

THE LICENSED BUILDING BLOCK FOR SAS MODERNIZATION

SAS Language Engine

Decades of DATA steps, PROCs and macros encode business logic nobody wrote down. This engine turns them into a precise, typed model — the foundation for inventory, data lineage, documentation and transpilation.

2018

PROCESSING SAS CODE
SINCE

23

PROC STEPS MODELLED

323

TYPED AST NODE
CLASSES

151

AUTOMATED TESTS

LEAVING SAS STARTS WITH UNDERSTANDING SAS

The pressure is familiar: license costs, a mandate to move to Python, Spark or a cloud platform, and a SAS estate no one dares touch. Before you can move it, you have to read it — every program, every macro, every PROC SQL. The SAS Language Engine, our SAS parser, does that mechanically: a typed AST with source positions, preserved comments and positioned issues instead of failures. You build the inventory, the lineage graph or the transpiler on top.

WHAT YOU GET

Macros made tractable

A complete parser of SAS would have to be a complete interpreter. Our pragmatic, best-effort macro parsing recognises macros without expanding them, handles nested and parameterized definitions, and parses bodies lazily.

The DATA step, in full

28 statement kinds — ARRAY, SET, MERGE, KEEP/DROP/RENAME/RETAIN, IF/THEN/ELSE, DO loops, INPUT in four modes.

Built for documentation

Comments are first-class annotations attached to the nodes they describe.

PROC SQL in depth

The largest sub-grammar: SELECT INTO :macrovar, CREATE TABLE/VIEW/INDEX, all join types, window functions, CASE/CAST/EXISTS, and pass-through SQL captured and optionally re-parsed.

It never gives up on your code

Real codebases are messy: you get a partial AST plus positioned issues, never an exception that stops the run.

Proven on transpilation

We have built transpilers from SAS to SQL, Python, Spark and Databricks.

WHY STRUMENTA

Fixed price, guaranteed

The quote we give once scope is agreed is the final cost. One-year guarantee, extendable at your convenience.

Language engineering is all we do

Turin, since 2017. Clients include ADP, CGI, the European Space Agency.

Open foundations

Apache-2.0 StarLasu AST framework (Kolasu, Pylasu, Tylasu, SharpLasu). No proprietary IR, no lock-in.

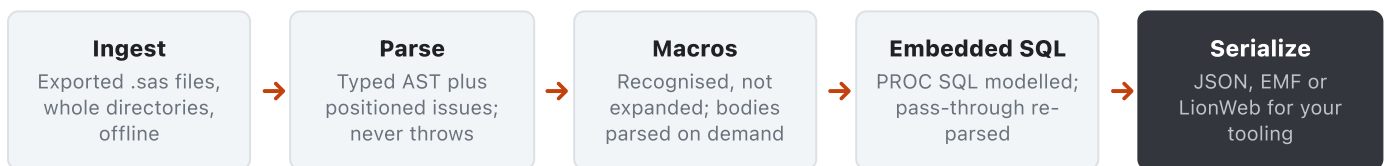
Try before you buy

Run your own code through the engine in the Strumenta Playground.

COVERAGE AND DELIVERY

Language	DATA step with 28 statement kinds; 23 PROC steps, with deep support for PROC SQL, DATASETS, FREQ, MEANS, SORT, TRANSPOSE and TIMESERIES; the full macro language (%MACRO/%MEND with parameters, %LET, %IF/%DO/%GOTO, %INCLUDE, %SYSFUNC, nested && variables, %STR/%NRSTR); LIBNAME, OPTIONS, FILENAME, TITLE, ODS.
Embedded SQL	PROC SQL, the largest sub-grammar: SELECT INTO :macrovar, CREATE TABLE/VIEW/INDEX, ALTER TABLE, MERGE, window functions, and pass-through CONNECT/EXECUTE with the native SQL captured and optionally re-parsed.
Serialization	JSON, EMF/Ecore (XMI), EMF-JSON and LionWeb 2024.1 over Protobuf; metamodel export. LionWeb output and the drop-in StarLasu pipeline component ship in v1.7.1.
Delivery	JVM library (Java 8+) for Kotlin and Java, plus a CLI fat jar with parse, jsonast, model, metamodel. Runs anywhere — no SAS installation, no SAS server connection: analyse exported .sas files offline, on laptops or CI servers.
Documentation	Dedicated docs site sas.strumenta.com : versioned AST reference, interactive AST viewer, and a cookbook for data lineage, program inventory, macro analysis and cross-file dependencies.

HOW IT WORKS



INTEROPERABILITY, NOT LOCK-IN

The metamodel is published as a LionWeb Language with 311 registered node classes and stable node IDs. LionWeb is the open interoperability initiative Strumenta co-founded with JetBrains MPS and itemis: SAS ASTs are consumable from Java, Python, C# and TypeScript tooling. A code formatter module is in early access.

Processing SAS since 2018 — clients in North America, Hong Kong and Europe. A transpiler built on this engine has processed tens of millions of lines of SAS.

Strumenta track record

ONE FAMILY OF ENGINES

Around 15 off-the-shelf parsers, one shared open-source AST framework, one LionWeb-compatible data model.

COBOL Language Engine

COBOL 85 and selected extensions, validated against the NIST COBOL-85 test suite.

RPG & DDS Language Engine

RPG IV, SQLRPGLE and DDS; battle-tested on 30+ million lines of production RPG code.

TALK TO US

Try it in the Strumenta Playground before you buy. Standard, Distribution and Service licenses, all with dedicated support and a one-year extendable guarantee. Full documentation: **sas.strumenta.com**

products@strumenta.com
strumenta.com