

**Patterson Warehouses (PAT)**

**Master Design Document**

**Design Document**

**v1.0**

****

**Deposco, Inc.**1-877-770-1110 | www.deposco.com  
© Copyright 2019. All Rights Reserved.

# 

# Document Revision History

Document Revision History

| **Revision By** | **Date** | **Version** | **Notes** |
| --- | --- | --- | --- |
| Brandon Baker | July 30, 2025 | v1.0 | Document Created |

# Table Of Contents

[**Table Of Contents 2**](#_po8yvhm44ya5)

[**Overview 10**](#_lkz0lmqbhugu)

[**1. Overview 10**](#_3t7ocuq9d6a3)

[1.1. Introduction 10](#_xm7zvpozngth)

[1.2. Scope 10](#_do6k9vv5kdrc)

[1.3. Success Criteria 11](#_fnq4evb5j9x4)

[1.4. Terminology 11](#_3efwtnbulut4)

[1.5. Acronyms 13](#_8wfpgkclnc6w)

[**2. Master Data 15**](#_xpm3s3hc0sxz)

[2.1. Facility 15](#_r9qp8oj3j2lh)

[2.2. Company 15](#_upsu1gzaedut)

[2.2.1. Key Assumptions 16](#_j29jakngpiyv)

[2.3. Trading Partners 16](#_ezdnnnl628ju)

[2.3.1. Key Assumptions 18](#_6wj2zmz6pviy)

[2.4. Locations / Zones / Storages 18](#_kwmvt7750axo)

[2.4.1. Locations 18](#_631sxse1ikfk)

[2.4.1.1. Location Definition 20](#_nf61t0rrxctp)

[2.4.1.1.1. Key Assumption 20](#_jmc8fjfejfe1)

[2.4.2. Zones 20](#_543myktktapt)

[2.4.3. Storages 21](#_thls6u8xl9xb)

[2.4.4. Key Assumptions 21](#_w9nwapr7hj3j)

[2.4.5. Steps To Add/Remove a Location 21](#_3i13v2z1rqcr)

[2.5. Items 22](#_7nj830qbpoug)

[2.5.1. Key Assumptions 23](#_356osjd9g8hj)

[2.6. Packs 23](#_affgb1xm85la)

[2.6.1. Key Assumptions 24](#_7ozsbmbrgp9p)

[2.7. Users 25](#_pw66pjvstiu1)

[2.7.1. Company Assignments 25](#_55xlgeewxmdf)

[2.7.2. Key Assumptions 25](#_w5lv0lrd08x8)

[2.8. Carton Types 26](#_ys4dqzcf31h8)

[2.8.1. Available Carton Types 26](#_1iwdn5y5jok5)

[2.8.2. Key Assumptions 26](#_pl3ea0czclde)

[2.9. Containers 27](#_u8z9s7d08n9l)

[2.9.1. Types 27](#_1u0bpz4j7alb)

[2.9.2. Key Assumptions 27](#_f81np87htiiq)

[2.10. Kit Headers and Details 28](#_hlvd1jivkz5j)

[2.10.1. Key Assumptions 28](#_wo9bl5hjbu0y)

[2.11. Work Orders 28](#_jsd4quyyr2fi)

[2.11.1. Key Assumptions 28](#_jwmjafb6n1ds)

[2.12. Components 28](#_wmbky2c68nlu)

[2.12.1. Key Assumptions 28](#_rrvyowodz22v)

[**3. Common System Functionality 29**](#_twr8mfmbv9ln)

[3.1. RuleService Process Configurations 29](#_ixsq2si7srjh)

[3.1.1 Printing 29](#_lks1e0fsry2p)

[3.1.2 Validations 29](#_hl5n2b7e2ncj)

[**4. Inbound Processes 30**](#_ci8yz5xly2rc)

[4.1. Pre-Receiving 30](#_nrbt77scnd74)

[4.1.1 Inbound Order Creation 30](#_aaxkugjzli8)

[4.2. Purchase Order Management 30](#_9raxjl9wwzgf)

[Types of Purchase Orders 30](#_edujsm19egmw)

[4.2.1. Purchase Orders 30](#_qf6aesm95mky)

[4.2.1.1 Key Assumptions 31](#_73cjnln1lg1r)

[4.2.2 Blind Orders 31](#_yqco8d6i6n2b)

[4.2.2.1 Key Assumptions 31](#_qthzgeyuqb14)

[4.2.4 Customer Returns 31](#_81dddj8z3jie)

[4.2.5 Over/Under Receiving 31](#_bhd2uuprzf3z)

[Configuration Options, Validations, Tipsat do not match the PO. Preventing over receiving of quantity can sometimes interrupt operations because it requires that a user stage the inventory without systematically receiving it; a standard recommendation is to allow over receiving but send an email to alert a contact. 31](#_81f4ju7da2ff)

[4.2.5.1 Key Assumptions 31](#_4xanbypdfnl5)

[4.2.6 Items Not on PO 32](#_tmfnz71f0jzf)

[4.2.7 PO/Customer Return Creation 32](#_80aj457wi0zw)

[4.2.8 Purchase Orders 32](#_nvhyapyaz4ny)

[4.2.9. LPN Receiving 33](#_wf62r3p8w1tf)

[4.2.10. Customer Returns 33](#_bsalr54791al)

[4.3 Receiving Processes 33](#_5w5xjywl3z0w)

[4.3.1. Loose Stock Receiving 34](#_h2cuoseng963)

[4.3.2. LPN/Pallet Receiving 1](#_kjsoapv9ds9e)

[4.3.3. Returns Receiving 1](#_wq5fgj15l7nj)

[User Stories 1](#_u8oe5wuhcfja)

[Key Assumptions 1](#_megna3982hzl)

[Process Flow 1](#_9gmnm059egm7)

[4.3.4. Inbound/Purchase Order Damages Receiving 1](#_2023leoeukbc)

[4.4. Patterson Specific Requirements 1](#_kue49b7f5nr7)

[4.4.1 Prompt for Location 1](#_czfjrw495csj)

[4.4.2 Entity Images 1](#_78nhcqspa61q)

[4.4.3 Item Attribute Tracking 1](#_7e9ab7jnhdu7)

[**4. Inventory Movement Processes 1**](#_dkl0gf0wqjt)

[4.5. Putaway Processes 1](#_u2y2m4mbdohh)

[4.5.1 Move Container 1](#_hqfy0jrtv79n)

[4.5.2. Move Item 1](#_u5eosrnjk5r9)

[4.5.3. Move Stock 1](#_4bvk7pj)

[4.5.4. Directed Putaway Logic 1](#_74lsjjxe20sm)

[User Stories 1](#_6g37x5rak4ty)

[Key Assumptions 1](#_vi2unoon7m7v)

[**5. Inventory Processes 1**](#_kd8zc8b0i85m)

[5.1. Inventory Transactions 1](#_vyagt9l28oun)

[Types of Transactions 1](#_z7ovz2v6vgs4)

[5.2. Inventory Counts 1](#_eilzyngjhoja)

[5.3. Physical Counts 1](#_bmiwloa7srhw)

[Strategy 1](#_wee1g12zlmue)

[End Count 1](#_nqpgmaug4nl6)

[5.4. Cycle Counts 1](#_4jbeok3y9j1g)

[5.5. Generation Frequency and Methods 1](#_pxuabjv8jwvo)

[5.5.1 System Generated Tasks 1](#_lk525tvppvu2)

[User Stories 1](#_rg0btgs2fokw)

[5.5.3 Prioritization 1](#_kkzmel1dzjn7)

[5.5.4 Managing Cycle Count Variances 1](#_e8uvodizuh62)

[Strategy 1](#_7i0qrnex4u5f)

[User Stories 1](#_1pznv2b8ukhy)

[Key Assumptions 1](#_9ne7d5drvo0t)

[5.6. Inventory Count Configurations 1](#_2s28m7z6h3kw)

[5.6.1. Bulk Count 1](#_i3yuoyt9wzix)

[5.6.2. Inventory Attributes in Counting 1](#_ual4km4afbe)

[5.7. Inventory Adjustments 1](#_m886rcbi9pfs)

[Strategy 1](#_57drmnr4gyfq)

[Process Flow 1](#_q5z0tkc773e4)

[User Stories 1](#_1qlfpqw82ss3)

[Key Assumptions 1](#_vsmkjt2jei3l)

[5.7.1. Patterson Warehouses Pack Conversion 1](#_eusyhr4kwi13)

[5.8. Replenishments 1](#_xjkwyourp9k4)

[5.9.1. Replenishment Strategy 1](#_fy9l17d14kph)

[Storage Based Replenishments 1](#_vxalar2jn0bl)

[Strategy 1](#_jiuiflb1sgoy)

[Process Flow 1](#_7w1c6tn4jfyf)

[5.9.2. Replenishment Configurations 1](#_u0e1py1587dx)

[5.9.2.1. Over Replenishment 1](#_jx1qefzgkosx)

[5.9.2.2. Replenishment Exceptions 1](#_sowuv9nlsh36)

[Replenishment Picking Exception 1](#_tprh6ups1hg1)

[Replenishment Putaway Exception 1](#_9lx6cbpbjz2m)

[5.9.2.3. Inventory Attributes in Replenishments 1](#_pmxgw9mmq1l1)

[5.9.2.4. Workorder Based Replenishments 1](#_tl33zsi1s5a7)

[User Stories 1](#_sebtyj4md683)

[Key Assumptions 1](#_zifgb1zbepc8)

[5.10. Kitting 1](#_639dfic7wfgi)

[Kitting Strategies 1](#_53k8zw23iabd)

[5.10.1. Kit to Stock 1](#_5e2f0m6ei2wo)

[5.10.2. Kit to Order 1](#_m6f1e31foeds)

[Kitting Master Data 1](#_rdj6pq2dnu1d)

[5.10.3. Item Components / Assemblies 1](#_ittsl1i2aty2)

[5.10.4. Kit Headers / Kit Details 1](#_ndl7z4f8wl32)

[5.10.5. Workorders 1](#_e7860rejg4dt)

[Kitting Processes 1](#_bgabcor9upbz)

[5.10.6. Product Conversion With Workorder 1](#_o1vtc4vkz4sl)

[Kitting Configurations 1](#_iwgttjdypa1i)

[5.10.7. Inventory Attributes in Kitting 1](#_xscror2o23a)

[5.11. Inventory Calculations 1](#_xmclhvchpwdg)

[**6. Outbound Processes 1**](#_ff6qq5guvq5c)

[6.1. Customer Order Management 1](#_zepi0s1qosa7)

[6.1.1. Strategy 1](#_frgyrqxpq5fn)

[6.1.3.1. Key Assumptions 1](#_8peg4fd32mgr)

[6.1.5.1. Process Flow 1](#_2oquu1tw2334)

[6.1.6. Customer Order Statuses 1](#_eovscl1hsym7)

[6.1.7. Exception Management 1](#_3yuo2vxrmhwm)

[6.1.8. Cancellations and Updates 1](#_okbnrdsxhe3e)

[6.1.8. User Stories 1](#_roohg9bqhgfl)

[6.1.8.1 Key Assumptions 1](#_fc1u3sccpfmm)

[6.5. Sales Order Management 1](#_ltt9w967yh7l)

[Process Flow 1](#_dt6jqhfxb3y)

[Sales Order Statuses 1](#_nsn9hhlr2nwq)

[Order Status 1](#_f9f9fvjkdcrq)

[Current Status 1](#_vrwnhr3rb0sw)

[Shipping Status 1](#_x0s67816b4jg)

[6.5.1. Sales Order Types and Origins 1](#_6qyztvgq9ufw)

[Sales Order 1](#_8j18nogs9l4a)

[6.6. Sales Order Pre-Processing 1](#_f7ehnaircryt)

[6.6.1. Automations 1](#_bb2ccwexnjla)

[6.6.2. Post Order Import 1](#_7cux4lk8dlq1)

[6.7. Sales Order Updates and Cancellations 1](#_ffhb4tc3b5nj)

[6.7.1. Sales Order Updates 1](#_2yt4078vj0dq)

