

QuiSuite Accounting

Installing on Linux

Setting up QuiSuite Accounting and its database on Ubuntu, Linux Mint, Debian and Fedora, on one PC or across an office

Version 1.28 - October 2026

© QuiSoft Limited 2024-2026. All rights reserved.

Contents

1. Installing QuiSuite in three steps.....	3
One PC or a whole office.....	3
2. What you need.....	4
Each PC that runs QuiSuite.....	4
The database server.....	5
Internet.....	5
3. PostgreSQL, the database.....	6
Is PostgreSQL already installed?.....	6
Installing PostgreSQL 18 on Ubuntu, Linux Mint and Debian.....	6
Installing PostgreSQL 18 on Fedora.....	6
Creating QuiSuite's login.....	6
Only the tools, for the other PCs.....	7
4. Installing QuiSuite.....	8
5. The first start.....	9
6. More PCs in an office.....	10
On the database server: let the PCs in.....	10
On each other PC.....	10
A shared folder for documents.....	10
7. Using it day to day.....	11
8. Keeping QuiSuite up to date.....	12
What the administrator does.....	12
What everyone else sees.....	12
9. If something is not right.....	14
10. Removing QuiSuite.....	15

1. Installing QuiSuite in three steps

QuiSuite is quick to set up. There is no long installation program to click through: on Linux it is a folder you copy onto the computer, and everything it needs is already inside that folder. The only other thing it needs is a **PostgreSQL** database server, which is free and installs in a few minutes.

Step	What you do	Where	How long
1. The database	Install PostgreSQL 18 and create a login for QuiSuite	Once, on one computer (this PC, or a server on your network)	About 10 minutes
2. The program	Copy QuiSuite onto each computer that will use it	Every PC	2 minutes each
3. First start	Start QuiSuite, connect it to the database, create your company	The first PC	5 minutes

One PC or a whole office

QuiSuite works the same way whether one person uses it or twenty:

- **On one PC:** PostgreSQL and QuiSuite are both on that PC. Follow the steps once and you are done.
- **In an office:** PostgreSQL goes on one computer that stays switched on (a server, or simply the main office PC). Every other PC gets its own copy of QuiSuite and connects to it over the network. Everyone then works on the same companies at the same time, each with their own login.

Mixing Windows and Linux

Windows and Linux PCs can share the same database: a Linux PC can use a database server on Windows, and the other way round. Each PC simply needs the QuiSuite for its own system.

2. What you need

Each PC that runs QuiSuite

	Minimum	Recommended
Linux	A desktop Linux from 2024 or later (glibc 2.38 or newer), 64-bit - see the list below	Ubuntu 24.04 LTS or 26.04 LTS
Processor	64-bit (x86-64) Intel or AMD, 2 cores, 2 GHz	4 cores or more
Memory	4 GB	8 GB or more
Disk	1 GB free in your home folder (the program is about 320 MB; updates need room alongside it)	2 GB free
Screen	1366 x 768	1920 x 1080 (Full HD) or larger
Desktop	Any desktop: GNOME, KDE Plasma, Cinnamon, Xfce, MATE...; X11 or Wayland	-
Other software	None: everything QuiSuite needs is inside its folder	-

It runs on

Linux	Versions	
Ubuntu	24.04 LTS, 24.10, 25.04, 25.10, 26.04 LTS	Tested on 24.04 and 26.04
Kubuntu, Xubuntu, Ubuntu MATE and the other Ubuntu flavours	24.04 and later	
Linux Mint	22, 22.1, 22.2 and later	
Debian	13 "Trixie" and later	
Fedora Workstation	39 and later	
Pop!_OS, Zorin OS, elementary OS	Versions based on Ubuntu 24.04 or later (Pop!_OS 24.04, Zorin OS 18, elementary OS 8)	

Older releases - Ubuntu 22.04, Linux Mint 21, Debian 12, Red Hat / Rocky / Alma 9 - are too old for QuiSuite's libraries. They can still hold the **database** (section 3 shows how), with QuiSuite itself on newer PCs.

64-bit only

QuiSuite is built only as a 64-bit program, so it needs 64-bit Linux on an Intel or AMD (x86-64) processor. It does not run on 32-bit Linux or on ARM computers such as the Raspberry Pi.

To check, open a terminal (Ctrl+Alt+T on most desktops) and type **uname -m**: it must answer **x86_64**. **ldd --version** shows the glibc version on its first line: it must be **2.38** or higher.

The database server

The computer that holds the PostgreSQL database. On one PC, that is the PC itself.

