



PrintManager

Print supervision, accounting & control

User & Administrator Manual

Complete guide to installing, operating and deploying PrintManager — the print-accounting suite where no printed document leaves without being traced.

Version 1.0.0 · Software edition 1.0

English edition

Table of contents

This manual documents the graphical application and every companion component (collection agents, web supervision server, Windows service, licensing). The product interface is delivered in French; each screen is shown as you will actually see it, and every control is explained in English with its French label quoted for reference.

1. Introduction & overview	3
2. System requirements & dependencies	5
3. Installation	6
4. Licensing, trial & activation	8
5. The main window at a glance	10
6. Dashboard tab	11
7. Printers tab	13
8. History tab	15
9. Network (SNMP) tab	17
10. Reports tab	19
11. Settings tab	21
12. E-mail notifications (SMTP)	24
13. Remote supervision: web server & agents	25
14. Headless & Windows-service deployment	28
15. Command-line reference	29
16. Data storage, backup & maintenance	30
17. Choosing a deployment strategy	31
18. Building executables & the installer	32
19. Troubleshooting	33
20. Frequently asked questions	34
21. Appendix: glossary, UI labels & reference	35

1 · Introduction & overview

PrintManager is a print-accounting and control suite for Windows environments. Its guiding principle is simple: **no printed document leaves the organisation without being traced**. Every job is captured, attributed to a user and a machine, counted in pages, stored, and made available for dashboards, reports and long-term auditing.

1.1 Why three data sources

There is no single technique that reliably counts every printout in every situation, so PrintManager combines three complementary collectors and consolidates them into one database. Each answers a different question:

Collector	What it captures	Best used on
Spooler monitor moniteur_spooler.py	Every job the moment it passes through the Windows print spooler — document name, user, machine, pages and status (printed / cancelled / error), across all installed printers (USB, network, shared).	Live monitoring on any workstation or print server.
Windows event log compteur_eventlog.py	Pages actually printed, read from the Windows print journal (event 307), broken down by user, printer and document — the most reliable historical record on Windows.	Retrospective import, especially on a shared print server.
SNMP hardware counter compteur_snmp.py	The machine's own total page counter over the network, any brand, via the standard OID <code>prtMarkerLifeCount</code> — including photocopies made directly on the device with no PC involved.	Networked printers and copiers of any make.

Because all three feed the same SQLite tables, reports cross-reference them. The gap between what Windows recorded and what the hardware counter reports reveals any printing that bypassed the PC entirely — the key to genuinely complete traceability.

1.2 What the application gives you

- A six-tab desktop interface (Dashboard, Printers, History, Reports, Network, Settings) built on Tkinter — nothing extra to install for the UI itself.
- Live spooler supervision with real-time event feed, top-printer and top-user rankings.
- Full Windows print-queue management: pause / resume, purge, set default, test page, and per-job cancel / pause / resume.
- Hardware page-counter polling for network printers of any brand, with automatic discovery by CIDR scan and per-reading deltas for billing.
- Professional exports: on-screen charts, a six-sheet formatted Excel workbook, and a ready-to-circulate A4 PDF report.
- Usage alerts (large jobs, daily quota) and printer-incident alerts (offline, jam, out of paper, low toner) by e-mail.
- Remote supervision through a self-contained web dashboard and JSON API, fed by lightweight collection agents deployed on remote workstations.
- Headless operation as a native Windows service for 24/7 collection with no open session.
- A 14-day trial and hardware-bound licence activation.

1.3 How the pieces fit together

The graphical application federates the three original command-line scripts rather than duplicating their logic: a set of wrapper modules imports them, runs them on background threads, and writes their results to the shared database. The same database can be fed 24/7 by a headless service or by remote agents, so the interface — launched at any time — always finds the full history waiting for it.

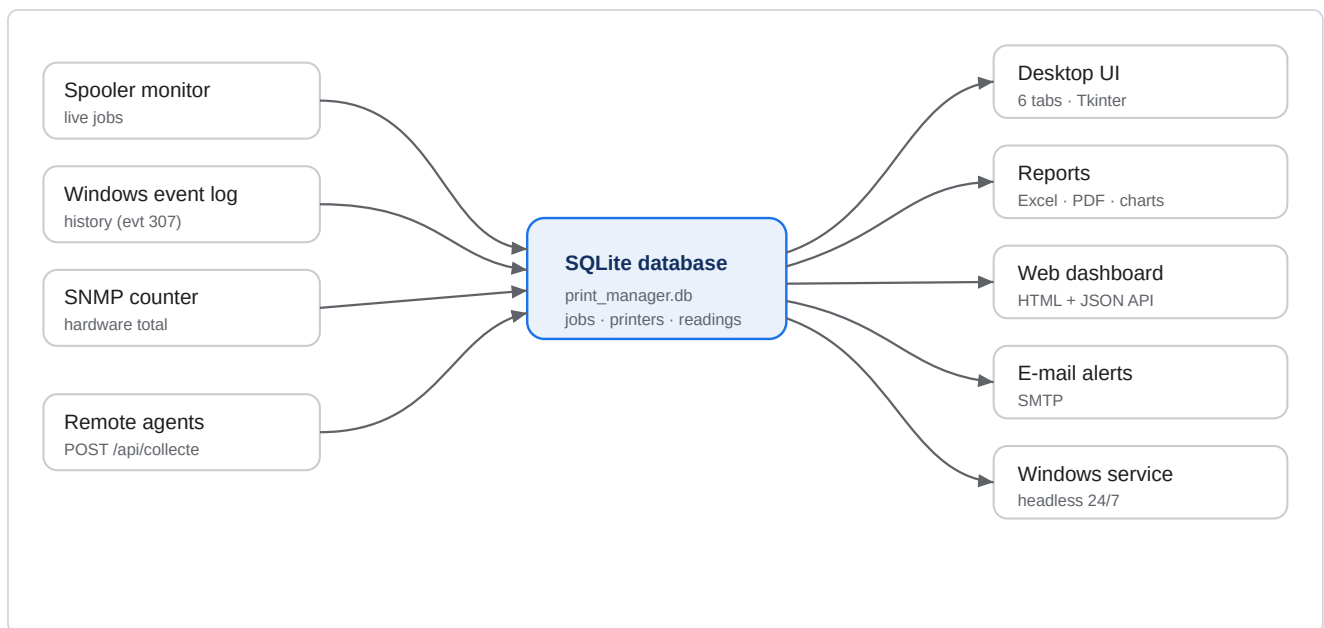


Figure 1 — All collectors feed one SQLite database; every consumer (interface, reports, web, e-mail, service) reads from it.

2 · System requirements & dependencies

PrintManager runs on Windows for full functionality. It also starts on Linux and macOS in a reduced mode where only the SNMP (network) features are operational and the interface remains usable. The application is designed to degrade gracefully: if an optional library is missing, the program still launches and the affected tab tells you exactly what to install.

2.1 Baseline

- **Operating system:** Windows 10 / 11 or Windows Server (2016 or later recommended) for spooler and event-log collection. SNMP works from any platform.
- **Python:** 3.9 or later, if you run from source. Tkinter and SQLite ship with the standard Python installer, so the interface needs nothing extra. (Packaged `.exe` builds embed their own Python — target machines need no Python at all.)
- **Privileges:** everyday supervision runs as a standard user. A few queue-management actions (set default printer, purge another user's jobs) may require running the application as administrator.

