

AADC Tracer

Validation evidence from the code that actually ran

Executive summary · Dmitri Goloubentsev, MatLogica · dmitri@matlogica.com · October 2026

Model documentation describes what a pricing model is *meant* to do. Validation has to establish what the code *actually* does, and today that means people reading source code by hand. The AADC Tracer records the calculation as it runs in your production binary, ties every step to the source line that performed it, and hands that record to an AI agent. The agent builds the validation evidence from what was computed, and every claim it makes can be checked.

One switch, no code changes

Set `AADC_TRACE=1` and run your normal pricing program. The run produces the same prices as always, plus a complete record of the calculation: every arithmetic step, comparison and input, each attributed to its file, line and function. Without the switch, the program runs exactly as before; with it off, pricing speed is unchanged.

What the record gives you

- **Source attribution for every step.** Any number in the result can be traced back across classes and libraries to the market quotes and constants it came from, and the record proves which inputs it does *not* depend on.
- **Every input identified and documented automatically.** Market data, calibrated parameters and every hard-coded number buried in the code (tolerances, cut-offs, day-count conventions, solver brackets) are listed with their value and the exact line that set them.
- **Every branch found and explored.** Each market-dependent decision is listed with its distance to switching. For each one, the agent finds the inputs that flip it, re-records the unmodified program there, and documents how the model behaves on the other side. Branches that cannot be reached within the agreed input ranges are reported, not assumed safe.

An agent that works from evidence, not from reasoning

Asked to read source code, a language model infers what the code probably does, the way a person does mental arithmetic, and it can be confidently wrong. Given the record, the agent reads what the code *did*: the record is its calculator. The output is a validation report (specification, input classification, dependency map, branch map, benchmarks, sensitivities, findings) in which:

- **every claim is verifiable:** each statement cites the file and line it rests on, and each formula links to the source lines that compute it;
- **every result is reproducible:** the record and the re-recordings are kept, so a reviewer or a regulator can re-run any finding;
- **people keep the decisions:** the tool produces the evidence; validators judge model fitness and sign off.

On real code

One QuantLib swaption price: 74,428 recorded steps across about 40 source files, every step attributed to its source line; 140 hard-coded numbers that reach the price, three day-count conventions in one valuation, and 85 market-dependent branches, all found automatically from a single 17 ms recording.

Who it is for

Model validation and audit teams (evidence from the binary that runs in production, the focus of PRA SS1/23, Fed SR 26-2 and the ECB guide), model documentation (generated from what the code computed, so it cannot drift), and quant developers (trace a wrong number to its source; see exactly what a code change did to the calculation). C++ and Python.

See it on your models: dmitri@matlogica.com