ON EBX™

EBX 6.3 · DATA CONNECTOR

INTEGRATION WITHOUT CODE

Connect EBX to every system your data lives in.

The EBX Data connector links governed master data to CRMs, data platforms, message buses, registers and APIs. Connections, data flows and field mappings are all configuration — designed graphically, tested safely, and audited end to end.

No code to deploy · Simulate before you write · Trust-aware

The Connection catalog in the Data connector perspective

111	11	35	23	0
connector types ready to configure	categories, from CRM to graph databases	built-in transformation steps	MCP tools for AI agents	lines of integration code to write

WHY IT MATTERS

Trusted data is only useful when it reaches the systems that run the business.

For business leadership

- **One trusted record, everywhere**
  Golden records reach Salesforce, Kafka consumers and the lakehouse the moment they change.
- **Better data from outside**
  Registers, data providers and address services enrich records — without silently overwriting what stewards decided.
- **Projects in days, not quarters**
  A new flow is a form and a mapping, not a development project.
- **Lower integration cost**
  One governed module replaces a patchwork of scripts and point-to-point jobs.

For technical leadership

- **Configuration, not code**
  Connections, flows and rules are EBX data: versioned, permissioned and changed without restart.
- **Safe by construction**
  Dry runs, change-only scopes, idempotent writes, replay of failed runs and a full synchronisation log.
- **Every entry point**
  User interface, REST API, workflows, scheduler, table triggers and MCP for AI agents.
- **Secure by default**
  Encrypted secrets or vault references, outbound host allow-list, OAuth 2.0, mTLS and SASL.

WHAT IT CONNECTS

Eleven categories, one way of working

24 Data platforms	Snowflake, Databricks, BigQuery, Redshift, Power BI, Collibra
20 Databases	PostgreSQL, Oracle, SQL Server, MySQL, SAP HANA, Teradata, any JDBC
18 Business applications	Salesforce, Dynamics 365, SAP, Workday, ServiceNow, HubSpot
12 Messaging and events	Kafka, RabbitMQ, JMS, Azure Service Bus, SQS, Pub/Sub
11 Cloud and file storage	S3, Azure Blob, Google Cloud Storage, SFTP, SharePoint
9 AI and machine learning	LLM enrichment as a low-trust source, EBX MCP servers
8 Corporate registries	GLEIF, Sirene, Companies House, Handelsregister, VIES
7 Graph databases	Neo4j, ArangoDB, Neptune, GraphDB, Cosmos DB Gremlin
6 Generic integration	Any REST, SOAP or GraphQL API; import an OpenAPI spec
4 Reference data	Address services such as Melissa, Loqate, Google, BAN
4 Risk and compliance	Dun & Bradstreet, Creditsafe, OpenSanctions
+ Your own	Add a connector type as a jar, without touching the module

Counts are the connections configured on the EBX 6.3 reference instance, by catalog category.

HOW IT WORKS

Three ideas carry everything.

EXTERNAL SYSTEMS

- CRM and ERP
- Lakehouse and warehouse
- Event streams
- Registers and data providers
- Any API

ON EBX™

Data connector

- **Connection** — endpoint, encrypted credentials, trust level
- **Data flow** — one operation on one EBX table, in or out
- **Mapping** — field links and transformation steps, drawn graphically
- **Provenance** — who wrote each field, with which trust
- **Log and deliveries** — every call, every run, replayable

GOVERNED IN EBX

- Master and reference data
- Match and merge golden records
- Steward review and conflicts
- Workflows and approvals
- Data catalog and lineage

User interface · REST API · Workflow step · Scheduled task · Table trigger · Webhook feed · MCP for AI agents

BUILT FOR THE SYSTEMS THAT MATTER MOST

Dedicated depth for Kafka, Salesforce, Snowflake and Databricks

MESSAGING · APACHE KAFKA

Master data as events.

Publish every change as a keyed event, or consume topics into governed tables.

