

What is a NetSuite transaction?

GL impact, posting rules, and the full transaction lifecycle

ORACLE
NetSuite
SuiteCloud Developer
Network Partner



What's inside

Introduction: What is a NetSuite transaction?

Page 3

01

Section 1: Service Orders: The Order-to-Cash Cycle

Page 6

02

Section 2: Manufacturing Orders (work orders)

Page 9

03

Section 3: Assembly Builds

Page 13

04

Section 4: Item Order Execution: The Purchase-to-Pay-cycle

Page 15

Key concepts to know

Page 19

Putting it all together: The interconnected transaction system

Page 20

Final tips for working with NetSuite

Page 21

Introduction:

What is a NetSuite transaction?

In NetSuite, a transaction is any record that creates a financial or operational event: including sales orders, purchase orders, work orders, item receipts, fulfillments, invoices, and assembly builds. Most NetSuite transactions are non-posting when created, meaning they have no GL impact until a downstream trigger (fulfillment, receipt, or completion) is hit.

The four key transaction types covered in this guide:

- ◆ **Service Orders (order-to-cash)**
- ◆ **Manufacturing and Work Orders**
- ◆ **Assembly Builds**
- ◆ **Item order execution (procure-to-pay)**

This guide covers the core transaction lifecycle in NetSuite's standard and manufacturing modules. Specific behavior may vary based on your enabled features, costing method, and configuration. It goes through how transactions move through NetSuite, what happens to inventory along the way, and how it all connects to your financial records.

If you're working with NetSuite, or in the process of evaluating it, one of the most important things to understand is how transactions flow through the system. A "transaction" in NetSuite isn't just an invoice. It's any record that creates a financial or operational event: a sales order, a purchase order, a work order, an item receipt, a fulfillment, an assembly build. They're all transactions, and they're all connected (and if they're not connected, then that's a whole separate conversation).



This guide walks through the full lifecycle of four key transaction types:

1. **Service Orders**
(Sales Order → Fulfillment → Invoice → Payment)
2. **Manufacturing and Work Orders**
(Planning → Production → Completion → Invoice)
3. **Assembly Builds**
(Simple BOM-based assembly without full manufacturing)
4. **Item Order Execution**
(Purchase Order → Receipt → Bill → Payment)

For each, we'll cover what the transaction does, how inventory is affected, and what exactly hits the general ledger.

Understanding NetSuite's transaction philosophy

Before diving into each type, there's one principal worth internalizing: most transactions in NetSuite are non-posting until a specific trigger is hit.

A sales order, for example, has zero accounting impact when it's created. It records a commitment, allocates inventory, and drives fulfillment workflows; nothing posts to your books until goods ship or services are completed. This is intentional, as it separates the operational record from the financial event.

Here's a quick reference to understand when key transactions post to the GL:

Transaction	Posts to GL?	When
Sales Order	No	— (operational only)
Item Fulfillment	Yes	When shipment is confirmed
Invoice	Yes	When created (increases A/R)
Customer Payment	Yes	When applied
Purchase Order	No	— (operational only)
Item Receipt	Yes	Increases Inventory Asset & Accrued Purchases
Vendor Bill	Yes	Moves from Accrued Purchases to A/P
Work Order	No	— (operational only)
Work Order Completion	Yes	Moves WIP to Finished Goods
Assembly Build	Yes	Consumes components, adds finished assembly

01

Section 1. Service Orders: The Order-to-Cash cycle

A Service Order in NetSuite is the standard sales cycle: a customer wants something, you commit to deliver it, you deliver it, you invoice them, and you collect payment. The backbone is the Sales Order, which then spawns fulfillment and billing transactions.

Step 1: Sales Order created (pending fulfillment)

A sales order records your commitment to sell items or services to a customer. At this stage:

- ◆ No GL impact—the order is non-posting.
- ◆ Inventory is soft-allocated (committed but not yet reduced).
- ◆ The order status is Pending Fulfillment.
- ◆ Available inventory drops (committed quantity increases), but on-hand inventory does not change.

Navigation:

Transactions → Sales → Enter Sales Orders

Sales Order status flow:

Pending Approval → Pending Fulfillment →

Partially Fulfilled → Fulfilled → Partially Billed → Billed/Invoiced

Step 2: Approval (optional but recommended)

If your account uses workflow-based approvals, the order moves through an approval queue before it can be fulfilled. Orders with status Pending Approval cannot be fulfilled or billed.