2.2 Dependency matrix

Install the two core libraries first, then add optional packages only for the features you need:

```
pip install pywin32 pysnmp
pip install matplotlib openpyxl reportlab    # reports (optional)
pip install flask                            # web supervision (server only)
```

Package	Required for	If absent...
<code>pywin32</code>	Spooler + Windows event log	Printers, live Dashboard feed and event-log import are disabled (Windows only).
<code>pysnmp</code>	Network (SNMP) counters	The Network tab is disabled and prompts you to install it.
<code>matplotlib</code>	Report charts	Reports tab shows a notice; charts are hidden but exports may still run.
<code>openpyxl</code>	Excel export	The Excel button is greyed out.
<code>reportlab</code>	PDF export	The PDF button is greyed out.
<code>flask</code>	Web supervision server	Remote web dashboard/API cannot start (agents unaffected).
<code>tkinter</code>	The graphical interface	Only headless mode (<code>--sans-ui</code>) is available.

Verify at any time

Run `python main.py --verifier` for a one-line diagnostic of every component — interface, spooler, SNMP, report libraries, web server and licence status.

3 · Installation

There are two ways to install PrintManager: from the Python source for evaluation and development, or from a packaged installer for production rollout. Choose whichever fits your situation.

3.1 From source (evaluation / development)

1. Copy the project folder to the target machine.
2. Install the core dependencies:

```
pip install pywin32 pysnmp
```

3. (Optional) add report and web libraries as needed:

```
pip install matplotlib openpyxl reportlab flask
```

4. Launch the interface:

```
python main.py
```

3.2 From the packaged installer (production)

For deployment across an organisation, PrintManager ships as three standalone executables that share one runtime folder and require no Python on the target machine:

Executable	Role
<code>PrintManager.exe</code>	Full graphical interface (no console window).
<code>PrintAgent.exe</code>	Collection agent for remote workstations (console).
<code>PrintService.exe</code>	Headless Windows collection service.

The Windows Setup program (built with Inno Setup — see chapter 18) offers, via check-boxes: a desktop shortcut, auto-launch of the interface at sign-in, installation and start of the collection service, and registration of the agent at start-up for remote workstations. On uninstall, the service is cleanly stopped and removed while the database and licence are kept by default, so a reinstall keeps your history and activation.

3.3 First launch & dependency check

The very first time you start the program, it opens the licence window in 14-day trial mode (chapter 4). After that, the main window appears. If any feature looks unavailable, run the diagnostic:

```
python main.py --verifier
```

```
=== Diagnostic PrintManager ===
[OK] tkinter (interface)
[OK] pywin32 (spouleur + journal Windows)
[OK] pysnmp (imprimantes réseau)
[OK] matplotlib (graphiques rapports)
[OK] openpyxl (export Excel)
[OK] reportlab (export PDF)
[OK] flask (supervision web à distance)
[OK] système Windows
[OK] licence (trial)
```

Where your data lives

Once installed under [Program Files](#) (read-only for standard users), the application stores its licence, trial state, [config.json](#) and the database in [%ProgramData%\PrintManager\](#) , a location writable by both the interface and the service. When run from source, everything stays in the project folder.

4 · Licensing, trial & activation

PrintManager uses a hardware-bound licence signed with HMAC-SHA256. A key is tied to one computer and cannot be forged without the vendor's secret, nor reused on another machine.

4.1 The 14-day trial

On first launch, the application starts a **free 14-day trial with every feature enabled**. A banner shows the days remaining and offers to continue the trial or activate a licence. As the end approaches, reminders appear; once the trial (or an activated licence) expires, the application locks and requires activation to continue.

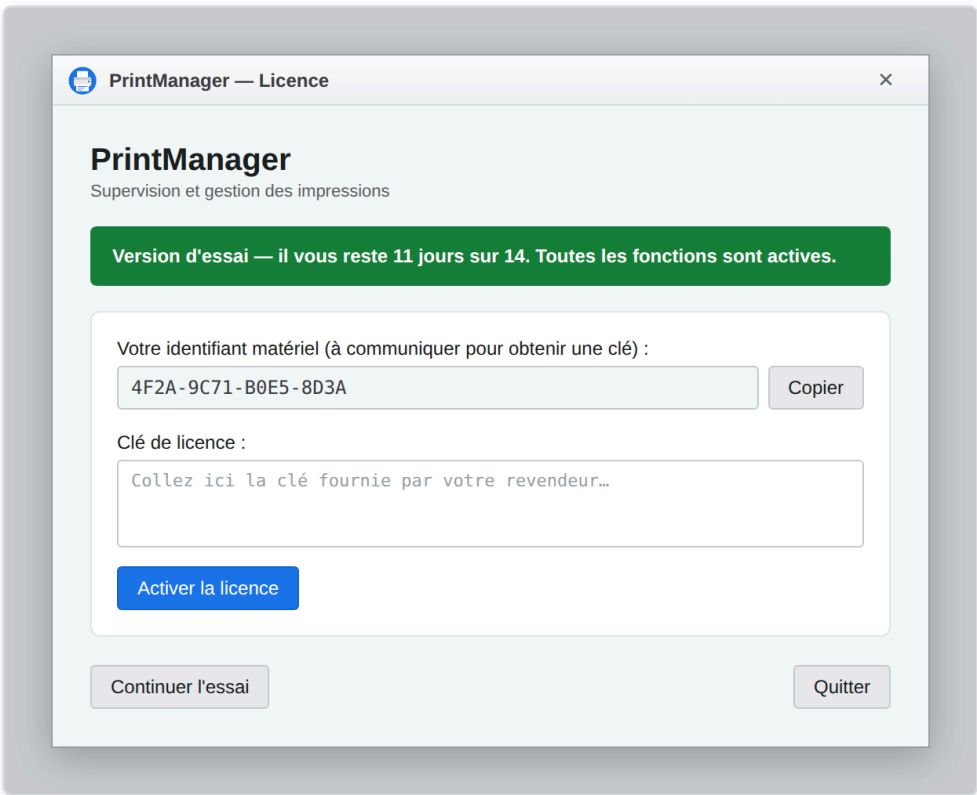


Figure 2 — The licence window in trial mode. The read-only field shows your hardware ID; the large field is where you paste the key supplied by your reseller.

4.2 Activating a licence

1. On the licence screen, read your **hardware ID** (French: *identifiant matériel*) and click **Copier** (Copy) to place it on the clipboard.
2. Send that ID to your reseller / vendor. They generate a key bound to your machine.
3. Paste the key into the **Clé de licence** (License key) box and click **Activer la licence** (Activate license).
4. On success, the confirmation appears and the main application opens. The key is stored with your data folder, so a reinstall keeps the activation.

Trial length	14 days, all features active
Binding	Cryptographically bound to the computer's hardware ID
On expiry	Application blocks until a valid key is activated
Copy hardware ID	"Copier" button on the licence screen
Continue trial	"Continuer l'essai" (while the trial is still valid)

For resellers

Keys are produced with an internal tool, `generateur_licence.py`, which is never distributed to customers. It records each issued key in `licences_emises.log`. Before your first distribution you must replace the shared secret `LICENCE_SECRET` in `licence.py` with your own random value and keep it private — anyone holding it can mint valid keys.

5 · The main window at a glance

The interface is organised as six tabs across the top, with a status bar along the bottom that always tells you what the background collectors are doing.

Tab (French label)	Purpose
Dashboard — <i>Tableau de bord</i>	Key figures for today and 30 days, a live event feed, and top-printer / top-user rankings.
Printers — <i>Imprimantes</i>	Manage the Windows print queue: printer state and actions, plus per-job actions.
History — <i>Historique</i>	Every stored job with filters, Windows event-log import and CSV export.
Reports — <i>Rapports</i>	On-screen charts and professional Excel / PDF exports.
Network (SNMP) — <i>Réseau (SNMP)</i>	Network printer fleet: add by IP, CIDR scan, hardware counters and reading history.
Settings — <i>Paramètres</i>	All configuration plus database maintenance.

