

ERP to live financial statements, *without an integration project.*

In Part 1 a finance team built statements, a dashboard and a planning app from three CSV files. This part replaces the files with live connections to the ERP, HRIS and data stores the company already runs — inside the planning product, with no second licence and no second project.

AUDIENCE	WORKED EXAMPLE	WHERE IT LIVES	READING TIME
Leaders of mid-size enterprises	NetSuite actuals, end to end	Integrations, inside EAConnect Planning	6 min read

Series Hours, Not Months · Part 2

Evidence EAConnect Integrations screens — Connections, Flow Builder, Processes, Planning Monitor — Sept 2026; connector catalogue as stated at time of writing

Standard product claims only where shown; opinion labelled as such

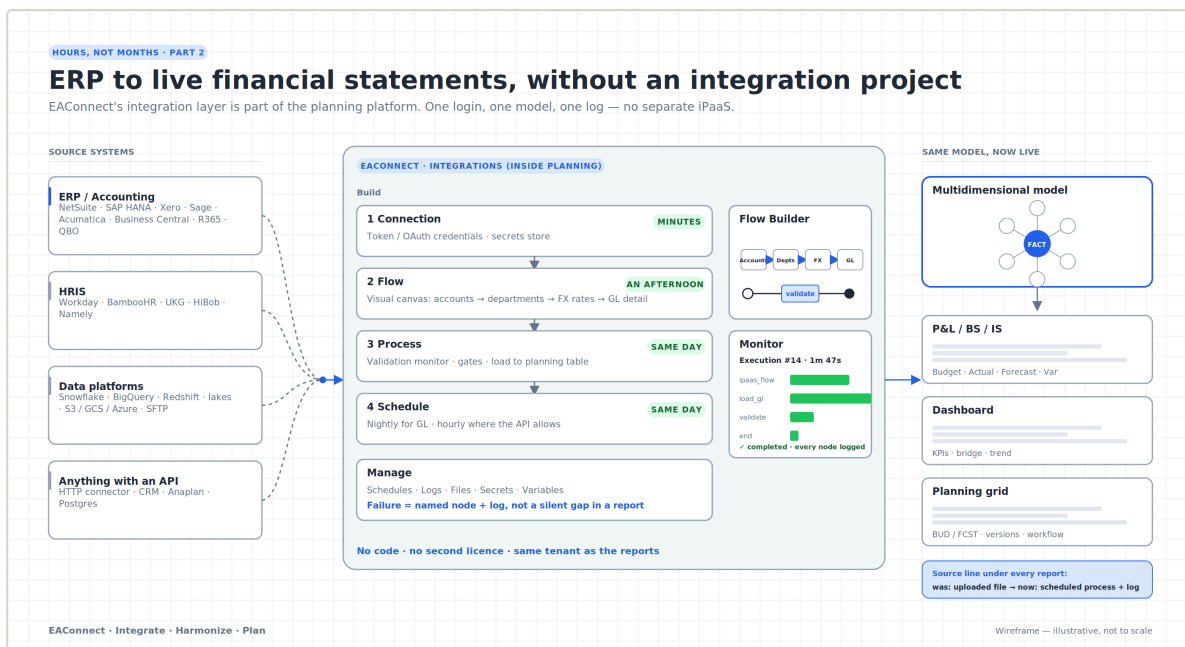


Figure 1. The integration layer, inside the planning platform

01 Files are a good start and a bad steady state

Starting from CSV exports was deliberate. It lets a finance team prove the outcome — a variance-analysed P&L, a dashboard, a budget grid — before anyone touches an integration. But files have a shelf life. Somebody has to export them, somebody has to load them, and the moment the ledger moves the report is stale. Month-end packs built on exports are correct for about a day.

The usual next step is where mid-market projects stall. Integration is bought as a separate product (an iPaaS licence), built by a separate skill set, and scheduled as a separate project. The analytics tool ends up waiting on the integration tool.

EACconnect was built the other way round: the integration layer is part of the platform. This article shows what that looks like in practice, what it takes to set up, and where the real effort goes.

02 What "built in" means here

Inside the Planning application there is an **Integrations** section with its own Monitor, Build (Connections, Tasks, Flows, Flow Builder) and Manage (Schedules, Logs, Files) areas. It is the same product, the same login and the same tenant as the reports. The target of an integration flow is a planning table or dataset in the same workspace; there is no export-to-file-then-import step between "integration" and "model".

