

AXIO Reader

Quick Guide — Run, Step Run, Context and Audit Trail

AXIO executes processes described in an `.axim` file. What sets AXIO apart from an ordinary tool: **the process is the audit trail**. Every step is recorded as it happens — not reconstructed afterwards.

Two ways to run

AXIO offers two ways to execute the same process. Both produce the same evidence — they differ in **who sets the pace**.

	Run	Step Run
Purpose	Carry out the process. The normal case.	Follow the process — pause at every step and inspect the context as it changes.
How it runs	Runs from start to finish. Pauses only when the process asks for input (<code>await</code>).	Pauses after every step. You decide when it continues.
What you see	The result at the end — and the complete trail.	After every step: which element ran, and what changed in the context.
Typically for	Everyday use: carrying the process out once, cleanly.	Review, training, troubleshooting, signing off a new process.

In short: **Run** is for *doing*. **Step Run** is for *understanding* — it shows you why the process did what it did.

The context — what the process currently knows

The **context** is the running process's memory: a set of named values. Every step reads from it, and most write to it. It starts empty (or with the initial values defined in the flow) and grows with each step.

- The right-hand pane shows the context **as it stands right now** — not the final state, but the current one.
- References (refs) in the context are **clickable**: a web address opens in your browser, a file in its default application.
- When a step requires input (`await`), the process pauses and asks you. Your answer becomes a value in the context — and thereby part of the evidence.

In a step run the context is the real observation window: after each step you see exactly what changed.

The audit trail — the evidence

On the left, the **Trail** builds up: one line per executed step, in the order it happened. It is created *while* the process runs — it is not a report someone writes afterwards.

If a process completes in full and without error, AXIO also stores the run as an `.axiotrace` file. It contains:

- a unique **run identifier**, start and end timestamps, and the status
- the **identifier and version of the process** — which process, in which revision, was executed
- one entry per step, with the **state of the context** after that step

This answers what matters in an audit: *who executed which process, in which revision and when; what was entered; and how the result came about*.

Reordering steps — and saving

You can change the order of steps directly in the **Diagram**: grab a step with the mouse and drag it to its new place. Within its own section it moves freely; across a group boundary the move is **clamped** — the structure of the process stays untouched, only the order changes.

As soon as something has changed, the application shows ● **unsaved**. Two ways to commit it:

- **Save** (⌘/Ctrl + S) — writes the new order back into the open .axim file.
- **Save as...** (⇧ ⌘/Ctrl + S) — stores a revision under a new name. The original stays untouched.

“Save as...” is the clean way to try out a variant of a process without touching the approved revision. **Each revision carries its own version — and every run records in the evidence which revision it executed.**

In three sentences

- **Run** gets the process done. **Step Run** explains it.
- The **context** is what the process currently knows.
- The **Trail** is what it did — and that is the evidence.

AXIO Reader — Quick Guide • As of: July 2026

© 2026 AINA Technology GmbH. All rights reserved. AXIO, AXIO Reader and AXIO Studio are products of AINA Technology GmbH.

aina.technology

A full manual (including AXIO Studio) will follow. This quick guide describes the functionality as of the date given above; later revisions of the application may differ.