5.1 The status bar

The bar at the bottom of every screen reports, from left to right, the spooler monitor state, the SNMP polling state, the web dashboard URL when the server is running, and the path to the active database. A typical line reads:

```
Spouleur : surveillance ACTIVE | SNMP : relevé auto ACTIF |  
Web : http://192.168.1.10:8080/ | Base : C:\ProgramData\PrintManager\print_manager.db
```

Spouleur : surveillance ACTIVE = spooler monitoring active · *relevé auto ACTIF* = automatic SNMP polling on · *Base* = database file.

Light & dark themes

The whole interface can switch between a light theme (*clair*) and a dark theme (*sombre*) from the Settings tab. The screenshots in this manual use the light theme; the theme change takes effect after restarting the application.

6 · Dashboard tab

The Dashboard (*Tableau de bord*) is the first screen you see and the pulse of the system. It combines instant key figures, a live feed of spooler activity, and two rankings.

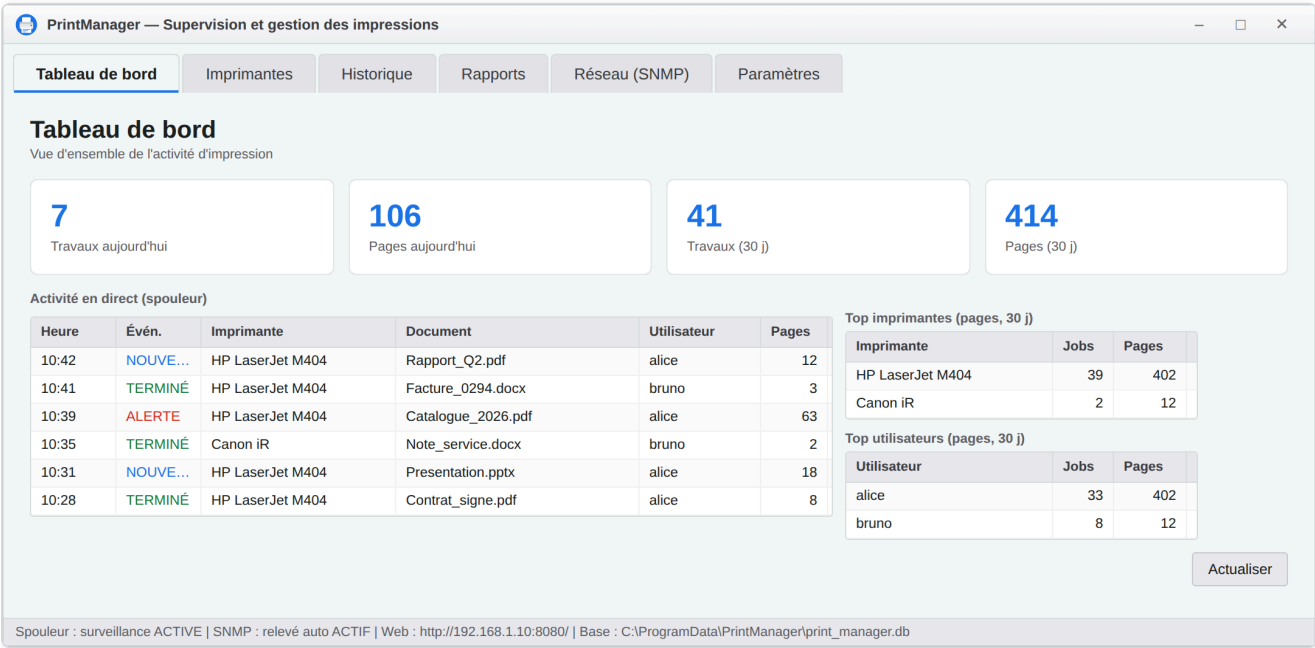


Figure 3 — The Dashboard: four key-figure cards, the live spooler feed on the left, and top-printer / top-user rankings on the right.

6.1 Key-figure cards

The four cards across the top summarise activity at a glance:

Card (French)	Meaning
Travaux aujourd'hui	Number of print jobs recorded today.
Pages aujourd'hui	Total pages printed today.
Travaux (30 j)	Jobs over the last 30 days.
Pages (30 j)	Pages over the last 30 days.

6.2 Live activity feed

The central table, *Activité en direct (spoleur)*, streams events straight from the spooler monitor as they happen. Each row is colour-coded by event type:

- **NOUVEAU** (blue) — a new job has just entered the spooler.
- **TERMINÉ** (green) — a job finished printing; the key figures refresh.
- **ALERTE** (red) — a job exceeded the "large job" page threshold set in Settings.

Columns show the time, event, printer, document, user and page count. The feed keeps the most recent 200 events on screen.

6.3 Rankings

On the right, *Top imprimantes* and *Top utilisateurs* list the busiest printers and users over the last 30 days, each with job and page totals — an immediate view of where your consumption concentrates. The **Actualiser** (Refresh) button at the bottom recomputes every figure on demand; the feed also refreshes automatically whenever a job completes.

Where the numbers come from

Dashboard figures are read from the shared database, so they include everything captured by the live monitor, imported from the Windows event log, and pushed by remote agents — a single consolidated picture.

7 · Printers tab

The Printers tab (*Imprimantes*) manages the Windows print fleet installed on this computer — local, USB, network and shared printers — and the jobs waiting in their queues.

The screenshot shows the 'Imprimantes' (Printers) tab in the PrintManager application. The interface includes a toolbar with buttons for 'Actualiser', 'Pause', 'Reprendre', 'Purger la file', 'Définir par défaut', and 'Page de test'. Below this is a table of installed printers with columns for 'Imprimante', 'État', 'File', 'Port', 'Pilote', and 'Défaut'. The first row, 'HP LaserJet M404', is highlighted in blue and marked as the default printer with a star. Other printers include 'Canon iR' (En pause), 'Microsoft Print to PDF' (Prête), and 'Brother HL-L2350' (Plus de papier). Below the printer list is a section for the selected printer's queue, 'File d'attente de l'imprimante sélectionnée', with buttons for 'Annuler le job', 'Pause job', and 'Reprendre job'. This section contains a table of jobs with columns for 'Job', 'Document', 'Utilisateur', 'Pages', and 'Statut'. Two jobs are listed: Job 204 (Rapport_Q2.pdf, 12 pages, En impression) and Job 205 (Annexe_budget.xlsx, 4 pages, En attente). The status bar at the bottom indicates 'Spouleur : surveillance ACTIVE | SNMP : disponible | Base : C:\ProgramData\PrintManager\print_manager.db'.

Imprimante	État	File	Port	Pilote	Défaut
HP LaserJet M404	Prête	1	192.168.1.50	HP Universal Print	★
Canon iR	En pause	0	192.168.1.51	Canon Generic Plus	
Microsoft Print to PDF	Prête	0	PORTPROMPT:	Microsoft Print To PDF	
Brother HL-L2350	Plus de papier	2	USB001	Brother HL-L2350DW	

Job	Document	Utilisateur	Pages	Statut
204	Rapport_Q2.pdf	alice	12	En impression
205	Annexe_budget.xlsx	alice	4	En attente

Figure 4 — The Printers tab: the fleet with live state on top (the highlighted row is selected), its queue and per-job actions below.

