
stratus
carta

SALESFORCE GO FOR REVENUE CLOUD

High Tech Order Orchestration

Salesforce Go for Revenue Cloud Series · Episode 3

Khaled El Halabieh

INTRODUCTIONS

Khaled El Halabieh

- Principal Technical Architect, Stratus Carta
- 10 years in Salesforce: Revenue Cloud and Steelbrick CPQ (Quote-to-Cash)
- 3 years on the Revenue Cloud Practice Lead team, Delivery Excellence (Salesforce)



stratus carta

OVERVIEW

Agenda

1. Order orchestration in two minutes
2. What the template installs, and how it runs
3. Demo: install it and run an order
4. Issues, limitations, and how I'd use it

ORDER ORCHESTRATION

IN TWO MINUTES

DYNAMIC REVENUE ORCHESTRATOR

From what you sell to what you deliver

Order	Decompose	Plan	Execute	Outcome
WHAT IT MEANS				
What the customer bought	What was sold becomes what to deliver and bill	The actions needed to fulfill your goods and services	Teams and systems do the work, in the right order	Delivered to the customer, and billed
IN SALESFORCE				
An order with commercial products	Decomposition rules map them to technical products	Fulfillment scenarios link products to step definitions	Auto tasks, manual tasks and milestones	Assets, billing schedules and invoices

WHAT IS THE

HIGH TECH ORDER ORCHESTRATION TEMPLATE?

HIGH TECH ORDER ORCHESTRATION TEMPLATE

A ready-made fulfillment design for high tech orders

- A Salesforce Go data pack for Dynamic Revenue Orchestrator
- Built for orders that mix project work, hardware assembly, software provisioning, billing and revenue scheduling
- Gives you technical products, fulfillment steps, dependencies and automations
- A starting point you adapt to your own products and processes

Demo path

Setup › Salesforce Go › Revenue Cloud
Filter: Data Packs
High Tech Order Orchestration › Set Up

Track the install

Go Accelerate › Solution Deployment Monitoring

Editions

Enterprise, Unlimited or Developer

TECHNICAL PRODUCTS

Six technical products, one per kind of work

- **Project**
- **Hardware Assembly**
- **Tenant Provisioning**
- **Software Provisioning**
- **Billing**
- **Revenue Schedule**

Each technical product has its own group of fulfillment steps. Teal: delivery work. Orange: billing and revenue.

WHAT'S IN THE DATA PACK

Seven building blocks, ready to adapt

Component	What it does
Technical products (6)	What your commercial products decompose into
Step definition groups	One per technical product, holding its steps
Fulfillment step definitions	Auto tasks, manual tasks and assetization steps
Step dependencies	Set step order, within and across groups
Product fulfillment scenarios	Map each technical product to its step group
Fulfillment workspace (1)	Organises the steps in the workspace designer

Not included: decomposition rules from your commercial products. You build those.

HOW IT RUNS

One order, four stages

01

Decompose

When the order is submitted, your decomposition rules map commercial products to the template’s technical products

02

Plan

Product fulfillment scenarios pick the matching step groups and build a plan that keeps their dependencies

03

Execute

People complete manual tasks and milestones; automated tasks call the billing flows; assetization steps create assets

