

03 — Traxcope User Guide

TraxProbe & Traxcope User Manual · Version 2.0.0 · Updated 2026-10-07

This chapter covers everything on the desktop side that is *not* specific to one view: installing Traxcope, creating and configuring probes, connecting and streaming, recording and reviewing sessions, workspaces, and the concepts shared by all views (Live vs Review mode, view synchronization). The individual views are covered in the feature chapters (10–22); TraxAI in Chapter 23.

0. Installing Traxcope

Supported hosts: **Windows 10 64-bit** and **Ubuntu 22.04+ (x86_64)**. Qt is bundled; you do not install Qt separately.

Windows

Run Traxcope-<version>-Setup.exe. Accept the EULA. **Full installation** includes the TraxProbe firmware SDK; **Application only** omits it. Launch from the Start menu.

	Path
Application	C:\Program Files\Embedya\Traxcope
TraxProbe SDK	C:\Program Files\Embedya\Traxcope\sdk\TraxProbe
App data, logs, recordings	%APPDATA%\Embedya\Traxcope

Uninstall from Windows **Apps & features**.

Checking for updates

When a newer version is published, Traxcope shows a status-bar banner: **Traxcope X.Y.Z available — Click to update**. You can also use **Menu → Check for Updates...**, or **Preferences → Updates**.

Choosing **Update now** downloads the installer for this host. Traxcope closes, the installer upgrades the existing copy, and Traxcope reopens. Your license, recordings, and workspaces stay in the per-user data directory (Windows: AppData; Ubuntu: ~/.local/share/Embedya/Traxcope).

- **Windows** asks for administrator permission (UAC) and runs

Traxcope-<version>-Setup.exe over C:\Program Files\Embedya\Traxcope.

- **Ubuntu** downloads traxcope_<version>_amd64.deb when this copy was installed with dpkg, otherwise Traxcope-<version>-linux-x86_64.run. A system install under /opt prompts for your password (PolicyKit). A --user install under ~/.local does not need root.

Later hides the banner until the next launch. **Skip this version** stops reminding you until a newer version than the one you skipped appears.

Automatic checks at startup can be turned off under **Preferences → Updates**. Air-gapped machines that set TRAXCOPE_DISABLE_TELEMETRY never contact the update server.

Ubuntu

Install **either** the Debian package or the self-extracting installer:

```
sudo dpkg -i traxcope_<version>_amd64.deb
sudo apt-get install -f          # pulls missing system libraries if dpkg reports them
sudo bash Traxcope-<version>-linux-x86_64.run
```

Without root: bash Traxcope-<version>-linux-x86_64.run --user --accept-eula.

Launch with traxcope or from the application menu.

	System (.deb / default .run)	Home (--user)
Application	/opt/Embedya/Traxcope	~/ .local/opt/Embedya/Traxcope
TraxProbe SDK	.../sdk/TraxProbe	.../sdk/TraxProbe
App data, logs, recordings	~/ .local/share/Embedya/Traxcope	same
Remove	sudo apt-get remove traxcope	~/ .local/opt/Embedya/Traxcope/uninstall.sh

Serial ports. Add your user to dialout, then log out and back in, or UART connect will fail even when the port appears in the list:

```
sudo usermod -aG dialout "$USER"
```

J-Link / RTT. Install the SEGGER J-Link software for your OS. Traxcope loads the J-Link library from the SEGGER J-Link folder (JLink_x64.dll on Windows, libjlinkarm.so on Linux) — set **Preferences → Tools → J-Link folder**: if auto-detect misses it. Empty means the newest installed version. UART and TCP need no extra host software beyond the serial/network stack (on Ubuntu, dialout as above).

ST-LINK / RTT. Traxcope drives the ST-LINK itself; no OpenOCD or STM32CubeProgrammer is needed to stream. On Windows the ST-LINK needs ST's USB driver (STSW-LINK009), which STM32CubeIDE and STM32CubeProgrammer install. On Linux the installer adds a udev rule (Chapter 01).

