

Formata

Installation and Administration Guide

Document ID	IM.FORMATA.INT.02
Document version	1.0
Date of issue	2026-10-06
Applies to	Formata 1.6.0
Language	English
Publisher	iMobile

Contents

- 1. General information 3
- 2. Environment requirements 3
- 3. Installation 4
 - 3.1. Preparation 4
 - 3.2. Running the installation script 4
 - 3.3. The pgcrypto extension and database roles 4
 - 3.4. Service parameters 4
 - 3.5. Loading the CRM application 5
 - 3.6. Installation in a container 5
- 4. Initial configuration 6
- 5. Verification 7
- 6. Updating 8
- 7. Backup and restore 8
- 8. Removal 9
- 9. Troubleshooting 9
 - 9.1. Logs and monitoring tools 9
 - 9.2. Typical failures 9

About this document

This guide covers installation, initial configuration, verification, updating, backup, removal and troubleshooting of Formata version 1.6.0 with the Formata CRM application on the customer's server.

It is written for an administrator experienced with Linux and PostgreSQL. Commands are given for Ubuntu 24.04 LTS; the paths `/opt/formata/core` and `/opt/formata/crm-app` are examples and may be changed.

1. General information

The product consists of the `crm_server` process, a PostgreSQL database, the web client (the `web` directory) and the CRM application (a set of SQL files). The server process runs as the systemd service `crm-fusion` under the `formata` account, accepts HTTP requests on one TCP port and serves the web-client files to browsers.

There are no pre-built binary packages: the executable is built on the server by `install.sh` from the source code in the kit. The program needs no activation or licence key and does not contact the vendor's servers.

Names created by the installation script:

Object	Default name
systemd service	<code>crm-fusion</code>
Operating-system account of the service	<code>formata</code>
Database	<code>crm_fusion</code>
Database role that owns the tables	<code>crm</code>
Database role for scripts	<code>crm_script</code>
TCP port	8080

2. Environment requirements

- Operating system: Ubuntu 24.04 LTS or Ubuntu 22.04 LTS, administrator rights (`sudo`), access to the Ubuntu package repositories and to the Python package index (PyPI) during installation.
- Hardware for a pilot installation: 2–4 CPU cores, 4–8 GB of RAM; disk space according to the size of the database and attachments.
- PostgreSQL as shipped with the operating system, with the contrib package (the `pgcrypto` extension is mandatory: password checking relies on it).
- The installation script installs the packages needed to build and run the server: `build-essential`, `cmake`, `pkg-config`, `libpqxx-dev`, `nlohmann-json3-dev`, `libcpp-httpplib-dev`, `python3-dev`, `python3-pybind11`, `pybind11-dev`, `postgresql`, `postgresql-contrib`, `curl`, `python3-pip`, the Pango and HarfBuzz libraries and DejaVu fonts.
- The packages `libhiredis-dev` and `libssl-dev` are required for the build but are not installed by the script; install them before running it.
- A reverse proxy with TLS (for example `nginx`) for user access over HTTPS.
- Workstations: a Chromium-based browser (Google Chrome, Microsoft Edge).

3. Installation

3.1. Preparation

1. Unpack the server kit into `/opt/formata/core` and the CRM application into `/opt/formata/crm-app`.
2. Install the packages that the installation script does not cover:

```
sudo apt-get update
sudo apt-get install -y libhiredis-dev libssl-dev
```

3.2. Running the installation script

```
cd /opt/formata/core
sudo CRM_PREFIX=/opt/formata/core bash install.sh
```

The script performs four steps:

1. Installs operating-system packages and uses pip to install the print-form libraries (WeasyPrint, segno, python-barcode) into `$CRM_PREFIX/pylibs`. `CRM_PREFIX` must be set: without it the libraries go to a directory that does not exist on the customer's server.
2. Starts PostgreSQL, creates the role `crm` with the password `crm` and the database `crm_fusion`, and applies the system schema `sql/schema.sql`.
3. Builds the executable `build/crm_server` (CMake, Release configuration).
4. Creates the `formata` account, writes the service file `/etc/systemd/system/crm-fusion.service`, then enables and starts the service on port 8080.

Note. The password that the script sets for the `crm` role is publicly known. Change it before going into production (see below).

3.3. The pgcrypto extension and database roles

```
sudo -u postgres psql -d crm_fusion -c "CREATE EXTENSION IF NOT EXISTS pgcrypto"
sudo -u postgres psql -c "ALTER ROLE crm PASSWORD '<owner-password>'"
cd /opt/formata/core
sudo -u postgres psql -v pw="'<script-password>'" -f deploy/role_bootstrap.sql
```

The first command enables the extension without which users cannot sign in. The third creates the `crm_script` role: Python worker processes use it to reach the database; it does not own tables and has no access to the system tables that hold passwords and the action log.

