MigryX | SAS → Polars

SAS MODERNIZATION · 3-PAGE BROCHURE

BASE SAS · EGP · DI STUDIO · SAS VIYA · DATAFLUX → POLARS

Retire SAS. Run on high-performance Polars.

MigryX converts Base SAS, EGP, and DI Studio workloads directly to **Polars — the Rust-backed DataFrame library** that's 10–100× faster than pandas on a single node, with lazy evaluation, query optimization, and out-of-core streaming for datasets larger than memory.

99%	60%+	10×	100%
AI ACCURACY	COST SAVINGS	FASTER	ON-PREM CAPABLE

Everything we ingest from your SAS estate

PROGRAMS & CODE

- Base SAS (.sas)
- DATA · SET · MERGE
- PROC SQL · PROC steps
- SAS Macros · %INCLUDE
- Formats / Informats

PLATFORM & TOOLS

- DI Studio jobs
- Enterprise Guide (.egp)
- SAS Viya · CAS engine
- Stored Processes
- SAS Grid Manager

DATA SOURCES

- .sas7bdat datasets
- SAS/ACCESS connectors
- Oracle · SQL Server
- Teradata · DB2
- DataFlux dfPower

🚀

10–100× Faster

Rust-backed multi-threaded execution. Single-node throughput rivals Spark for analytical workloads.

💰

No Cluster Required

Process datasets larger than memory on a single machine. Eliminate Spark cluster overhead for many workloads.

🤖

Merlin AI Augmentation

Resolves macros, runtime parameters, and ambiguous logic. Pushes accuracy to 99%.

📊

Audit-Ready Validation

Row-level + aggregate matching. Side-by-side parity proof for go-live confidence.

Schedule a Demo → Email us →

TRANSLATION MAPPING

Every SAS construct. Native Polars output.

Deterministic mapping from SAS source to Polars-native targets. Every transformation auditable, every dependency preserved, every macro expanded. 95%+ parser accuracy out of the box, 99% with Merlin AI augmentation.

SAS CONSTRUCT	POLARS OUTPUT	ACCURACY
DATA step · SET / MERGE	pl.concat() · join() · with_columns()	99%
PROC SQL · pass-through	Polars SQL context · pl.SQLContext	99%
SAS Macros · %macro	Python functions with Polars expressions	95%+
PROC SORT/MEANS/FREQ	sort · group_by.agg · value_counts	99%
PROC FREQ / TABULATE	pivot · group_by · describe	99%
.sas7bdat datasets	Parquet · Arrow · streaming scan	99%

NATIVE POLARS OUTPUTS

Production-ready Polars from day one.

COMPUTE & LOGIC

- Polars DataFrames
- LazyFrames
- Streaming queries
- Expression API
- SQL context
- User-defined functions

STORAGE & PIPELINES

- Parquet · Arrow · IPC
- CSV · NDJSON
- Delta Lake reader
- Iceberg reader
- S3 / GCS / ADLS scan

GOVERNANCE & AI

- MLflow integration
- OpenLineage hooks
- Reproducible pinned envs
- Local DuckDB
- Hugging Face datasets

10–100× Faster Than pandas

Rust-backed columnar engine with multi-threaded query execution. Single-node throughput rivals Spark.

Lazy Evaluation + Optimization

Build queries lazily; the optimizer fuses operations and pushes predicates. Like SQL planners for DataFrames.

Out-of-Core Streaming

Process datasets larger than memory via streaming. No need to provision a Spark cluster for analytical workloads.

PyFluent Portability

Same logic runs on Polars locally and on Snowpark/PySpark in production via MigryX PyFluent.

5-STAGE METHODOLOGY

From SAS estate to production Polars — in five proven steps.

Every migration follows the same five-stage pipeline — fully automated, runs entirely in your environment, produces audit-ready evidence at each stage.

1 Analyze
Inventory SAS programs, EGP, DI jobs. Visual lineage + risk scoring.

2 Convert
Parser-driven SAS → Polars translation. Auto-documentation included.

3 Execute
Visual execution on Polars with centralized logs and orchestration.

4 Validate
Row-level + aggregate matching SAS legacy ↔ Polars.

5 Go-Live
Cutover with hyper-care monitoring and Merlin AI optimization.

99%	10-100×	60%+	🚀
CONVERSION ACCURACY	FASTER THAN PANDAS	COST SAVINGS	RUST-POWERED

RUN THE PILOT YOURSELF

MIGRATION READINESS

Discovery & Insights

1 wk DURATION · 100K LOC DISCOVERY

- ✓ Inventory + dependency mapping
- ✓ Visual lineage + risk scoring
- ✓ Self-service · runs in your env

FULL PILOT · END-TO-END

Convert · Execute · Validate

4–6 wk DURATION · 10K LOC CONVERSION

- ✓ Discovery + pilot conversion
- ✓ Data matching + validation
- ✓ Enterprise data workflows

LARGE SCALE PILOT

Enterprise Migration

2–4 mo DURATION · 100K LOC CONVERSION

- ✓ 1M LoC discovery scope
- ✓ Full project + JCL reports
- ✓ Production execution + cutover

Deployment · On-Premise · Air-Gapped

No outbound connections. Zero external API calls. Source code and data stay entirely on your infrastructure. Deploy in under an hour.

DOCKER · 8 CORES · 32 GB
Single-command install on Linux or Windows VM. Self-service pilot — no consultants required.

CLOUD · AWS · AZURE · GCP
Runs inside your VPC/VNet with private endpoints. Container images from your private registry.

K8S & ENTERPRISE
Kubernetes / OpenShift deployment. SOC 2 certified. Role-based access, full audit trails.

See your SAS code land on Polars — live, in 30 minutes.

No slides. No generic demos. Send a sample of your actual SAS code — Base SAS, EGP, DI Studio, or Viya — and we'll convert it, deploy it, and return column-level lineage. Free, no commitment, runs entirely in your environment.

Schedule Your Demo →