

VRS X Manuals

- [\[LINUX\] VRS X Setup Guide](#)
- [MacOS setup guide](#)
- [Windows \(OMG\) Manual](#)

[LINUX] VRS X Setup Guide

Pre-start checklist

1. Make sure your OS has kernel 5.0 or newer
2. Make sure OpenSSL 1.1.1 or 3.0+ (supporting TLS 1.2/1.3) is installed.

Sudo is not necessary to start VRS X up, but to set up systemd service

1. Download zip file from the link given in e-mail, for example `linux-arm64.zip`
2. Unzip the file using command `unzip linux-arm64.zip` (or `-x64` if on Intel/AMD machine)
3. Enter the directory using command `cd linux-arm64` (or `-x64` if on Intel/AMD machine)
4. Make binary executable using command `chmod +x VRSX`
5. Start it up: `./VRSX` for desktop version, `./VRSX --headless` for headless mode
6. Enter IP.OF.YOUR.MACHINE:8085 and provide licence key provided in an email

Your instance is ready to go!

Set up systemd service, sudo is necessary!

Assuming, you are using `youruser` user on machine and VRS X is unzipped into your home directory, for example `/home/youruser/linux-arm64/`, you can simply follow the instructions below, but overwrite `youruser`` with a real username in your OS

1. Create a systemd service file `sudo touch /etc/systemd/system/vrsx.service`
2. Copy and paste the service configuration using for example nano (`sudo nano /etc/systemd/system/vrsx.service`):

```
[Unit]
Description=VRSX
After=network.target

[Service]
Type=simple
```

```
User=youruser
WorkingDirectory=/home/youruser/linux-arm64 ## change to linux-x64 if you are on Intel/AMD
machine
ExecStart=/home/youruser/linux-arm64/VRSX --headless ## change to linux-x64 if you are on
Intel/AMD machine

# If your system is RHEL-based, for example Rocky Linux/CentOS, please remove ExecStart line
above and uncomment the line below to avoid issues with SELinux:
# ExecStart=/bin/bash -c '/opt/linux-x64/VRSX --headless'
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Once done, click Ctrl+O (save), Ctrl+X (exit)

WARNING: Old Linux joke: If you want to create a strong admin password, try to edit this file using `vi`, LOL.

3. Make sure you have **overwritten** `youruser` with your username and removed all comments (after `##`)
4. Reload systemd using command `sudo systemctl daemon-reload`
5. Start and enable VRS X service using command `sudo systemctl enable vrsx --now`
6. Visit your VRS X instance on port 8085 (i.e. 192.168.100.100:8085) and activate it using the licence key given in "MY Account" section in ADS-B.Pro RadarView

MacOS setup guide

Sudo is not necessary anywhere!

COMPATIBILITY INFORMATION

Only Apple Silicon is supported

1. Download `osx-arm64.zip` package from releases
2. macOS may automatically unzip this zip file, so please move `osx-arm64` to any directory you want
3. Using Finder (or any other file manager, but Finder is preferred), enter `osx-arm64` directory, make sure you can see VRSX.App and then, right-click on the folder in the path bar at the bottom or the folder itself in the parent view"
4. Disable Gatekeeper using command `xattr -dr com.apple.quarantine ./VRSX.App`
5. Open the app by double-clicking on VRSX.App in Finder or using command `open VRSX.App/`
6. NOTE: There is a bug (I love Apple *including fruits*, but hate for this!) and there won't be any VRSX icon in the dock or top bar - it's mostly working like a daemon. To kill VRSX, you need to execute command `killall VRSX`
7. That's all! Enter 127.0.0.1:8085 (or any other IP of your mac) and activate VRSX - licence key is present in "MY Account" section in ADS-B.Pro RadarView
8. Enjoy!

Any systemd, automation?

macOS has `launchd` which is something "like" systemd, but no idea how it's working. First, it will require me to do some changes in Info.plist file, but, hey!

If you want VRS X to start automatically, just add VRSX.App to your Login Items in System Settings -> General -> Login Items. It's not a full daemon service, but it works for 99% of users!

NOTE: I have never tried it. Mac is my main development machine, but my testing server is a Raspberry Pi.

Windows (OMG) Manual

Nah, I hate Windows, but lemme try to help ya

Workflow is as simple as Windows, as buggy as Windows and as problematic as... yeah! You guessed correctly! As Windows!

BEFORE YOU START, READ IT!

Due to TLS requirements, there are certain Windows versions supported:

SUPPORTED

- Windows 10
- Windows 11
- Windows Server 2022
- Windows Server 2025
- (And newer)

MAY WORK, BUT MAY NOT - NO GUARANTEE

- Windows 7
- Windows 8
- Windows 8.1
- Windows Server 2016
- Windows Server 2019

WILL NOT WORK

- Windows XP (and Older)
- Windows Vista
- Windows Server 2008 (and older)
- Windows Server 2012.

But to the point!

1. Download `win-x64.zip` and extract it
2. Move win-x64 directory anywhere you want

3. Enter `win-x64` directory and simply double click on VRSX.exe (it has no icon, yet)
4. You will receive a huge Windows SmartScreen popup: "Windows protected your PC - Microsoft Defender SmartScreen prevented an unrecognized app from starting". Simply click "More info" and then "Run anyway". I'm not paying Microsoft \$500/year for a code-signing certificate just for a beta. Screw it! ;)
5. You are more than welcome! Up and running.
6. Firewall: If a Windows Firewall popup appears, allow access for Private Networks.
7. Done! Your browser should open automatically. If not, go to 127.0.0.1:8085.
8. Activate your VRSX using licence key available under "MY Account" section in ADS-B.Pro RadarView

How to make a service?

Windows doesn't have a simple systemd, because... well, it's Windows. If you want VRS X to run in the background without a console window staring at you, you have two options:

The "Lazy" Way: Create a shortcut to VRSX.exe, press Win+R, type shell:startup, and drop the shortcut there. It will start when you log in.

The "Pro" Way (NSSM for old-but-bald Windows admins):

1. Download NSSM (Non-Sucking Service Manager). It's a tiny tool that turns any .exe into a real Windows Service.
2. Open Command Prompt as Admin.
3. Type `nssm install VRSX`.
4. Select VRSX.exe path and set the arguments to `--headless`.
5. BANG! It's now a background service that starts with the OS.