[6.7.2. Sales Order Cancellation 1](#_ttym8p8qw826)

[6.8. PAT Order Classifications 1](#_3uvcd73sws33)

[6.8.1. Order Release Profiles 1](#_we9ieo9264r7)

[6.8.2. Allocation/Wave Release 1](#_1gyfgp75vn6k)

[6.8.3. Stock Roll-up Functionality 1](#_by1fkqw889xr)

[6.8.3.1 Key Assumptions 1](#_gtrlj6f2i776)

[6.8.4. Pick Wave Grouping 1](#_byh2ioo1at3r)

[6.9. BackOrder Management 1](#_muht6wodm7o1)

[User Stories 1](#_wd0e0wh1uwvn)

[Key Assumptions 1](#_mq7iuia0qeb)

[6.10. Waving 1](#_pp9cw8cghst7)

[Order Fulfillment Profiles 1](#_c3u51nrx894x)

[Build And Release Pick Waves (Legacy) 1](#_l1h7p6pxn8b4)

[Wave Profiles 1](#_ulk49uxbey63)

[Ready To Ship (RTS) 1](#_ypowr3glwzuc)

[Not Ready To Ship (NRTS) 1](#_kqskttxykof0)

[Mixed 1](#_55ahf7b5r4uf)

[Freight 1](#_pop87rr9oq4s)

[Exception Re-Pick 1](#_fne5gh4ku4pr)

[Bulk 1](#_2ciek153btmp)

[Exception Orders 1](#_qmfr0g2ls3h7)

[6.11. Pre-Manifest/Shipping Label Generation at the Wave 1](#_ehlf3i7ephv1)

[Data Requirements 1](#_l9008eao199)

[Order Release Profiles 1](#_wipme8l7u8o)

[Parcel Order Fulfillment 1](#_1tuee74)

[Parcel Overview 1](#_8yau48snav6)

[6.12. Ready To Ship Picking (Batch Picking) 1](#_184mhaj)

[Key Assumptions 1](#_os5s57m0j49y)

[6.13 Not Ready To Ship (Batch Picking) 1](#_3s49zyc)

[Key Assumptions 1](#_ox8r6f3kyll8)

[6.15 Mixed (Batch Picking) 1](#_elg7i8jfhool)

[Key Assumptions 1](#_ehgtub4kw5gg)

[6.15 Bulk (Single Scan Picking) 1](#_46gxtzsg5gk2)

[Key Assumptions 1](#_axl8met5bq59)

[6.16 Exceptions During Picking 1](#_279ka65)

[Packing 1](#_embyzclq1ssr)

[6.17.1 Ready To Ship Packing (Single Container Packing) 1](#_ftik4hgehqah)

[Process Flow 1](#_7v3e6u7jka7e)

[Key Assumptions 1](#_gtmxb3h7sanu)

[6.17.1 Over Packing (Single Scan Packing) 1](#_cs812v5ft8xl)

[Shipping Error Handling 1](#_run7yg7jmf64)

[Key Assumptions 1](#_8qvd32ycziwx)

[6.17.1 Bulk Packing (Bulk Packing) 1](#_65pn1s1fhx3r)

[Key Assumptions 1](#_bobvbvmf20hc)

[6.32. Shipping Requirements 1](#_7da6qkyrik79)

[6.32.5. Manifest 1](#_xn3g9k9g7qt6)

[6.32.8. Freight Terms Type 1](#_oeo1pbyfcfym)

[6.32.9. Prepaid 1](#_s21pd4nspw3j)

[6.32.10 Third Party 1](#_7t244lqz4s62)

[6.32.11. Shipping Options 1](#_42ueh451z3hg)

[6.33. Ship From Address 1](#_mcqve1fclgod)

[6.34 Reference Field Mappings 1](#_bcq4rehc3jmh)

[6.35 International Shipping Options 1](#_vpc00unryxhm)

[6.36 VAT 1](#_4lqo7s1uy3ae)

[6.37 Duties Paid By 1](#_d0o7rshs25l5)

[Special Services 1](#_nuuju88dqd8z)

[6.38 Hazmat 1](#_m5nomc51gbcv)

[**7. Freight/Retailer Processes 1**](#_ey0ht0jp4k6q)

[Warehouse Process Flow Overview 1](#_nj7302vcrvbp)

[Allocation Overview 1](#_xglkhrao7bvx)

[Wave Release 1](#_g08uvvaomqgw)

[Picking 1](#_qzm11twpx97m)

[Freight Packing 1](#_kwb3xvryrnr8)

[Trip Management 1](#_heowq5460vbx)

[Screen Flow 1](#_5i6wqqzg8hjp)

[Key Assumptions 1](#_1gw21bxuu8s2)

[Retail Requirements 1](#_4qr5x9r7gxy9)

[Carton Labels 1](#_1tsc3yxxh4cy)

[Print and Apply 1](#_o4cw3gkdi5zj)

[**8. Billing 1**](#_os91fn93upqg)

[8.2. Scheduler-Based Billing Configs 1](#_ag19ifvp4hae)

[Billing Charges 1](#_g7z2xto29rhv)

[8.3. AdHoc Billing Billing Configs 1](#_wluzrskhkvgo)

[AdHoc Billing Charges 1](#_s7sxim4dyhkp)

[**9.0 Integrations 1**](#_jr64h2snrrcm)

[UPS 1](#_mcatnsde3y16)

[FedEx 1](#_b6o8lae7gi9h)

[Stamps.com 1](#_n9q0fihh8i2e)

[Rest API Integrations 1](#_qpyspxjnxn7z)

[Deposco Integration Setup 1](#_y3sor167fjcl)

[Channel Cross-References 1](#_5999logez1i)

[Channel Listings 1](#_rurxwdfzsn0c)

[Webhook 1](#_mw21kr94gnb8)

[Verify the webhook 1](#_p91uksgs81t1)

[Master Data Interfaces 1](#_ppqkp19snmbv)

[Item Creation 1](#_zcqq5ieg16na)

[Key Assumptions 1](#_et67km3xkl8t)

[User Stories 1](#_j5txir1eagbt)

[Item Update 1](#_1qk3ak7xl2y7)

[Key Assumptions 1](#_mz93uau87riz)

[Trading Partner 1](#_mwyup749ifoy)

[Key Assumptions 1](#_o4xc4i755xp)

[Inbound Process Interfaces 1](#_349xftr46rtt)

[Purchase Order 1](#_h8s3ue1ozdph)

[Key Assumptions 1](#_njvgfk8f1d3o)

[User Stories 1](#_ypk2i9waub2w)

[Receipt Confirm 1](#_29axa8o274jx)

[User Stories 1](#_hdljci6skt8z)

[Outbound Process Interfaces 1](#_p9jutobnio9d)

[Customer Order 1](#_5txxwxbgce3i)

[Key Assumptions 1](#_mmpd2usnk1gf)

[User Stories 1](#_m8qgjqf4fdas)

[Ship Notice 1](#_w5dmg3da3i3d)

[Key Assumptions 1](#_p32bmffmsh5j)

[User Stories 1](#_t1cmgja8mo53)

[**Reporting 1**](#_vqctn9m24cwk)

[Process Reports 1](#_cawi6vw3lcfy)

[**UI Requirements 1**](#_zdrb2mtxu6t1)

[External/Internal Images & Notes 1](#_i25a3sym0aj0)

[**Customer Agreement 1**](#_pegce4l7umnd)

# 

# 

# Overview

# Overview

## Introduction

​Patterson Warehouses, Inc. is a third-party logistics provider headquartered in Memphis, TN. Established in 1856 as Patterson Transfer Company moving cotton from riverboats to storage in stagecoaches, they have since grown into one of the region's most trusted third-party logistics providers. Patterson Warehouses is a privately owned business with a primary focus in retail manufacturing, food and beverage, raw materials and web fulfillment.

Patterson Warehouses offers a comprehensive suite of logistics solutions, including service parts logistics, finished goods distribution, inbound logistics, inventory optimization, cross-docking, network optimization, and transportation management.

Currently operating in 6 facilities across 2 states with 2.5 million square feet of warehouse space.

The purpose of this document is to define the scope and processes required to meet the specific business requirements of Patterson Warehouses. This document is considered a living document and serves as a reference throughout the project for confirmation of approach.

## Scope

The full scope of this project is outlined in the Master SaaS Agreement and Statement of Work (SOW). This document is used to align on functionality to be implemented and is not considered a new Scope or Change Request (CR). A separate task list is provided with estimates of effort and duration to implement functionality outlined in this document. Estimates are not fixed, nor to be considered maximum/ ‘Not To Exceed’ (NTE) estimates and may change based on new information, design gaps, functionality gaps, or other. If estimates change before or during a task’s execution, Deposco notifies the customer to review options.

Changes in functionality requested after Master Design Document sign off that require significant rework may lead to overages beyond original estimate; therefore, a CR is required to execute so all parties are aware of the change. The purpose of a CR is to define all requirements of change to mitigate future rework.

Summary of licensed Deposco Products included in this project:

| **Product** | **Notes** |
| --- | --- |
| **Bright Warehouse** | Warehouse Management/WMS |
| **Bright Performance** | Reporting and Insights |
| **Bright Socket** | Integrations |
| **Bright Order** | Order Management/OMS |
| **Data Archive** | Data Store and Warehousing |

## Success Criteria

The success criteria of this project are as follows:

* Operational processes improvement
* Improved customer service & inventory management
* Improved labor efficiency
* Standardization for business units

## Terminology

| **Terminology** | **Definition** |
| --- | --- |
| **ASN** | Advanced Shipment Notice.  Receiving objects indicating a unique inbound shipment of inventory for the warehouse. |
| **Batch/Lot** | Group of items manufactured together |
| **BOL** | Bill of Lading.  A legal document used for the transportation of goods (non-parcel). |
| **CO (Customer Order)** | An order placed by a customer including header level information like destination address and line level information like item and quantity. |
| **SO (Sales Order)** | Sales orders are used for fulfillment processing through waving, picking, packing, and shipping. As a result, sales orders are also referred to as fulfillment orders. |
| **Business Unit** | A division or segment of an organization generally treated as a separate profit-and-loss center. Business units are defined in Deposco through the company entity. |
| **Error Message** | An error message that does NOT allow the user to continue processing after viewing the error. |
| **ATP** | Available to Promise. Inventory calculation used during allocation and when communicating inventory to an external system. |
| **Item/SKU** | Sellable product. Usually item name, item description, and item barcode are three unique values. |
| **Non LPN-Controlled Location** | Case-Pick or Active Locations that do not track LPN inventory (which may include FEFO for expiry inventory). |
| **Pallet** | A physical surface (skid) to stack LPNs on top of. Can be tracked systemically as an LPN or assists in tracking product movement where the group of LPNs move together. Can have one barcode for all inventory on the pallet or unique LPNs for each case (nested LPNs). |
| **Putaway** | The systematic process of locating products to a permanent location. |
| **Reserve Location** | Storage locations with either full or partial LPNs. |
| **Ship Notice** | Notice that is sent to an external system through a socket integration to notify the external system that part or all of an order was fulfilled. Ship notices are sent on a real-time basis as each shipment is created. |
| **Shipment** | For outbound processes, a shipment contains the shipping information for a sales order (SO), including the dimensions of the shipping container and the tracking number. A shipment may include part of an order, one order, or more than one order, depending on the business scenario. |
| **Work Group** | A portion of orders in a pick wave. For example, if a pick wave includes 100 orders, and pickers use a cart that can hold between one and eight bins, then the 100 orders can be separated into work groups that contain between one and eight orders each so that each picker's cart is not overfilled by the end of picking. Large orders can be separated into a work group of two orders per cart, while small orders can be separated into a work group of eight orders per cart. |
| **Pick Tasks** | Within a work group, the details of the item and quantity that must be picked from a pick location. |
| **Socket** | A pre-connected integration package to a key business system such as an e-commerce shopping cart, online marketplace, ERP and accounting system, or shipping carrier. Sockets are named for the connecting system, such as Amazon Seller Central, Shopify, or NetSuite. You can create one or more instances of each socket, which are called socket integrations. |
| **Scheduler Task** | Specifies when to run a background process in the Deposco application, such as the creation of a list of replenishment tasks by priority. A scheduler task can run at any time and can update data sets for entities. |
| **Screen Message** | A screen message is an informational message that does not prevent the user from continuing to process their task. |
| **Pick Wave** | Distribution Order selection process to release work to the warehouse. A Pick wave follows selection, allocation, and task creation. |
| **Ready To Ship** | Items that can be shipped in their original container; in other words, they do not require packing in a separate shipping container. |
| **PAL** | Process Action Link. Used in the Deposco UI to initiate processes ad hoc. |
| **View** | A defined set of filters for an entity list. Admin users can create and edit views for any entity that they have permission to modify. Non-admin users can create their own views and use the standard shared views, but cannot edit the shared views created by admin users. |
| **Layout** | A predefined set of fields that appear on an entity detail page. Admin users can create and edit the standard layouts for entities. Users can create personal layouts for entities. |
| **Order Release Profile** | Wave Processing Type.  Determines the allocation strategy for a Pick Wave (e.g. Bulk Item Pick) |
| **Work Order** | Work orders in deposco are specific to kitting activities. Customer requested work, outside of kitting, is handled via Ad Hoc Billing transactions. |