3.4. Service parameters

Create the attachment directory and a parameter file readable by root only:

```
sudo mkdir -p /var/lib/formata/uploads /etc/formata
sudo chown formata:formata /var/lib/formata/uploads
sudo touch /etc/formata/crm.env
sudo chmod 600 /etc/formata/crm.env
```

Contents of `/etc/formata/crm.env`:

```
CRM_DB=host=127.0.0.1 dbname=crm_fusion user=crm password=<owner-password>
CRM_SCRIPT_DB=host=127.0.0.1 dbname=crm_fusion user=crm_script password=<script-password>
CRM_PYLIBS=/opt/formata/core/pylibs
CRM_UPLOAD_DIR=/var/lib/formata/uploads
CRM_TELEPHONY_KEY=<random string>
```

Attach the file to the service by creating `/etc/systemd/system/crm-fusion.service.d/override.conf`:

```
[Service]
EnvironmentFile=/etc/formata/crm.env
```

Remove the line `Environment="CRM_DB=..."` with the default password from `/etc/systemd/system/crm-fusion.service`.

3.5. Loading the CRM application

```
cd /opt/formata/crm-app
CRM_PGDB=crm_fusion CRM_PGUSER=crm \
  CRM_PGPASSWORD='<owner-password>' bash apply_crm_app.sh
sudo systemctl daemon-reload
sudo systemctl restart crm-fusion
```

`apply_crm_app.sh` applies the application SQL files in the required order, grants the `crm_script` role rights on the application tables and prints the number of forms, queries, scripts, jobs and reports loaded. It can be run repeatedly.

Script variables:

Variable	Default	Purpose
CRM_PGDB	crm_app	Database name
CRM_PGHOST, CRM_PGPORT	127.0.0.1, 5432	PostgreSQL server address
CRM_PGUSER, CRM_PGPASSWORD	crm, crm	Owner role and its password
CRM_SEED	1	1 loads the initial data; 0 does not
CRM_DEMO	0	1 adds demonstration data for the warehouse, processes and bulk loading

Note. The initial data (`CRM_SEED=1`) is meant for the first installation on an empty database: loading it empties the company, contact, lead, deal and task tables. On a database with production data run the script with `CRM_SEED=0` only.

The initial data contains task types with their status sets, demonstration records and four accounts: `admin` (roles `admin, manager`), `ivanov` and `petrova` (role `sales`), `boss` (role `manager`). The password of each account equals its login.

3.6. Installation in a container

The kit includes a `Dockerfile`: an image based on Ubuntu 24.04 that contains `crm_server` and the `web` directory; the database address is set with `CRM_DB`. The print-form libraries and the CRM application are not included in the image; prepare the database with the commands from the previous sections.

4. Initial configuration

1. Sign in as `admin`, open Form designer in the Designer section of the menu, go to Users & roles, change the passwords of all four accounts with Set password and clear Active for the accounts you do not need. A password set here is stored as a hash immediately.
2. In the designer, under App shell → Login screen, clear Show demo-login hint, set the brand and tagline and choose the authentication mode.
3. Delete the demonstration records (companies, contacts, leads, deals, tasks), create users, roles and groups, and set the form permissions of roles in the Access rights form.
4. Configure the reverse proxy. Example for nginx:

```
server {
    listen 443 ssl;
    server_name crm.example.org;
    ssl_certificate      /etc/ssl/certs/crm.example.org.pem;
    ssl_certificate_key  /etc/ssl/private/crm.example.org.key;

    location = /metrics { deny all; }
    location / {
        proxy_pass http://127.0.0.1:8080/;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header Connection "";
        proxy_buffering off;
        proxy_read_timeout 3600s;
        client_max_body_size 256m;
    }
}
```

Buffering is switched off and the read timeout is long because notifications arrive over an event stream. Close port 8080 to external networks with a firewall: the service listens on all network interfaces.