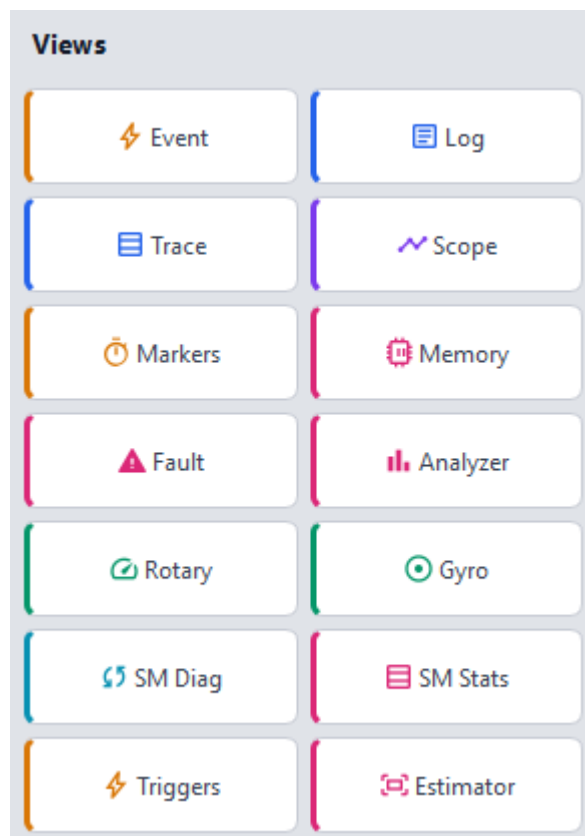
First launch and license

Streaming live data is license-gated. On first launch, use **License** to activate with the key from Embedya, or start the free trial. A machine with no internet at all activates by signed file exchange instead, and teams can lease a shared seat from a license server — both in Chapter 04. When the trial ends Traxcope keeps running as the free **Community Edition**: 3 minutes of live streaming per rolling hour, one connected probe at a time, and no new recordings (existing recordings still open and export). It never hard-locks. See also Chapter 01.

1. The main window

Traxcope is built around **dockable views**. The left sidebar hosts four panels:

Panel	Purpose
Probes	Create, configure, connect, stream, and record probes
Recordings	Load and manage recorded sessions for offline review
Faults	Archived crash dumps, one card per fault; a badge counts new ones — see Chapter 20
Views	Open analysis views (Event, Log, Trace, Scope, Markers, Memory, Fault, Analyzer, Rotary, Gyro, SM Diag, SM Stats, Triggers, Estimator)



Views panel listing all available view types

Screenshot: Views panel with the full list of view types

A view your license does not include is drawn faded, with the tooltip *Not included in your license. Click to see which license has it.* Clicking it tells you which license includes it.

Views can be docked, tabbed, floated, and opened in multiple instances (e.g. two Scope views watching different variable sets). The complete layout is saved in the workspace (§6).

2. Probes

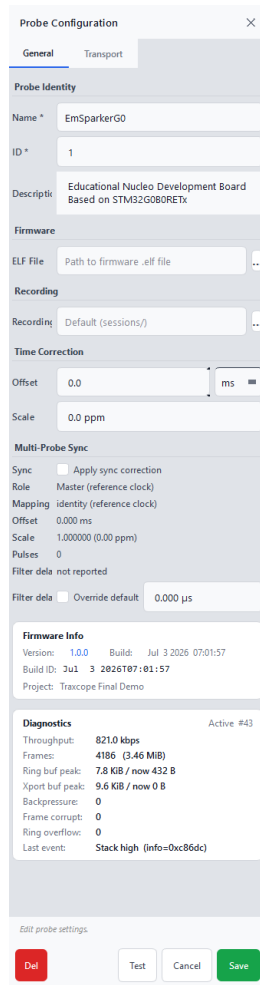
A **probe** in Traxcope represents one connection to one device running TraxProbe firmware.

2.1 Creating and configuring a probe

Sidebar → **Probes** → **Add**. Or import an existing configuration file via **Import** (.traxcfg, a JSON file you can keep in version control and share with your team). The counterpart is right-click on a probe → **Export Config...**, which saves the probe's current configuration as a .traxcfg you can hand to a colleague.

The probe configuration has four tabs: **General**, **Transport**, **Trace**, and **Streams**.