## Acronyms

| **Acronym** | **Definition** |
| --- | --- |
| 3PL | Third-party Logistics |
| ASN | Advanced Shipping Notice |
| DC | Distribution Center |
| DEP | Deposco |
| CO | Customer Order |
| BOL | Bill of Lading |
| FTL | Full truckload. Also, referred to as TL |
| ATP | Available To Promise |
| UPC | Universal Product Code |
| LPN | License Plate Number |
| VAS | Value Added Service |
| LTL | Less Than Truckload |
| PO | Purchase Order |
| SO | Sales Order. Also, referred to as Fulfillment Order |
| RTS | Ready to Ship |
| UOM | Unit of Measure |
| B2B | Business-to-Business |
| B2C | Business-to-Consumer |
| DTC | Direct-to-Consumer |
| RMA | Return Merchandise Authorization |
| SCAC | Standard Carrier Alpha Code |
| COO | Country of Origin |
| UI | User Interface |
| PAL | Process Action Link |
| EL | Event Log |
| ESL | Event Subscription Log |
| SKU | Stock Keeping Unit |
| UA | Test/Sandbox Environment |
| UAT | User Acceptance Testing |
| PROD | Production Environment |

# Master Data

## Facility

A facility is a physical location for a company, typically a building such as a warehouse/distribution center or a retail store. A facility entity is created in Deposco for every facility that is involved in warehouse and order management activities.

Facilities are added manually in the Deposco User Interface by navigating to Setup > Facilities. The following facilities are in scope for the initial implementation:

| **Facility** | **Description** |
| --- | --- |
| PAT001 | Building A |
| PAT002 | Building B |
| PAT003 | Building C |
| PAT004 | Building D |
| PAT005 | Building E |
| PAT007 | Marathon |

## Company

A company is the root entity in Deposco. All other entities in Deposco are assigned to a company. For example, each facility, zone, product category, item, order, and so on is assigned to a company. In a 3PL setting, a Company is also referred to as a “Business Unit” or a “Brand”.

Companies are added manually in the Deposco User Interface by navigating to Setup > Companies. The following Business Units are in scope for the initial implementation:

| **Business Unit** | **Description** |
| --- | --- |
| Patterson | Master Business Unit |
| HKC | Initial go live business unit |

The following company attributes are available to drive logic in certain workflow processes:

| **Field Label** | **Field Name** | **Values** | **Description** |
| --- | --- | --- | --- |
| Record Accounting Data | recordAccounting | TRUE/FALSE | Determines if Inventory General Ledger records are created. Suggested to always be set to TRUE. |
| UCCEAN Company # | ucceanCompanyNumber | Integer | Evaluated during SSCC label generation. |

### Key Assumptions

* While the concept of Parent and Child Business Units exists, it is not included in the scope of this project.

## Trading Partners

A trading partner, also known as consignee or consignee partner, in Deposco represents an external party to the company that is typically associated with an order. The primary types of partners include:

* Customer – Used for partner companies that are purchasing products from the company and are therefore associated with sales orders. Also used for general shipping purposes.
* Vendor – Used for partner companies that a company is buying products from and are therefore associated with purchase orders. Also used for partners that provide drop shipping services.

| **Trading Partner** | **Description** |
| --- | --- |
| Walmart | Trading Partner |
| Home Depot | Trading Partner |
| Lowe’s | Trading Partner |
| Amazon | Trading Partner |
| Build.Com | Trading Partner |
| Clark Core Services | Trading Partner |
| Eaccess | Trading Partner |
| Essendant/CPO COmmerce | Trading Partner |
| HD Supply | Trading Partner |
| Home Depot | Trading Partner |
| Jet.com | Trading Partner |
| Meijer | Trading Partner |
| Menards | Trading Partner |
| Mule/Shopify | Trading Partner |
| Northern Tools | Trading Partner |
| Prominence Home | Trading Partner |
| Ollie’s | Trading Partner |
| Overstock | Trading Partner |
| Sherwin Williams | Trading Partner |
| Spreetail | Trading Partner |
| Tamalpais Paint | Trading Partner |
| Tractor Supply | Trading Partner |
| Wayfair | Trading Partner |
| Houzz | Trading Partner |

The following trading partner attributes are available to drive logic in certain workflow processes:

| **Attribute** | **Values** | **Description** |
| --- | --- | --- |
| Packing Slip Report | String | Evaluated during packing. Determines the packing slip report used. |
| ZPL Label Report | String | Evaluated during packing. Determines the carton label report used. |
| Pallet SSCC Label | String | Evaluated during packing. Determines the pallet label report used. |
| SSCC Nbrs Reqd for Trip | TRUE/FALSE | Evaluated during packing. Determines if an order requires SSCC label generation. |
| UCCEAN Company # | Integer | Evaluated during SSCC label generation. |
| Expiration Threshold in Days | Integer | Amount of days between the release of a pick task to a stock unit with expiration date attached. e.g. 540, 180 |

### Key Assumptions

* A Trading Partner exists on all inbound and outbound orders, regardless of fulfillment type (e.g. B2B, DTC).
* All Trading Partner/Vendor requirements are maintained by Patterson Warehouses.
* All ucc/sscc labels required must be provided to Deposco by Patterson Warehouses

## Locations / Zones / Storages

### Locations

Locations are specific areas in a facility that are used for tracking movement of physical objects such as inventory, containers, and shipments. A location can be a specific inventory storage shelf, or a location might represent an area like the receiving or shipping area of the facility.

The following standard locations are defined:

| **Location** | **Type** | **Storage Type** | **Description** |
| --- | --- | --- | --- |
| Receiving | Staging | Receiving | Standard location where inventory is staged after receiving and before putaway. |
| Returns | Staging | Staging | Standard location where inventory is staged after receiving a customer return. |
| Damages | Damaged | Damaged | Standard location where inventory is staged if identified as Damaged during receiving. |
| Kitting Workcell | Staging | Staging | Standard location where inventory is staged in order to complete a kitting activity (e.g. work order). |
| WHSE-Floor | Staging | Staging | Standard location where inventory is staged during the picking process. |
| Shipping | Staging | Shipping | Standard location where inventory is staged after picking and before packing. |
| Receiving Lanes | Staging | Receiving | Receiving locations where inventory is staged after receiving and before putaway. |
| Packing Stations | Staging | Shipping | Pack stations to be used for staging inventory after picking. |
| Each Picking Locations | Permanent | Split Case Picking | Forward picking locations used to pick loose stock (eaches). Commonly used in D2C fulfillment operations. |
| Case Picking Locations | Permanent | Full Case Picking | Picking locations used to pick full case quantities. Commonly used in B2B fulfillment operations. |
| Reserve Locations | Permanent | Reserve Storage | Reserve/Bulk locations used to store full pallet quantities. Commonly used as suggested locations for inbound putaway and as source locations for replenishment. Certain order types can pick from these locations. |

#### Location Definition

The Location’s locator value contains 4 sections, each divided by a separator character. The locator should not exceed 11 alphanumeric characters.

| **Description** | **Field Length** | **Separator** |
| --- | --- | --- |
| Aisle | 2 |  |
| Bay | 3 | - |
| Level | 3 | - |
| Position | 1 | - |

##### **Key Assumption**

* The Location Number is displayed on all handheld process screens. The user scans the locator.
* All locations must have a pick sequence.

### Zones

A zone is an area in a facility that is dedicated to a specific functional task, such as picking, receiving, or shipping.

The following zones are defined as part of this project. These are standard zones and can be adjusted as needed throughout the project life cycle. Any adjustments to zone master data may require additional configuration to other workflow processes.

| **Zone** | **Type** | **Description** |
| --- | --- | --- |
| Receiving | Staging | Standard zone to control all receiving locations. |
| Quality | Staging | Standard zone to control all damaged or quarantined locations. |
| Returns | Staging | Standard zone to control all returns receiving locations. |
| Shipping | Staging | Standard zone to control all shipping locations. |
| Kitting | Kitting | Standard zone to control all Kitting Locations. |
| Staging | Staging | Standard zone to control WHSE-Floor or temporary locations. |
| Floor Picking | Picking | Picking zone. All locations in this zone are eligible for picking. Only lower-level locations are included in this zone. Replenishments are driven to locations within this zone. |

### Storages

Storage entities are used to assign an item (more specifically, the pack for an item) to a picking location. If multiple items are stored in a single location, a storage entity must be created for each item in the location.

As part of the definition of a storage entity, the minimum and maximum inventory levels for the item in that picking location must be specified. When the stock in the location falls below the minimum, a warehouse replenishment task can be generated to move inventory from a reserve location so that the quantity of inventory at the picking location equals but does not exceed the specified maximum.

### Key Assumptions

* All Locations are barcoded and can be scanned using an RF device.
* Locations are defined as single SKU or mixed SKU based on the Mixed Item Threshold field: 1=Multi SKU; 0=Single SKU.
* Location pick sequence and putaway sequence are included as part of the Location Master Data and are maintained by the customer team. These fields, along with the zones to which each location is assigned, play a key role in how stock is allocated and pick tasks are defined and sorted for work groups on a pick wave.
* All Locations have a predefined, alphanumeric, mask. This is to be determined by the customer team.
* Any adjustments to zone master data may require additional configuration to other workflow processes.
* All locations must be associated with a zone.

### Steps To Add/Remove a Location

* A Location can be added to the system by being uploaded via Data Exchange or created manually in the Deposco User Interface.
* It is not recommended that Location Master Data is removed from the system once in use, as there is historical process data tied to the specific location. Instead, the Location can be repurpose/updated or the status can be adjusted to an inactive status, such as “Disabled” or “On Hold”.
* In order to make the warehouse location pick sequencing flexible, it is recommended that the numbers are the same length and a few trailing zeros are added to each numerical value. For example, if a location needs to be added in between two existing locations and the current pick sequence is 10001000, 10002000, it is much easier to simply upload a location with a pick sequence of 10001500 than it would be if the original pick sequence did not have trailing zeros (e.g. 10001 10002).

Locations, Zones, and Storages are uploaded via Data Exchange or created manually in the Deposco User Interface. Specific mapping requirements are outlined in more detail in the Data Exchange section of this document.

## Items

An item, or stock keeping unit (SKU), is a distinct type of item for sale, including all attributes that distinguish the item type from other item types, such as manufacturer, description, material, size, color, and packaging.