Main service parameters (environment variables):

Variable	Default	Purpose
CRM_PORT	8080	TCP port of the service
CRM_DB	—	Connection string of the table owner
CRM_SCRIPT_DB	—	Connection string of the script role
CRM_DB_POOL	16	Number of database connections
CRM_DB_WAIT	120	Seconds to wait for the database at start-up
CRM_THREADS	32	Number of request-handling threads
CRM_PY_WORKERS	4	Number of Python worker processes
CRM_SCRIPT_TIMEOUT_MS	30000	Script run-time limit, ms
CRM_SESSION_TTL	1800	Lifetime of an idle open form, s
CRM_LOGIN_MAX_FAILS	5	Failed sign-ins before lockout
CRM_LOGIN_WINDOW_S	900	Window for counting failed sign-ins, s
CRM_LOGIN_LOCK_S	600	Lockout duration, s
CRM_UPLOAD_DIR	uploads	Attachment directory
CRM_STORAGE_MAX_MB	256	Size limit of an uploaded file, MB

Variable	Default	Purpose
CRM_PYLIBS	—	Directory of additional Python libraries
CRM_REDIS_URL	—	Redis address for shared sessions
CRM_TOKEN_TTL	28800	Token lifetime when Redis is used, s
CRM_OIDC_URL, CRM_OIDC_REALM, CRM_OIDC_CLIENT	—	Identity provider for domain sign-in
CRM_PUBLIC_INTAKE, CRM_PUBLIC_KEY	—	Enables website request intake and sets its key
CRM_TELEPHONY_KEY	—	Key for telephony event intake
CRM_ANTHROPIC_KEY	—	Key of the external AI service used for lead scoring

Security notes:

- without Redis, access tokens are kept in process memory: they are valid until the user signs out or the service restarts;
- `/api/telephony/event` accepts requests without a key until `CRM_TELEPHONY_KEY` is set;
- `/healthz` and `/metrics` respond without authentication;
- if `CRM_ANTHROPIC_KEY` is set, a scheduler job sends the fields of unscored leads (title, company, contact person, email address, phone, source, budget) to the external AI service every 15 minutes. Without a key the score is calculated on the server by built-in rules;
- the passwords of mailboxes connected in the Mail section are stored in the database in clear text;
- the right to change scripts and queries is equivalent to access to the system's data; give the `admin` role to trusted staff only.

Regional features: the `REGION` setting accepts `AUTO` (by the user's language), `RU` and `INTL`; individual features are switched on in the Regional features form.

5. Verification

1. Check the service and its response (commands below). Expected: `active` and a JSON response with `"ok":true, "db":true, "version":"1.6.0"`.
2. Check sign-in through the API (command below). Expected: a JSON response with a `token` field.
3. Open the product address in a browser, sign in, make sure menu items are shown as words and not as keys of the form `@t:...`, and open Dashboard.
4. Attach a file to a deal card and download it; open a warehouse document and produce a print form.
5. As a user with the `admin` role open Designer → Monitoring (the `/monitor.html` page): the status is Healthy and there are no script errors.

Commands for step 1:

```
systemctl is-active crm-fusion
curl -s http://127.0.0.1:8080/healthz
```

Command for step 2:

```
curl -s -X POST http://127.0.0.1:8080/api/login \
-H 'Content-Type: application/json' \
-d '{"login":"admin","password":"<password>"}'
```

6. Updating

There is no automatic update. A new version is delivered as a new set of files.

1. Back up the database and the attachment directory.
2. If stock CRM forms, queries or scripts were changed in the designer, export them: in the designer, Service → Config transfer → Build a set.
3. Stop the service: `sudo systemctl stop crm-fusion`.
4. In `/opt/formata/core` replace the `server`, `web`, `sql` and `deploy` directories and the `CMakeLists.txt` file with those of the new kit; keep the `build` and `pylibs` directories.
5. Rebuild the server and apply the system schema (commands below).
6. Apply the SQL files of the new application without the initial data (commands below).
7. Start the service with `sudo systemctl start crm-fusion` and run the verification steps.
8. If needed, restore your own changes: Config transfer → Apply from file.

Commands for step 5:

```
cd /opt/formata/core
cmake -B build -S . -DCMAKE_BUILD_TYPE=Release
cmake --build build -j$(nproc)
PGPASSWORD='<owner-password>' psql -h 127.0.0.1 -U crm -d crm_fusion \
-v ON_ERROR_STOP=1 -f sql/schema.sql
```

Commands for step 6:

```
cd /opt/formata/crm-app
CRM_SEED=0 CRM_PGDB=crm_fusion CRM_PGUSER=crm \
CRM_PGPASSWORD='<owner-password>' bash apply_crm_app.sh
```

Note. The apply script writes the stock forms, queries and scripts over the existing ones. Changes that were not exported before the update are lost. Your own forms and scripts with different identifiers are not affected.

Browsers receive the new shell files the next time the product is opened. If a user still sees the previous shell, reset it by opening the address with the `?nosw=1` parameter.