General tab



Probe configuration, General tab

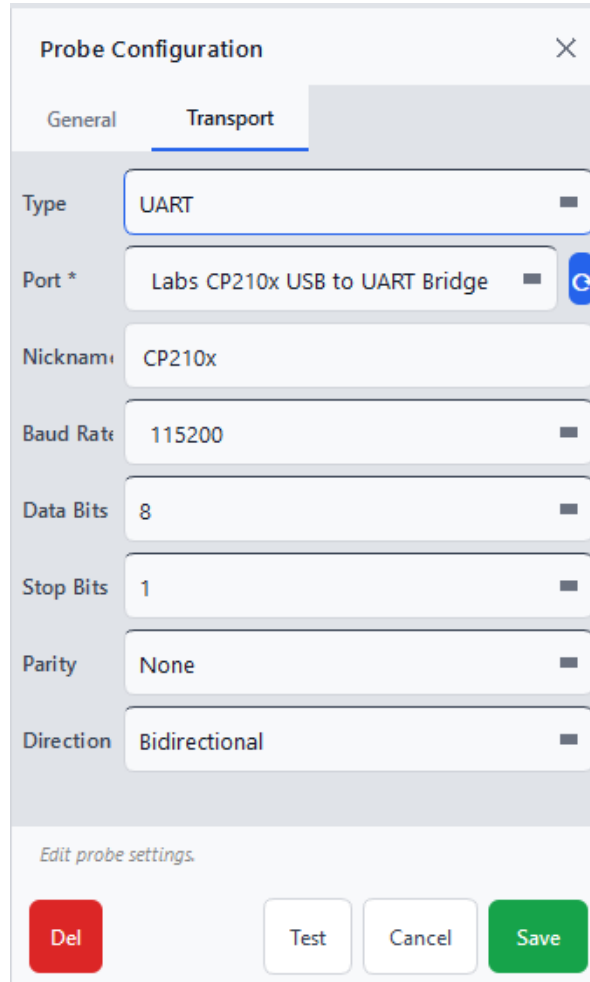
Screenshot: Probe config General tab — name, ELF path, firmware info card, diagnostics card

- **Name / description** — how the probe appears throughout the UI.
- **ELF file** — path to the firmware .elf. Traxcope reads the .trax_* metadata sections from it: log format strings, variable names/units/colors, marker deadlines, state machine tables. Mandatory if the firmware was built with TRAX_META_ELF_ONLY; recommended always.
- **Recording directory** — optional override of where this probe's recordings are stored.
- **Time correction** — manual time offset and scale (ppm) applied to this probe's timestamps; changes apply live while streaming.
- **Multi-Probe Sync** — the workspace-wide sync correction toggle plus a read-only status panel (role, mapping, offset/scale, pulse count, filter delay) for this probe — see Chapter 18.
- **Firmware info card** — fills in live after connection: project name, version, build ID as reported by the firmware (TRAX_CFG_PROJECT_NAME / _BUILD_VERSION / _BUILD_ID). Use the build ID to verify the ELF matches the running firmware.
- **Diagnostics card** — live firmware self-diagnostics (throughput, ring-buffer usage, drop/corruption counters), updated every second from the device's diagnostic reports.

Transport tab

Four transport types:

Transport	Settings	Pairs with firmware
RTT	Probe (J-Link or ST-LINK, ☺ lists both), speed; J-Link also device name and interface; <i>Advanced RTT</i> : up/down channels, control block address (0 = search), reset & run on connect	TRAX_TRANSPORT_RTT (library default)
UART	Serial port (refreshable list, nicknames), baud rate, data/stop bits, parity. Windows: COMn. Linux: /dev/ttyUSB*, /dev/ttyACM* (membership in dialout required — \$0)	Custom UART transport
TCP	IP address, port, timeout	Custom TCP transport
UDP	IP address, port, optional fixed local port (default: auto)	Custom UDP transport



Probe Configuration [X]

General **Transport**

Type: UART

Port *: Labs CP210x USB to UART Bridge [G]

Nickname: CP210x

Baud Rate: 115200

Data Bits: 8

Stop Bits: 1

Parity: None

Direction: Bidirectional

Edit probe settings.

[Del] [Test] [Cancel] [Save]

Probe configuration, Transport tab with UART selected

Screenshot: Probe config Transport tab, UART selected, COM port list open

Test Connection verifies the transport before you save.

