AP 3-Way Match

Minimum Specification & Installation Guide

What your computer needs, and step-by-step setup for both the Personal and Server editions - Windows one-click installer or manual Docker setup.

12 October 2026

Table of Contents

Minimum & Recommended Specification 3

Personal edition vs Server edition - what's different . . . 3

Installation & Deployment 5

Which path is right for you? . . . 5

Option A — Windows: the one-click installer . . . 5

Option B — Manual setup (Mac, Linux, or Windows by hand) . . . 7

Everyday operations . . . 11

Backing up your data . . . 12

Troubleshooting . . . 12

Minimum & Recommended Specification

This is the baseline a computer needs before installing either edition. There is no separate spec for Personal vs Server beyond what's called out in the comparison table below — the underlying software (Docker Desktop, the application image, the database) is the same either way; what differs is whether that database is PostgreSQL (Server) or a single SQLite file (Personal).

Component	Minimum	Recommended
Operating System	Windows 10/11 (64-bit), or a 64-bit Linux/macOS for the manual (Docker) setup	Windows 11 64-bit, fully updated
CPU	Any 64-bit CPU from the last ~8 years with hardware virtualization support	4+ modern cores
RAM	4 GB free (Docker Desktop's own floor)	8 GB+ free - more if you also run a local AI model (see below)
Disk space	5 GB free (application image, Docker's own files, database, logs)	20 GB+ free, on an SSD
Network	Internet connection for one-time setup (downloading Docker and building the app image)	A wired or reliable Wi-Fi connection if several people will reach this over your office network
Browser	Any current version of Chrome, Edge, Firefox, or Safari	Same - keep it updated
Virtualization	Must be enabled in the computer's BIOS/UEFI (required by Docker Desktop/WSL2)	Already on by default on most machines bought in the last few years

Do: treat "4 GB RAM" as the bare floor for Docker Desktop to start at all, not a comfortable running size. Once the app and its database are both actually running and processing invoices, 8 GB is a much safer baseline - especially on the Server edition, where PostgreSQL keeps its own memory cache.
Don't: plan to run a local AI model (Ollama, LM Studio, vLLM - see the AI Provider chapter in the User Manual) on the same minimum-spec machine without extra RAM/VRAM budgeted specifically for that model on top of everything above - a vision-capable local model typically wants several GB of its own, sometimes a GPU, depending on which one you pick.

Personal edition vs Server edition - what's different

Both editions are the same application, the same screens, and the same workflow once you're signed in - the User Manual is identical either way. What differs is entirely about where your data lives and how many people can use it at once:

	Personal edition	Server edition
Who it's for	One person, on their own laptop or desktop	Several people, reaching it over a network, at the same time
Windows installer	`AP3WayMatchSetup-Personal.exe`	`AP3WayMatchSetup.exe`
Where data lives	One file, `exceptions.db`, inside the install folder	A PostgreSQL database server, running inside Docker
Default install folder	`C:\ProgramData\AP3WayMatchPersonal`	`C:\ProgramData\AP3WayMatch`
Backing up your data	Copy the one `exceptions.db` file (app stopped first)	`docker compose exec db pg_dump ...` (see Backing up your data, below)
Uninstall behavior	No extra prompt - the database file is simply left in place	Asks whether to permanently delete the database (defaults to **No**)
Several people signed in at once	Not supported by design - it's a single-user database	Fully supported - this is what it's built for
Moving to the other edition later	Supported as a migration, not a one-click switch - ask whoever set this up for you	n/a

If you're not sure which one you need, ask: **"will anyone else ever open this on their own computer and sign in at the same time as me?"** Yes -> Server edition. No, just me -> Personal edition.

---

*The rest of this guide covers installing and operating either edition in detail. Chapter references below (e.g. "see the AI Provider chapter") point to the companion **User Manual** PDF, which covers every in-app screen once you're signed in.*

Installation & Deployment

<- Back to Table of Contents

This chapter is for whoever is standing this application up for the very first time, on a computer that doesn't have it running yet. It assumes **no technical background at all**.

If someone else (an IT provider, a technical colleague) already has this running somewhere and just gave you a web address to sign in at, you don't need this chapter — skip straight to Getting Started.

Which path is right for you?

- **Windows, no technical background, want this done in a few clicks**: use

Option A — the one-click installer below. Budget 10–15 minutes, most of it unattended.