7. Backup and restore

There are no built-in backup tools. Back up:

- the database, which holds the data, forms, scripts, queries, logs and mail attachments;
- the attachment directory (`CRM_UPLOAD_DIR`);
- the parameter file `/etc/formata/crm.env`, the service file and its drop-in.

Creating a backup:

```
sudo mkdir -p /var/backups/formata
sudo -u postgres pg_dump -Fc -f /var/backups/formata/crm_fusion.dump crm_fusion
sudo tar -czf /var/backups/formata/uploads.tgz -C /var/lib/formata uploads
```

Restoring:

```

sudo systemctl stop crm-fusion
sudo -u postgres dropdb crm_fusion
sudo -u postgres createdb -O crm crm_fusion
sudo -u postgres pg_restore -d crm_fusion /var/backups/formata/crm_fusion.dump
sudo tar -xzf /var/backups/formata/uploads.tgz -C /var/lib/formata
sudo systemctl start crm-fusion

```

The roles `crm` and `crm_script` must exist before the restore. The backup schedule and retention period are decided by the customer.

8. Removal

The kit has no removal script. Removal procedure:

```

sudo systemctl disable --now crm-fusion
sudo rm /etc/systemd/system/crm-fusion.service
sudo rm -r /etc/systemd/system/crm-fusion.service.d
sudo systemctl daemon-reload
sudo -u postgres dropdb crm_fusion
sudo -u postgres psql -c "DROP ROLE crm_script" -c "DROP ROLE crm"
sudo userdel formata
sudo rm -r /opt/formata /var/lib/formata /etc/formata

```

These commands irreversibly destroy the database and attachments; make a backup first. Operating-system packages installed by the script are removed with operating-system tools if no other software needs them.

9. Troubleshooting

9.1. Logs and monitoring tools

Source	Content
<code>journalctl -u crm-fusion</code>	Service messages: start-up, request errors with a request number, worker process failures
<code>/healthz</code>	Database availability, version, task queue depth
<code>/monitor.html</code>	Requests and response times, pool usage, jobs, script errors
<code>/metrics</code>	Counters in Prometheus format
<code>sys_audit table</code>	Sign-ins, opening and saving of forms, script runs and errors, access denials
Designer: Service → Monitoring	Slow queries and script statistics

An error response from the server contains a `request_id` field; use it to find the matching line in the service journal.

9.2. Typical failures

Symptom	Cause	Action
The build stops while looking for hiredis or OpenSSL	libhiredis-dev, libssl-dev are not installed	Install the packages and run <code>install.sh</code> again
The service does not start; the journal shows a database connection error	PostgreSQL is unavailable for longer than <code>CRM_DB_WAIT</code> , or <code>CRM_DB</code> is wrong	Check PostgreSQL and the parameter file, start the service

Symptom	Cause	Action
Sign-in fails with an error about the <code>crypt</code> function	The <code>pgcrypto</code> extension is not enabled	Run <code>CREATE EXTENSION pgcrypto</code> in the database
A message about too many failed sign-in attempts	The lockout after failed sign-ins was triggered	Wait <code>CRM_LOGIN_LOCK_S</code> seconds or restart the service (without Redis the counter is kept in memory)
Menus and forms show <code>@t:...</code> keys	The server version is older than the application version	Update the server and the <code>web</code> directory to the kit version
A <code>column ... does not exist</code> error when a form is opened or a file is uploaded	The database schema does not match the server version	Apply <code>sql/schema.sql</code> and the application SQL files from the kit of the same version; if the error persists, contact iMobile support
Only two languages in the language selector	The browser still holds the previous shell	Reload the page; open the address with <code>?nosw=1</code>
Response 503 saying file storage is unavailable	The attachment directory is not accessible to the <code>formata</code> account	Check <code>CRM_UPLOAD_DIR</code> and directory permissions, restart the service
Response 413 when uploading a file	The file exceeds <code>CRM_STORAGE_MAX_MB</code> or the proxy limit	Raise both limits
A print form is not produced	The print libraries are missing or <code>CRM_PYLIBS</code> is not set	Check the <code>pylibs</code> directory and the variable, restart the service
A message that a script ran longer than the allowed time	<code>CRM_SCRIPT_TIMEOUT_MS</code> was exceeded	Fix the script or raise the limit
A script is marked DISABLED	Five failures within five minutes triggered automatic disabling	Fix the cause and re-enable the script with <code>Enable</code> again on the monitoring page
Notifications arrive late	The proxy buffers the event stream	Switch off buffering for the product address

Note. Many error messages produced by the server itself are in Russian regardless of the interface language.

For questions not covered by this guide, contact iMobile support.

Document history

Document version	Date	Product version	Change
1.0	2026-10-06	1.6.0	First issue