	Minimum	Recommended
Database	PostgreSQL 18	PostgreSQL 18 (QuiSuite is developed and tested on it)
Processor	64-bit, 2 cores	64-bit, 4 cores or more for 10 users or more
Memory	4 GB	8 GB, or 16 GB for 10 users or more
Disk	10 GB free	An SSD with 50 GB free, leaving room for the companies, their files and backups
Network	The PCs can reach it on port 5432	Wired network, 100 Mbit/s or faster; a fixed address for the server

PostgreSQL itself can run on Windows or Linux, whichever you prefer: the PCs do not need to match it.

Internet

QuiSuite does not need the internet to keep your accounts. It is used for: **updates** (from the QuiSuite web site), sending and receiving **email**, exchange rates, and the built-in **web browser**. Licences are checked on the PC, without going online.

3. PostgreSQL, the database

PostgreSQL is the free database that holds your accounts. Install it on the one computer that will hold the data. In an office, the other PCs only need its small set of tools (at the end of this section).

Everything below is typed in a **terminal** (Ctrl+Alt+T, or Terminal from the applications menu). Commands starting with **sudo** ask for your own password the first time.

Is PostgreSQL already installed?

```
psql --version
systemctl status 'postgresql*'
```

psql (PostgreSQL) 18.x and **active (running)** mean PostgreSQL 18 is installed and running. "command not found" means it is not installed; a version below 18 means an older one - install 18 as below (both can live side by side). Press **q** to leave the status display.

Installing PostgreSQL 18 on Ubuntu, Linux Mint and Debian

PostgreSQL's own package source always has version 18, whichever of these you have. Copy these lines into the terminal - they work unchanged on Ubuntu, Mint, Debian, Pop!_OS, Zorin and elementary:

```
sudo apt install -y curl ca-certificates
sudo install -d /usr/share/postgresql-common/pgdg
sudo curl -fsSo /usr/share/postgresql-common/pgdg/apt.postgresql.org.asc
https://www.postgresql.org/media/keys/ACCC4CF8.asc
echo "deb [signed-by=/usr/share/postgresql-common/pgdg/apt.postgresql.org.asc]
https://apt.postgresql.org/pub/repos/apt $(. /etc/os-release; echo ${UBUNTU_CODENAME:-
$VERSION_CODENAME})-pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg.list
sudo apt update
sudo apt install -y postgresql-18
```

PostgreSQL 18 is now installed and running, and starts by itself whenever the computer starts.

Installing PostgreSQL 18 on Fedora

```
sudo dnf install -y https://download.postgresql.org/pub/repos/yum/repos/pms/F-$(rpm -E
%fedora)-x86_64/pgdg-fedora-repo-latest.noarch.rpm
sudo dnf install -y postgresql18-server
sudo /usr/pgsql-18/bin/postgresql-18-setup initdb
sudo systemctl enable --now postgresql-18
```

Creating QuiSuite's login

QuiSuite logs in to PostgreSQL with a login of its own, called **A4LUser**, which is allowed to create the databases your companies are kept in. Open PostgreSQL's command line as its administrator:

```
sudo -u postgres psql
```

Then type this line, putting your own password between the single quotes, and press Enter:

```
CREATE ROLE "A4LUser" LOGIN CREATEDB PASSWORD 'choose-a-password';
```

It answers **CREATE ROLE**. Type **\q** and press Enter to close it. Keep A4LUser's password: QuiSuite asks for it once on each PC.

The double quotes matter

Type "A4LUser" with its double quotes. Without them PostgreSQL turns the name into lower case, and QuiSuite would not find it.

Only the tools, for the other PCs

QuiSuite takes its backups with PostgreSQL's tools, so the PCs that take backups or upgrade the databases (the administrator's, at least) need them even when the database is elsewhere. Add PostgreSQL's package source as above, then install only the tools:

```
sudo apt install -y postgresql-client-18    # Ubuntu, Mint, Debian
sudo dnf install -y postgresql18            # Fedora
```

4. Installing QuiSuite

QuiSuite comes from QuiSoft as one file, **QuiSuite-1.28-linux-x64.tar.gz** (the number is the version). Put it in your Downloads folder, then in a terminal:

```
cd ~/Downloads
tar -xzf QuiSuite-1.28-linux-x64.tar.gz
cd QuiSuite-1.28-linux-x64
./install.sh
```

That is all. install.sh:

- puts QuiSuite in your home folder, in **~/local/share/QuiSuite/QuiSuite** (to choose another folder, give it: `./install.sh ~/QuiSuite`);
- adds **QuiSuite Platform**, with its icon, to your applications menu;
- on Ubuntu 24.04 and later, asks for your password once, to let QuiSuite's built-in web browser keep web pages in a safe sandbox, as web browsers do. You may skip it with Ctrl+C: QuiSuite works either way.

It does not need administrator rights for anything else, and changes nothing outside your home folder. You can then delete the downloaded file and the unpacked folder.