7.1 The fleet list

Each printer row shows its name, current **state** (*État*: ready, paused, jam, out of paper, low toner...), the number of jobs in its queue (*File*), its port, its driver, and a star in the **Défaut** column marking the default printer. Paused printers are shown in red.

Printer actions (top toolbar)

Button (French)	Action
Actualiser	Refresh the fleet and queue.
Pause / Reprendre	Pause or resume the selected printer.
Purger la file	Cancel all pending jobs on the printer (asks for confirmation).
Définir par défaut	Make the selected printer the Windows default.
Page de test	Send a Windows test page to the selected printer.

7.2 The selected printer's queue

The lower table, *File d'attente de l'imprimante sélectionnée*, lists the jobs of the printer selected above — job ID, document, user, pages and status — and refreshes every few seconds. The second toolbar acts on the selected job:

Button (French)	Action
Annuler le job	Cancel the selected job.
Pause job / Reprendre job	Hold or release the selected job.

Administrator rights

Some queue actions — setting the default printer, or purging / cancelling jobs owned by other users — require elevated rights. If an action fails, the application suggests relaunching "as administrator". Start `PrintManager.exe` with *Run as administrator* when you need full control of a shared print server.

8 · History tab

The History tab (*Historique*) is the searchable record of every job stored in the database — whether captured live by the spooler monitor, imported from the Windows event log, or pushed by remote agents. This is where day-to-day auditing happens.

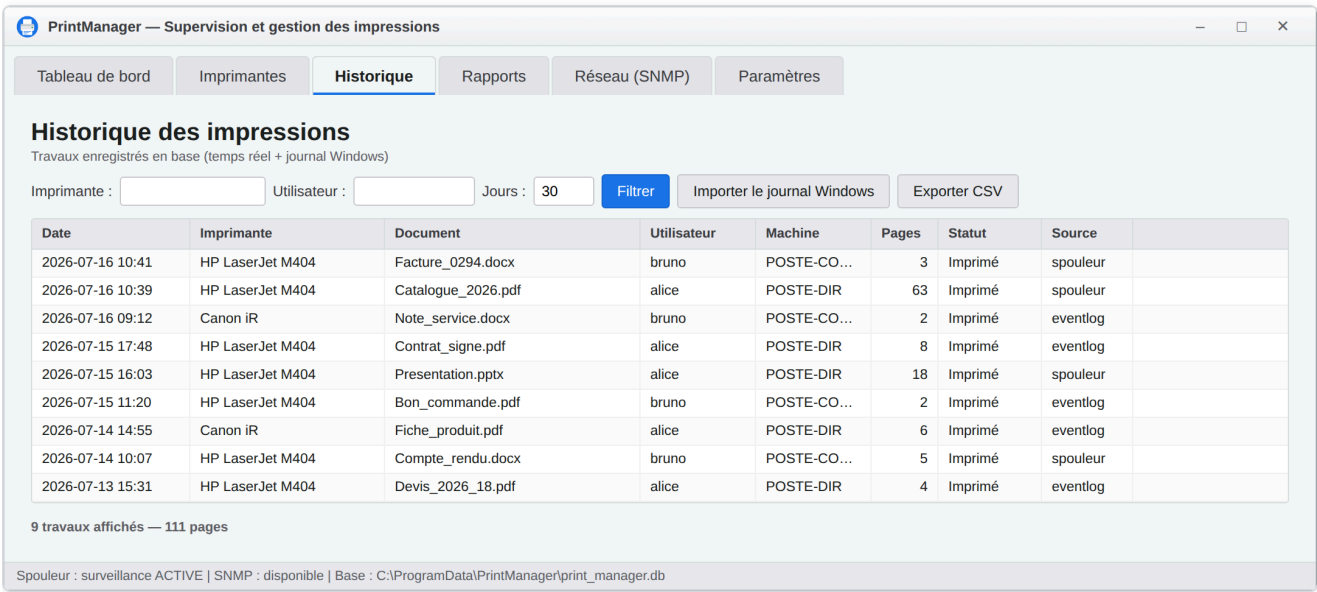


Figure 5 — The History tab: filters and action buttons on top, the job table below, and a running total in the summary line.

8.1 Filtering

The filter row lets you narrow the list before displaying it:

Field (French)	Effect
Imprimante	Restrict to a printer (partial name matches).
Utilisateur	Restrict to a user.
Jours	Look-back window in days (0 = all history).
Filtrer	Apply the filters and reload the table.

Each row shows the date, printer, document, user, machine, page count, status and the **source** of the record (*spouleur* = live monitor, *eventlog* = Windows journal). The summary line beneath reports how many jobs are shown and their total pages.

8.2 Importing the Windows event log

Click **Importer le journal Windows** (Import Windows log) to pull historical jobs from the Windows print journal into the database. The import runs in the background and de-duplicates automatically, so you can repeat it safely. The look-back depth is the value set in Settings.

Enable the print journal once

The Windows print journal must be switched on (once) in an **administrator**

PowerShell before import will find anything:

```
wevtutil sl Microsoft-Windows-PrintService/Operational /e:true
```

Run on a Windows

print server

, the event-log import counts jobs from every workstation on the network that prints through that server — centralised accounting with a single import.

8.3 Exporting to CSV

Click **Exporter CSV** to save the currently filtered list as a semicolon-separated, UTF-8 CSV file (columns: Date, Printer, Document, User, Machine, Pages, Status, Source). This is ideal for spreadsheets or feeding another accounting tool. For formatted Excel and PDF, use the Reports tab (chapter 10).

9 · Network (SNMP) tab

The Network tab (*Réseau (SNMP)*) manages networked printers and copiers as physical devices, reading their built-in page counters directly over the network. Because it uses the standard OID `prtMarkerLifeCount` (1.3.6.1.2.1.43.10.2.1.4.1.1, RFC 3805), it works with virtually any brand — HP, Canon, Epson, Brother, Ricoh, Kyocera, Xerox and more — and counts even the copies made directly on the machine with no PC involved.

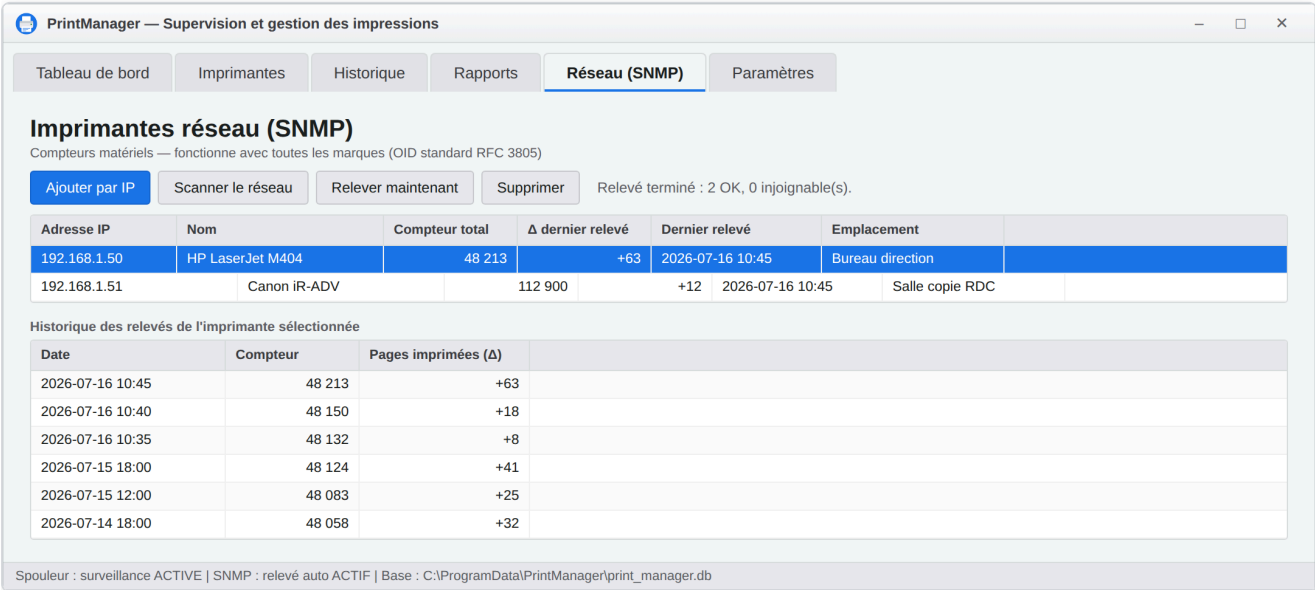


Figure 6 — The Network tab: the declared fleet with total counters and per-reading deltas on top, and the reading history of the selected printer below.

