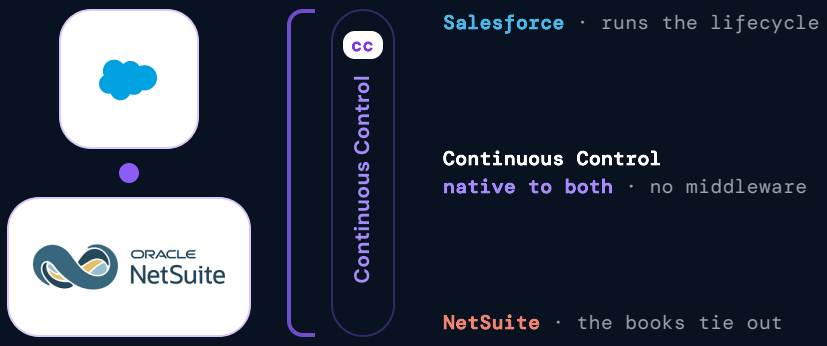
The lightweight path from Salesforce revenue to the NetSuite ledger.

The customer, and nearly everything they do, stays on Salesforce. Continuous Control posts transactions to the NetSuite general ledger, or runs revenue out of NetSuite, **with or without NetSuite ARM**. Prebuilt native apps on both platforms. No middleware, no custom build.



WHY IT WINS

It removes the "but our books are in NetSuite" blocker from Agentforce Revenue Management deals.

Every customer interaction runs on Salesforce, rep-driven or Agentforce. The full quote-to-cash lifecycle stays on platform, and the general ledger never moves. The integration stops being the reason the deal stalls.

THREE REASONS

Unlocks Agentforce Revenue Management adoption

The large NetSuite-finance installed base can adopt RCA/RCB without an integration project. The books objection disappears in the first call.

Keeps the lifecycle on platform

Billing, collections, and payments stay in Salesforce in every recommended method. More platform footprint, more Agentforce surface.

Lightweight, not another project

Prebuilt app on both sides, prebuilt mappings, no middleware. The easy attach to an Agentforce Revenue Management deal, not an integration project.

THE SPLIT

SALESFORCE	Quote · Order · Usage · Bill · Collect · Pay · **the whole lifecycle**
NETSUITE	General Ledger · **everything posts here** · Revenue and A/R optional

No integration black box, and nothing new to build

ARCHITECTURE

SALESFORCE

OWNS THE CUSTOMER · NATIVE PACKAGE

Salesforce record
Order · Invoice · Payment · Credit Memo

↓

Finance Staging
one row per record, status tracked

↓

Finance Sync Job
maps, validates, writes · one-click retry

CONTINUOUS

THE PREBUILT SYNC

Config fetch first
subsidiaries · currencies · countries · categories, so every transaction lands in the right place

↓

Order graph + A/R documents out
to NetSuite, matched on Salesforce IDs

↓

Ids, status + revenue back
on every record · recognized revenue to the Contract

NETSUITE

OWNS THE BOOKS · NATIVE SUITEAPP

ARM Contract → Orders → Lines
the mirrored order graph

↓

Performance Obligations
lines grouped by rule · Revenue Plans + Schedules

↓

Journal Entries → General Ledger
Dr Deferred · Cr Revenue, per period

Staging, jobs, status, and retry are all native to the Salesforce org. Already standardized on **MuleSoft**, Celigo, Boomi, or Workato? The same certified mappings ship as prebaked recipes on the middleware you own. **What the customer stops buying:** a middleware license, an SI integration build, the maintenance contract that follows it, a separate billing engine, and a separate revenue engine.

THE FOUR DEAL TYPES

SALESFORCE FOOTPRINT ← WHERE REVENUE, A/R, AND BILLING LIVE → NETSUITE FOOTPRINT

METHOD 1
Salesforce-First
VERSION: LEDGER

SF · Customer 360 + A/R + Revenue (RightRev)
NS · General Ledger

Integrating transactions to the ledger, everything else on Salesforce

+ Whole customer and revenue lifecycle on Salesforce; full Agentforce
– Revenue on Salesforce needs RightRev (separate purchase)

METHOD 2
NetSuite Revenue
VERSION: REVENUE

SF · Customer 360 + A/R subledger
NS · Revenue + General Ledger

Revenue out of NetSuite, with or without ARM, while A/R stays on Salesforce

+ Revenue in NetSuite without buying a Salesforce-side revenue engine
– Split subledgers: A/R on Salesforce, revenue in NetSuite