Step 3: Item Fulfillment (inventory decreases here)

The Item Fulfillment is the transaction that officially records the shipment of goods. This is where inventory actually moves.

GL impact → Item fulfillment:

Debit: Cost of Goods Sold (COGS)	\$X
Credit: Inventory Asset	\$X

The cost booked to COGS is calculated based on your costing method (Average Cost, FIFO, LIFO, or Specific Identification). The order status moves to Shipped.

Pick, Pack, Ship (with Advanced Shipping enabled): If you use the Advanced Shipping feature, the fulfillment process breaks into three discrete steps:

1. Pick: Items are picked from the warehouse bins.
2. Pack: Items are packed for shipment.
3. Ship: Shipment is confirmed, Item Fulfillment record is created.

Each step updates the order status and creates operational records. The inventory impact (COGS debit, Inventory credit) happens at the ship step.

Navigation:

Transactions → Sales → Fulfill Orders

Step 4: Invoice created → AR posted

The Item Fulfillment is the transaction that officially records the shipment of goods. This is where inventory actually moves.

**GL impact → Invoice:**

Debit: Undeposited funds (or Bank)	\$X
Credit: Sales revenue	\$X

GL impact → Cash Sale:

Debit: Accounts receivable (AR)	\$X
Credit: Revenue	\$X

With Advanced Shipping enabled, the fulfillment and invoice are separate transactions. Without it, invoicing happens automatically at fulfillment.

Navigation:

Transactions → Sales → Bill Sales Orders

Step 5: Customer Payment (AR Cleared)

When the customer pays, you apply the payment against the open invoice.

GL impact → customer payment:

Debit: Bank / Undeposited Funds	\$X
Credit: Accounts Receivable (AR)	\$X

The invoice status becomes Paid and the order cycle is complete.

Complete Service Order flow

[Sales Order Created]

↓ (inventory committed, no GL impact)

[Approval]

↓

[Item Fulfillment / Pick-Pack-Ship]

↓ (GL: Debit COGS / Credit Inventory)

[Invoice Generated]

↓ (GL: Debit A/R / Credit Revenue)

[Customer Payment Applied]

↓ (GL: Debit Bank / Credit A/R)

[Order Closed]

02

Manufacturing Orders (work orders)

A work order is the instruction to manufacture something. It is the central record driving material consumption, production tracking, and cost accumulation in NetSuite manufacturing.

Important: Full work order management requires the Manufacturing module (a paid add-on). Basic assembly production is available through Assembly Builds (see Section 3 to learn more about Assembly Builds).

Step 1: Work order created (planning)scape

Work orders can originate from multiple sources:

- ◆ Manually by a production planner.
- ◆ From a Sales Order (make-to-order: a customer ordered the item).
- ◆ From MRP (Material Requirements Planning, based on demand/reorder points).
- ◆ From planned orders converted to firmed work orders.

When created, NetSuite automatically:

- ◆ Explodes the Bill of Materials (BOM), listing every required component and quantity.
- ◆ Checks component availability against current stock.
- ◆ Calculates expected completion date based on routing and lead times.

Type	Description
Standard	Produce a quantity for stock (make-to-stock)
Make-to-Order	Linked to a specific sales order
Make-to-Stock	Driven by forecast or reorder points
Disassembly	Breaks a finished assembly back into components

Step 2: BOM explosion & component management

The work order reads the BOM and calculates all required component quantities scaled to the production quantity.

Example: BOM for "Widget A" requires: 2× Steel Plate, 4× Bolt M6, 1× Motor Unit
Work Order for 100× Widget A therefore requires: 200 Steel Plates, 400 Bolts M6, 100 Motor Units

- ◆ Reserve inventory: allocate existing stock so other transactions can't consume it.
- ◆ Back-flush: don't reserve upfront; consume components automatically at completion based on BOM quantities.
- ◆ Manual issuance: issue components to the work order as they become available (most control).

Best practice: Back-flush only low-value, high-volume items (fasteners, packaging). Explicitly issue high-value components for better WIP accuracy.

Step 3: Release (work begins)

Releasing the work order signals to the production floor manufacturing can start. Released orders appear on shop floor dashboards and production schedules.