9.1 Building the fleet

Button (French)	Action
Ajouter par IP	Add one printer by its IP address; the device is queried immediately and, if it answers, added with a first reading.
Scanner le réseau	Discover printers automatically across a CIDR range (for example <code>192.168.1.0/24</code>). A full scan can take a minute or two.
Relever maintenant	Poll the counters of every declared printer now.
Supprimer	Remove the selected printer from the fleet (historical readings are kept).

9.2 Counters & deltas

The fleet table shows each printer's IP, name, **total counter** (*Compteur total*), the change since the previous reading (*Δ dernier relevé*), the timestamp of the last reading, and its location. The delta is the number of pages the machine physically produced between two readings — the figure you use for billing or consumption tracking.

9.3 Reading history

Select a printer to see its full reading history below (*Historique des relevés*): date, counter value and pages printed since the previous reading. Over time this builds an authoritative record of real output per device.

Requirements & automation

SNMP v1/v2c must be enabled on the printer (almost always the default, community [public](#)). You can change the community and the default scan range in Settings, and enable

automatic periodic polling

so counters are read on a schedule without any manual action.

10 · Reports tab

The Reports tab (*Rapports*) turns the accumulated data into charts on screen and into professional, shareable documents. All three data sources are cross-referenced, and exports are generated on a background thread so the interface never freezes.

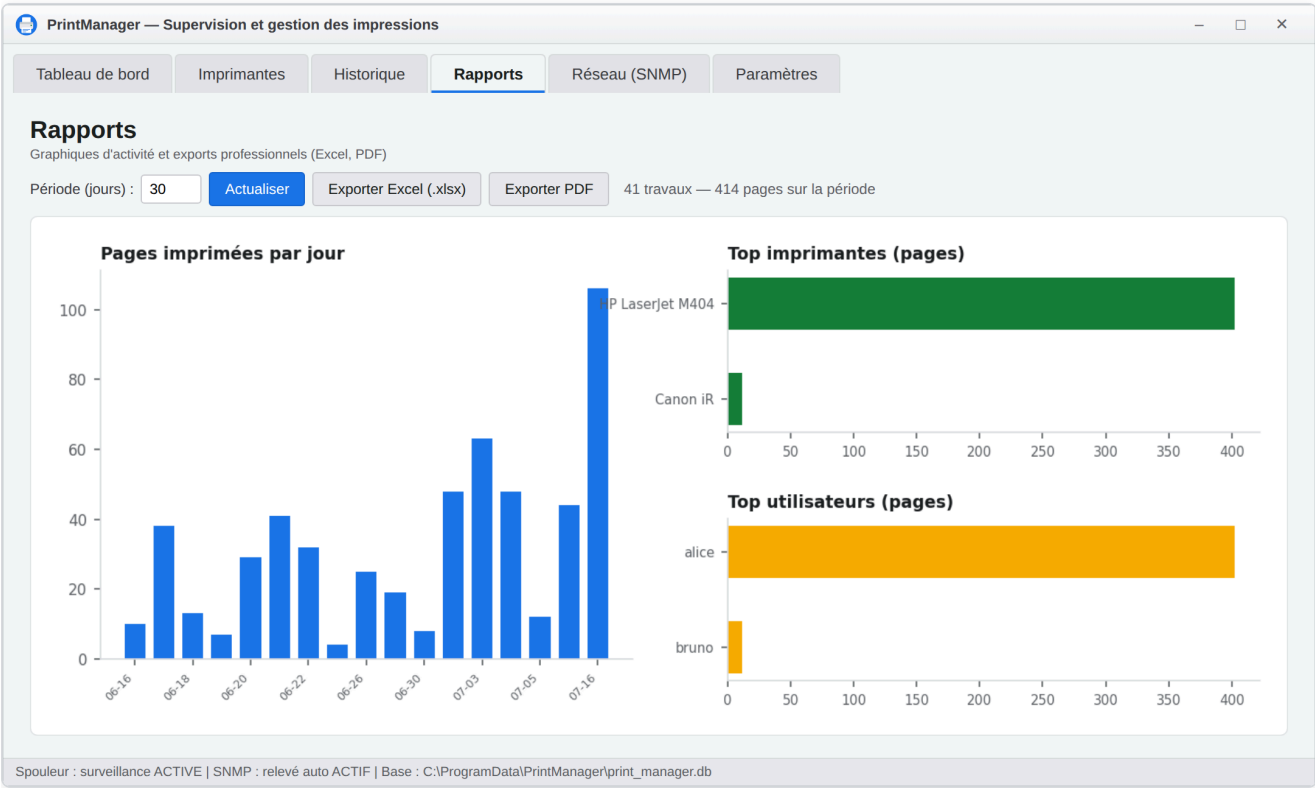


Figure 7 — The Reports tab: choose a period, view the three charts, and export a formatted Excel workbook or a ready-to-circulate PDF.

10.1 On-screen charts

Set the look-back window with **Période (jours)** and click **Actualiser** (Refresh). Three charts, drawn in the theme colours, appear: pages printed per day, top printers, and top users. The summary text reports the jobs and pages over the chosen period.

10.2 Excel export (.xlsx)

Exporter Excel (.xlsx) produces a formatted six-sheet workbook, complete with coloured headers, frozen panes, auto-filters and native Excel charts you can edit in Excel:

Sheet (French)	Contents
Synthèse	Key figures for the period.
Par jour	Pages and jobs per day.
Par imprimante	Totals per printer.
Par utilisateur	Totals per user.
Compteurs SNMP	Hardware counter readings.
Détail complet	The full job-by-job detail.

10.3 PDF export

Exporter PDF produces a print-ready A4 report: key figures, the three charts, printer and user rankings, SNMP counters, and a numbered, dated footer — ideal for circulating to management or attaching to an invoice.

Optional libraries

Charts need [matplotlib](#) ; Excel needs [openpyxl](#) ; PDF needs [reportlab](#) . If any is missing, the tab tells you exactly what to install and disables only the affected button — everything else keeps working.

11 · Settings tab

The Settings tab (*Paramètres*) gathers every configurable option in one place. Changes are written to `config.json` when you click **Enregistrer** (Save), and the background monitors restart immediately to match. **Réinitialiser les défauts** restores factory defaults.