The following Item attributes are configured as part of the initial implementation:

| **Label** | **Field** | **Type** | **Source** | **Description** |
| --- | --- | --- | --- | --- |
| HS Code | harmonizationCode | true/false | API | Used as input criteria in Planned Putaway Date calculation |
| Eligible Countries of Origin | originCountry | string | API | Used to validate the origin countries for a given item during the receiving process |

### 

The following inventory attribute tracking is available as a part of the initial implementation:

| **Inventory Attribute** | **In Scope** |
| --- | --- |
| UPC(s) | Yes |
| Lot Tracking | Yes |
| Expiry Tracking | Yes |
| Serial Tracking | Yes |
| Born on Date Tracking | No |
| Hazmat | Yes |

### 

### Key Assumptions

* Lot Number Tracking and Serial Tracking is configurable on Item Master.
* Items are not required to have an item UPC if using Pack upc’s.
* Packs are required to have a pack upc/gtin if that is to be used for the business unit.
* All upc’s must be unique in the system.

The Item master is defined in a third-party system, integrated via data exchange. Item updates may occur in the Deposco User Interface.

## Packs

Packs enable you to manage the stock of an item in multiple units of measure. For example, you may stock individual beverage cans (which are called *Each* packs or *Eaches*), and also cases of 12 beverage cans. A single item can have multiple packs of a given type (e.g. case–12, case–24, pallet–100, pallet–150). The weights and dimensions of each item are stored in the packs for the item.

The following pack types are supported:

| **UOM** | **Definition** |
| --- | --- |
| **Each** | The smallest UOM for a given item within the warehouse. |
| **Case** | A UOM containing more than one Each pack quantity for a given item within the warehouse. |
| **Inner Pack** | A UOM containing more than one Each pack quantity for a given item within the warehouse, typically smaller than a case. |
| **Pallet** | Typically, the largest UOM for a given item within the warehouse. A UOM containing more than one Each or Case pack. |

### Key Assumptions

* Every item must have at least one pack with a quantity value of 1 (each), which represents the most granular unit by which the item is managed.
* With the exception of the Each pack, a given item can have multiple pack records of the same type with different sizes (e.g. Case–12, Case–24).
* If performing cartonization, Pack Length, Width, Height, Weight and UOM are required fields.
* Pack dimensions are measured in Imperial units (i.e. inches, lbs), by default. Other UOMs can be configured (e.g. centimeters, kgs). However, Deposco is not responsible for conversion of Pack UOMs.
* Cartonization can support both Imperial and Metric UOMs. The following UOMs are supported:
  + ounces, oz
  + pound, lb, lbs
  + kg, kilo, kilos
  + g, gram, grams
  + in, inch, inches
  + ft, feet, feets
  + yd, yard, yards
  + cm, centi, centis
  + m, meter, meters
* A single Pack record should have all UOMs in one measurement system (e.g. weight in lbs, length in inches).
* There is only one RTS Pack Per Item

Packs are defined in a third-party system, integrated via data exchange. Pack creation and updates may occur in the Deposco User Interface.

## Users

Deposco provides two different features for creating and managing users and passwords: *User Management*, which provides full user management functionality and requires membership in the Admin group, and *Associate User*, which can be configured to provide more limited user management functionality and is available on an application menu.

The Associate User feature is intended for users such as warehouse department managers or assistant store managers, who may need the ability to create and manage users for specific facilities but who should not have full administrative access to Deposco.

### Company Assignments

Company assignments limit users to viewing data and performing tasks for only the companies to which they are assigned. For example, entity lists are automatically filtered to display only entities for companies that the user has permissions for, and picking workflow processes can be configured to require the user to select one of their assigned tenant companies and then display only orders for the selected tenant.

Use the following strategies to set up company assignments for a user account:

* If a user should be assigned to a single tenant company, then select the facility in the User of Tenant field on the User Information tab for the user account.
* If a user should be assigned to multiple companies, then enable company assignments in the user management options. Then select the primary company in the User of Tenant field on the User Information tab for the user account, and assign additional companies on the Company Assignments tab.
* If a user should be able to access data for all tenant companies, then leave the User of Tenant field blank on the User Information tab for the user account.

### Key Assumptions

* Users, Groups, and Permissions are set up through the Deposco User Interface to ensure the right users have access to the right functionality/permissions.
* Associate Users can be created manually in the User Interface.
* Tenant users are set up using the same process as internal users.

## Carton Types

Carton types typically represent the different sizes of shipping boxes, and are used to facilitate the process of entering box dimensions during packing. Carton types are required for cartonization, which is the process of evaluating the contents of an order to determine the number and size of each shipping carton required for the order.

### Available Carton Types

The following carton types are supported in the initial implementation:

| **Name** | **Notes** |
| --- | --- |
| **8x8x8** | PAT to add tare weights |
| **22x14x14** | PAT to add tare weights |
| **21.5x11.5x8** | PAT to add tare weights |
| **19.5x10.25x8** | PAT to add tare weights |
| **18x12x12** | PAT to add tare weights |
| **12x12x12 DW** | PAT to add tare weights |
| **12x12x12 SW** | PAT to add tare weights |
| **10x10x10** | PAT to add tare weights |
| **6x6x6** | PAT to add tare weights |
| **26x16x14** | PAT to add tare weights |

### Key Assumptions

* Carton Types have accurate dimensions defined.
* Outer Dimensions are used when communicating with shipping carrier APIs.
* Inner Dimensions are used when communicating with the Deposco cartonization API.
* All active Carton Types are considered during cartonization and packing processes. The option is available in the UI to disable a specific Carton Type.
* Additional carton types can be added by the Patterson Warehouses team.

Carton Types are uploaded via Data Exchange or created manually in the Deposco User Interface. Specific mapping requirements are outlined in more detail in the Data Exchange section of this document.

## Containers

A container is any object that holds items, such as a cart, bin, pallet, or shipping box. A container is also called an LPN (license plate number) since it has a preprinted, sequential barcode label for tracking.

### Types

* **Cart** - typically used for picking carts or for carts that are used for moving inventory. Carts are reusable containers.
* **Pallet** - typically used for inventory containers. Inventory may be received, put away, and moved while on a pallet container.
* **Transient** - typically used for reusable containers such as totes or bins.
* **Shipping** - typically created automatically by the system for containers that are shipped, including both parcel shipping containers and pallets that are used for shipping. Shipping pallets are typically identified with a secondary container type of Pallet.

### Key Assumptions

* A container is associated with a single location at a time.
* Multiple stock units can be assigned to a single container.
* Containers can be assigned to another container (e.g. shipping boxes or LPNs on a pallet).
* During receiving, a container can be associated with multiple receipt lines.
* During shipping, the container is associated with a shipment before the shipment completes. After the shipment completes, the container becomes a container history (ContainerHist) entity, which is a separate entity from a container.
* All totes, bins, and carts are uploaded to the system. This allows the system to perform validations on user scans during handheld processes.
* All totes, bins, and carts are barcoded and can be scanned using an RF device.

Containers are uploaded via Data Exchange or created manually in the Deposco User Interface. Specific mapping requirements are outlined in more detail in the Data Exchange section of this document.

## Kit Headers and Details

A kit header is a parent record that identifies a kit that is built from items. Each kit header has one or more kit details, which describe the individual items that compose the kit. Both input (items consumed by kit creation) and output (items created from kitting) items are held in kit detail records.

### Key Assumptions

* A kit header must exist in Deposco before a work order can be created using that kit header.
* By default, kitting activities take place in a Kitting Workcell location.

Kit Headers and Details are uploaded via Data Exchange or created manually in the Deposco User Interface. Specific mapping requirements are outlined in more detail in the Data Exchange section of this document.

## Work Orders

A Work order is created in order to build kits. Each work order has an associated kit header, which serves as the BOM for the work order.

### Key Assumptions

* A kit header must exist in Deposco before a work order can be created using that kit header.
* The work order entity is solely used for kitting purposes.

## Components

A component is a record that establishes the relationship between a kit item and the items that compose of the kit. Once this entity is created, the Kit To Stock workflow process can be used to assemble kits. In addition, the Component entity drives Kit To Order logic as the Sales Order allocates to inventory tied to Item Components.

### Key Assumptions

* By default, kitting activities take place in a Kitting Workcell location.
* A component item cannot also be a kit item. Meaning, nested kits are not supported.

Components are uploaded via Data Exchange or created manually in the Deposco User Interface. Specific mapping requirements are outlined in more detail in the Data Exchange section of this document.

# Common System Functionality

This section outlines requirements for core system features that can be attached to any process. This section outlines key features that need to be configured during the implementation cycle (or post implementation) and does not outline all common functionality (such as users, permissions, UI navigation, reporting, etc.). For information on these general topics, please refer to [help.deposco.com](http://help.deposco.com/).

## 3.1. RuleService Process Configurations

All features outlined below are configurations in Deposco RuleService that can be applied on top of handheld processes to modify or enhance behavior. These are initially configured by Deposco, and Patterson Warehouse SuperUsers are then able to modify. SuperUsers can configure these features if they have enough comfort level with RuleService.

### 3.1.1 Printing

Any reports or labels in Deposco can be printed automatically during a process based on configurable logic.

### 3.1.2 Validations

Users can be prevented from proceeding through handheld screen flows with a corresponding error message, based on configurable logic. Common applications of this are to:

* Prevent specific users from moving inventory to zones.
* Prevent mixing of items within specific locations.
* Require additional information to be captured for specific customers.
* Prevent accidental processing of an order or shipment until specified date or flag is marked on order.

# Inbound Processes

This section outlines all inbound related processes. This includes receiving at the warehouse, returns processing, staging logic, putaway/user directed moves, and PO management.

## 4.1. Pre-Receiving

### 4.1.1 Inbound Order Creation

The below table outlines the different types of inbound orders and their creation method.

| **Order Type** | **Description** | **Creation Method** |
| --- | --- | --- |
| **Purchase Order** | Stock ordered from a vendor or supplier is tracked and managed by using a purchase order. | API/User Interface |
| **Inbound Shipment** | Shipment that is created outside of Deposco when a supplier sends stock to fulfill a purchase order. Commonly referred to as an Advanced Ship Notice (ASN). | API, User Interface, Handheld |
| **Customer Return** | Stock sent back for an exchange or refund. Commonly referred to as a Return Merchandise Authorization (RMA). | API or Handheld |

#### 

## 4.2. Purchase Order Management

PO management outlines how purchase orders (inventory coming into the warehouse) are defined, managed, and monitored in Deposco. A PO can refer to an actual PO, to an external vendor, or an internal PO such as for in-house manufacturing.

### Types of Purchase Orders

#### 4.2.1. Purchase Orders

Inventory is listed on a PO with items, pack types, and quantities listed as separate lines. As inventory is received against these lines, unique receipts are tied to the PO line, and the net received quantity is updated.

### 4.2.1.1 Key Assumptions

* Patterson receiving team creates a purchase order with the container number (orderHeader.notificationCode1) and footage prior to receiving the containers (orderHeader.notificationCode2)

#### 4.2.2 Blind Orders

There is no existing information in Deposco on what is to be received, and users scan any items/quantities against a user-entered or auto-generated next up number. Additional information can be captured during the Receiving process and logged against the Blind Order for reference later. A Blind PO is later typically merged with a new/existing PO manually by a receiving supervisor.

### 4.2.2.1 Key Assumptions

* 85% of Patterson Warehouse receipts are blind receipts.

#### 4.2.4 Customer Returns

Returns can be received using the same Receiving process as normal receiving, except that it is processed against an RMA that is typically

#### 4.2.5 Over/Under Receiving

Over or under receiving thresholds can be set as a fixed value or percentage rule globally, by vendor, by item, or any logic set. This can be used to prevent users from accidentally receiving quantities tied to a previous Sales Order (SO). An RMA does not have to be tied to an existing SO. Customer Returns prompt for additional information such as inventory condition; any additional information can also be captured during the receipt if needed.

### Configuration Options, Validations, Tipsat do not match the PO. Preventing over receiving of quantity can sometimes interrupt operations because it requires that a user stage the inventory without systematically receiving it; a standard recommendation is to allow over receiving but send an email to alert a contact.

### 4.2.5.1 Key Assumptions

* Users are allowed to overreceive any quantity.
* Over-receiving can only be controlled if there is a purchase order to receive against.

#### 4.2.6 Items Not on PO

Receiving items not listed on the PO can be prevented or allowed and added to the PO as an additional line. If the item was not on the original PO, the order quantity shows zero and the received quantity reflects actual.