- **Mac, Linux, or you'd rather see/control every step yourself (including on Windows)**:

follow Option B — manual setup instead. Budget about 30–60 minutes, and comfort typing a few commands exactly as given.

There's no one-click installer for Mac or Linux yet — Option B is the only path there for now.

Which Windows installer do I need?

The one-click installer comes in two editions — same application, same wizard, same look once you're signed in. What differs is where your data lives, and that follows directly from who's going to use it:

- `AP3WayMatchSetup.exe` (the **Server** edition) — for an install that several people will

reach over your network from their own computers (an office server, a shared machine). Your data is kept in a proper database server (PostgreSQL) that safely handles several people saving things at the same moment.

- `AP3WayMatchSetup-Personal.exe` (the **Personal** edition) — for one person, on their own

laptop or desktop, using it themselves. Your data is kept in a single file on that computer, which makes backing up as simple as copying one file — see Everyday operations below.

If you're not sure, ask: "will anyone else ever open this on their own computer and sign in at the same time as me?" Yes -> Server edition. No, just me -> Personal edition. You can move from Personal to Server later if it turns out more people need in — ask whoever set this up for you about migrating, rather than reinstalling from scratch (it's a supported path, just not a one-click one).

Installing the wrong edition into a folder that already has the other one running is blocked — setup will tell you plainly and stop, rather than overwrite it.

Option A — Windows: the one-click installer

A single installer file (see above for which one) installs Docker Desktop if it isn't already on this computer, builds and starts the application, and walks you through a short wizard — no command line, no editing any files by hand. The steps below are identical for both editions except where noted.

What you'll need: the same as Option B below — a computer that can stay switched on, an internet connection for this one-time setup, and the installer file itself (provided to you by whoever you got this software from).

1. **Run** `AP3WayMatchSetup.exe`**.** Windows will ask for administrator permission (a UAC prompt) — this is expected, since the installer needs to install Docker Desktop if it's missing and write files under `C:\ProgramData`. Click **Yes**.

2. **Click through the Welcome and installation-folder pages.** The default location — the Server edition uses `C:\ProgramData\AP3WayMatch`, the Personal edition uses `C:\ProgramData\AP3WayMatchPersonal` — is right for almost everyone; this isn't your regular `Program Files` folder because this app needs to keep writing to its own folder while it runs (uploaded documents, logs), which `Program Files` increasingly locks down against.

3. **Create your admin account** — just an email address. Setup generates a secure temporary password for you automatically; you don't type one in here.

4. **Optionally add a Gemini API key** to enable AI-powered invoice extraction immediately. Leave this blank to skip it — you can add Gemini, or switch to OpenAI, Anthropic Claude, or your own local model, from Settings once you're logged in. Only Gemini can be set up at this step; see AI Provider & Local Model Setup for the others.

5. **Wait.** Setup now works through several stages automatically, shown on a progress page: checking/installing Docker Desktop, building the application's image (several minutes on this first run — it's downloading and compiling everything), starting it, then waiting for it to respond as healthy. This is the same work Option B does by hand, just automated.

6. **Copy your login details from the Finish page.** It shows your admin email and the generated temporary password in a selectable field — click into it and press **Ctrl+C** before closing the wizard, since it's shown once and not saved anywhere you can get back to. Leave "Open 3 Way Match in your browser" checked to sign in right away.

7. **Sign in** with that email and temporary password. You'll be required to set a real password immediately — from there, follow the first-time setup checklist in Getting Started.

If Windows needs to restart partway through (this happens on a computer that's never had Docker's underlying "WSL2" feature enabled before): setup tells you plainly and stops. Restart the computer and sign back in as the same user — setup resumes automatically from where it left off, with no need to run the installer again.

If something goes wrong, setup still reaches the Finish page rather than crashing outright, with a message pointing at the install log — `C:\ProgramData\AP3WayMatch\logs\install.log` (Server edition) or `C:\ProgramData\AP3WayMatchPersonal\logs\install.log` (Personal edition). Common causes:

What you see	Likely cause	Fix
Docker Desktop couldn't be made ready	Something blocked Docker's own install/startup (rare — antivirus interference, a pending unrelated Windows update)	Install Docker Desktop yourself from docker.com, then re-run setup

What you see	Likely cause	Fix
Setup could not build or start the application	Often a missing internet connection during the very first build, or something else on this computer already using port 5000	Check your internet connection; check what else is using port 5000 (`docker compose ps` if it's another Docker app) and stop it, then re-run setup
The application was started but did not respond in time	It may still be finishing its first startup — this can occasionally take longer than the wait	Wait a minute and open `http://localhost:5000` yourself; if it still doesn't load, check the install log