04

Invoice

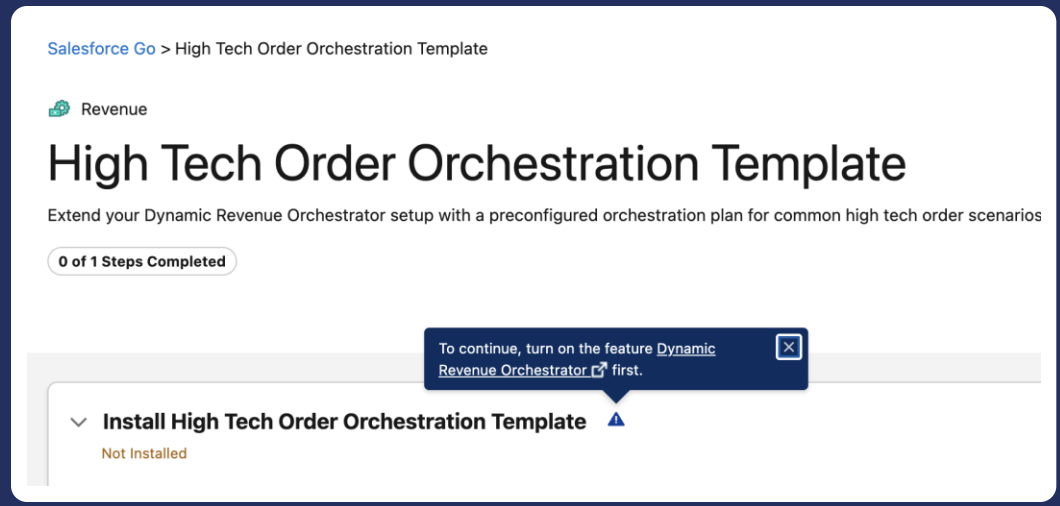
Once the invoice document is generated, the final billing task sends it to the customer

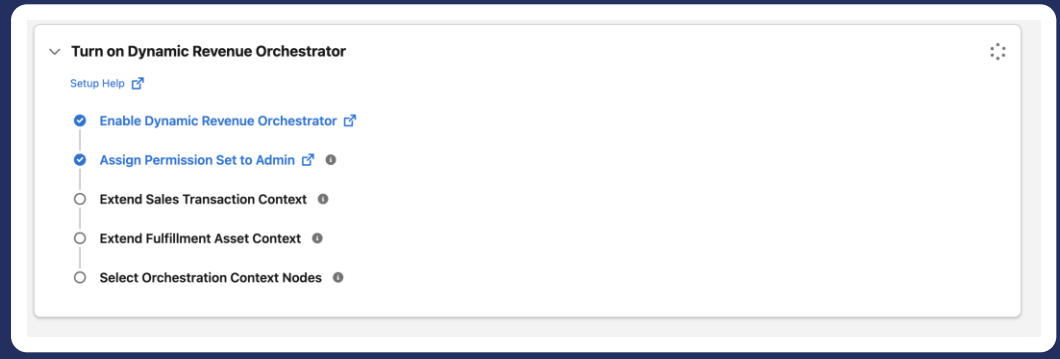
DEMO · PART 1

Turn on Dynamic Revenue Orchestrator first

DRO is its own Salesforce Go feature. Until it's on, the template can't be installed.

1. Enable Dynamic Revenue Orchestrator
2. Assign Permission Set to Admin
3. Extend Sales Transaction Context
4. Extend Fulfillment Asset Context
5. Select Orchestration Context Nodes



