ring group transfers

• Missed call counts fixed using **`COUNT(DISTINCT uniqueid)`**

Performance Issues

Symptoms: Slow report generation, application lag, frozen UI

Checklist:

• Check CDR table size: **`SELECT COUNT(*) FROM cdr;`**

• Reduce date range for queries

• Verify network connectivity to PBX

• Check PBX system load and resources

• Monitor client RAM usage and disk space

• Review database query logs for slow queries

Solutions:

• Archive old CDR data (older than 2 years)

• Schema v34 (v1.9.1+) maintains composite indexes auto-applied on launch - upgrade if still on older version

• Increase client RAM to 8+ GB

• Reduce polling interval for auto-refresh

• Use manual refresh instead of continuous monitoring

AI Features Not Working

Symptoms: AI Brain buttons disabled or "Feature unavailable"

Checklist:

• Verify outbound HTTPS (port 443) firewall rule

• Check daily limit not exceeded (see AI section)

• Verify license tier supports AI

• Check application logs for API errors

• Try again after UTC midnight (daily limit reset)

Solutions:

• Verify network access to q5000.co.za (HTTPS) â€” all AI traffic goes through the Q5000 proxy; direct Gemini access is not used

• Confirm license tier (Starter 50/day, Business 100/day, Professional unlimited)

• If disabled intentionally, contact admin to re-enable

• Check GEMINI.md server configuration

12. Maintenance

Daily Tasks

• **Monitor logs** for errors: **`%LocalAppData%\CDRReporter\logs\`**

• **Review budget alerts** for issues

• **Check data connectivity** - verify PBX/database accessible

• **Monitor disk space** on PBX CDR database server

Weekly Tasks

• **Check disk space** on client workstations (500 MB minimum)

• **Verify PBX connectivity** - test SSH tunnel

• **Test report generation** - generate sample report

• **Review error logs** for patterns or issues

• **Monitor database size** on PBX

Monthly Tasks

• **Archive old logs** - compress and store

• **Review PIN mappings** - update obsolete entries

• **Update department budgets** - Q-end/month-end adjustments

• **Backup local databases** - copy cdrreporter.db to safe location

• **Review cost rules** - verify rates current

• **Check disk space** on PBX and clients

Quarterly Tasks

• **Rotate passwords** - change PBX SSH and MySQL passwords

• **Review permissions** - audit PBX user and database access

• **Update cost rules** - refresh carrier rates if changed

• **Audit policy violations** - review compliance trends

• **Backup review** - verify backups are valid and recent

• **AI usage review** - check AI lookup patterns for anomalies

• **Review Call Identifier Mode** - confirm auto-detected mode matches your PBX setup

• **Review N-level Department Hierarchy** - verify parent/child structure is optimal for your organization

• **Run Review Budgets** (v2.8.0+) - Departments -> Review Budgets re-sizes department budgets from the last three months of actual spend; the application raises a dashboard reminder on this same quarterly cadence

Annual Tasks

• **Security audit** - review all access and credentials

• **Archive database backup** - long-term storage

• **License validation** - verify license still current

• **Disaster recovery test** - practice restoration procedure

• **Documentation review** - update runbooks and procedures

Updates

When new versions release:

1. **Backup:** Copy **`%LocalAppData%\CDRReporter\cdrreporter.db`** to safe location

2. **Close:** Ensure all instances of CDR Reporter closed

3. **Install:** Run the new **`CDRReporter-Setup-vX.Y.Z.msi`** (upgrades in-place)

4. **Verify:** Launch application and test key features

5. **Validate:** Generate sample report, check dashboards

6. **Notify:** Inform users of update completion

No additional configuration needed - upgrades are backward compatible.

v2.0.5+ cost recalculation on upgrade: If the upgrade carries a cost-rule change, the next startup will show "Recalculating call costs under current rates..." on the splash screen while it re-prices historic **`local_cdr`** rows. This is driven by the v42 **`pending_cost_recalc`** flag and runs once per upgrade, regardless of the auto-sync-on-startup setting. No action required.

v2.1.1 license token refresh: Background validation on startup uses **`NeedsTokenRefresh`** (5-day token age check) rather than the broader **`NeedsValidation`** flag. This prevents unnecessary re-validation calls. Automatic; no admin action required.

v2.4.0 scheduled billing jobs: billing schedules created before v2.4.0 are deactivated during the upgrade (they never actually ran - see Section 5.5, migration v45). If users relied on them, recreate the schedules from **Billing -> Scheduled Jobs** after upgrading.

v2.7.1 license re-validation: after a backup restore, or when the offline grace period has expired, the application re-validates the saved license with the server automatically instead of prompting for the key again. **Settings -> License** includes a **Validate Now** button for an on-demand re-check. No admin action required.

v2.7.2 update source restriction: update packages are only downloaded from the official **q5000.co.za** server; a download offered from any other host is rejected.

v2.7.2 interval auto-sync: the **Automatic CDR Sync** interval now runs app-wide (whichever screen is open), and changes saved in Settings take effect within a minute - no application restart is required.

13. Quick Reference

File Locations

```
Install:  C:\Program Files\Intellivoss\Q5000\
Data:     %LocalAppData%\CDRReporter\
DB:       %LocalAppData%\CDRReporter\cdrreporter.db
Logs:     %LocalAppData%\CDRReporter\logs\
```

Default Ports

• SSH: 22 (on PBX) - one tunnel per configured PBX server

• MySQL: 3306 (on PBX, via SSH tunnel)

• HTTPS: 443 to **q5000.co.za** - licensing, email relay, and AI features (no client-side SMTP)

Command-Line Reference

Silent Install:

```
msiexec /i CDRReporter-Setup.msi /quiet /norestart
```

Silent Uninstall:

```
msiexec /x CDRReporter-Setup.msi /quiet
```

Log Install:

```
msiexec /i CDRReporter-Setup.msi /quiet /l*v install.log
```

Support

• **Email:** support@q5000.com

• **Documentation:** See User Guide and Quick Start Guide

• **Logs: `%LocalAppData%\CDRReporter\logs\cdr-reporter-YYYYMMDD.log`**