Afterward: the Start Menu gets four shortcuts — **Open**, **Start**, **Stop**, and **View Logs** — under "3 Way Match", so day-to-day use never needs a command line either. See Everyday operations below (same underlying commands, either path). **Uninstalling** removes the application's program files and stops/removes its Docker containers — it never deletes your uploaded invoices or vendor documents, which stay in place if you ever reinstall. On the **Server edition**, you'll also be asked whether to permanently delete your database — this defaults to **No**, so answering the prompt (or an unattended uninstall, which always picks the default) keeps your data either way. Only choose **Yes** if you genuinely want to start over. The **Personal edition** has no separate prompt — its database is a single `exceptions.db` file in the install folder, left in place exactly as before.

Option B — Manual setup (Mac, Linux, or Windows by hand)

Follow this instead of Option A if you're not on Windows, or you'd simply rather see/control every step. Budget about 30–60 minutes — if you've never used a "Terminal" or "Command Prompt" before, this section explains every step in the order you need it, including exactly what to type.

What you'll need

• **One computer that can stay switched on** for as long as your team needs to use this

app — a spare desktop, an office server, or a cloud virtual machine all work. It doesn't need to be powerful, but it does need to stay running; if it's someone's personal laptop that gets closed and carried home every evening, nobody else can reach the app while it's closed.

• **An internet connection** on that computer, at least for the one-time setup below (to

download the installer and the app's own files). The app itself doesn't need the internet to keep running afterward, except to reach whichever AI provider you choose (see AI Provider & Local Model Setup) and, optionally, Gmail.

• **The application's files**, provided to you by whoever you got this software from — as a

ZIP file, or a link to download one. Extract it into its own folder (e.g. `C:\ap-three-way-match`) on the computer from the point above.

> **Do:** pick a computer that's connected to your office network by cable if possible, or a reliable Wi-Fi connection — this is what everyone else's browser will be talking to.
> **Don't:** use a laptop that goes to sleep automatically or that someone might disconnect from the network — the app becomes unreachable for everyone the moment that happens.

Step 1 — Install Docker Desktop

Docker is the one piece of software you need to install. It packages this entire application — no separate database software, no separate "Python" install, nothing else to configure — and it's what actually runs the app for you.

1. Go to docker.com/products/docker-desktop

and download the installer for your operating system (Windows, Mac, or Linux).

2. Run the installer, accepting the default options. On Windows, it may ask to enable a

feature called "WSL2" or "Virtual Machine Platform" — accept this; it's required and Docker's own installer sets it up for you.

3. **Restart the computer** if the installer asks you to.
4. Open **Docker Desktop** from your Start Menu / Applications folder. The first launch takes

a minute or two. You'll know it's ready when its icon (a whale) appears in your system tray/menu bar and says "Docker Desktop is running."

> **Don't** worry about learning Docker beyond this — you'll only ever type two or three Docker commands, all given to you exactly below. You don't need to understand how it works internally.

If the installer complains about virtualization not being enabled: this is a setting in your computer's BIOS/UEFI, which varies by manufacturer. Search "[your computer's brand] enable virtualization" or ask a technical colleague — this is the one step in this whole chapter that's genuinely hardware-specific and worth asking for help with if you get stuck.

Step 2 — Create your configuration file

The app reads its settings from one plain text file named `.env`, in the same folder you extracted the application's files into.

1. Open that folder. You should see a file named `.env.example`.
2. Make a copy of it, and rename the copy to exactly `.env` (no `.example`, and no `.txt` at

the end — see the Don't below).

3. Right-click `.env` and open it with **Notepad** (Windows) or **TextEdit** (Mac — open

TextEdit first, then use File -> Open, since double-clicking may try something else).

You'll see a list of lines like `SOMETHING=`. Fill in the blank ones as follows.

> **Do:** use a plain text editor (Notepad, TextEdit, VS Code, Notepad++). **Don't** use Microsoft Word or a rich-text editor — it can silently save invisible formatting that breaks the file.
> **Don't** let Windows add a hidden `.txt` extension when renaming — if File Explorer is hiding file extensions, you might end up with `.env.txt` without realizing it. Turn on "File name extensions" in File Explorer's View tab first, or check by opening the folder's **Details** view, to confirm the file is really named `.env` with nothing after it.