For operations that do not want to allow receipt of items not on PO (especially if inbound shipments have mixed POs so it's difficult to know which PO the item mistakenly came on), it's recommended to have users receive these against a Blind PO so that later a receiving supervisor can review and reconcile manually.

### 

#### 4.2.7 PO/Customer Return Creation

* Manual creation through the desktop
  + Tenant creation
  + Patterson Super User creation
* PO Entry
* POs are transmitted to DEP via the following methods:
  + Creation via custom CSV import
  + Creation via EDI
  + Creation through PO entry

#### 4.2.8 Purchase Orders

* An email is sent to the CSR group upon completion of a PO being received. This displays over/under receipts. This email is triggered and sent when the user selects the “finished” button in the general receiving process.
* When items not on the PO arrive, a supervisor must either update the PO with new po lines or create a separate independent PO.

#### 4.2.9. LPN Receiving

Patterson uses the receiving process to receive stock and assign it to an LPN which is staged in a receiving location. Once the stock and LPN have been received operators use the directed putaway process to be routed to the optimal putaway location.

**4.2.9.1 Key Assumptions**

* Deposco prints out LPN Labels at the time of receiving with the below information populated on a 4x3 zpl label:

- Business Unit

- Container #

- Date Received

- Item #

- Qty on the LPN

#### 4.2.10. Customer Returns

* Customer Return POs are created using similar methods as PO.
* The operator receiving the return determines the disposition code at the time of receipt. They can determine if the item is damaged, hazmat, quarantine etc

## 4.3 Receiving Processes

The Receiving processes are used to perform the receipts of the POs outlined above and to create inventory in Deposco. The receiving process can be used real time as a part of the trailer unload or may be received after all inventory is physically unloaded, as is common with live unloads where carrier charges may apply if a certain duration is exceeded. Inventory can be received to a container (typically pallet) or location and be auto-directed or user directed/scanned. Scanning takes more time than if the staging is automatic but has the benefit of improving accuracy.

Any receiving strategies can be used in various combinations, so long as criteria can be defined of when a strategy should apply.

#### 4.3.1. Loose Stock Receiving

Each receipt is received as loose inventory into a location within the Receiving zone. A user initiates the receiving process by navigating to the Inbound menu on the handheld and selecting ‘Loose Stock Receiving’.

The user is then prompted to scan a putaway location per receipt. The putaway location must reside within their active zone assignment (i.e. Receiving). Once the putaway location is scanned, the inventory is staged for operations to put away to a permanent location or move to a special services location.

1. The user enters/scans the PO Number.
2. The Receive Item Screen displays the item number and the allowable quantity to be received. The user then scans the item number to be received.
3. The user is prompted to select the pack type and quantity to receive. If multiple pack types are configured for the item being received, the user has the ability to select any available pack type (e.g. each, case, pallet).
4. If the item has inventory tracking attributes enabled (e.g. Lot Number, Expiration Date), the user is prompted to enter the applicable attributes on the Item Capture Screen.
5. The user is prompted to scan a putaway location within their active zone assignment.
6. Once the inventory is received, the receiptLine is systematically created.
7. If more items are available to be received on the PO, the user is brought back to the Receive Item Screen. If no more items are available to be received on the PO, the user is brought back to the Inbound menu and the PO is updated to a status of “Received”.

#### 4.3.2. LPN/Pallet Receiving

Each receipt is assigned a unique, system-generated inbound LPN (iLPN). A user initiates the receiving process by navigating to the Inbound menu on the handheld and selecting “LPN Receiving”.

As each iLPN is received, it is systematically created as a container with a type of “Pallet”. The user is then prompted to scan a putaway location per receipt. The putaway location must reside within their active zone assignment (i.e. Receiving). Once the putaway location is scanned, the container is staged for operations to put away to a permanent location or move to a special services location.

1. The user enters/scans the PO Number.
2. The Receive Item Screen displays the item number and the allowable quantity to be received. The user then scans the item number to be received.
3. The user is prompted to select the pack type and quantity to receive. If multiple pack types are configured for the item being received, the user has the ability to select any available pack type (e.g. each, case, pallet).
4. If the item has inventory tracking attributes enabled (e.g. Lot Number, Expiration Date), the user is prompted to enter the applicable attributes on the Item Capture Screen.
5. The LPN is pre-populated for the user to verify prior to printing.
6. The LPN label is printed and applied to the pallet.
   1. Each Label contains the following information
      1. - Business Unit
      2. - Container #
      3. - Date Received
      4. - Item #
      5. - Qty on the LPN
7. The user is prompted to scan a putaway location within their active zone assignment.
8. Once the LPN is put away the receiptLine is systematically created.
9. If more items are available to be received on the PO, the user is brought back to the Receive Item Screen. If no more items are available to be received on the PO, the user is brought back to the Inbound menu and the PO is updated to a status of “Received”.

#### 4.3.3. Returns Receiving

Returns Receiving is handled through Deposco’s Receiving process.

The types of customer returns are detailed in the below table:

| **Type** | **Definition** | **Creation** |
| --- | --- | --- |
| Customer Return | Customer return that is associated with a return merchandise authorization (RMA) | API, User Interface |
| Blind RMA | Customer return that is not associated with a return merchandise authorization (RMA) | Real-time during Receiving process |

During the returns receiving process, reason codes are captured and stored on the Receipt Line “Disposition” field.

### User Stories

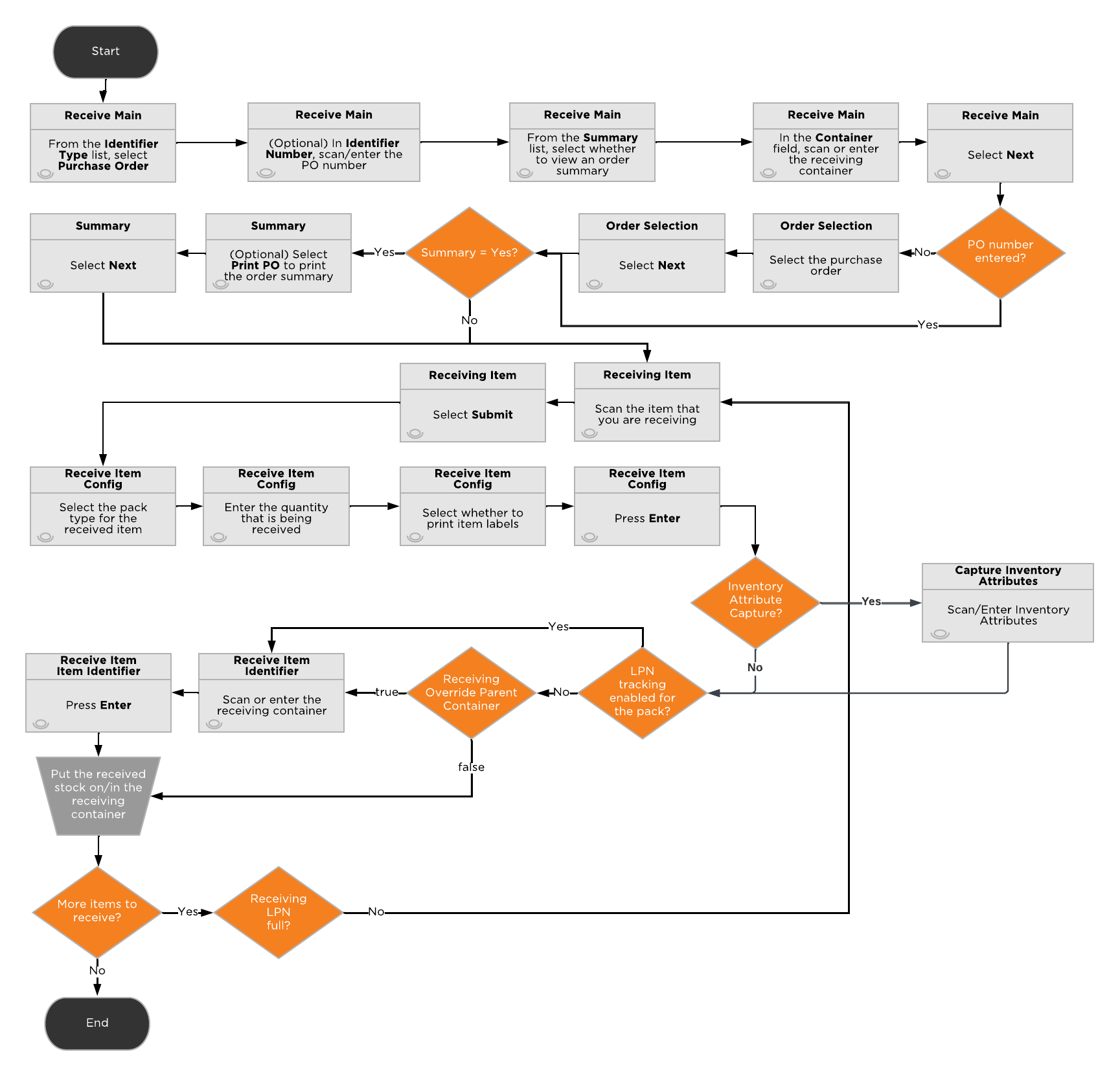
As a receiving user, I have the ability to…

* Receive inventory based on an existing customer return.
* Create a customer return based on the tracking number.
* Blindly receive a customer return.
* Capture disposition codes for each item being returned.

### Key Assumptions

* Inventory is created as part of this process in one of the configured customer returns staging locations.
* Inventory is received loosely into a location within the Returns zone
* The locations within the Returns zone allow for multiple lot numbers, expiration dates, and disposition codes to exist for the same item.
* The locations within the Returns zone allow for multiple items to exist in the location at the same time. Storage records do not exist in the returns zone.
* Return labels are not printed using this process.

### Process Flow



#### 4.3.4. Inbound/Purchase Order Damages Receiving

If damaged inventory is identified, it is received using the Damages Receiving HH option.

During the Damages Receiving process, reason codes are not captured on the handheld device. If a receipt is flagged as damaged, a code of “Damaged” is stored on the Receipt Line “Disposition” field. If a receipt is flagged as damaged, the inventory is directed to be staged at a location within the Damages zone.

## 4.4. Patterson Specific Requirements

Patterson requires an emailed receipt report to be sent to the customer service team whenever a purchase order or blind receipt has been received. Images are taken during returns and damages receiving process and then saved to the order level.

#### 4.4.1 Prompt for Location

After the user receives the inventory for a customer return, they are prompted for a location to put the inventory to. The user selects a disposition code such as “Damage” which systematically places the received stock into the location that is suggested by the associated disposition code.

4.5.1.1 Key Assumptions

* Patterson has suggested locations for customer returns, blind receipts and purchase orders

#### 4.4.2 Entity Images

During the receiving process, the user can take receipt level pictures in the “Damaged Receiving” process. These pictures are be attached to the PO entity to be viewable from the desktop UI.

#### 4.4.3 Item Attribute Tracking

Patterson Warehouses does NOT track the below base stock unit fields for each applicable item that is configured through master data for Phase 1 go live.