Status flow:

Planned → Released → In Process → Built → Closed

Step 4: Component issuance → first inventory impact

As production begins, components are issued from raw materials inventory to the work order.

GL impact → component issuance:

Debit: Work-In-Process (WIP)	\$X
Credit: Raw materials inventory	\$X

Labor costs (if routing is configured) also accumulate in the WIP as operators record time against each routing step.

Step 5: Work order completion → WIP moves to Finished Goods

When finished goods come off the production line, record a work order Completion for the quantity produced.

GL impact → Work Order Completion:

Debit: Finished Goods Inventory	\$X
Credit: Work-In-Process (WIP)	\$X

Partial completions are supported—a work order for 100 units can be completed in batches over multiple days.

Step 6: Scrap and variance handling

Manufacturing isn't perfect. Some of the offsets that need to be recorded include:

- ◆ Component scrap: damaged material goes to a Scrap Expense account (not into the finished good cost).
- ◆ Yield loss: if 100 items were planned but only 95 were produced, the 5 lost units' WIP cost posts to a Yield Loss account.
- ◆ Production variances: actual vs. standard cost differences are posted to variance accounts at close.

Step 7: Work order closed

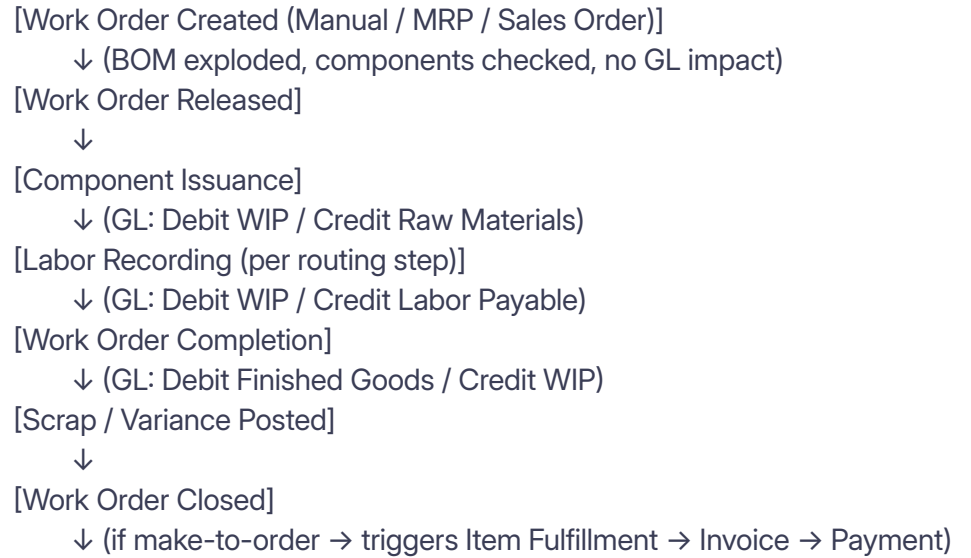
Once all quantities are completed (or cancelled), the work order is closed with the following actions taking place:

- ◆ Remaining WIP balances are written off to variance accounts.
- ◆ Component allocations are released.
- ◆ The work order becomes non-editable.

Step 8: Invoicing (make-to-order)

For make-to-order work orders linked to a sales order, completion triggers the fulfillment and billing cycle as described in Section 1.

Complete manufacturing order flow



03

Assembly Builds

An Assembly Build is NetSuite's simpler, single-step alternative to full work orders. It's available in the base NetSuite license (no Manufacturing module required).

Think of it as: select the finished item, enter the quantity to build, and NetSuite consumes the components and adds the finished assembly to inventory, all in one step.

When to use Assembly Builds vs. Work orders

	Assembly Build	Work order
License required	Base NetSuite	Manufacturing module
Routing / multi-step	No	Yes
WIP tracking	No	Yes
Labor recording	No	Yes
Partial completions	No	Yes
Scheduling	No	Yes
Best for:	Kitting, light assembly	Complex manufacturing

The Assembly Build transaction

Navigation: Transactions → Manufacturing → Build Assemblies

1. Select the assembly item
2. Enter the quantity to build
3. NetSuite reads the BOM and verifies component availability
4. Save; components are consumed and finished assemblies are added to inventory

**GL impact → Assembly Build:**

Debit: Assembly/Finished Goods Inventory \$X (at sum of component costs)

Credit: Component Inventory (each component) \$X (per BOM quantities)

The cost of the assembly is the sum of component costs at the time of the build, using your costing method. There's no WIP—the transaction is instantaneous.