RTT through an ST-LINK

Pick an ST-LINK in the **Probe** list and the RTT transport reads the target through it: the on-board ST-LINK of a Nucleo or Discovery board, or a stand-alone one. Traxcope drives the probe itself, so it needs no target device name and attaches without halting the core.

Probe	Fastest SWD clock	Clean RTT throughput
ST-LINK/V2, V2-1	4 MHz	about 125 KB/s
STLINK-V3 (V3E, V3SET, V3MINIE, V3PWR)	24 MHz	about 460 KB/s

The figures were measured on Windows with an ST-LINK/V2 and an on-board STLINK-V3E; the other models in each row use the same USB speed and should be close. Above these figures the target's RTT buffer fills and TraxProbe marks the lost interval as a data gap; nothing is lost silently on the PC side. The **Speed** field is an upper limit: the probe uses its closest rate at or below it. Selecting an ST-LINK raises the 4000 kHz J-Link default to 24000 kHz. In a virtual machine, USB pass-through can halve the throughput.

The list marks a probe Traxcope cannot use yet:

Marker	Meaning
- in use	Another program holds the probe (an STM32CubeIDE debug session, STM32CubeProgrammer, OpenOCD). Close it; the ST-LINK serves one program at a time.
- no driver	Windows: ST's ST-LINK USB driver is not installed (STSW-LINK009).
- no permission	Linux: the udev rule is missing; see Chapter 01.

Things worth knowing:

- **Reset & run on connect** (Advanced RTT, on by default) resets the target. The firmware boots, and Traxcope finds the RTT control block again. With it off, Traxcope attaches to the running target. A core a previous debug session left halted is resumed.
- The control block is found by searching target RAM, which takes well under a second. Set **Block Address** to the ELF's `_SEGGER_RTT` address only if your RAM is outside the usual Cortex-M ranges.
- ST-LINK/V2 firmware older than V2J22 cannot change the SWD clock and runs at 1.8 MHz; older than V2J15 is refused. Update it with STM32CubeProgrammer (*Firmware upgrade*).
- An ST-LINK reflashed with SEGGER's J-Link firmware is listed as a J-Link and uses the J-Link path.
- On dual-core parts (STM32H7 dual-core, STM32MP1) RTT is read on the core behind access port 0 only.
- **Flashing:** in *Build & Deploy*, a deploy command can name the same probe with `${STLINK_SERIAL}`, for example `STM32_Programmer_CLI -c port=SWD sn=${STLINK_SERIAL} mode=UR -w "${ELF}" -v -rst`. Keep **Disconnect while deploying** on, so Traxcope releases the probe while the flasher uses it.

Trace and Streams tabs

Trace tab — *RTOS Trace Groups*: one checkbox per group of events the probe traces. **Interrupts** and **Main loop** exist on any firmware. The RTOS section adds Task switches, Task states, Priority changes, Semaphores & mutexes, Message queues, Software timers, Task notifications, Stream & message buffers, Event groups, Heap & stack usage, and Low power (tickless idle); it is hidden when the probe reports bare metal.

- A change applies at once while connected, and again at every session start.

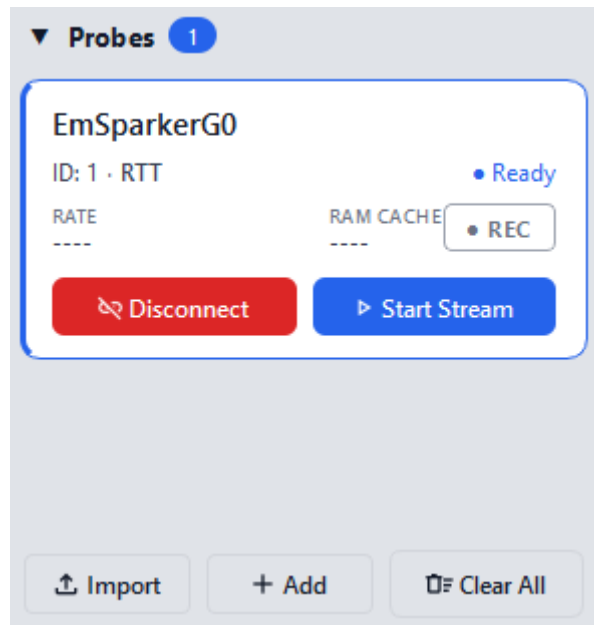
- Untraced periods show as empty lanes. Task and object create/delete are always traced.
- The probe reports which groups it can switch when a session starts. A group it cannot switch at run time is greyed out. The line below the boxes says which groups the probe does not trace.

