MigryX | SAS → pandas

SAS MODERNIZATION · 3-PAGE BROCHURE

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

Retire SAS. Run on Python & pandas.

MigryX converts Base SAS, EGP, and DI Studio workloads directly to **production-grade Python with pandas** — DataFrames, scikit-learn, statsmodels, NumPy, and SciPy — with Anaconda-governed environments and Jupyter notebook output for data-science teams.

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× Faster Migration

Automated parser-driven conversion. Eliminate 3–5 years of manual rewrites.

🐼

Hire from Anywhere

Modern data scientists know pandas. Eliminate the SAS talent cliff.

🤖

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 pandas output.

Deterministic mapping from SAS source to pandas-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	PANDAS OUTPUT	ACCURACY
DATA step · SET / MERGE	pd.concat() · merge() · assign()	99%
PROC SQL · pass-through	pandas query · pandasql · DuckDB	99%
SAS Macros · %macro	Python functions with parameters	95%+
PROC SORT/MEANS/FREQ	sort_values · groupby.agg · value_counts	99%
PROC FREQ / TABULATE	pivot_table · crosstab · describe	99%
.sas7bdat datasets	Parquet / Feather / CSV	99%

NATIVE PANDAS OUTPUTS

Production-ready pandas from day one.

COMPUTE & LOGIC

- pandas DataFrames
- NumPy · SciPy
- scikit-learn
- statsmodels
- xgboost · lightgbm
- Jupyter Notebooks

STORAGE & PIPELINES

- Parquet · Feather
- CSV · Excel · JSON
- SQLAlchemy · pandas.read_sql
- DuckDB local
- S3 / GCS / ADLS

GOVERNANCE & AI

- Anaconda Server
- conda environment.yml
- MLflow
- Model Registry
- Pinned reproducibility
- Air-gapped repo

Pythonic DataFrames

ETL logic converted to idiomatic pandas — readable, testable, hire-able from any Python data team.

Statistical Models Preserved

PROC REG, LOGISTIC, GLM, and PRINCOMP migrate to scikit-learn or statsmodels with equivalent semantics.

Anaconda-Governed

Pinned conda environment.yml per project. Air-gapped repository support. No public PyPI calls.

Jupyter Notebook Output

Self-documenting notebooks with inline lineage, transformation comments, and parameter docs.

5-STAGE METHODOLOGY

From SAS estate to production pandas — 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 → pandas translation. Auto-documentation included.

3 **Execute**
Visual execution on pandas with centralized logs and orchestration.

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

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

99%	10×	60%+	🐼
CONVERSION ACCURACY	FASTER THAN MANUAL	COST SAVINGS	PYTHONIC OUTPUT

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 pandas — 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 →

python.migryx.com · hello@migryx.com · (617) 512-9530