The screenshot displays the EACConnect application interface. On the left is a dark sidebar with a menu containing: Monitor, Build (with a sub-menu arrow), Connections (highlighted with a blue bar), Agents, Tasks, Variables, Secrets, Flows, Transformations (with a sub-menu arrow), Run (with a sub-menu arrow), Manage (with a sub-menu arrow), and Documentation. The main content area is titled 'Connections' and features a search bar and a list of existing connections with columns for NAME and a description. The list includes: sftpTestTegria, SalesforceLargeDatabase, PostgresConnection, Sage, NetsuiteTest, AnaplanNewTest, NStest2, SageTestMW, AzureBlob, and AzureKinders. An 'Edit Connection' modal is open on the right, showing fields for: Name (filled with 'NStest2'), Description (filled with 'Netsuite2'), Connection Type (filled with 'Netsuite_Oauth1'), Account (empty), Client_Id (empty), Client_Secret (empty), Token_Id (empty), and Token_Secret (empty). Each field is preceded by a red asterisk indicating it is required. Below the modal, a blue banner contains the text: 'Creating a NetSuite connection — token-based auth, alongside Sage, Salesforce, Postgres, Anaplan, Azure Blob and SFTP'. The EACConnect logo is in the bottom left corner of the banner.

Figure 2. Creating a NetSuite connection

Connectors available at the time of writing include:

- **ERP / accounting** — NetSuite, SAP HANA, Xero, Sage, Acumatica, Microsoft Dynamics 365 Business Central, Restaurant365, QuickBooks Online (via the HTTP connector)
- **HRIS** — Workday, BambooHR, UKG, HiBob, Namely
- **Data platforms** — Snowflake, BigQuery, Redshift and other databases; data lakes; GCS, Azure Blob, S3 and SFTP storage
- **Other** — Salesforce, Anaplan, Postgres, and a generic HTTP connector for any system with a REST API

The list grows, so treat this as a snapshot rather than a ceiling. If a system has an API, the HTTP connector usually covers it.

03 A worked example: NetSuite actuals

Here is the sequence a finance-systems person follows to get GL actuals flowing from NetSuite into the planning model. Nothing below requires code.

1. Create the connection

A connection is a named set of credentials. For NetSuite that is token-based authentication: account ID, client ID and secret, token ID and secret. Secrets are stored in the platform's secrets store, not in the flow definition. The connection list in the screenshot above also shows Sage, Salesforce, Postgres, Anaplan, Azure Blob and SFTP connections living side by side.

What this actually takes: a few minutes in EAConnect, plus whatever time your NetSuite administrator needs to create the integration record and token on their side. That NetSuite step is often the longest part of day one, and it is outside EAConnect's control.

2. Build the flow

Flow Builder is a visual, BPMN-style canvas. The example below pulls master data: the account list, the department list and exchange rates, in sequence.

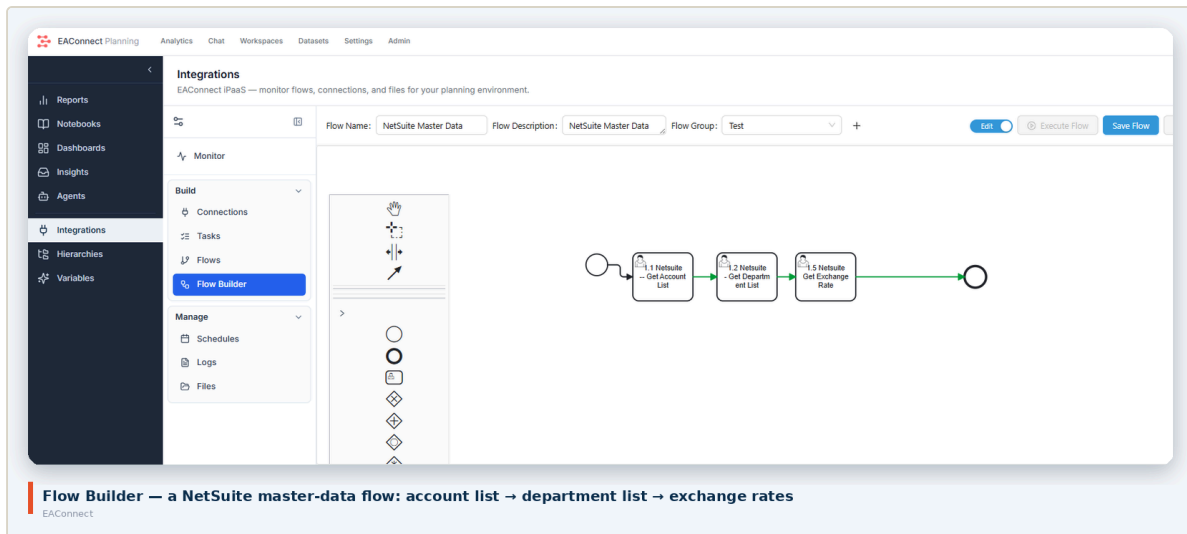


Figure 3. Flow Builder — NetSuite master data

Each step is a connector task with its own configuration (which object, which fields, which filters). Flows are grouped, versioned and can be executed on demand from the same screen while you are building them, so the loop of "run it, look at the rows, adjust" is short.