Fill in your AI provider key

Pick **one** of the four lines near the top (`GEMINI_API_KEY=`, `OPENAI_API_KEY=`, `ANTHROPIC_API_KEY=`, or leave all four blank if you're using a local model) and paste in a key from that provider. If you don't have one yet, or don't know what this means, see AI Provider & Local Model Setup — do that first, then come back here.

Choose your admin sign-in

Find these two lines and replace the placeholder values with your own:

```
APP_USERNAME=admin
APP_PASSWORD=choose-a-real-password
```

This becomes the very first login for this app — the one you'll use to sign in and set everything else up. Pick a real password here, not the placeholder text.

Generate your two required security keys

Two lines need a value that isn't something you make up yourself — a long, random string:

```
FLASK_SECRET_KEY=
FIELD_ENCRYPTION_KEY=
```

The app will not start at all without both of these filled in. Generate them using Docker itself — you don't need to install anything else to run this:

1. Open a **Command Prompt** (Windows: press the Windows key, type `cmd`, press Enter) or **Terminal** (Mac/Linux).
2. Run this command exactly as written:

```
docker run --rm python:3.11-slim python -c "import secrets; print(secrets.token_hex(32))"
```

The first time, this downloads a small helper image (one-time, a minute or so) and then prints a long string of letters and numbers. Copy it and paste it after `FLASK_SECRET_KEY=` in your `.env` file.

3. Run this second command the same way:

```
docker run --rm python:3.11-slim python -c "import os, base64; print(base64.urlsafe_b64encode(os.urandom(32
```

Copy its output after `FIELD_ENCRYPTION_KEY=`.

> **Do:** run each command again to generate a fresh value any time — there's no downside to generating a new one before you've gone live with real data.
> **Don't** reuse or share these two values, and don't commit them to any shared/public place — anyone who has both can read this app's stored data and forge sessions. Once you've gone live with real vendor data, don't casually change `FIELD_ENCRYPTION_KEY` again (see the comment above that line in `.env.example` for why).

The rest of the file

Leave everything else at its default (commented-out) value for now — the `POSTGRES_...` lines already have sensible defaults, and everything under "Optional" (`HOST`, `PORT`, Gmail, Single Sign-On) can wait until you actually want those features. Save the file when you're done.

Step 3 — Start the application

Back in your Command Prompt/Terminal, navigate into the folder holding the app's files (the same one with `.env` in it) and run:

```
docker compose up -d
```

The first run downloads and builds everything — this can take several minutes depending on your internet connection. You'll see a lot of text scroll past; that's normal. The `-d` means it keeps running in the background even after you close this window.

How to tell it worked: run `docker compose ps` — you should see two services listed (`app` and `db`), both saying "Up" or "running (healthy)". If instead `app` keeps restarting or shows an error, see **Troubleshooting** below.

> **Do:** keep Docker Desktop itself running (it can sit minimized/in your system tray) — the app only runs while Docker Desktop is running.

Step 4 — Find the address to open it at

This is the part that trips people up, because **there's no single fixed address** — it depends entirely on where your computer is and who needs to reach it.

- **Just you, on the same computer that's running it**: open a browser and go to `http://localhost:5000`.
- **Your team, on other computers on the same office network** (the common case for an office server): you need that computer's own network address.

• **Windows**: open Command Prompt and run `ipconfig`. Look for **"IPv4 Address"** — it looks like `192.168.1.50`.

• **Mac**: open System Settings -> Network, click your active connection, and read the IP address shown. Or in Terminal: `ipconfig getifaddr en0`.

• **Linux**: run `hostname -I` in a terminal.

Whatever that address is, your team (and your vendors — see below) open `http://<that-address>:5000` — for example `http://192.168.1.50:5000`.

> **Don't** assume this address never changes. Most office networks hand out addresses that *can* change when the computer restarts. Ask whoever manages your office router/network to set a "static IP" or "DHCP reservation" for this specific computer — a one-time request that keeps the address permanent. Until that's done, double-check the address after every restart of the computer running this app.

What your vendors use: exactly the same address, with `/vendor/login` on the end instead — e.g. `http://192.168.1.50:5000/vendor/login`. This is a completely separate sign-in from your team's (see the Vendor Portal Guide); once you create a vendor account and send their invite (see Admin Guide -> Managing vendors), the invite email is built automatically using whichever address you're reached at — you never need to type it in by hand for them.