Streams tab — the VAR streams the probe reported, listed once it starts a session (Chapter 12). Each row shows channel count, sample rate, bandwidth, and state (*sending, muted by Traxcope, stopped by the application*).

- Uncheck a stream to stop the probe sending it; it applies at once. Use this to free link bandwidth for the streams you are looking at.
- Sampling on the target is untouched. The muted span shows as a gap.
- The selection is kept for this firmware build only.

2.2 Connect → Stream → Record

Each probe card exposes the three-stage pipeline:



A probe card showing Connect / Stream / Record buttons

Screenshot: Probe card with Connect, Stream, Record buttons

- 1. Connect** — opens the transport and performs the session handshake. The firmware's `SESSION_START` header arrives, carrying firmware info, timestamp parameters, sync role, and (in `TRAX_META_IN_FLASH` mode) all static metadata.
- 2. Stream** — sends the start-streaming command. The firmware gates producers, waits for the ring to go idle, then **clears the application ring and the transport TX buffer** so the new session starts empty. `SESSION_START` is sent next; live views begin updating from that point. Events that happened before you clicked are not flushed to the host.
- 3. Record** — writes everything arriving from this probe to disk. You can toggle recording on and off during a live session.

Health indicators on the card warn about recording overflow, blocked drains, and similar conditions. If the firmware stops streaming on its own (buffer overflow, transport failure, heap exhaustion), Traxcope displays the stop reason reported by the device.

3. Recordings

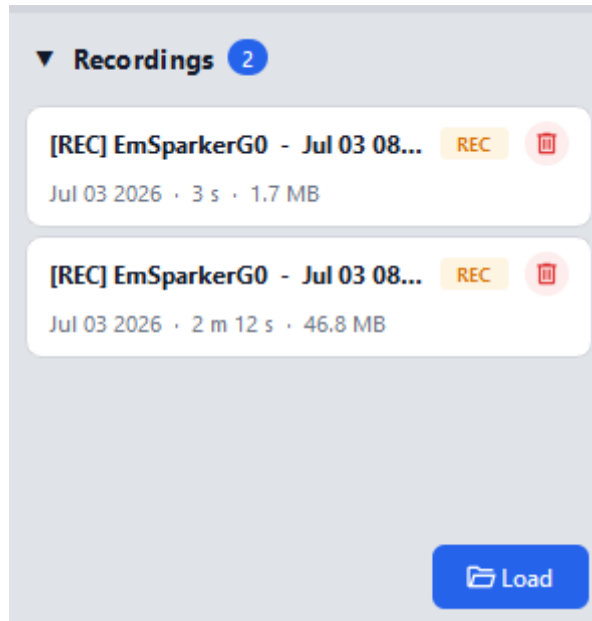
3.1 Creating a recording

Click **Record** on a connected, streaming probe. Traxcope writes chunked trace data plus multi-resolution level-of-detail (LOD) files into a per-probe session folder (default under the application data directory, configurable per probe), described by a `session.json` / `recording.json` manifest.

LOD files are what make review fast: views can render hours of data zoomed out without touching every raw sample, and progressively load detail as you zoom in.

3.2 Reviewing a recording

Sidebar → **Recordings** → **Open / Load**, then pick the recording folder. The recording is registered alongside live probes — views can display it exactly like a live probe, but in Review mode.



Recordings sidebar with one loaded recording and the detail panel

Screenshot: Recordings panel + detail side panel with metadata and time correction fields

The detail panel shows name, source probe, date, duration, size, chunk and LOD info, and offers:

- **Time offset (ms)** and **scale (ppm)** — manual time correction, useful to align recordings that were captured without hardware sync.
- **Remove** — unregister the recording (files stay on disk).

For multi-probe recordings, sync anchors captured during the session are persisted in the manifest, so synchronized timelines survive into offline review (Chapter 18).

