



AI SUPPORT AGENT GUIDE

Inventory AI Agent Guide

Knowledge guide for supporting merchants with inventory configuration, operations, stock movement, fulfillment, cycle counts, troubleshooting, and agent guardrails.

Document Type: AI Support Agent Guide

Module: Inventory 1.2.1

Platform: Indition Commerce

Audience: Store admins, buyers, managers

Knowledge document for a support chatbot assisting merchants with the Indition Commerce Inventory module (core/modules/Shop/modules/Inventory). It describes how to configure, operate, and troubleshoot inventory. Everything below reflects the implemented behaviour in the module; do not promise capabilities not listed here.

Audience for your answers: store administrators, buyers, and managers using the Inventory admin area. Keep answers plain-language; name the exact screen or setting. Escalate to a human for data-repair operations (bulk migrations, availability rebuilds affecting live storefronts, negative-stock incidents).

1. What the module is

Inventory adds warehouse- and location-based stock control on top of Shop. Instead of one stock number per SKU, stock is counted per location, costs are tracked in FIFO layers, and every movement is written to an audit ledger. It also covers purchasing, receiving, transfers, cycle counts, serial numbers, reporting, and optional pre-cutting of bulk stock.

Key principles to remember when answering:

- Adoption is per-SKU. Enabling the module does not force every product into tracking. SKUs are brought under management deliberately.
- Menu gating. Until the module is enabled and its tables exist, the Inventory menu shows Settings only. A merchant who "can't find the Inventory screens" usually has not enabled the module yet.
- Availability equals fulfillability. What a shopper sees as available for a store is the stock that can actually ship to that store.

2. Core stock model

Three numbers describe a SKU's stock in a location:

Number	Meaning
On Hand	Units physically present (qty_on_hand).
Reserved	Units promised to open orders (qty_reserved).
Available	Sellable = On Hand – Reserved.

Where the data lives:

- <site>_shop_inventory_item - one row per SKU-and-location, holding qty_on_hand and qty_reserved. This is the source of truth.
- SubProduct rollup - location balances roll up into stock_total and stock_reserved on the SubProduct so the rest of Shop keeps working. The sentinel value 999999999 in stock_total marks an untracked SKU (not a real quantity).
- Product in_stock - the parent product's in-stock flag is recomputed from

the sum of its SKUs' available stock.

- `<site>_shop_inventory_store_availability` - the **per-store availability

cache: one row per (subproduct_id, store_id) with qty_on_hand, qty_reserved, qty_available, in_stock, last_update. Absence of a row means zero available for that store.** Storefront listing and product pages read from this cache.

3. Per-store fulfillment (important)

A store's orders are fulfilled only from locations eligible for that store. There is no separate "fulfillment" switch - eligibility follows from warehouse and location store assignments.

A location is eligible for a store when it is active AND either:

- 1 it inherits its warehouse's store assignments and that warehouse serves **the store; or**
- 1 it has its own store subset (not inheriting) that includes the store.

Consequences to explain to merchants:

- Stock sitting in a location that is not eligible for a store does not count toward that store's availability.
- Availability shown for a store = summed available across that store's eligible locations. This is why "available" always equals "fulfillable."
- After changing which stores a warehouse or location serves, the **per-store availability cache** should be rebuilt (globally or per warehouse) so the storefront reflects the new mapping.

Most common per-store symptom: "SKU has stock in the warehouse but shows 0 available for store X." -> The stock is in a location not eligible for store X, or the availability cache is stale. Check location active flag, warehouse store list / location store subset, then rebuild the cache.

4. Cost tracking (FIFO) and the ledger

- Cost layers live in `<site>_shop_inventory_cost_layer` - each received batch records qty_received, qty_remaining, and unit_cost (plus source_type, e.g. receipt). As stock is consumed, cost is drawn oldest-first (FIFO).
- Layer sources: **PO receipts, manual single-SKU receipts, manual purchase-cost adjustments, and transfer cost layers.**
- Last purchase cost and an average cost (from remaining layers) are tracked per SKU and shown on the SKU's Purchase History tab. Missing opening layers can be backfilled from latest purchase-cost data when historical stock already exists.
- Stock-movement ledger - `<site>_shop_inventory_transaction` records every

movement. Transaction reasons: adjustment, receipt, reserve, unreserve, shipment, move_out, move_in, return, precut_out, precut_in.

5. Stock operations (how things move)

Operation	Where	Effect
Receive	Inventory Management > Receive, or SKU Purchase History > Manual Inventory Adjustment	Adds stock into a location (no PO). Creates a cost layer; serials required for serialized SKUs.
Adjust	Inventory Management > Adjust	Positive/negative quantity change with reason (Adjustment or Return) + note. Positive serialized needs a serial per unit; negative removes oldest serials.
Move	Inventory Management > Move	Shifts stock between locations (even across warehouses) without a transfer document.
Order stock movement	Automatic (POS / fulfillment)	Sale debits on-hand (reason shipment); refund credits (reason return); linked to the order.
Reserve / Release	Order sourcing (e.g. Service & Repair)	Increments/decrements reserved without moving on-hand; reserve caps at available and reports any shortfall.
Bulk adjust / move	Inventory Management (select rows)	One quantity change or a shared move destination across rows. Bulk positive adjustments are blocked for serialized SKUs.