- ✓ JSON by default; Confluent Avro with Schema Registry is one property
- ✓ Golden records published with their merged sources and deduplicated contacts
- ✓ Offsets committed only after EBX has written the batch; dead-letter topic
- ✓ PLAINTEXT, mTLS, SASL PLAIN / SCRAM and Confluent Cloud API keys

Acceptance: 7 of 7 Kafka scenarios passed — publish, non-committing preview, consume, restart without duplicates, malformed-message recovery.

BUSINESS APPLICATIONS · SALESFORCE

CRM and MDM, in step.

Accounts in, governed companies out — both directions from one connection.

- ✓ OAuth 2.0 client credentials or JWT bearer with a certificate
- ✓ Upsert by external ID, so a rerun never creates a duplicate
- ✓ Schema discovery from Salesforce describe, straight into the designer
- ✓ Catalog, profile and sample Salesforce objects for the data catalog

Verified live on a Developer Edition org: simulate, bounded live upsert, read-back in Salesforce, idempotent rerun, reciprocal inbound check.

DATA PLATFORMS · SNOWFLAKE

The warehouse, under governance.

Import, merge and catalog Snowflake through its SQL API — no JDBC driver to install.

- ✓ Programmatic access token, OAuth or key-pair (JWT) authentication
- ✓ Incremental imports from a watermark, reviewed by a steward before they land
- ✓ Trusted write policy: a weaker source raises a conflict instead of overwriting
- ✓ MERGE outbound with VARIANT columns; catalog, usage and column lineage in

Live on the reference instance: 631 records processed, 151 conflicts raised for stewards to decide.

DATA PLATFORMS · DATABRICKS

Data products, published to Unity Catalog.

Governed EBX data lands as managed Delta tables, with its meaning attached.

- ✓ OAuth token federation or machine-to-machine service principals
- ✓ All-or-nothing publication: staged, committed, recorded per run
- ✓ Column comments, tags and lineage written to Unity Catalog
- ✓ Read back into EBX through a maker / checker review

Verified on a live Databricks workspace: guided setup checks, published tables, comments, tags and lineage.

TRUST, PROVENANCE, CONFLICTS

A weaker source never overwrites a better one silently.

Every connection has a trust level. Every write is recorded in field provenance. When a less trusted source disagrees, its proposal becomes a conflict for a steward — on screen, in a workflow task, or through the API.

CONFLICT · INDIVIDUAL — **OPEN**

Employer name differs

CURRENT VALUE
Written by the policy system · trust 75

PROPOSED
Warehouse import · trust 40

DECISION
Take the proposed value · Keep the current value · Decide later

Illustrative conflict

- **Only if empty** — Fills gaps once. Never overwrites a value already there.
- **Always** — The external system is master of the field.
- **Trusted** — Writes when empty or when this source is at least as trusted; otherwise logs a conflict.

OPERATE WITH CONFIDENCE

Simulate first. Send only what changed. See everything.

Design graphically
Drag links, add transformation steps, test on a sample record.

Simulate, then apply
A dry run shows every planned change; a guard confirms real runs.

Change-only scopes
Not sent yet, changed since the last send, failures to retry.

Schedule it
One scheduled-task class for every flow, with scope and predicate.

Prove each delivery
Per record: last send, failures since, and the run that did it.

Measure every connection
Inbound, outbound, unchanged, conflicts and retries.

UP AND RUNNING IN MINUTES

From an empty catalog to events on the topic

1. **Create the connection**
   Pick the type, enter the endpoint and credentials. Test it — diagnostics never write.
2. **Create the data flow**
   Choose the operation, the direction and the EBX table it applies to.
3. **Design the mapping**
   Link fields and add transformation steps in the graphical designer.
4. **Run and schedule**
   Simulate, run for real, then put it on a schedule and watch the log.

Trusted data, connected.

See the EBX Data connector running against your own systems.

Request a demo