Why in the home folder?

QuiSuite updates itself, so it must be somewhere the person using it may change. In your home folder it always can, with no password. Each person who uses QuiSuite on a shared Linux PC runs install.sh once in their own account.

5. The first start

Start QuiSuite on the first PC. The first time, it asks how to reach the database:

Field	What to type
Server	localhost if PostgreSQL is on this PC; otherwise the database server's name or address (for example office-server or 192.168.1.10)
Port	5432 (leave it as it is)
Master database	qs4l (leave it as it is)
User	A4LUser (the login you created in section 3)
Password	The password you gave A4LUser
Shared data folder	Leave it blank on one PC. In an office, see section 6.
PostgreSQL tools folder	Leave it blank (Ubuntu, Debian, Mint). On Fedora: /usr/pgsql-18/bin . QuiSuite uses these tools for backups.

1. Press **Test**. It should say **Connected successfully**, or that the server was reached but has no database "qs4l" yet - which is exactly right for a new installation.
2. Press **OK**. QuiSuite offers to create the master database (the list of your companies): answer **Yes**. It takes a few seconds.
3. The login window opens and says there is no company yet. Press **Companies...** and type A4LUser's password.
4. Press **New Company...**, type your company's name, and press **Create**. Tick **Demo company** instead if you would like a company full of example data to try things out on first.
5. QuiSuite creates the company and says: **Log in as Admin / Admin**. Close the Companies window, choose the company, and log in with the login **Admin** and the password **Admin**.
6. Change the Admin password straight away: **Administrator > Users**. Add a login for each person who will use QuiSuite there too.

Every new company starts with a 30-day trial with every module switched on. Your licence key, when you have it, is installed from the same Companies window (**Licence...**).

The connection settings are kept for each person on each PC, in `~/.config/QuiSoft/QuiSuite/connection.ini`; QuiSuite fills them in again by itself on the next start.

Start QuiSuite from the applications menu (search for **QuiSuite**), or from a terminal with `~/.local/share/QuiSuite/QuiSuite/QuiSuite`.

6. More PCs in an office

On the database server: let the PCs in

A new PostgreSQL only accepts connections from its own computer. To let the office PCs in, on the database server:

1. Make PostgreSQL listen on the network: open its settings file (on Fedora the files are in `/var/lib/pgsql/18/data/` instead):

```
sudo nano /etc/postgresql/18/main/postgresql.conf
```

Find the line **`#listen_addresses = 'localhost'`**, change it to the line below (without the `#` at the front), and save with `Ctrl+O`, `Enter`, `Ctrl+X`:

```
listen_addresses = '*'
```

2. Allow your office network: open the access file and add this line at the end, using your own network's numbers (most office networks are `192.168.1.something` or `192.168.0.something`):

```
sudo nano /etc/postgresql/18/main/pg_hba.conf
```

```
host      all      all      192.168.1.0/24      scram-sha-256
```

3. Open the firewall to the office network if it is switched on, and restart PostgreSQL:

```
sudo ufw allow from 192.168.1.0/24 to any port 5432 proto tcp      # Ubuntu, Mint,
                                                                    Debian (if ufw is on)
sudo systemctl restart postgresql
```

```
sudo firewall-cmd --permanent --add-service=postgresql && sudo firewall-cmd --reload
# Fedora
sudo systemctl restart postgresql-18
```

Give the database server a fixed address on your network (your router or IT supplier can do this), so the PCs always find it.

On each other PC

1. Install QuiSuite (section 4).
2. Start it. In the connection window, type the **database server's name or address** as the Server, and A4LUser and its password. Press **Test**, then **OK**.
3. The companies are already there: choose one and log in with your own login.

A shared folder for documents

Each company also keeps files - letters, PDFs of invoices, imported spreadsheets, backups - in a data folder. On one PC that is beside the program. In an office, make a folder on the server that everyone may change, shared over the network (a Samba or NFS share, or a network drive), mount it in the same place on every PC (for example `/srv/quisuite-data`), and type that as the **Shared data folder** in the connection window on every PC (Connection Settings... in the login window).

7. Using it day to day

- Start QuiSuite from the applications menu and log in with your own login and password.
- Several companies? Choose one at login; **File > Switch Company** moves to another.
- Press **F1** anywhere for help about the window you are in.
- **Administrator > Backup Databases** takes a backup whenever you want one.

8. Keeping QuiSuite up to date

QuiSuite updates itself. New versions are published by QuiSoft on the QuiSuite web site; nothing is downloaded behind your back in the middle of work. An update has two parts, done in this order:

	Who	What happens
1. The databases	An administrator, once for the whole office	Administrator > Upgrade Databases brings the master database and every company up to the new version.
2. Each PC	Nobody - it happens by itself	The next time someone logs in on a PC, QuiSuite sees the databases are newer, fetches the new version and restarts.