Order-driven movement detail (POS/fulfillment): it runs the full engine - a row lock, the negative-stock guard, the ledger, cost layers (COGS), stock_total, product in_stock, and the per-store availability cache. For serialized SKUs it skips the engine's own serial-status changes because the order caller (e.g. POS) owns marking serials sold/returned with order linkage.

5b. Cycle counts (physical audits)

Lifecycle: open > in_progress > (pending_review) > posting > completed, plus cancelled. One active count per location.

- Start with a scope (all / categories / random sample / zero-qty / search /

not-counted-recently / ABC class) - the SKU set is frozen at start. Counts can be blind (counter never sees expected qty; stripped server-side), have a due date, and an assignee (emailed when notifications are on).

- Counting is progressive: lines auto-save with counter + timestamp; barcode

scan box (SKU/UPC/alternate barcodes); found items can be added. Blank counted qty = not counted (line untouched); an explicit 0 zeroes the shelf. Serialized SKUs: scan every serial present - counted qty derives from the list.

- Submit values lines (shortages at actual FIFO layer costs) and checks

tolerances (_QTY/_PCT/_VALUE); approval-required or out-of-tolerance counts wait in review where approvers post, skip lines, or request recounts (first count preserved). Otherwise variances post immediately, set-to-target under row locks, chunked and resumable; stock_moved flags lines whose



on-hand changed since counting.

- Serial reconciliation: unscanned in-stock serials adjust out by id; unknown scanned serials adjust in; serials living elsewhere are rejected at submit.
- Program: ABC classes (annualized shipment value; A $\approx$ top 80%, B next 15%)

recompute when stale; schedules generate due counts via the daily

Inventory.Inventory.GenerateDueCycleCounts job. QuickBooks journal (CYCLECOUNT-<id>, Inventory Shrinkage vs Inventory Asset) posts on completion when enabled and the Orders QuickBooks connection is live.

- Reports: Accuracy, Coverage, SKU Variance History, Counter Performance, monthly Trend (with chart).

Agent tips: if a user says a count "won't submit", check for lines still flagged recount, serialized lines counted above system without serials, or a foreign serial named in the error. A count stuck in posting resumes safely from the view page (already-posted lines are skipped).

6. Oversell and negative stock

- INVENTORY_ALLOW_NEGATIVE_STOCK is the platform-wide floor. Off (default, safer) blocks any movement that would drive a location below zero; on permits oversells/backorders.
- Serialized SKUs are always limited by the serials that actually exist.
- POS has its own layer on top: a POS_BLOCK_OVERSELL check and an optional inventory override. When POS override sells a tracked out-of-stock item, it receives the shortfall through the same receive engine first, so on-hand reconciles with an audit trail instead of going negative. (POS-specific - do not attribute POS oversell behaviour to the core Inventory setting.)

7. Serial numbers

- Enablement is governed by INVENTORY_SERIALIZATION_ENABLE_RULE: require zero stock, allow with serial import, or always allow.
- Once enabled, one serial per unit is required on receive, positive adjustment, and count-over.
- Serial statuses: In Stock, In Transit, Sold, Returned, Adjusted Out.
- Trace via Serial Number Lookup (by serial, SKU, status, or location) and the Serial Number History / Sold Serial reports.
- Alternate barcodes let multiple scans resolve to the same SKU.

8. Settings reference (keys)

All configured under Inventory -> Settings.