3. Compose the process

Integration flows are one node type in a broader **Process** model. The *Netsuite Actuals* process below runs the master-data flow, then a *Load NS GL Detail* step that takes the tabular source and writes it to a planning table.

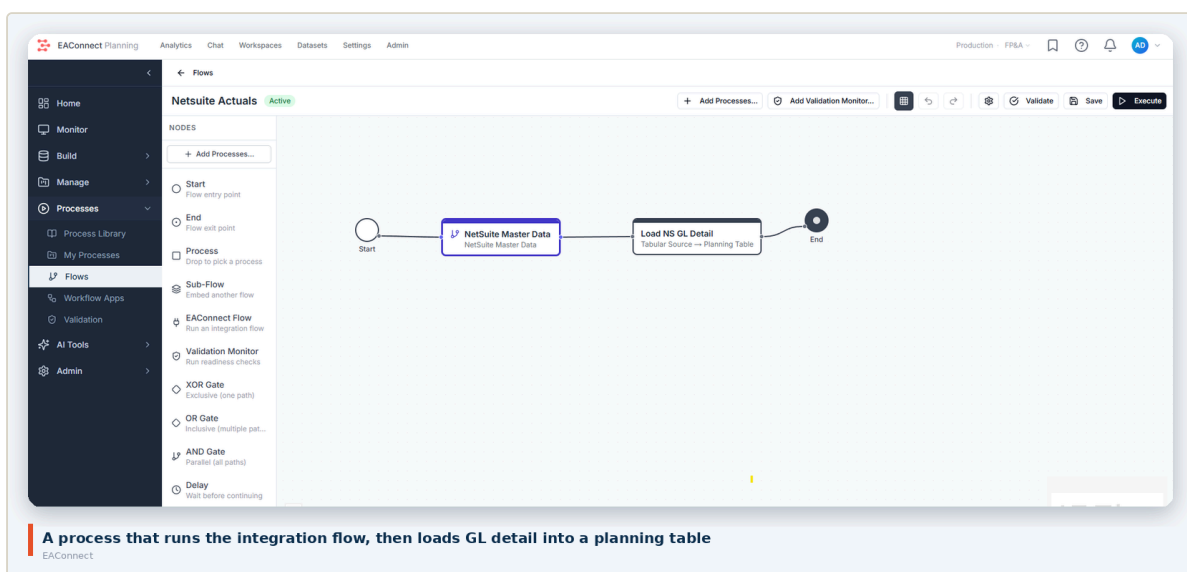


Figure 4. Process — NetSuite actuals

The node palette on the left is worth noticing: sub-flows, validation monitors (readiness checks before a load is accepted), XOR/OR/AND gates for branching and parallel paths, and delays. This is the same process engine that runs planning workflows later in the cycle, so the skill you build here transfers.

4. Execute and watch it

Every execution is recorded in the Planning Monitor with a trigger type, start time, duration and node-by-node status, plus a timeline view.