If vendors or staff need to reach this from outside your office (a different physical location, over the public internet) — that's a meaningfully bigger step: it needs a real domain name, a security certificate (HTTPS), and careful router/firewall configuration, or a cloud server instead of an office computer. This is the one part of setup genuinely worth paying an IT provider for, or asking your vendor for guidance on, rather than attempting it purely from this chapter — getting it wrong can expose your data to the open internet. `SETUP.md` in the application's own files has the technical detail for whoever ends up doing this.

Step 5 — Sign in and finish setup

Open the address from Step 4 and sign in with the `APP_USERNAME`/`APP_PASSWORD` you set in your `.env` file. From here, follow the first-time setup checklist in Getting Started — Company Profile, choosing where your Purchase Orders come from, and so on.

Everyday operations

If you installed via Option A, the Start Menu's **Stop 3 Way Match**/**Start 3 Way Match**/**View Logs** shortcuts do exactly what the commands below do — use whichever you find easier; there's no difference underneath.

• **Is it running right now?** `docker compose ps` in the app's folder.

• **Stop it** (e.g., to move the computer, or for maintenance): `docker compose down` — your data is untouched, it's just not reachable until you start it again.

• **Start it again**: `docker compose up -d`, same as Step 3.

• **The computer was restarted and now nobody can reach the app**: Docker Desktop needs to

be running for the app to run. Option A already enables Docker Desktop's own "start on login" setting for exactly this case, so this normally recovers on its own after a minute or two; if not (or if you're on Option B, which doesn't enable that setting), open Docker Desktop yourself, wait for it to say it's running, then `docker compose up -d` in the app's folder if it hasn't already started back up.

Backing up your data

How you back up depends on which edition you installed (see Which Windows installer do I need? above) — do this on a schedule (weekly, at minimum, more often once you're relying on this for real invoices).

Personal edition: your entire database is one file, `exceptions.db`, directly inside the app's own folder (`C:\ProgramData\AP3WayMatchPersonal` by default). Back it up by copying that one file somewhere else — a USB drive, a shared network drive, a cloud backup folder. Stop the app first (`docker compose down`, or the **Stop 3 Way Match** shortcut) so the file isn't being written to while you copy it, then start it again afterward.

Server edition: data lives inside a proper database server (PostgreSQL) running inside Docker, not as a single file you can just copy. Back it up like this instead:

```
docker compose exec db pg_dump -U ap_app ap_three_way_match > backup_2026-09-08.dump
```

(Replace the date in the filename with today's.)

Both editions: keep a few weeks of these backups someplace other than the same computer, so a hardware failure on this one machine doesn't take your only copy with it. Also back up the `uploads` and `vendor_uploads` folders inside the app's own folder the same way — every invoice and vendor document you've ever received lives there, not in the database.

> **Do:** actually test a restore once, before you need it for real — ask a technical colleague or your vendor to walk through it with you once, so you're confident the backups you're taking are actually usable.

Troubleshooting

What you see	Likely cause	Fix
`docker compose up -d` fails immediately, or Docker Desktop won't open	Docker Desktop isn't installed correctly, or virtualization isn't enabled	Reopen Docker Desktop and check its own status message; see the virtualization note in Step 1
The `app` service keeps restarting in `docker compose ps`	Almost always a missing or wrong value in `.env` — most commonly `FLASK_SECRET_KEY` or `FIELD_ENCRYPTION_KEY` left blank	Run `docker compose logs app` and read the last few lines - it names the exact missing value
Browser says it can't connect / "This site can't be reached"	Wrong address, or the app isn't actually running yet	Re-check Step 4's address, and confirm with `docker compose ps` that `app` shows "Up"

What you see	Likely cause	Fix
It worked yesterday, nobody can reach it today	The computer's network address changed (see Step 4's warning), or the computer/Docker Desktop wasn't left running	Re-check the address with `ipconfig`/`ifconfig`; make sure Docker Desktop is open
Login page loads, but signing in never seems to work, no error shown	`FORCE_HTTPS_COOKIES` is set to `true` while you're reaching this over a plain `http://` address with no real TLS in front of it	Leave it unset/`false` in `.env` unless you've genuinely set up HTTPS in front of this (see Step 4's note on going outside the office)

Where to go next

- Every-role basics once you're signed in: Getting Started
- Every admin-only screen: Admin Guide
- Choosing and configuring what reads your invoices: AI Provider & Local Model Setup