PrintManager — Supervision et gestion des impressions

Tableau de bord

Imprimantes

Historique

Rapports

Réseau (SNMP)

Paramètres

Paramètres

Le `config.json` est enregistré dans `config.json`

☒ Démarrer la surveillance au lancement

Intervalle de scrutation (s) :

Journal d'événements Windows

Profondeur d'import par défaut (jours) :

Activation du journal (PowerShell admin) :
wevtutil sl Microsoft-Windows-PrintService/Operational /e:true

Alertes et quotas

alerter si un job dépasse (pages, 0 = off) :

Quota total de pages / jour (0 = off) :

Notifications e-mail (SMTP)

☒ Alertes par e-mail (jobs volumineux, quota)
☒ Incidents imprimantes (hors ligne, toner...)

Serveur SMTP :

Port : Utilisateur :

Mot de passe :

Destinataires :

☒ STARTTLS (587) — décochez pour SSL (465)

Envoyer un e-mail de test

Imprimantes réseau (SNMP)

Communauté SNMP :

Plage de scan par défaut :

☒ Relevé périodique automatique des compteurs

Intervalle des relevés (s) :

Supervision à distance (web)

☒ Activer le serveur web au lancement

Port :

Jeton (collecte des agents) :

Tableau de bord : (l'arrêt du serveur prend effet au redémarrage)

Interface

Thème :

(le thème s'applique au redémarrage)

Maintenance de la base

Fichier : C:\ProgramData\PrintManager\print_manager.db

Vider l'historique des jobs

Vider les relevés SNMP

Enregistrer

Réinitialiser les défauts

Spoleur : surveillance ACTIVE | SNMP : relevé auto ACTIF | Web : http://192.168.1.10:8080/ | Base : C:\ProgramData\PrintManager\print_manager.db

Figure 8 — The Settings tab groups every option into labelled sections across two columns, with Save / Reset at the bottom.

11.1 Real-time monitoring & the Windows journal

Démarrer la surveillance au lancement	Start spooler monitoring automatically when the app opens.
Intervalle de scrutation (s)	Seconds between spooler polls (default 2.0).
Profondeur d'import (jours)	Default look-back for the event-log import (default 30).

11.2 Alerts & quotas

Alerter si un job dépasse (pages)	Warn when a single job exceeds N pages (0 = off; default 50). Triggers the red ALERTE feed entry and, optionally, an e-mail.
Quota total de pages / jour	Daily page budget for the whole site (0 = off). When reached, the app warns you and you can pause printers from the Printers tab.

11.3 Network (SNMP)

Communauté SNMP	SNMP community string (default <code>public</code>).
Plage de scan par défaut	Default CIDR range for network discovery.

Relevé périodique automatique	Enable scheduled counter polling.
Intervalle des relevés (s)	Seconds between automatic readings (default 300).

11.4 Interface & database maintenance

Thème	Light (<i>clair</i>) or dark (<i>sombre</i>); applied on restart.
Vider l'historique des jobs	Permanently clear the jobs table.
Vider les relevés SNMP	Permanently clear the SNMP readings table.

The two remaining sections — **E-mail notifications** and **Remote supervision (web)** — are covered in chapters 12 and 13.

Maintenance is irreversible

"Vider..." permanently deletes rows from the database. Take a copy of `print_manager.db` first if you might need the data later (chapter 16).

12 · E-mail notifications (SMTP)

PrintManager can e-mail alerts so administrators are informed without watching the interface. Configuration lives in the *Notifications e-mail (SMTP)* section of the Settings tab.

12.1 Setting up SMTP

Serveur SMTP	e.g. <code>smtp.office365.com</code> , <code>smtp.gmail.com</code>
Port	587 for STARTTLS, or 465 for SSL
Utilisateur / Mot de passe	The sending account's credentials
Destinataires	One or more recipients, comma-separated
STARTTLS (587)	Keep ticked for port 587; untick for SSL on 465

Use the **Envoyer un e-mail de test** (Send a test e-mail) button to confirm the settings before relying on them.

12.2 The two families of alert

- **Usage alerts** (*Alertes par e-mail*) — a job exceeding the large-job page threshold, or the daily quota being reached.
- **Printer incidents** (*Incidents imprimantes*) — offline, error, jam, out of paper, low or empty toner, door open. One e-mail is sent when the problem appears and one when it clears, with nothing in between, so you are never spammed.

Password storage

The SMTP password is protected on disk. On Windows it is encrypted with DPAPI (machine scope) so both the interface and the service can read it locally, but it cannot be read if `config.json` is copied to another machine. Even so, prefer a dedicated account or an "app password" rather than a primary mailbox password.

13 · Remote supervision: web server & agents

For sites with several workstations, PrintManager offers a central web dashboard fed by lightweight agents. A central machine (a print server or the administrator's PC) runs the supervision server and holds the central database; each remote workstation runs an agent that pushes its completed jobs to that server.

13.1 Starting the server

There are two ways to run the supervision server:

- **Integrated** — in Settings → *Supervision à distance (web)*, tick *Activer le serveur web*, choose a port and set a **token**, then Save. The URL appears in the status bar.
- **Standalone / headless** — `python serveur_web.py` or `python main.py --web`, ideal on a server alongside the Task Scheduler.