1. Serial Number
2. Lot Number
3. Expiration Date

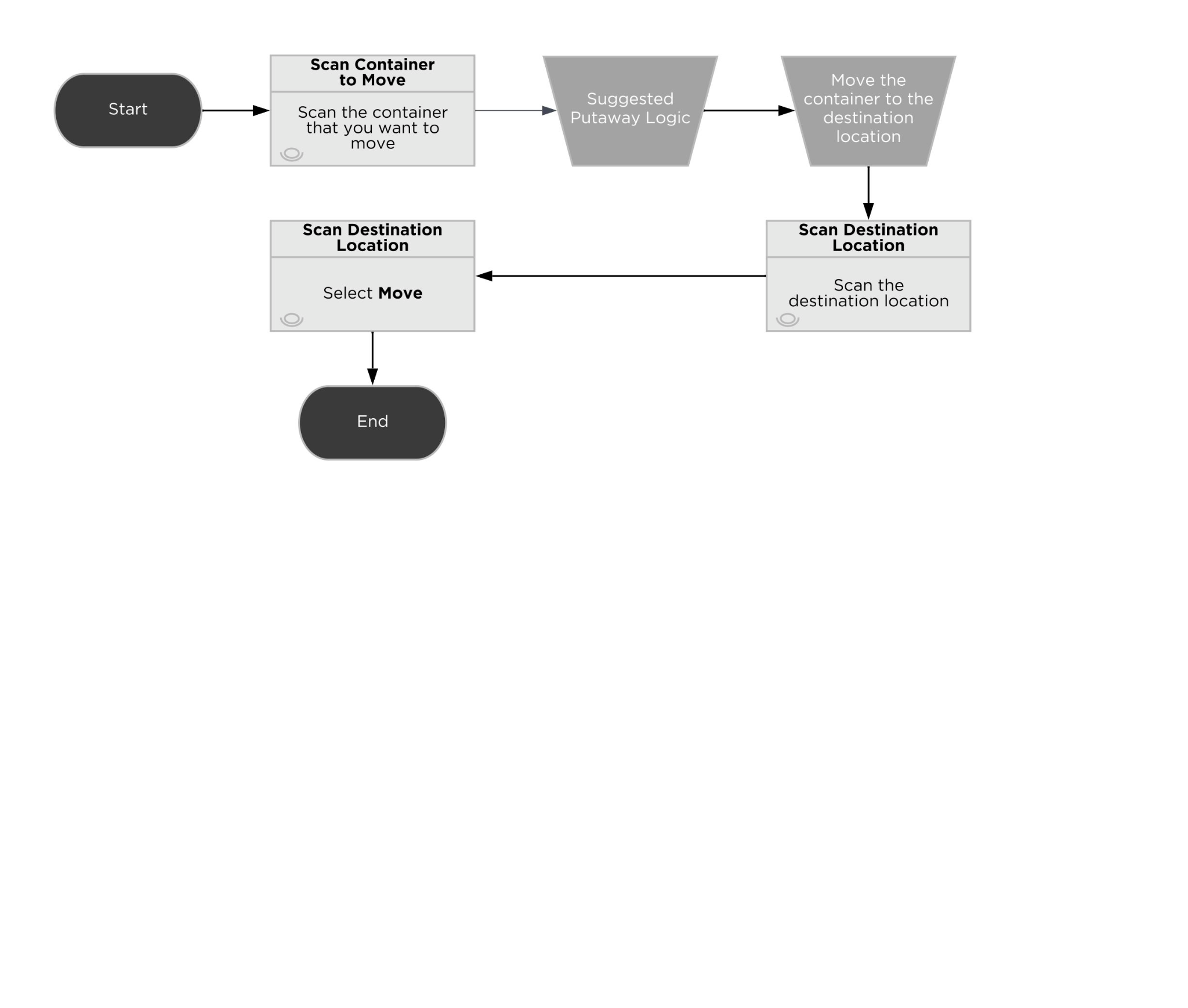
# 4. Inventory Movement Processes

## 4.5. Putaway Processes

#### 4.5.1 Move Container

Allows the user to scan a container then the destination location; simple, fast method to move a container between locations. Deposco suggests different putaway locations in Move Container than in Move Stock. If the PAT team uploads storage details for packs/items then Deposco overrides the suggested putaway screen to show empty locations within a zone or show the location with storage details for a specific pack. This process should be used to move received LPN’s from receiving to either testing locations or reserve locations. There are no validations for inventory attributes, such as lot/serial number provided through base solution.

Process Flow:

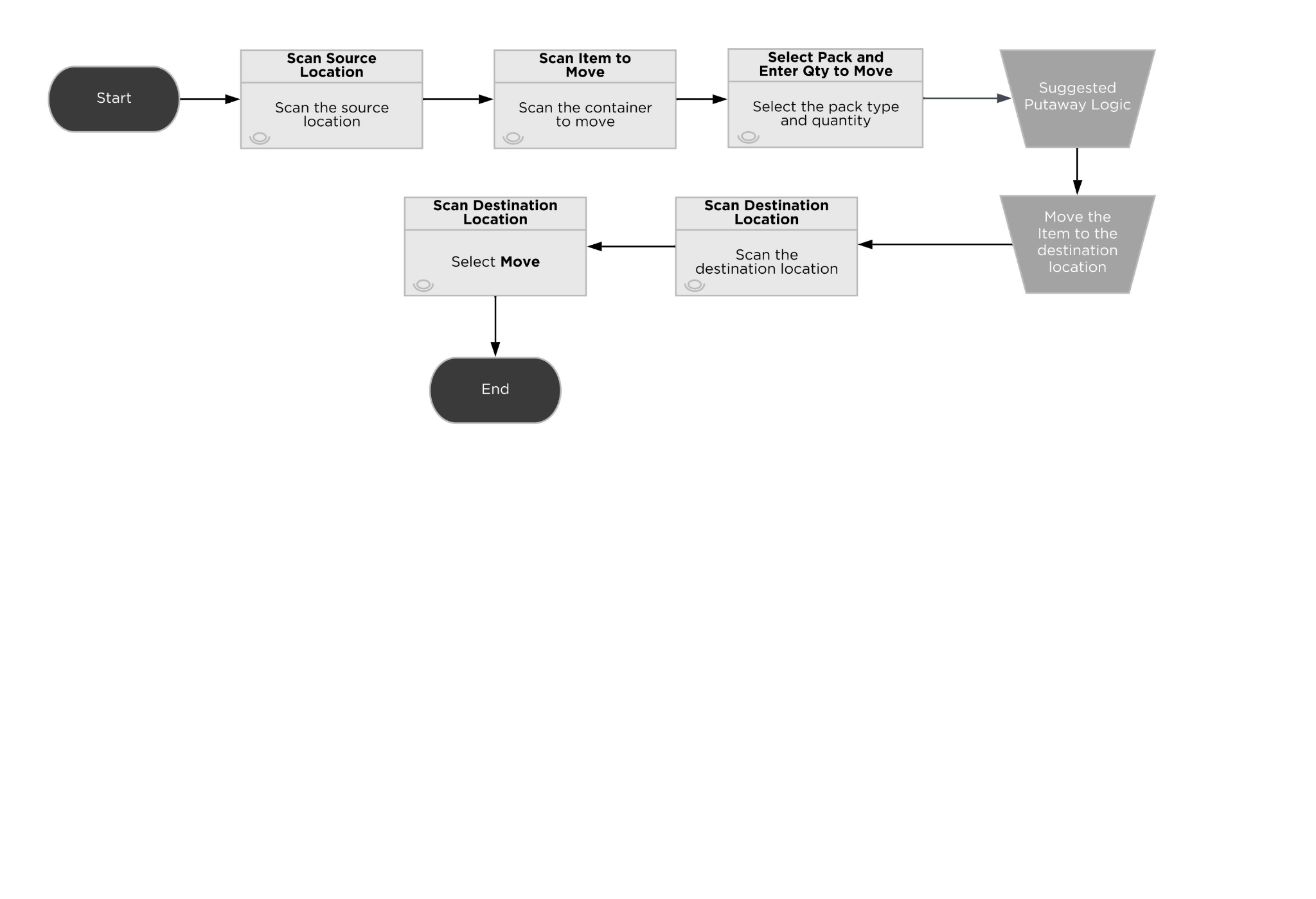
**

#### 4.5.2. Move Item

Allows users to scan a location, enter a pick quantity, and scan a destination location; used for simple non-LPN inventory movements.

Moves one item at a time to one location.

Process Flow:

**

#### 4.5.3. Move Stock

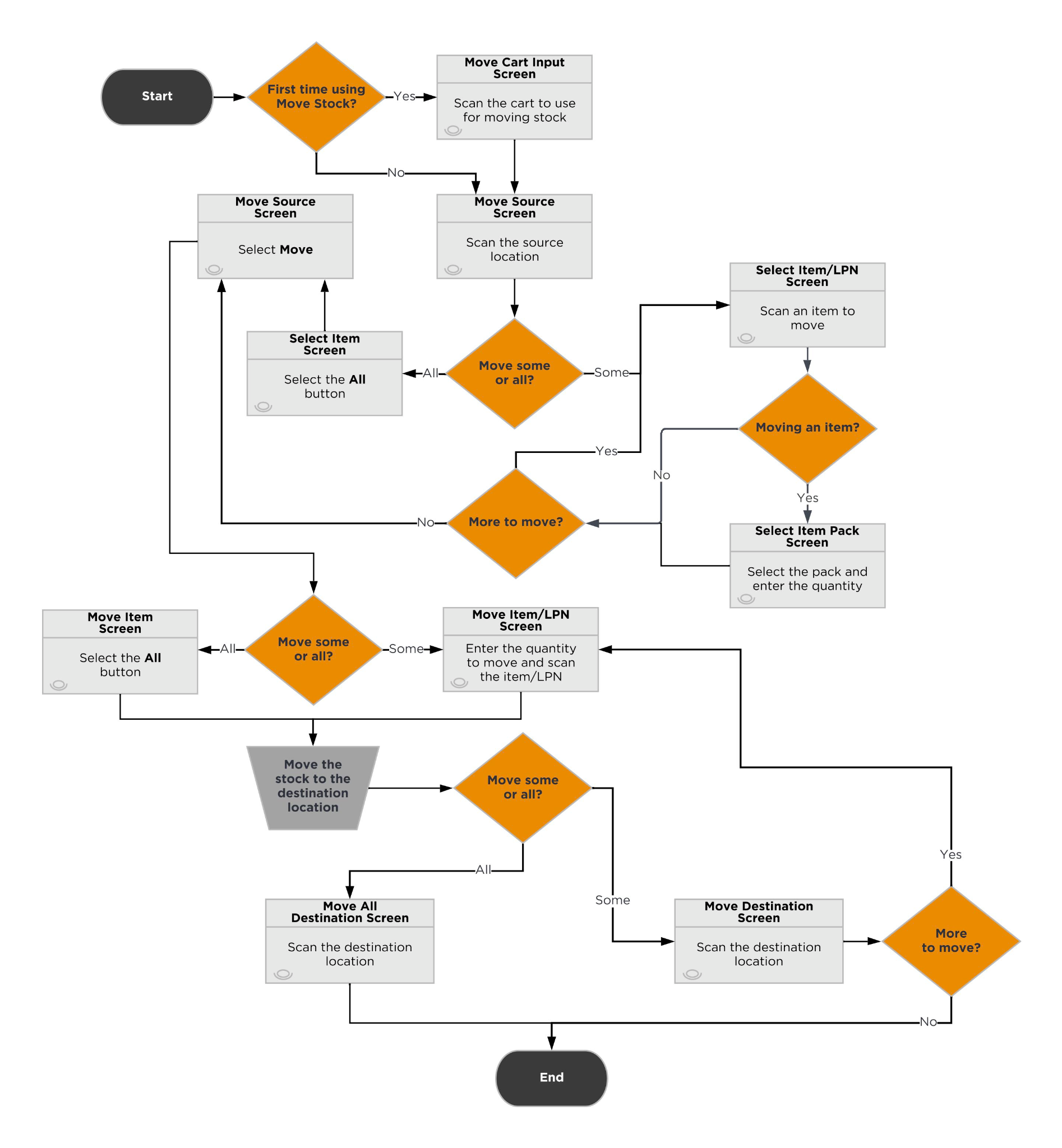
PAT uses the Move Stock process to conduct item movements. This process is used to move loose stock from the Receiving location as well as moving stock between permanent bin locations. The user is first prompted to enter a cart. This is the “Parent” cart (typically forklift or pallet jack) that holds the items needing to be moved.

The user’s cart is the actual equipment the operator is using to move, or simply a virtual cart that identifies they have the inventory. An advantage of scanning into the equipment is that it is handed off to a different user to continue/finish the move. Once the cart has been scanned on the Move Cart Input Screen, the user sees a cart summary on the Move Source Screen. This cart summary includes a count of items that are currently associated with the cart. As items are added/removed, these values update. Multiple items are built into the Cart and putaway individually, partially, or all at once.