Disassembly

Assembly Builds can be run in reverse to disassemble a finished item back into components; useful for rework, recycling, or inventory rebalancing. The BOM runs backwards: the finished assembly is consumed, and components are returned to inventory.

04

Item Order Execution: The purchase-to-pay cycle

Item Order Execution covers how you acquire inventory from vendors: from raising a purchase order, through receiving goods into stock, to paying the vendor. This is the procure-to-pay cycle.

Step 1: Purchase Order created (no GL impact)

A Purchase Order (PO) is your commitment to buy from a vendor. Like a sales order, it has no GL impact when created, as it is non-posting.

What happens at PO creation:

- ◆ A formal record is created for the vendor.
- ◆ Items are flagged as "on order" (visible in inventory reports).
- ◆ Approval routing can be enforced before the PO is sent to the vendor.

Navigation: Transactions → Purchases → Enter Purchase Orders

Step 2: Item Receipt (inventory increases here)

When the vendor delivers your goods, you record an Item Receipt against the open PO. This is the critical moment when inventory physically (and financially) enters your warehouse.

GL impact → Item Receipt (with Advanced Receiving):

Debit: Inventory Asset	\$X	(increases your assets)
Credit: Accrued Purchases	\$X	(liability — you received goods but haven't been billed yet)

The Accrued Purchases account acts as a clearing/bridge account to handle the timing gap between receiving goods and receiving the vendor's bill. The on-hand quantity in your inventory records updates immediately.

Navigation: Transactions → Purchases → Receive Orders

After receiving:

- ◆ PO status moves to Pending Billing.
- ◆ Items are immediately available in inventory.
- ◆ Items can be automatically allocated to open sales orders, backorders, or production builds.

Step 3: Vendor bill created → AP posted

When the vendor's invoice arrives, you record a Vendor Bill against the item receipt. This is when Accounts Payable is formally recorded.

GL impact → vendor bill:

Debit: Accrued Purchases	\$X (clears the bridge account)
Credit: Accounts Payable (A/P)	\$X (now you formally owe the vendor)

The Accrued Purchases account is now zeroed out. The bill appears on the Open Bills report and the AP Aging report.

Navigation: Transactions → Payables → Enter Bills (or Bill from the PO)

Step 4: Vendor bill variances (when quantities or prices differ)

In practice, the PO price, received quantity, and vendor invoice amount don't always match. NetSuite handles this through Vendor bill variances:

- ◆ Quantity variance: received more or fewer units than ordered.
- ◆ Price variance: vendor billed at a different price than the PO.
- ◆ Exchange rate variance: for multi-currency POs.

Variances post to dedicated variance accounts (set on the item record), keeping your inventory costing clean.

Navigation: Transactions → Payables → Post Vendor Bill Variances

Step 5: Vendor payment → AP cleared

When you pay the vendor:

GL impact → vendor payment:

Debit: Accounts Payable (AP) \$X

Credit: Bank Account \$X

The bill status becomes Paid and the procure-to-pay cycle is complete.

Complete item order execution flow

[Purchase Order created]

↓ (items flagged "on order," no GL impact)

[Vendor Approval / PO Sent to Vendor]

↓

[Item Receipt Recorded]

↓ (GL: Debit Inventory / Credit Accrued Purchases)

↓ (on-hand quantity increases immediately)

[Vendor Bill Created]

↓ (GL: Debit Accrued Purchases / Credit A/P)

[Vendor Bill Variances Posted (if any)]

↓ (GL: Variance accounts adjusted)

[Vendor Payment Made]

↓ (GL: Debit A/P / Credit Bank)

[PO closed]