4. Live vs Review — the one concept every view shares

Every data view operates in one of two modes, switchable from its toolbar:

	Live	Review
Data source	Streaming probe, follows incoming data	Frozen window into the store (live session history or a loaded recording)
Scrolling	Auto-follows latest data	Free navigation: pan, zoom, jump, search
View sync	Not participating	Fully participating

Typical workflow: watch live, spot something odd, switch that view to Review, navigate to the anomaly, then bring other views to the same instant via view sync.

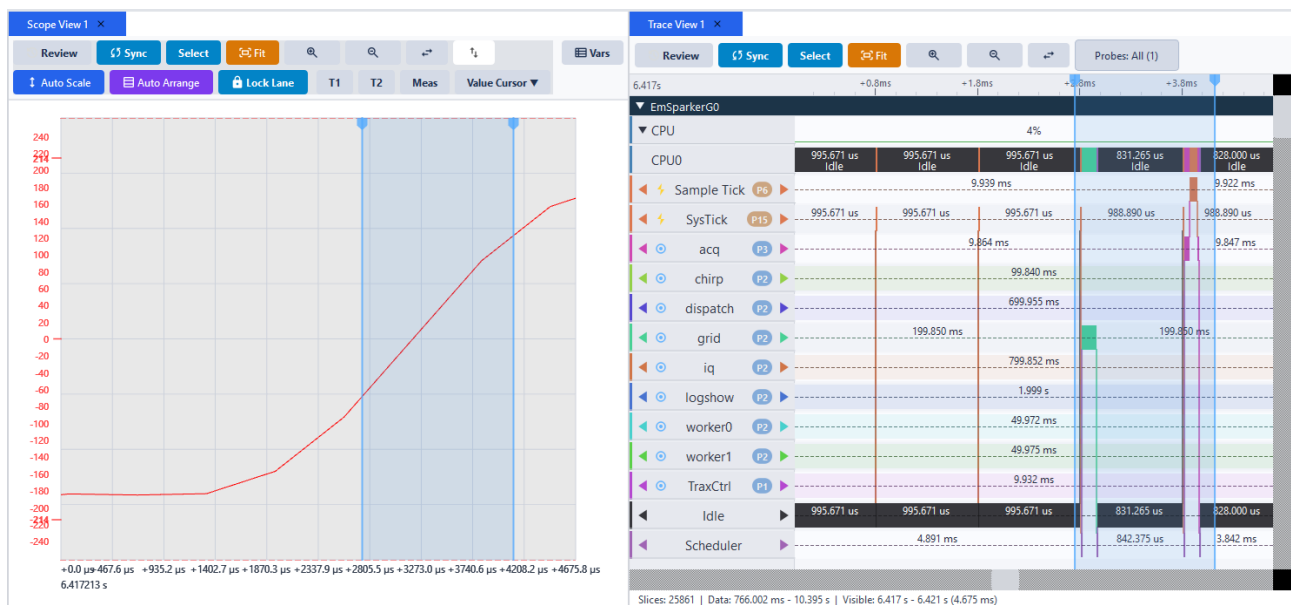
In the Trace View a **trigger firing** automatically drops the view out of live mode and centers on the trigger point (Chapter 17).

5. View synchronization (ViewSync)

Views that show time-based data (Scope, Trace, Log, Event, Marker Stats, SM Diagram, Spectrum Analyzer...) can share their navigation through the sync bus. Each view has a **Sync** toggle in its toolbar.

What is shared between synced views in Review mode:

Shared item	Meaning
Time selection	A single instant (light-blue marker line). Click in one view, every synced view jumps to and marks the same instant
Range selection	A light-blue highlighted range (e.g. T1–T2 cursors in Scope, a selected interval in Marker Stats)
Hover cursor	Live mouse-position tracking across views
Visible range	Pan/zoom one view, the others follow



Two views side by side with a synchronized time selection

Screenshot: Scope and Trace views side by side, same light-blue time selection in both

Row-based views participate too: selecting a log line or an event row broadcasts that timestamp to the bus, so you can click a suspicious error log and instantly see what every task was doing at that moment in the Trace View.

Live-mode views neither send nor receive sync — switch to Review to join the conversation.