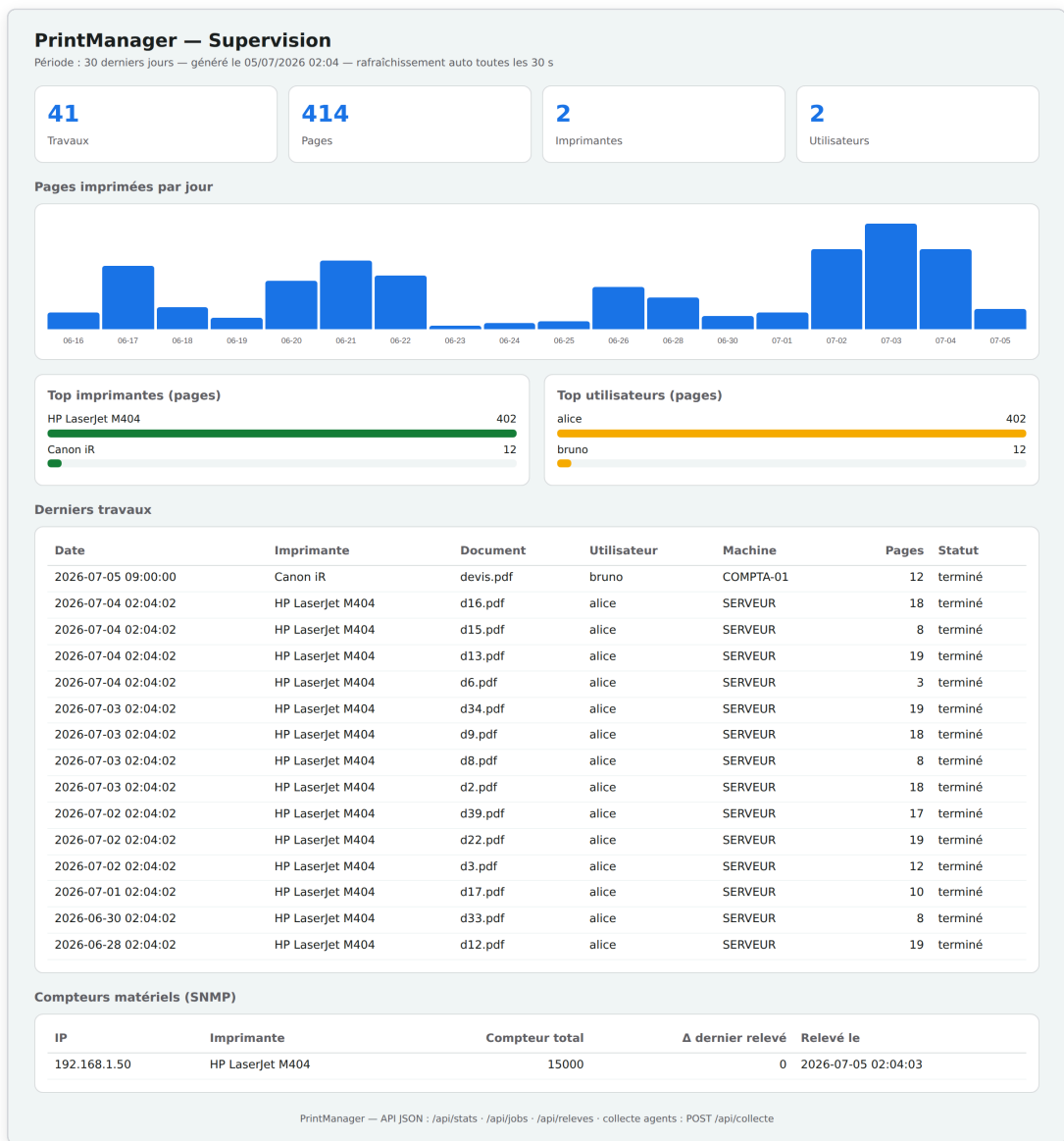


Figure 9 — The self-contained web dashboard (<http://SERVER:8080/>): key figures, pages-per-day, top printers and users, latest jobs and SNMP counters, refreshed every 30 seconds.

13.2 The web dashboard

Open <http://SERVER-IP:8080/> in any browser on the LAN. The page shows key figures, pages per day, top printers and users, the latest jobs and SNMP counters, and refreshes automatically every 30 seconds. It is 100%

self-contained — the charts are pure CSS and no internet resources are required. Append `?jours=90` to change the period.

13.3 The JSON API

Read-only endpoints let you plug PrintManager into an existing monitoring tool:

Endpoint	Returns
<code>/api/stats</code>	Key figures and rankings.
<code>/api/jobs?limite=&jours=&imprimante=&utilisateur=</code>	Filtered job list.
<code>/api/relevés</code>	SNMP readings.
<code>/api/santé</code>	Health check.

13.4 Deploying agents

On each workstation whose printing you want centralised, run:

```
python agent_collecte.py --serveur http://192.168.1.10:8080 --jeton MYTOKEN
```

The agent watches the local spooler and pushes each completed job to the central database. If the network drops or the server is down, jobs are buffered in `agent_tampon.json` and re-sent automatically — nothing is lost — and server-side de-duplication means you can restart the agent without creating duplicates. For production, start it automatically via Task Scheduler ("at sign-in") or install it as a service with `nssm`. In packaged form the agent is `PrintAgent.exe`.

Security

Every write (collection) requires the header `X-Jeton` equal to the server's configured token; with no token set, collection is refused. This tool is designed for a trusted local network — do not expose it directly on the internet. Place it behind a reverse proxy with HTTPS and authentication if remote access is required.

14 · Headless & Windows-service deployment

For continuous, unattended collection, PrintManager can run without the interface — either as a one-off console process or as a native Windows service that survives reboots and needs no open session. Both write to the same database, so the interface, launched later, finds the full history.

14.1 Console (headless) mode

```
python main.py --sans-ui
```

This starts spooler monitoring in the console and logs each event to the database. Useful for a quick unattended run or inside a scheduled task. Press **Ctrl + C** to stop.

14.2 Native Windows service

`service_windows.py` runs the whole collection stack — spooler, SNMP, web server and e-mail, according to `config.json` — as a system service, with no user logged in, against the same database as the interface. From an **administrator** command prompt:

```
python service_windows.py install
python service_windows.py start      (stop / remove / debug)
sc config PrintManagerCollecte start= auto
```

Once packaged, the same commands apply to the executable:

```
PrintService.exe install
PrintService.exe start
PrintService.exe stop
PrintService.exe remove
```

Recommended production layout

Install the service on the print server for round-the-clock centralised collection, enable the web server for supervision, and let administrators open the interface on demand to review history and generate reports.

15 · Command-line reference

15.1 The application entry point

Command	Purpose
<code>python main.py</code>	Launch the graphical interface.
<code>python main.py --verifier</code>	Print a dependency & licence diagnostic, then exit.
<code>python main.py --sans-ui</code>	Headless spooler monitoring (service / scheduled task).
<code>python main.py --web</code>	Run only the web supervision server.

15.2 The standalone collector scripts

The three original scripts remain usable on their own from the command line:

```
# Windows event-log counting (history)
python compteur_eventlog.py                # entire history
python compteur_eventlog.py --jours 30      # last 30 days
python compteur_eventlog.py --imprimante "HP" # filter by printer
python compteur_eventlog.py --csv rapport.csv # detailed CSV export

# Real-time spooler monitoring + cumulative CSV log
python moniteur_spooler.py
python moniteur_spooler.py --intervalle 1 --csv journal.csv

# SNMP hardware counters, any brand
python compteur_snmp.py --ip 192.168.1.50      # single reading
python compteur_snmp.py --scan 192.168.1.0/24  # auto-discovery
python compteur_snmp.py --ip 192.168.1.50 --suivi 300 # poll every 5 min + deltas
```

15.3 Companion components

```
# Web supervision server
python serveur_web.py

# Remote collection agent
python agent_collecte.py --serveur http://192.168.1.10:8080 --jeton MYTOKEN

# Windows service (administrator prompt)
python service_windows.py install | start | stop | remove | debug
```

16 · Data storage, backup & maintenance

16.1 Where everything is stored

All state lives in a small set of files. When the application is installed under `Program Files`, they are placed in `%ProgramData%\PrintManager\`; when run from source, they stay in the project folder.

File	Contents
<code>print_manager.db</code>	SQLite database: printers, jobs, SNMP readings.
<code>config.json</code>	All settings; SMTP password and web token are encrypted (DPAPI on Windows).
<code>agent_tampon.json</code>	An agent's offline buffer of unsent jobs.
Licence / trial files	Activation and trial state, bound to the machine.

16.2 The database schema

The database holds three tables that all three collectors feed:

Table	Holds
<code>imprimantes</code>	The declared fleet — local Windows printers and network (SNMP) devices.
<code>jobs</code>	Every captured job: date, printer, document, user, machine, pages, status and source (<i>spouleur</i> or <i>eventlog</i>).
<code>relevés_snmp</code>	Periodic hardware-counter readings with per-reading deltas.

16.3 Backup

To back up the system, copy `print_manager.db` and `config.json` while the application is closed (or the service stopped). To migrate to another machine, copy the database — but reconfigure secrets, because DPAPI-encrypted values cannot be decrypted on a different computer.

16.4 Housekeeping

The Settings tab offers **Vider l'historique des jobs** and **Vider les relevés SNMP** to clear the respective tables. Both are permanent; back up first if in doubt. SQLite handles concurrent writes with a file lock, so the monitoring threads, the service and the interface can all run against the same database safely.

17 · Choosing a deployment strategy

Because the three collectors answer different questions, the right combination depends on your environment. Use this table as a starting point:

Environment	Recommended approach
Fleet behind a Windows print server	Event-log import on the server gives centralised per-user accounting for every workstation that prints through it. Add the service for continuous collection.
Standalone workstations	Run the live monitor (or event-log import) on each machine, or deploy the collection agent to a central server.
Networked copiers / printers used without a PC	SNMP polling is the only method that captures copies made directly on the device.
Complete control / audit	Combine "who printed what" (event log / spooler) with "real total output" (SNMP). The gap between the two reveals any printing that bypassed Windows.