Inventory impact summary at a glance

Transaction	Inventory effect	GL account movement
Sales Order Created	Soft commit (no physical change)	None
Item Fulfillment	Decreases on-hand	Debit COGS/Credit Inventory
Purchase Order	Increases "on order" quantity	None
Item Receipt	Increases on-hand	Debit Inventory/Credit Accrued Purchases
Assembly Build	Components decrease, finished good increases	Debit Assembly/Credit Component Inventories
Work Order Component Issuance	Components decrease	Debit WIP/Credit Raw Materials
Work Order Completion	Finished goods increase	Debit Finished Goods/ Credit WIP
Customer Return (RMA)	Increases on-hand	Debit Inventory/Credit COGS
Vendor Return	Decreases on-hand	Debit AP or Accrued Purchases/Credit Inventory

Key concepts to know

Costing methods

NetSuite calculates inventory cost using one of these methods, set per item:

- ◆ Average Cost: the default; COGS = average of all units in stock.
- ◆ FIFO: (First In, First Out) oldest units sold first.
- ◆ LIFO: (Last In, First Out) newest units sold first.
- ◆ Standard Cost: a pre-set expected cost; variances are posted separately.
- ◆ Specific Identification: cost tracked per individual lot or serial number.

The costing method determines what gets debited to COGS on every fulfillment and assembly transaction.

Lot and serial tracking

For regulated industries or high-value items, NetSuite tracks individual units by lot number or serial number. These flow through every transaction—receipts, builds, fulfillments—giving full traceability from supplier to customer.

Multi-location inventory

With the Advanced Inventory module, inventory is tracked by location. A transfer order moves stock between locations, creating a two-sided inventory transaction:

Debit: Destination Location Inventory

Credit: Source Location Inventory

Non-posting transactions

Remember: Sales Orders, Purchase Orders, and Work Orders are non-posting. They are operational commitments that drive workflows but have no direct GL impact until a downstream transaction (fulfillment, receipt, completion) is triggered.

Putting it all together: How the transaction cycles interconnect

In a manufacturing company, these cycles often interlock in a single chain:

1. **A Sales Order is placed for a finished product.**
2. **The sales order triggers a Work Order (make-to-order).**
3. **The work order's BOM reveals a component shortage, triggering a Purchase Order.**
4. **The PO is received via an Item Receipt (inventory increases).**
5. **Components are issued to the Work Order (inventory shifts to the WIP).**
6. **Production completes and Finished Goods are received from the WIP.**
7. **The finished goods are fulfilled against the Sales Order (inventory decreases).**
8. **An Invoice is generated and the customer pays.**
9. **The Vendor Bill is matched and paid.**

Every step in this chain is a transaction in NetSuite, and every transaction either moves inventory, posts to the GL, or both. The power of a properly configured NetSuite implementation is that all of this flows automatically, from demand signal to cash receipt, with a complete audit trail at every step.

Final tips for working with NetSuite transactions

1. **Don't skip transaction steps.** Jumping from a work order straight to "in process" bypasses the release checkpoint. The release gate exists to prevent the starting of work that can't be completed due to material or capacity constraints.
2. **Close work orders promptly.** Open work orders with remaining WIP balances distort inventory valuation. If a work order won't be completed, cancel the remaining quantity and close it.
3. **Match bills to receipts.** Enable "Match Bill to Receipt" on purchase order lines. This ensures variance accounts are populated correctly and simplifies reconciliation.
4. **Understand your costing method.** Every fulfillment and build transaction posts a cost to the GL based on your costing method. Errors in item setup (wrong costing method, wrong cost) are hard to fix after the fact.
5. **Use the GL impact view.** Every posting transaction in NetSuite has a "GL impact" tab showing exactly which accounts were debited and credited. This is your best debugging tool when something looks off in the books.
6. **Leverage saved searches.** NetSuite's transaction reporting can feel overwhelming. Saved searches across transaction types (e.g., "all open work orders with WIP > \$0", "items received but not billed") are essential operational tools.



About Charted

Charted clears the path for finance teams with AP Automation solutions built directly into your NetSuite environment, providing real-time clarity on every invoice, approval, and payment. Eliminate manual data entry with AI-powered capabilities and automation designed to accelerate your month-end close. Enable the most complex approvals and global, multi-entity workflows. Charted was born from years of hands-on implementation and consulting expertise, bringing you a one-stop-shop for everything NetSuite. To find out more, please visit www.charted.com.

©2026 Charted (070626)



101 Arch St 8th floor
Boston MA, USA
+1 800-779-6285
charted.com

ORACLE
NetSuite
SuiteCloud Developer
Network Partner