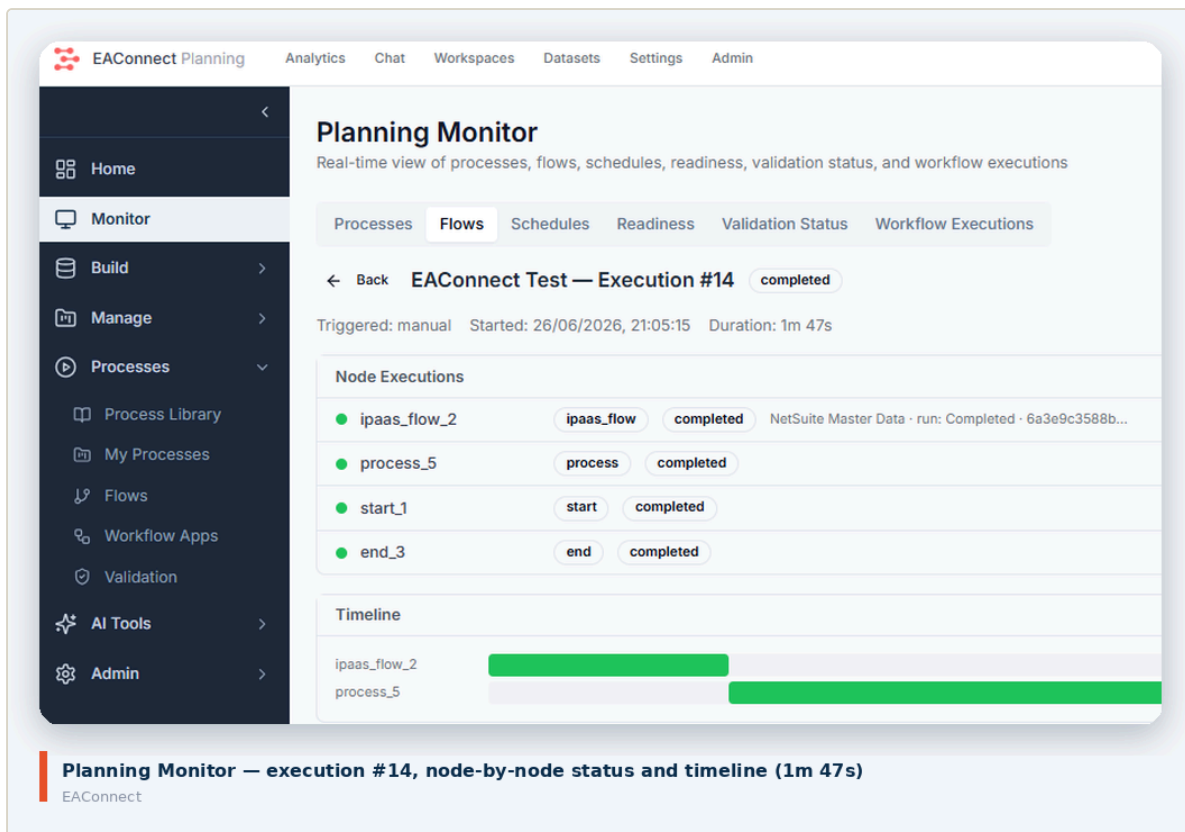


Figure 5. Execution monitor

Execution #14 in the screenshot ran manually and completed in 1 minute 47 seconds. When something fails — an expired token, a changed field, an API limit — it fails at a named node with a log, rather than as a silent gap in a report.

5. Schedule it

Once a run is clean, it goes on a schedule. Nightly is typical for GL actuals; hourly is possible for operational data where the source API allows it.

04 Honest time expectations

For someone who knows their ERP and has admin cooperation on the source side:

STEP	TYPICAL EFFORT
Source-side setup (integration record, tokens, permissions)	Depends on your ERP admin — minutes to a day
Connection in EACconnect	Minutes
Master data + GL detail flow	An afternoon, including test runs
Process with validation and schedule	Same day
Historical backfill	Governed by data volume and the source API's rate limits — plan for it to run overnight for multi-year history

The decisions that take longest are not technical: which subsidiaries, which accounts, which posting periods count as "closed", how to treat adjustments after close. EACconnect does not make those decisions for you; it makes them cheap to change once made.

05 Beyond the ERP

The same mechanism covers the other sources a planning model needs:

- **HRIS** (Workday, BambooHR, UKG, HiBob, Namely) for headcount, compensation and start/end dates — the input to personnel-cost planning, which is usually the largest opex line.
- **CRM** (Salesforce) for pipeline, feeding revenue forecasting.
- **Warehouses and lakes** (Snowflake, BigQuery, Redshift, GCS, Azure, S3) when the company already has a data platform and EACconnect should read from it rather than from operational systems directly.
- **Data replication** for keeping a full copy of a source table in sync, rather than extracting a report.

There is also an **MCP Servers** area under Connections. That is the channel through which the platform's AI agents reach data, and it is the subject of a later article.

06 Where this leaves the Part 1 outcome

The P&L, dashboard and planning grid from Part 1 do not change. What changes is the "source" line underneath them: from a file somebody uploaded to a scheduled process with a log. The team that spent day one proving the outcome on CSVs spends day two wiring in the ERP, and the report they showed the CFO yesterday is the one that refreshes tonight.

Next in the series: the features mid-market teams are usually told need a bigger platform — versioned hierarchies, multiple calendars, currency translation, broadcasting, insight agents — and what each looks like in EAConnect.

EAConnect Planning · Research library · Hours, Not Months

EAConnect Integrations screens — Connections, Flow Builder, Processes, Planning Monitor — Sept 2026; connector catalogue as stated at time of writing

1,303 words · 6 min read