Setting	Key
Module enabled	INVENTORY_MODULE_ENABLED
Enabled stores	INVENTORY_ENABLED_STORE_IDS
Enabled product types	INVENTORY_ENABLED_PRODUCT_TYPE_IDS
Warehouse mode (shared / per-store)	INVENTORY_WAREHOUSE_MODE
Allocation method (default_bin / fifo)	INVENTORY_ALLOCATION_METHOD
Allow negative stock	INVENTORY_ALLOW_NEGATIVE_STOCK
Serialization enable rule	INVENTORY_SERIALIZATION_ENABLE_RULE
PO starting number / next number	INVENTORY_PURCHASE_ORDER_NUMBER_START / INVENTORY_PURCHASE_ORDER_NEXT_NUMBER
PO approval required	INVENTORY_PURCHASE_ORDER_APPROVAL_REQUIRED
PO approver users / roles	INVENTORY_PURCHASE_ORDER_APPROVER_USER_IDS / INVENTORY_PURCHASE_ORDER_APPROVER_ROLE_NAMES
Vendor payment terms	INVENTORY_VENDOR_PAYMENT_TERMS
PO disclaimer toggle / text	INVENTORY_PURCHASE_ORDER_ADD_DISCLAIMER / INVENTORY_PURCHASE_ORDER_DISCLAIMER_TEXT
Transfer carriers enabled / list	INVENTORY_TRANSFER_CARRIERS_ENABLED / INVENTORY_TRANSFER_CARRIERS
Pre-Cuts enabled	INVENTORY_PRECUT_ENABLED
Pre-Cut category restriction / fractional sizes	INVENTORY_PRECUT_RESTRICT_CATEGORY_IDS / INVENTORY_PRECUT_FRACTIONAL_SIZES
Fractional/decimal quantities	INVENTORY_FRACTIONAL_ENABLED
Legacy migration sync threshold	INVENTORY_LEGACY_MIGRATION_SYNC_THRESHOLD
Cycle counts blind by default	INVENTORY_CYCLE_COUNT_BLIND_DEFAULT
Cycle count variance tolerances (qty / % / \$)	INVENTORY_CYCLE_COUNT_TOLERANCE_QTY / _PCT / _VALUE
Cycle count approval required / approvers	INVENTORY_CYCLE_COUNT_APPROVAL_REQUIRED, INVENTORY_CYCLE_COUNT_APPROVER_USER_IDS / _ROLE_NAMES
Cycle count email notifications	INVENTORY_CYCLE_COUNT_EMAIL_NOTIFICATIONS
Cycle count QuickBooks shrinkage journal	INVENTORY_CYCLE_COUNT_QB_JOURNAL

9. Setup walkthrough (condensed)

- 1 Enable the module and choose enabled stores; pick warehouse mode (shared vs per-store). Structure is created and a background legacy-stock migration is queued.
- 1 Set core rules: allocation method, allow-negative-stock, serialization



rule, PO starting number.

- 1 Warehouses - code, name, delivery details, store assignments. First save creates a default location (Primary Location, default + default-receive).
- 1 Locations - code, name, capacity; set default / default-receive; choose inherit-stores or a store subset.
- 1 Per-store fulfillment - verify warehouse store lists and location subsets; rebuild the availability cache if eligibility changed after stock existed.
- 1 Track SKUs - from the SKUs Without Inventory Tracking report, enable tracking (with optional starting quantity). Tracked SKUs gain Inventory and Purchase History tabs.
- 1 Receive stock - direct receive or against a PO; enter quantity, unit cost (creates a FIFO layer), and serials where applicable.

10. Troubleshooting playbook

"I only see Settings under Inventory."

The module is not enabled, or its runtime tables are not ready. Enable `INVENTORY_MODULE_ENABLED` and save; the full menu appears once runtime is ready.

"SKU has warehouse stock but shows 0 available for a store." Stock is in a location not eligible for that store, or the availability cache is stale. Check: location `is_active`; the warehouse serves the store (for inheriting locations) or the location's store subset includes the store; then rebuild the per-store availability cache (globally or for the warehouse).

"Available doesn't match On Hand."

Available = On Hand – Reserved. Some units are reserved to open orders. Check the SKU's reserved quantity and open orders.

"The Inventory / Purchase History tabs are missing on a product." Those tabs appear only when the runtime is ready and the SKU is inventory-tracked. Enable tracking for the SKU (e.g. from the untracked-SKUs report).

"A fractional order quantity became a whole number in inventory." Expected. The ledger quantity columns are whole numbers, so fractional line quantities are truncated on write - same on POS and e-commerce paths. Known platform limitation pending a column-widening migration; not a per-channel bug.

"Available shows 999999999."

That is the untracked sentinel, not a real count. The SKU is not under inventory tracking. Enable tracking (with a real starting quantity) if it should be managed.

"Costs/valuation look wrong after go-live on existing stock." Opening cost layers may be missing. They can be backfilled from latest purchase-cost data. Escalate for a backfill if valuation is materially off.

"Stock went negative" / "an oversell got through."

Check `INVENTORY_ALLOW_NEGATIVE_STOCK`. If off, the guard should have blocked it - escalate as a possible incident. If the sale was at POS, POS override may have received the shortfall first (that is by design, with an audit trail).



11. Guardrails for the agent

- Do not tell a merchant to edit database tables directly. Route everything through the Inventory admin screens.
- Do not promise store-split of legacy stock at enable time - legacy stock seeds the first enabled store's warehouse only.
- Availability-cache rebuilds and legacy migrations affect the live storefront; recommend running them deliberately and escalate for large catalogs.
- When unsure whether a behaviour is core Inventory vs POS, say so and separate the two - the POS oversell/override behaviour is distinct from the core negative-stock setting.

Reflects the implemented Inventory module (v1.2.1). Verify against the User Manual and Technical Documentation in this docs/ folder for field-level detail.