The point of combining sources

The Windows journal tells you the document's page count; the SNMP counter is authoritative for the real total the machine produced (some drivers report pages × copies differently). Reconciling the two is what makes traceability genuinely complete.

17.1 A note on responsible use

PrintManager is intended for administering your own printer fleet. Monitoring users' printing must respect local regulations — in particular, informing employees where required. Configure retention and alerting in line with your organisation's policy.

18 · Building executables & the installer

This chapter is for whoever packages PrintManager for distribution; end users can skip it.

18.1 Building the executables (PyInstaller)

In the `empaquetage\` folder, double-click `construire.bat` (or run `pyinstaller --noconfirm --clean PrintManager.spec`). The result lands in `empaquetage\dist\PrintManager\` as three executables that share one `_internal` library folder: `PrintManager.exe` (GUI), `PrintAgent.exe` (agent) and `PrintService.exe` (service). No Python is required on target machines.

18.2 Compiling the Setup (Inno Setup)

1. Build the executables first (above).
2. Install Inno Setup (free): <https://jrsoftware.org/isdl.php>.
3. Open `empaquetage\installateur.iss` in Inno Setup.
4. Choose **Build** → **Compile** (F9).
5. The installer is produced at `empaquetage\sortie\PrintManager_Setup_1.0.0.exe`.

The generated installer offers, by check-box: a desktop shortcut, auto-launch of the interface at sign-in, installation and start of the collection service, and registration of the agent at start-up for remote workstations (remember to edit the server URL and token in the agent's `Run` registry command afterwards). Uninstalling stops and removes the service cleanly; the database is kept by default (adjustable in the `[UninstallDelete]` section of the `.iss`).

Before your first distribution

Replace `LICENCE_SECRET` in `licence.py` with your own random value (32+ bytes) and never share it. The licence generator imports the same secret, so change it in one place. The generator is not included in the Setup, so the secret does not ship to customers — but it remains inside the executables, so keep control of your distribution.

19 · Troubleshooting

Symptom	Likely cause & fix
Printers / Dashboard feed empty; "pywin32 not installed" notice	Install pywin32 (<code>pip install pywin32</code>) and ensure you are on Windows.
Event-log import finds nothing	The print journal is off. Enable it once in an admin PowerShell: <code>wevtutil sl Microsoft-Windows-PrintService/Operational /e:true</code> , then re-import.
A printer doesn't answer over SNMP	Check the IP, that SNMP v1/v2c is enabled on the device, and that the community matches (default <code>public</code>) in Settings.
Network tab disabled	Install pysnmp (<code>pip install pysnmp</code>).
Report buttons greyed out	Install the matching library: <code>matplotlib</code> (charts), <code>openpyxl</code> (Excel), <code>reportlab</code> (PDF).
"Action impossible... try administrator"	Queue actions on shared/other users' jobs need elevation. Relaunch <code>PrintManager.exe</code> as administrator.
Web dashboard unreachable	Confirm the web server is enabled and Flask is installed (<code>pip install flask</code>); check the port and any firewall rule; use the exact URL in the status bar.
Agent seems idle	Verify the server URL and that the <code>--jeton</code> matches the server token; queued jobs are held in <code>agent_tampon.json</code> until the server is reachable.
Test e-mail fails	Re-check server, port and TLS/SSL choice (587 STARTTLS vs 465 SSL) and the account credentials; try an app password.
Application locked at start-up	Trial or licence expired — activate a valid key (chapter 4).
Interface won't start (Tkinter missing)	Reinstall Python with tcl/tk ticked, or run headless with <code>python main.py -s -sans-ui</code> .

20 · Frequently asked questions

Does anything print without being recorded?

That is exactly what the combination of collectors prevents. The spooler monitor and event-log import record what Windows prints; SNMP records the machine's true total, including walk-up copies. The difference between them surfaces anything that bypassed the PC, so every page is accounted for.

Can it count printing for a whole office from one place?

Yes. Run the event-log import (or service) on a Windows print server to centralise every workstation that prints through it, and/or deploy collection agents that push jobs to the central web server.

Does it work with our brand of printer?

For counters, yes — SNMP uses the standard RFC 3805 OID supported across HP, Canon, Epson, Brother, Ricoh, Kyocera, Xerox and others. Windows queue management works with any printer installed on the machine.

Do remote workstations need Python?

No. Distribute `PrintAgent.exe`; it is self-contained.

Is a job lost if the network drops?

No. Agents buffer unsent jobs in `agent_tampon.json` and resend automatically, and the server de-duplicates, so restarts never create duplicates.

Where are settings and history kept?

In `%ProgramData%\PrintManager\` when installed (`config.json` and `print_manager.db`), or in the project folder when run from source.

Can I run it without the interface?

Yes — headless with `--sans-ui`, or as a Windows service for 24/7 collection. The interface, opened later, reads the same database.

Is the web dashboard safe to expose to the internet?

No. It is built for a trusted LAN. Writes require a shared token, but for remote access put it behind a reverse proxy with HTTPS and authentication.

The interface is in French — will an English build be provided?

The current release ships a French interface. This manual documents every control in English with the French label quoted, so an English-speaking administrator can operate it fully. Use the appendix glossary as a quick reference.

21 · Appendix

A · French ↔ English UI glossary

French (on screen)	English
Tableau de bord	Dashboard
Imprimantes	Printers
Historique	History
Rapports	Reports
Réseau (SNMP)	Network (SNMP)
Paramètres	Settings
Actualiser	Refresh
Enregistrer	Save
Réinitialiser les défauts	Reset to defaults
Pause / Reprendre	Pause / Resume
Purger la file	Purge the queue
Définir par défaut	Set as default
Page de test	Test page
Annuler le job	Cancel the job
Filtrer	Filter
Importer le journal Windows	Import the Windows log
Exporter CSV / Excel / PDF	Export CSV / Excel / PDF
Ajouter par IP	Add by IP
Scanner le réseau	Scan the network
Relever maintenant	Read now
Compteur total	Total counter
Dernier relevé	Last reading
Emplacement	Location
Essai / Licence	Trial / License
Identifiant matériel	Hardware ID
Activer la licence	Activate the license
Continuer l'essai	Continue the trial
Prête / En pause / Bourrage / Plus de papier	Ready / Paused / Jam / Out of paper

B · Technical reference

SNMP OID (page counter)	1.3.6.1.2.1.43.10.2.1.4.1.1 (prtMarkerLifeCount, RFC 3805)
--------------------------------	---

SNMP version / community	v1 / v2c, default community <code>public</code>
Windows print journal	<code>Microsoft-Windows-PrintService/Operational</code> , event 307
Default web port	8080 (host 0.0.0.0 = whole LAN)
Collection auth header	<code>X-Jeton</code> = configured web token
Service name	<code>PrintManagerCollecte</code>
Licence signature	HMAC-SHA256, hardware-bound
Database engine	SQLite (<code>print_manager.db</code>)

C · Key files

File	Role
<code>main.py</code>	Application entry point.
<code>config.py</code> / <code>config.json</code>	Configuration & persistence.
<code>database.py</code>	SQLite persistence layer.
<code>ui/</code>	The six tabs, theme and licence dialog.
<code>utils/</code>	Wrappers bridging the collector scripts to the UI.
<code>serveur_web.py</code> / <code>agent_collecte.py</code>	Web server & remote agent.
<code>service_windows.py</code>	Native Windows service.

PrintManager — User & Administrator Manual · Version 1.0.0 · English edition. Product interface language: French. © The PrintManager author. All product and company names are the property of their respective owners.