METHOD 3
Unified Books
VERSION: BOOKS

SF · Customer 360
NS · A/R + Revenue + General Ledger

Front office on Salesforce, books fully consolidated in NetSuite

+ Automatic: Continuous Control syncs RCA/RCB in; A/R, revenue and GL all in NetSuite
– Subledgers run in NetSuite, not on the Salesforce platform

METHOD 4 · CUSTOM
Back-Office Sync
VERSION: BACK OFFICE

SF · Customer 360 + Sales · status synced back
NS · Native billing + A/R + Revenue + GL

Customers set on billing natively in NetSuite, a custom path

+ Supported with a standard, certified sync (order out, invoice / A/R status back)
– Standard NetSuite limits & more NetSuite work; blocks a future Salesforce billing opportunity

The books (the General Ledger) always run from NetSuite. Revenue on Salesforce uses RightRev. **Methods 1–3 keep billing on Salesforce**, the recommended, fully-supported path.

Every object we provide a prebuilt mapping for

CUSTOMER & CATALOG

M1	**Account** the customer who signed	→ ACCOUNTID	**Customer** native NetSuite customer
M2	**Product** the catalog items sold	→ PRODUCT2ID	**Item** non-inventory sale item

AGREEMENT & ORDER GRAPH

M2	**Contract** the master agreement & term	→ CONTRACTID	**ARM Contract** customrecord_ct_agreement
M2	**Order · Amendment Order** original deal + later changes	→ ORDER ID	**ARM Order** customrecord_ct_arm_order
M2	**Order Product** a line on the order	→ ORDERITEM ID	**ARM Order Line** customrecord_ct_arm_order_line
M2	**Order Products, grouped** by item across the contract	→ DERIVED	**Performance Obligation** customrecord_ct_perf_ob

BILLING PLAN

M2	**Billing Schedule · Group · Period** the plan: what bills, and when	→ BS / BSG / BP	**ARM Billing Schedule / Group / Period** customrecord_ct_billing_schedule · _bs_period

INVOICES & CASH

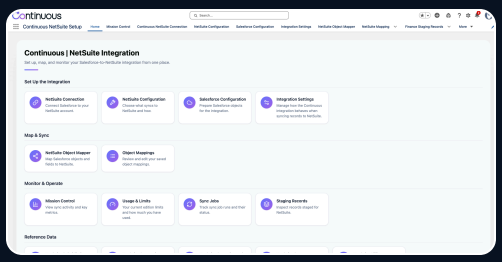
M3	**Invoice** a posted Salesforce invoice	→ INVOICE ID	**Invoice** native NetSuite invoice
M3	**Credit Memo** posted credit	→ CREDITMEMO ID	**Credit Memo** native NetSuite credit memo
M3	**Payment** cash received	→ PAYMENT ID	**Customer Payment** native NetSuite payment
M3	**Refund** cash returned to the customer	→ REFUND ID	**Customer Refund** native NetSuite customer refund
M3	**Invoice lines → obligations** links billed amounts to POBs	→ DERIVED	**Obligation Detail** customrecord_ct_ob_detail

BACK TO SALESFORCE · REVERSE SYNC

NS→SF	**Contract · Performance Obligations** CT_POB__c · Booked / Billed / Recognized	← WRITES BACK	**Performance Obligation** customrecord_ct_perf_ob

Badges show the method each mapping unlocks at; mappings are cumulative, so Method 3 includes everything. Derived records are built in NetSuite by the Continuous engine. **Every record matches on its Salesforce ID, so re-syncing never duplicates.**

See it before you show it

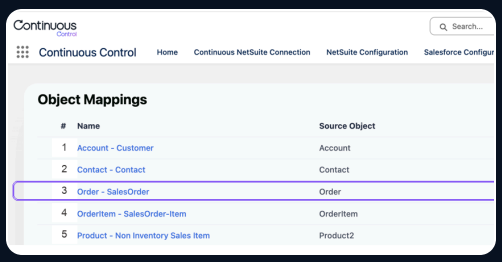


01 · THE PLACEMAT

Set up, map, and monitor from one place

The Continuous home in Salesforce: connect NetSuite, choose what syncs, map objects, and watch it run. Set Up · Map & Sync · Monitor & Operate · Reference Data, all on one screen.

