



Subscription Products — Admin User Guide

Day-to-day management of live subscriptions.

Subscription Products Documentation · August 4, 2026

 © 2026 Indition Corp. All rights reserved. Indition™ and the Indition logo are trademarks of Indition Corp. All other product names, logos, and brands are the property of their respective owners.

The screens

SCREEN	PATH	WHAT IT IS FOR
Subscriptions	<code>/admin/SubscriptionProducts/admin/subscription/index</code>	Every agreement, filterable, with headline numbers
Subscription Products	<code>.../subscription/products</code>	Which SKUs are set up for subscription
Inventory Planning	<code>.../subscription/inventoryPlanning</code>	Projected demand vs stock — see its own guide
Settings	<code>.../subscriptionSettings/index</code>	Configuration
A single subscription	<code>.../subscription/view/id/<id></code>	Everything about one agreement, and the actions

Subscriptions grid

Six headline figures across the top: Active + Trial, Past Due, Failed, Due in 7 Days, Renewal Success (30 days), and estimated monthly recurring revenue. The revenue figure normalises every live agreement to a 30-day month — a weekly £10 subscription counts as about £43 — so it is an estimate for sizing the programme, not an accounting number.

Filter by category (multi-select), product, SKU, customer, status and next renewal date. Filters reload the grid without a full page load, and the page size selector sits with the pager above and below the table.

Clicking **Subscribers** from the Subscription Products grid scopes this screen to one SKU. When scoped, the product and SKU move to a panel above the grid and their columns drop out — they would say the same thing on every row. "All subscriptions" leaves the scope; **Reset** clears the filters but keeps it.

Managing one subscription

The detail screen shows status, product, SKU, customer, billing interval, price, next renewal, cycles, failed attempts, payment profile, the originating and last orders, and where renewals ship. Below that, the full event history.

Actions

ACTION	AVAILABLE WHEN	WHAT HAPPENS
Renew Now	Active, trial or past due	Raises the renewal order immediately, ignoring the due date — and, with auto-charge on, charges the card . Advances the cycle count and moves the next renewal date a full interval forward from its existing date. Use for "customer wants it early" or a missed cron run.
Skip Next Renewal	Active, trial or past due	Pushes the schedule on one interval without billing
Pause	Active, trial or past due	Stops billing. Optionally give a date to resume automatically; blank pauses indefinitely
Resume	Paused	Back to active, dunning counter reset. If the renewal date has lapsed it is moved forward
Restart / Reactivate	Cancelled, completed, or payment-failed	Brings an ended agreement back, due to bill today, with the cancellation cleared
Cancel	Anything not already ended	Ends it, with a reason recorded
Edit Terms	Always	Price, quantity, cadence, next renewal date, cycle cap, price lock
Renewal Shipping Address	Always	Send renewals somewhere other than the original order's address

Renew Now is not a dry run. It creates a real order and, with auto-charge on, takes real money — early. The confirmation says so.

Restarting a completed subscription whose cycle cap is used up is refused, because it would bill once and immediately re-complete. Raise Max Cycles under Edit Terms first, then restart.

Editing terms

Blank fields are left alone — you only fill in what you are changing. Every change is written to the history against your user. Changes take effect from the next renewal; nothing is prorated. The next renewal date cannot be set in the past.

Terms are editable even on ended subscriptions, because they can be restarted and a completed one *needs* its cycle cap raised first.

Reading the history

The event trail is append-only and is the answer to "why did it do that". It records creation, every renewal order and charge, every failure with the gateway message, reminders, pauses and resumes (including automatic ones), skips, payment-method changes, term edits, and completion.

Two entries are load-bearing rather than informational: `renewal_order_created` is how the system recognises a renewal order and avoids spawning a second subscription from it, and how a retry finds the order it already raised. If you see gaps there, read the Operations Runbook.

Customers tab

Admin → Customers → a customer shows a **Subscriptions** tab when they have any, past or present.

Related

- [Inventory Planning](#)
- [Operations Runbook](#)