Per-view **probe filters** (the probe selector popup in each view's toolbar) control which probes' data a particular view instance shows — so two Trace Views can watch two different boards while staying time-synchronized.

6. Workspaces

Traxcope automatically saves your entire working environment on exit and restores it on the next launch:

- All probe configurations
- The dock layout and all open views with their per-view settings (Scope channels, filters, sync state, ...)

- Sidebar state and window geometry

You always come back to exactly where you left off — no manual saving required. To share a setup with a colleague, export the probe configurations (§2.1) and hand over the `.traxcfg/ELF` files.

7. Exports

Export capabilities are view-specific:

7.1 Export Data... (raw samples, events, logs)

Menu → Export Data... writes the underlying data of one source — a live probe or a loaded recording — to CSV.

The dialog is **non-modal on purpose**: its time range live-tracks the cross-view ViewSync TimeSpan, so you can leave it open, drag a span in any synced view, and watch the export range follow. **Use Selection** re-applies the current TimeSpan, **Full Range** takes the source's whole data range, and both fields can always be typed in by hand.

Choose	Options
Source	Any registered probe or recording
Time range	From the shared TimeSpan selection, the full range, or typed manually
Variable channels	Per-channel selection, with All / None
CSV layout	Long (one row per sample) · Wide (one column per channel) · Separate (one time_s, value file per channel)
Events / Logs	Exported as their own files alongside the variable data

One file is written per selected category, named from the base path you pick. The export runs on a worker thread with progress and cancel. Export works on every license, Community included. Data captured on Community (or with an expired license) is watermarked: each CSV starts with a comment line such as `# Traxcope Community — free edition`. A loaded recording keeps the license tier it was recorded with.

7.2 Per-view exports

View	Export
Marker Statistics	HTML report, PDF report, CSV verdicts, JUnit XML (CI integration), stats table CSV — see Chapter 15
State Machine Statistics	CSV — see Chapter 16

JUnit XML deserves a special mention: it turns marker deadline compliance into a test report your CI can consume, making "the control loop must finish in 600 µs" a regression-tested property.

8. Menu, Preferences, license, TraxAI

The **Menu** button holds, in order: **Toggle Sidebar**, **TraxAI...**, **Preferences...**, **Conversion Library...**, **Export Data...**, **Announcements...**, **Report a Bug...**, **Manage License...**, **Open TraxProbe SDK Folder...**, **Check for Updates...**, **About Traxcope**, and **Exit** (Ctrl+Q).

- **Toggle Sidebar** — shows or hides the left sidebar panel (§1).
- **TraxAI...** — opens TraxAI, the in-app assistant. It reads the current session (tasks, logs, variables, faults, recordings) and helps you interpret a capture. It can also work on files and run commands in your project; anything that changes something asks you first. Choose the model under **Preferences → TraxAI**: a local model (Ollama) or a cloud or company server. **Preferences → AI apps (MCP)** lets AI apps you already use (Claude, Codex, Copilot, ...) call TraxAI's tools. TraxAI is included in Standard and Professional licenses; it is not available on Community. Setup, privacy, and model choice: Chapter 23.

- **Preferences** — application-level settings (theme, behavior, updates; **Tools → J-Link folder:** for the SEGGER J-Link software, §0).
- **Conversion Library** — the machine-wide store of reusable **formulas** and **filters**, shared across every workspace and probe. A recipe kept here can be linked onto any variable of any probe; linked entries follow the library **live**, so fixing a calibration in one place fixes it everywhere it is used. Entries appear in the per-variable pickers (and in the Estimator Studio's Formula/Filter columns) prefixed **Library:** Detach a linked entry to make it a local copy. See Chapter 11.
- **Export Data...** — raw samples, events, and logs to CSV (§7.1).
- **Manage License...** — streaming live data is license-gated; the License dialog manages the trial, node-locked activation (online or by offline file exchange), and floating seats (hosted, or an on-premise license-server URL). First launch: §0. Everything about licensing: Chapter 04.
- **Announcements..., Report a Bug..., Open TraxProbe SDK Folder..., Check for Updates...** (§0), **About Traxcope** — the remaining entries.
- **Exit** — closes Traxcope and saves the workspace, like closing the window (§6).