The program and the databases always carry the same version number (1.28, say). That is how every PC knows when it is out of date.

What the administrator does

When QuiSoft tells you a new version is out (or whenever you like, to see):

1. **Update your own PC first.** Log in, then **Help > Check for Updates....** If there is a new version, QuiSuite downloads only the files that changed, closes, puts them in place and starts again, saying "QuiSuite has been updated to version ...".
2. Choose a quiet moment, then **Administrator > Upgrade Databases.** The window lists the master database and each company, with its version and the changes it needs.
3. Press **Upgrade** and confirm.
4. Everyone else using QuiSuite sees a red notice: **"SAVE YOUR WORK NOW: QuiSuite closes itself in 4:59..."**. It counts down for five minutes and then closes QuiSuite on their PC. Nobody can log in while the upgrade runs. Your window carries on by itself as soon as everyone has gone.
5. QuiSuite **backs up every database first**, into each company's Backups folder (before_upgrade_<version>_<date>.zip). It then upgrades the master database and each company, showing each result.
6. When it says it has finished, people can log in again.

If something goes wrong

Each change is made as a whole or not at all, so a database is never left half changed, and the backup taken at the start holds everything as it was. The window says what failed; nothing is lost. Call QuiSoft support before trying again.

What everyone else sees

Nothing needs doing on the other PCs. The next time someone logs in after the upgrade, QuiSuite says:

This company's database has been upgraded to version 1.29,
but this copy of QuiSuite is version 1.28.

1. Press **Update Now.** (Always choose this: the older program may not understand the upgraded database.)
2. A progress window shows the new files arriving. Only the files that changed are downloaded, and each is checked as it arrives.

3. QuiSuite closes, the new files are put in place, and it starts again as the new version, ready to log in.

If anything goes wrong while the files are being put in place, every old file is put back, so the PC is never left with half an update; QuiSuite starts as it was and says where its update log is.

Each person's copy updates itself

QuiSuite is installed in each person's own home folder, so it can always update itself without a password. If several people log in to the same Linux PC, each one's copy updates when they next log in to QuiSuite.

9. If something is not right

What you see	What to do
"Connection failed: ... password authentication failed for user "A4LUser""	The password is wrong, or the login was made without the double quotes (section 3) and is really called a4luser. Make it again with the quotes.
"Connection failed: ... could not connect to server" / "Connection refused"	PostgreSQL is not running, or (from another PC) the server does not allow the network yet: see section 6.
"... no pg_hba.conf entry for host ..."	The server does not yet allow this PC's network: add the line in section 6 and restart PostgreSQL.
"The master database could not be created: permission denied to create database"	A4LUser was made without CREATEDB. In psql: ALTER ROLE "A4LUser" CREATEDB;
A new company fails with "unrecognized configuration parameter"	The database server runs an older PostgreSQL. Install PostgreSQL 18.
Backups fail: "pg_dump: error: server version: 18...; pg_dump version: 16..."	This PC's PostgreSQL tools are older than the server. Install the PostgreSQL 18 tools (section 3).
"This company's database has been upgraded ..." at login	Not a fault: press Update Now (section 8).
"tar: ... Cannot open: No such file or directory"	The file is not in Downloads, or its name differs: type ls ~/Downloads to see it, and use that name.
QuiSuite does not appear in the applications menu	Log out and back in once; the menu reads new entries at login on some desktops.
"Ident authentication failed for user "A4LUser"" (Fedora)	In /var/lib/pgsql/18/data/pg_hba.conf change ident to scram-sha-256 on the host lines, then sudo systemctl restart postgresql-18.
"error while loading shared libraries ... GLIBC_2.38 not found"	This Linux is older than QuiSuite needs (section 2). Use a newer release for the PCs; this one can still hold the database.
The other PCs cannot connect, but the server itself can	Check listen_addresses, the pg_hba.conf line and the firewall (section 6), and that the PCs use the server's right name or address.
Backups fail: "pg_dump: command not found"	Install the PostgreSQL 18 tools on this PC (section 3); on Fedora also fill in /usr/pgsql-18/bin as the PostgreSQL tools folder.

10. Removing QuiSuite

In a terminal:

```
~/local/share/QuiSuite/QuiSuite/uninstall.sh
```

It asks before removing anything, then takes away the program folder, its menu entry and icon, and the sandbox permission if install.sh added one. The companies stay safely in PostgreSQL, and your connection settings in ~/.config/QuiSoft are kept. To remove the databases as well, remove PostgreSQL (sudo apt remove postgresql-18) - only once you are sure you have a backup of anything you need.