Once finished loading Items onto the cart, the user then selects the Move button on the Move Source Screen. The next screen displays the current contents of the cart and asks the user to enter the quantity and Item they would like to move.

From the Move Source Screen, the user enters a Location or LPN. If a Location is scanned the user is prompted to scan an item from that location that they would like to move to the cart. Also, when scanning a location the user has the option to MoveAll contents to the cart.

Process Flow:

**

#### 4.5.4. Directed Putaway Logic

The PAT suggested location logic is:

1. Combine with existing stock
   * Existing stock location with available capacity
     + Finds a location with existing stock as well as space for the stock being put away
   * Open location closest to the first location with existing stock
     + Finds the closest location that does not have any stock closest to the first location with existing stock
     + Determines closest location based on putaway sequence

# 

### 

### User Stories

As a Putaway user, I want the ability to…

* Move inventory by pallet (container).
* Move inventory by individual units (loose stock).
* Have suggested putaway locations.
* Prevent On Hold inventory from being putaway

### Key Assumptions

* Because the receiving workflow requires an LPN, Directed Putaway is the primary putaway process.
* For accurate suggestions in Directed Putaway, single-item LPNs are recommended.
* A validation is in place to prevent mixing multiple SKUs in a single location, based on the location attribute “Mixed Item Threshold”.
* An LPN created during receiving has a single lot number.
* If an LPN is put away to a split case picking location, the container record is detached from the stock.
* Move Item is the secondary putaway process and can be used when moving loose stock.
* Move Stock is not to be used for putaway, but can be used to move inventory within the warehouse.

# 5. Inventory Processes

This section outlines all inventory management related processes. This includes inventory counts, adjustments, and replenishments/system directed movement.

## 5.1. Inventory Transactions

An inventory ledger is a document or computer record that tracks inventory balance transactions. The total of all transactions listed in this ledger should match the total balance for the corresponding account in the general ledger. An inventory ledger tracks changes in the balance of inventory, either quantity increases with a CR code or decreases with a DR code. Inventory ledger records are stored in the Inventory General Ledger (IGL) table. These records are commonly used to communicate inventory adjustments back to external ERP systems.

### Types of Transactions

| **Process** | **Action Type** | **Code** |
| --- | --- | --- |
| Manually Add stock from UI | MAD | CR |
| Kit to Stock Process | KIT | CR, DR |
| Receiving: PO, Blind Receipt | REC | CR |
| Ship out an Order using any packing process | SHP | DR |
| Customer Return: Receive a CR Order  VoidAndReprocessShipment: Shipping container and stock re-created | RET | CR |
| Adjust Inventory: Run this process to Create or Consume Inventory | ADJ | CR/DR |

## 5.2. Inventory Counts

Inventory counting is used to audit the inventory in the warehouse by going to a location and verifying the contents of the location. These can be system directed (Cycle Counts) or user directed (Physical Counts) depending on the need, and the processes allow for counting of the LPN, item/quantities, or both, depending on the configuration. Changes to inventory made during a count get recorded as Cycle Count or Physical Count adjustments and can be extracted or transmitted to other systems.

## 5.3. Physical Counts

In general warehousing terminology, a physical count refers to a count in which the entire facility or a zone is put on hold and counted all at once to ensure a complete picture of actual inventory. Despite its name, this process does not require a full facility count and can be performed on individual locations without having to put fulfillment on hold; this process is simply a count that is user directed (as opposed to the Cycle Counts).

#### Strategy

Regular use of Cycle Counting avoids the need for a full shutdown for a physical count and enables counts to be performed by a smaller team of operations. Full physical counts are expensive and require a larger group of users to perform counts to minimize the shutdown time, which tends to make many physical counts inaccurate.

If operations decides there is a need to perform a physical count, the following process is followed:

*Initiate Count*

* Supervisor initiates the count via the Physical Count Admin option on the handheld.

*Perform Count*

* Once initiated, associate users can begin performing the counting activities via the Physical Count handheld option.
* The system prompts users to count by location or by LPN within location based on the location’s LPN controlled configuration.
* Expected Qty is not displayed to the user.

#### End Count

* Supervisor ends the count via the Physical Count Admin option on the handheld.
* Physical counts are controlled counts. Handling any variances from the expected quantity is detailed in the Managing Cycle Count Variances section of this document.

## 5.4. Cycle Counts

Regular **Cycle Counts** ([link](https://docs.deposco.com/docs/html/Content/Warehouse_management/Inventory_management/Cycle_counts/Cycle_counts.htm?tocpath=Warehouse%20management%7CInventory%20management%7CCycle%20counts%7C_____0)) help to ensure that the inventory quantity in Deposco is as accurate as possible. During a cycle count, the quantity of inventory at a specific location is counted and then compared against the inventory quantity that is stored in Deposco.

If the quantity that is entered is different from the system quantity (a cycle count variance), then a supervisor can review the variance and either accept the quantity entered during the count, reject the quantity entered during the count and use the system quantity, or initiate a re-count.

The cycle count is system directed in that users are directed to perform counts based on pre-set tasks in a cycle count queue. Tasks are sequenced first by task priority, then by the pick sequence of the locations to count.

Please note that for a cycle count to be generated, the location and item must have Cycle Count = True (Shown Below). Automations are used to ensure all items have Cycle Count = true.



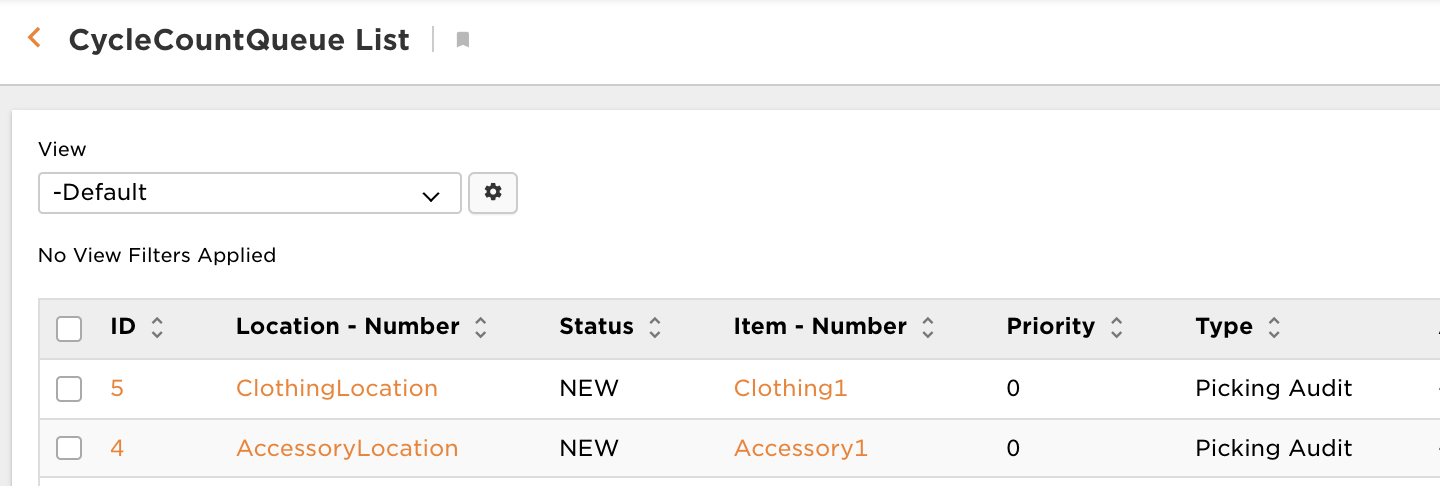
## 5.5. Generation Frequency and Methods

Cycle count tasks are generated for Patterson Warehouses in one of the following ways:

##### **5.5.1 System Generated Tasks**

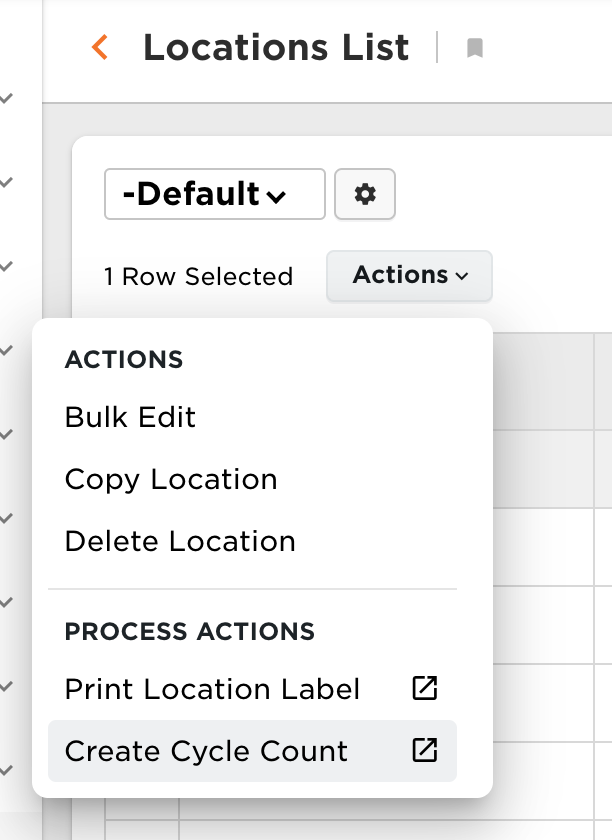
*Picking Exceptions*

* When a location is exceptioned during the picking or replenishment processes, the stockUnit is put on hold and a cycle count task is generated with priority = 0. This Cycle Count Queue task is created with Type = Picking Audit (Shown Below).

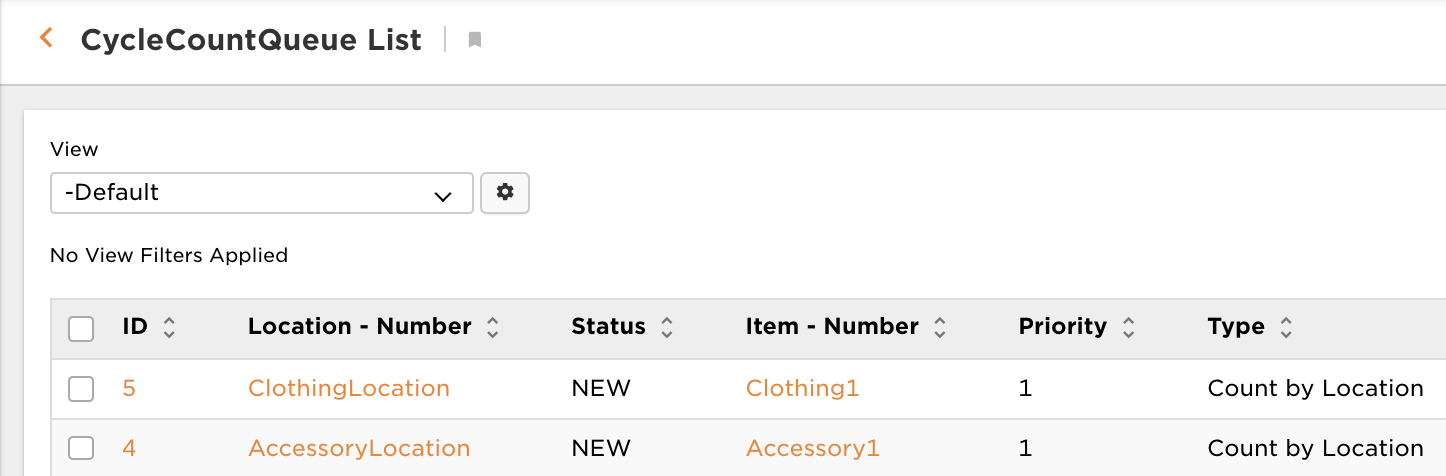


*Manual Creation*

* Location or Item (UI): At any time, locations are selected and cycle counts are created using Actions > Create Cycle Count (Shown Below).



* + These Cycle Count Queue tasks are created with Type = Count by Location (Shown Below).