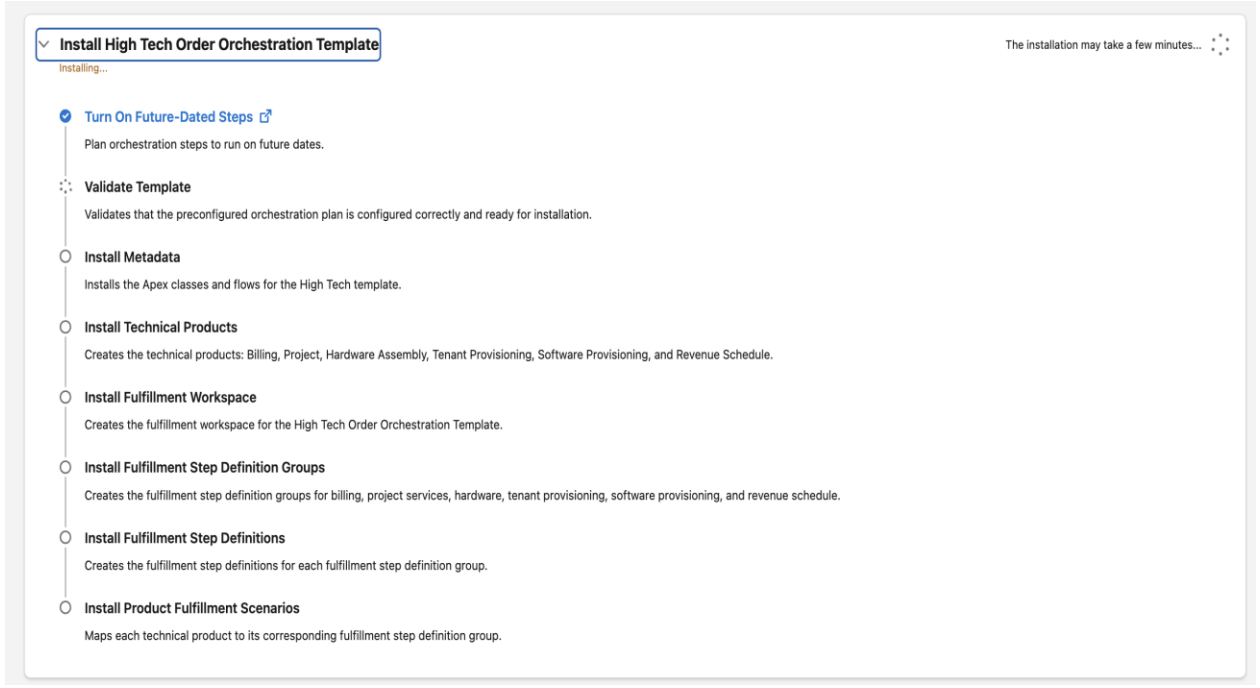


DEMO · PART 2

Installing the template

Salesforce Go › Revenue Cloud › Data Packs › High Tech Order Orchestration › Set Up

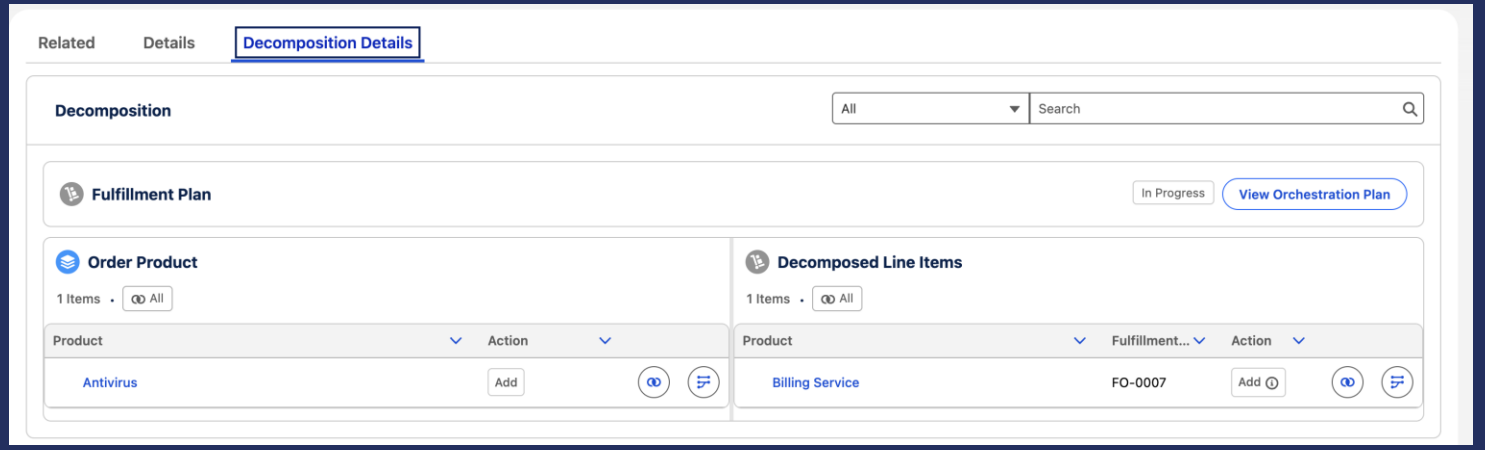
1. Turn on future-dated steps
2. Validate template
3. Install metadata (Apex classes and flows)
4. Install technical products
5. Install fulfillment workspace
6. Install step definition groups
7. Install step definitions
8. Install product fulfillment scenarios



DEMO · PART 3

Running an order through it

1. Map commercial products to the technical products
2. Create and activate an order
3. Open the orchestration plan
4. Complete a manual task; watch auto tasks run