Watch the demo →

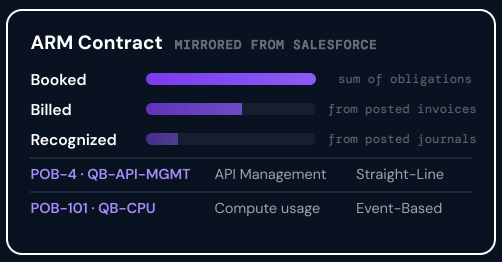


02 · PREBUILT MAPPINGS

Every object mapped out of the box

Account to customer, order to the ARM order graph, invoice to invoice. The mappings ship prebuilt and editable, with per-object toggles. No mapping spreadsheet, no field-by-field build.

Watch the demo →

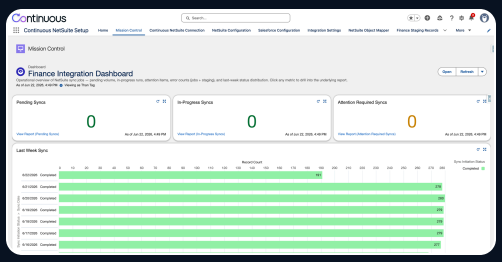


03 · THE NETSUITE OBJECTS

The books side: obligations and the waterfall

The SuiteApp holds the mirrored contract, groups lines into Performance Obligations, and rolls Booked / Billed / Recognized on every obligation and the contract. Revenue posts as journal entries to the GL.

Watch the demo →

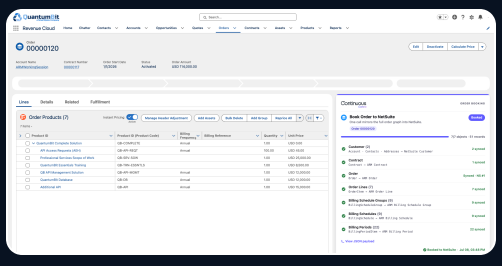


04 · MISSION CONTROL

Every transaction tracked to the books

Pending, in-progress, and attention-required syncs on one dashboard, with history. Errors are one click to retry. Nothing hidden in middleware; finance and admins see the same truth.

Watch the demo →



05 · BOOK ORDER, ONE CLICK

The whole order graph, in one call

From the Salesforce Order: customer, contract, order, lines, billing schedules and periods land in NetSuite in a single call, with per-object status and a deep link to the ARM Contract.

Watch the demo →

Versions match the deal type, one for one

THE FOUR VERSIONS

Ledger

VERSION 1 · METHOD 1

- Config fetch: subsidiaries, currencies, accounts
- Account → Customer
- Summary GL postings (one-way) with a reconciliation report

`Postings only; no Continuous objects in NetSuite`

Revenue

VERSION 2 · METHOD 2

- Everything in Ledger
- Native Continuous objects in NetSuite: ARM Contract / Orders / Lines / POBs / Revenue Plans + Schedules
- Rev-rec engine (or the native ARM bridge); A/R as summary JEs

`Adds the Continuous objects + revenue engine`

Books

VERSION 3 · METHOD 3

- Everything in Revenue
- A/R documents: invoices, credit memos, payments, refunds
- Obligation Details + the Billed waterfall

`Syncs the most objects into NetSuite`

Back Office

VERSION 4 · METHOD 4 · CUSTOM

- Standard certified sync only: order out, invoice / A/R status back
- Does not power native NetSuite billing; standard NetSuite limits apply
- Scoped per engagement

`We recommend billing on Salesforce`

QUALIFY IT IN ONE CONVERSATION

1 · Where should revenue run?

On Salesforce (with RightRev) → `Method 1`. In NetSuite, with or without ARM → `Methods 2-3`.

2 · Where should A/R live?

On Salesforce → `Methods 1-2`. Consolidated in NetSuite → `Method 3`.

3 · Set on native NetSuite billing?

That is `Method 4`, a custom path. Steer to billing on Salesforce when you can: it keeps the lifecycle (and the future billing opportunity) on platform.

WHEN YOU HEAR "OUR BOOKS ARE IN NETSUITE"

"The books stay in NetSuite."

Continuous Control is the certified, prebuilt connection between Agentforce Revenue Management and the NetSuite ledger, so the whole customer lifecycle can run on Salesforce. Ledger integration or NetSuite revenue, with or without ARM, on one certified app.

THE PLAY

Bring Continuous Control into your Agentforce Revenue Management deals.

1 · Spot the NetSuite objection → 2 · Qualify the method (3 questions) → 3 · Continuous demos with your team