Print to PDF

Non-Profit Fund Accounting System v9.0

Installation Guide – VirtualBox on Windows 11 (Ubuntu 24.04 LTS Guest)

July 22, 2025

Table of Contents

1. Prerequisites & Host Requirements 1
2. VirtualBox VM Setup & Ubuntu 24.04 Installation 2
3. Install Prerequisite Packages 3
4. Clone Application & Prepare Environment 4
5. Database Configuration & Initialization 5
6. Running the Application 6
7. Testing Checklist (v9.0) 7
8. Troubleshooting 8
9. Performance Tips 8
10. Security Notes 9
Appendix A – Useful Commands 9

Architecture overview

- ***Host OS:*** *Windows 11 (runs Oracle VM VirtualBox)*
- ***Guest OS:*** *Ubuntu Desktop 24.04 inside the VM*
- ***Application:*** *Non-Profit Fund Accounting System v9.0 installed in* `/opt/nonprofit-fund-accounting`

1. Prerequisites & Host Requirements

Host requirement	Minimum	Recommended
Host OS	Windows 10/11	—
CPU	4 cores (VT-x/AMD-V)	6+ cores
RAM	8 GB	16 GB (allocate ≥ 6 GB to guest)
Disk space	40 GB free	80 GB SSD/NVMe
Software	Oracle VirtualBox ≥ 7.0, Ubuntu 24.04 ISO	—

⚠ **Important:** Enable hardware virtualization (Intel VT-x/AMD-V) in BIOS/UEFI before proceeding.

2. VirtualBox VM Setup & Ubuntu 24.04 Installation

1. **Download**
   • VirtualBox: https://www.virtualbox.org/wiki/Downloads
   • Ubuntu 24.04 ISO: https://ubuntu.com/download/desktop
2. **Create a new VM**
   - Name: `Ubuntu24-FundAcct-v9_0`
   - Type: *Linux* → version *Ubuntu (64-bit)*
   - Memory: **6144 MB**

   - Processors: **4 vCPU** (System ➜ Processor)
   - Disk: **VDI**, dynamically allocated, **60 GB**
3. **Adjust settings**
   - Display ➜ Graphics Controller: **VBoxSVGA**, enable **3D Acceleration**
   - Storage ➜ Empty optical drive → **Choose a disk file…** select Ubuntu ISO
   - Network Adapter 1: **Bridged** *or* **NAT** (either works)
4. **Install Ubuntu 24.04** inside the VM
   - "Normal installation", enable third-party software (optional)
   - Disk setup: **Use entire disk** with **LVM** (default)
   - Username: **fundadmin** (sudo)
   - Reboot, login, run `Software Updater`

3. Install Prerequisite Packages

```
# Update system
sudo apt update && sudo apt -y upgrade

# Essential tools
sudo apt install -y git build-essential curl

# Node.js 20 LTS
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash
-
sudo apt install -y nodejs

# PostgreSQL 16
echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release
-cs)-pgdg main" | \
  sudo tee /etc/apt/sources.list.d/pgdg.list
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
sudo apt update
sudo apt install -y postgresql-16

# Verify versions
node -v    # v20.x
npm  -v    # 10.x+
psql -V    # 16.x
```

4. Clone Application & Prepare Environment

```
sudo mkdir -p /opt && cd /opt
sudo git clone https://github.com/tpfbill/nonprofit-fund-
accounting.git
sudo chown -R $USER:$USER nonprofit-fund-accounting
cd nonprofit-fund-accounting
```

Create `.env`:

```
PGHOST=localhost
PGPORT=5432
PGDATABASE=fund_accounting_db
PGUSER=npfadmin
PGPASSWORD=npfa123
PORT=3000
```

Install dependencies:

```
npm ci
```

5. Database Configuration & Initialization

5.1 One-step Role & Database Setup

Run the cross-platform helper script – it creates the **npfadmin** role, database, grants, and verifies connectivity in one go:

```
sudo -u postgres psql -f database/setup-database-cross-platform.sql
```

5.2 Load Base Schema

```
sudo -u postgres psql -d fund_accounting_db -f database/db-init.sql
```

`database/db-init.sql` now contains **all** required tables and columns – no manual `ALTER` steps needed.

5.3 NACHA Vendor Payments Schema (v9.0)

```
sudo -u postgres psql -d fund_accounting_db -f database/nacha-vendor-payments-schema.sql
```

This adds `vendors`, `vendor_bank_accounts`, `payment_batches`, `payment_items`, `company_nacha_settings`, and `nacha_files`.

5.4 (Optional) Load Demo Data

```
node database/load-principle-foundation-data.js
```

Creates The Principle Foundation multi-entity hierarchy with sample funds, accounts, and transactions.

6. Running the Application

Open **two terminals** inside the VM:

```
# Terminal 1 – backend API on port 3000
cd /opt/nonprofit-fund-accounting
node server.js
```

```
# Terminal 2 – static frontend on port 8080
cd /opt/nonprofit-fund-accounting
npx http-server . -p 8080 --no-cache
```

Alternative (single-command) approach
The project's *package.json* ships with helper scripts that can save time:

```
# Start only the static frontend (port 8080)
npm run client

# OR start both backend (port 3000) and frontend (port 8080)
concurrently
npm run dev
```

Using `npm run dev` is handy during active development because it launches both services in the background and streams their combined output.

Browse to **http://localhost:8080/index.html** (from within the VM or via host browser if using bridged network).

7. Testing Checklist (v9.0)

Test	Steps	Expected Outcome
Dashboard	Open `/index.html`	Summary cards & charts display
Vendor Directory	`Vendor Payments → Vendors`	Vendors list loads, "Add Vendor" opens modal
Payment Batch	`Vendor Payments → Batches → New Batch`	Entity & Fund dropdowns populate immediately
NACHA File	Create batch ➜ approve ➜ *Generate NACHA*	`.ACH` file appears in **Files** tab and downloads
API Health	`curl http://localhost:3000/api/health`	`{"status":"ok"}`
Documentation Tab	Click **Documentation**	`direct-docs.html` opens without styling issues

8. Troubleshooting

Symptom	Fix
"DB offline" badge	`sudo systemctl restart postgresql` and verify `.env` creds
Port 3000 already in use	`sudo lsof -i:3000` → `kill <PID>`
Empty dropdowns in New Batch modal	Ensure `database/nacha-vendor-payments-schema.sql` was run & server restarted
CSS not updating	Hard-refresh (Ctrl+F5) or clear browser cache

9. Performance Tips

1. Allocate extra vCPU/RAM in VM settings.
2. Enable **Nested Paging** & **KVM Paravirtualization**.
3. Store VDI on SSD/NVMe.
4. Tune PostgreSQL (`shared_buffers = 512MB`, `work_mem = 16MB`).
5. Use **Bridged Adapter** for faster host↔guest transfers.

10. Security Notes

- Change default passwords (`npfa123`) before production.
- Keep `.env` file **chmod 600**.
- Enable UFW:

```
sudo ufw allow 8080/tcp
sudo ufw allow 3000/tcp
sudo ufw enable
```

- Snapshot VM after successful install for quick rollback.

Appendix A – Useful Commands

```
# Stop servers
pkill -f http-server
pkill -f node

# Backup database
sudo -u postgres pg_dump -Fc fund_accounting_db >
fundacct_$(date +%F).dump

# Restore
sudo -u postgres pg_restore -d fund_accounting_db -c
fundacct_2025-07-19.dump
```

Enjoy your fully-functional Non-Profit Fund Accounting System v9.0 on Ubuntu 24.04!

*For more details see the in-app **Documentation** tab or the project README.*