BEST PRACTICE

Decomposition Best Practice

Design from the downstream systems

Each technical product is how one system, like billing or provisioning, sees what was sold

Keep technical products reusable

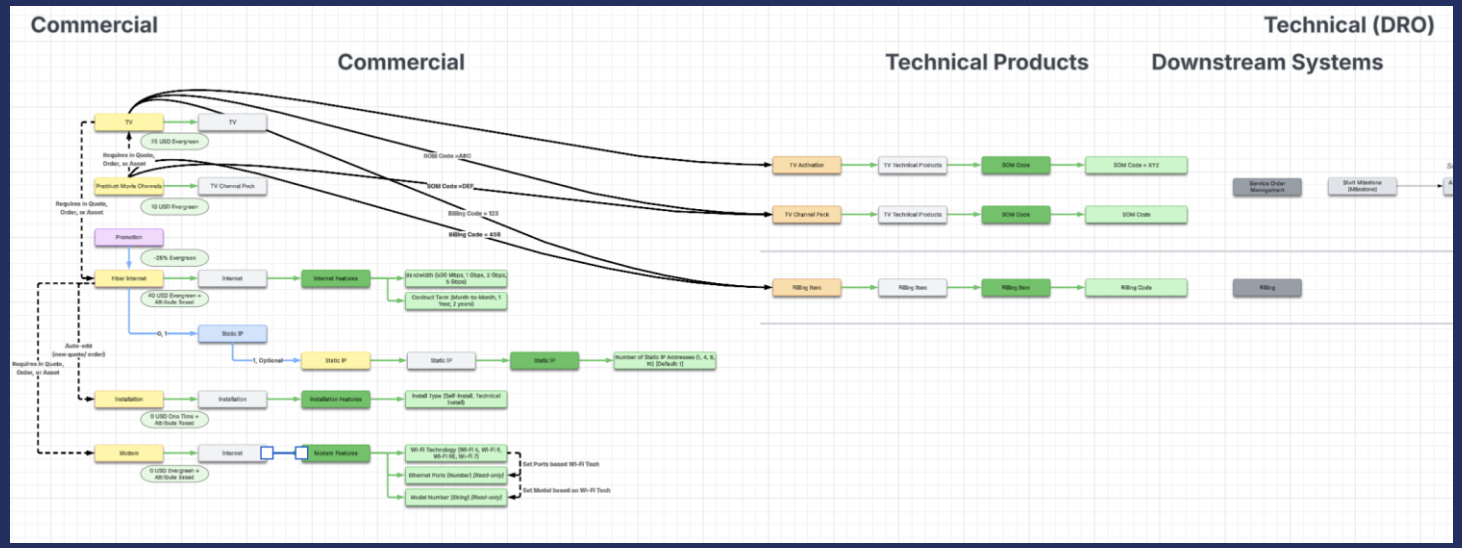
Many commercial products can share one, so the technical catalog stays small and stable

Pass the differences as data

Use field and attribute mapping for codes and options, not more technical products

Model it before you build it

Draw commercial, technical and downstream first, then write the rules



From the Stratus Carta product modeling template

FROM THE FIELD

Two issues to get ahead of

01 Test coverage on a demo org

Same as Episode 2: on a Revenue Cloud demo org, the existing test classes don't give enough Apex coverage, so the metadata deploy is blocked.

Tip: Fix test coverage in the org before you start the install.

02 Products created as commercial

If your profile's default product record type is Commercial, the template creates its six technical products as commercial products.

Tip: Switch your default record type to Technical before you install, or flip each product's record type afterwards.

LIMITATIONS

What to watch out for

Limitation	Impact
A template, not your design	Steps and dependencies need tailoring to your teams
No link to your catalog	You write the decomposition rules
Can’t be uninstalled	Install in a sandbox first

BEST PRACTICE

How I’d use it on a real project

Discovery demo

Show stakeholders an order flowing to an invoice before design starts

Design reference

Use its step groups and dependencies as a pattern for your own

Proof of concept

Map one real commercial product and run it end to end

Thank you



www.stratuscarta.com