* Location or Item (Handheld): At any time, a location can be scheduled for a cycle count using the Cycle Count by Location process. Users are prompted to scan or enter a location number and a priority to assign to the count. A task is added to the cycle count queue for the scanned location with the given priority. In addition, an item can be scheduled for a cycle count using the Cycle Count by Item process. A task is added to the cycle count queue for the scanned item with the given priority.
* On Demand Count (Handheld/Location): Users are able to create and execute a cycle count task for a particular location on an as needed basis using the “On Demand Cycle Count” button in the Handheld menu. Through the process, a location and priority are entered and the user is immediately afterwards prompted to complete the count of the scanned location.

*System Generated Tasks*

* Scheduler Task / Cycle Count Profile by Item
  + Cycle count tasks are generated for every location where the item exists and according to criteria in the cycle count profile. Cycle count = true must be enabled at the location and item level.
* Scheduler Task / Cycle Count Profile by Location
  + Cycle count tasks are generated for locations that meet specific criteria related to the cycle count profile.

### User Stories

As an Inventory user, I want the ability to…

* Cycle count an entire location.
* Cycle count an empty location.
* Cycle count a specific SKU in a location.
* Cycle count by LPNs in a location.
* Perform a blind count against a location.
* Create an on-demand cycle count.
* Create a cycle count using the Location or Item PAL in the UI.
* Perform a recount on a previously counted location.

**5.5.1.1 Key Assumptions**

* Counts can be item-based or location-based.
* All cycle counts are blind counts. Blind cycle counting shows all items/LPNs systematically at a location but does not show the quantities associated with them.
* Users are prompted to recount the location one additional time if they record a variance from the first count.
* LPN-based cycle counts cannot be performed by counting the number of LPNs. The scanning of each LPN is required.
* Patterson Warehouses does NOT capture item tracking attributes such as serial, lot and expiration dates during the cycle count process for phase 1.

##### **5.5.3 Prioritization**

When cycle counts are generated, a priority is assigned so that certain counts are completed first. When users perform Cycle Counting, they are directed in order of Cycle Count → Priority then by Location → Pick Sequence to provide an efficient travel path.

Cycle count tasks generated via a picking exception are higher priority than cycle counts generated manually so that they are completed first by the inventory team.

Prior to executing a full Physical Inventory, all other cycle count tasks must be completed.

##### **5.5.4 Managing Cycle Count Variances**

After cycle counts are completed, any variances are stored in the Manage Cycle Count Variance application. No variance record is recorded for a zero variance count.

### Strategy

From the cycle count variance screen, an Admin user must reconcile each variance.

*Accept Variance*

* Adjusts the inventory to the value of the variance of the cycle count.
* Any adjustments that are made in this process are written to the Inventory General Ledger with an Action Type of ADJ.

*Reject Variance*

* Leaves inventory levels as they currently stand.

*Recount Variance*

* Triggers the creation of a new cycle count task and the location remains in ‘On Hold’ status.

### User Stories

As an Inventory admin user, I want the ability to…

* Approve, reject, or initiate a recount on cycle count variance records.

### Key Assumptions

* If there is a difference between the inventory attributes counted at a location, two variance records are created – an increment for the inventory attribute counted and a decrement for the inventory attribute originally in the location.
* All cycle count variances must be approved, rejected or recounted.
* Reason codes cannot be captured in the UI by the user approving the variance.

## 5.6. Inventory Count Configurations

#### 5.6.1. Bulk Count

With Bulk Counts, the user is prompted to enter a quantity of each item scanned in a location during the count. If a variance is detected between what is entered and the inventory quantity in Deposco, the user is asked to confirm the counted quantity a second time.

#### 5.6.2. Inventory Attributes in Counting

Lot, serial and expiration date capture/confirmation **is not performed** during cycle counts for phase 1.

## 5.7. Inventory Adjustments

The AdjustInventory process is used to add or remove inventory from a location. This differs from a count in that it is incremental instead of a complete location verification and used primarily for adding found inventory to a location or removing damaged inventory.

### Strategy

The following adjustment processes are used:

* Consume by Item
* Consume by LPN
* Consume by Location
* Create Inventory

During the adjustment, the user is prompted for an adjustment code, which is stored against the transaction so that it can be extracted or transmitted to another system. The following adjustment codes have been identified:

| Adjustment Code | Description |
| --- | --- |
| DM | Damaged Goods |
| CC | Cycle Count Adjustment |
| RE | Receiving Error |
| PC | Pack Conversion |

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

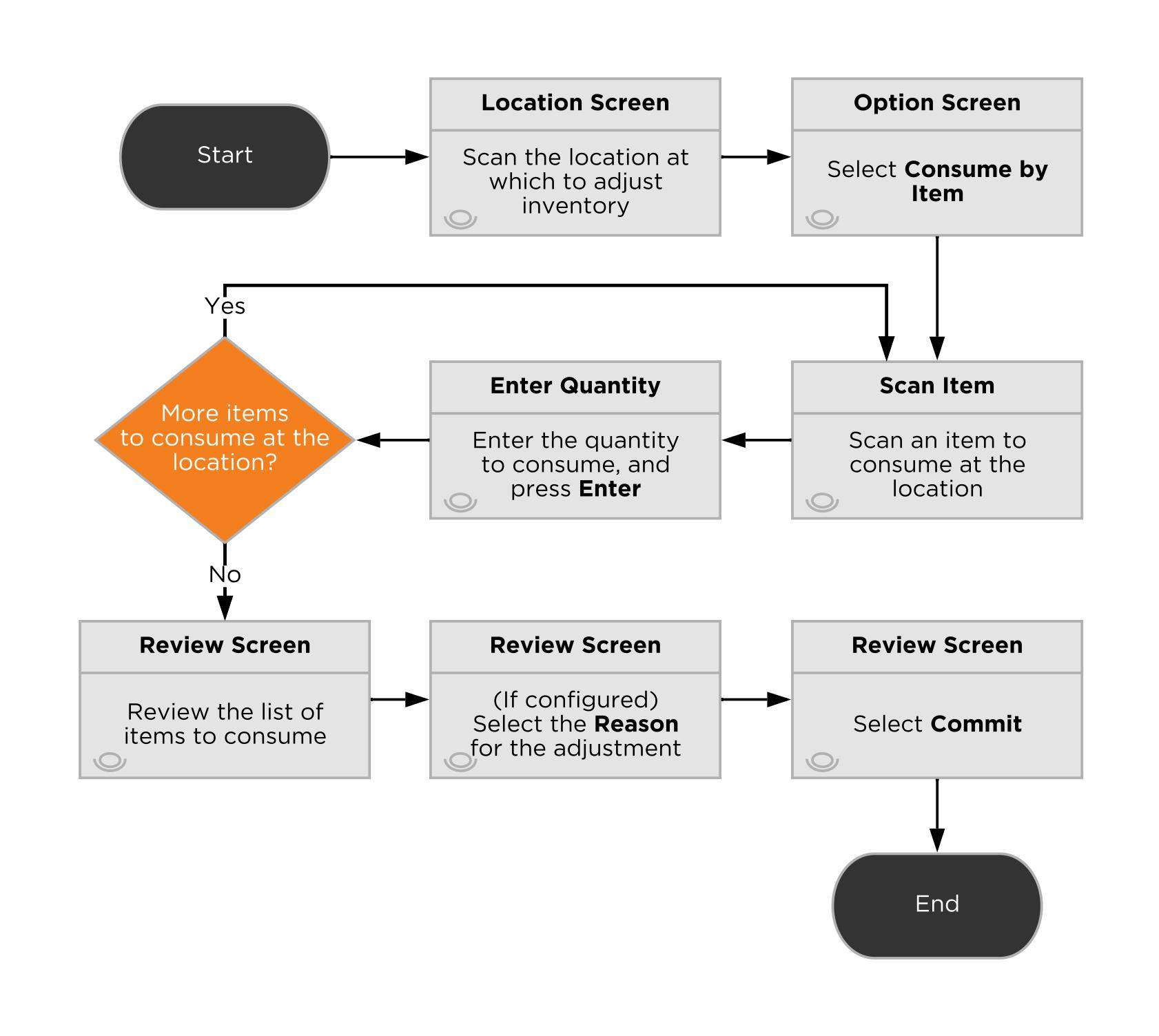
### 

### 

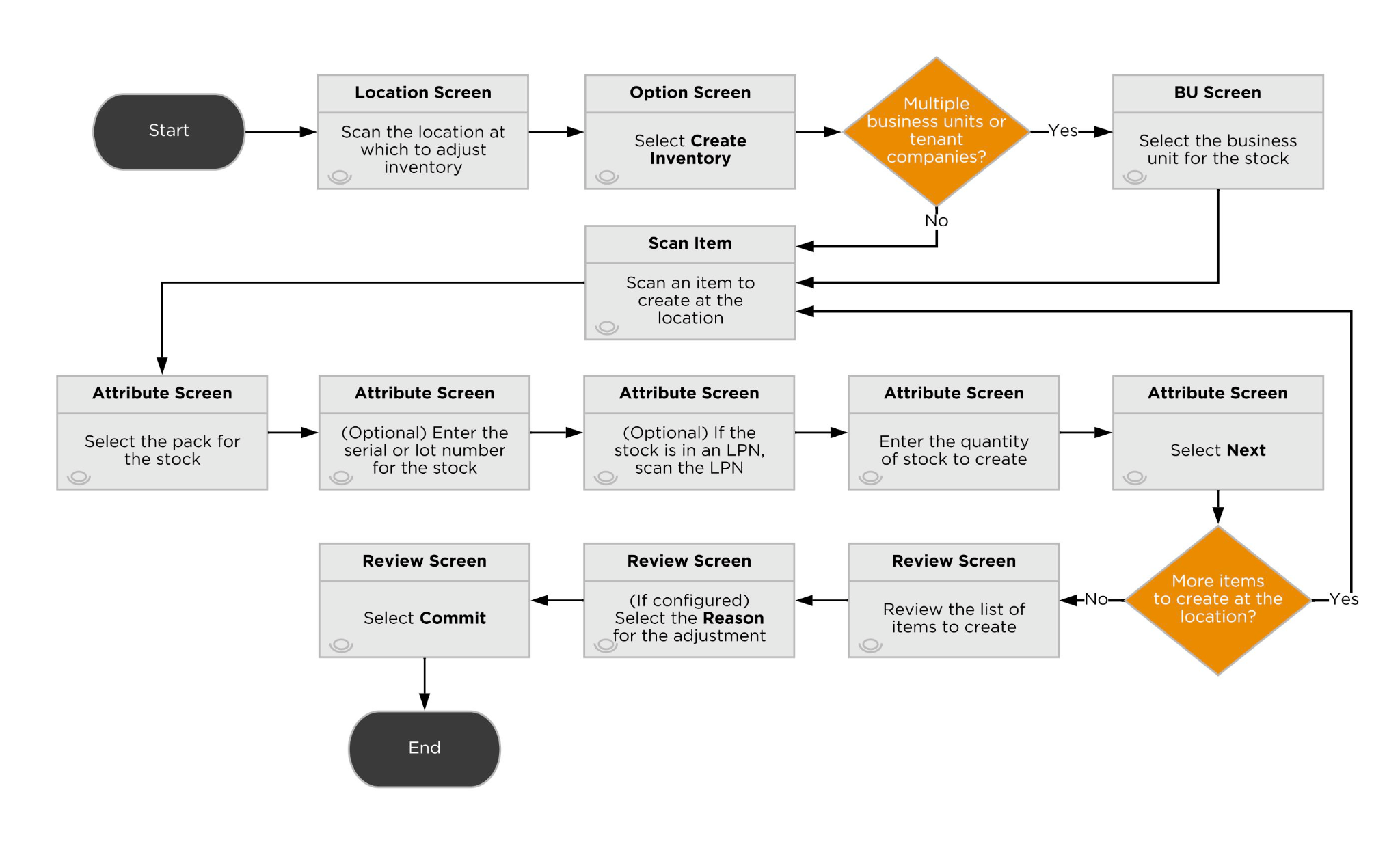
### 

### Process Flow

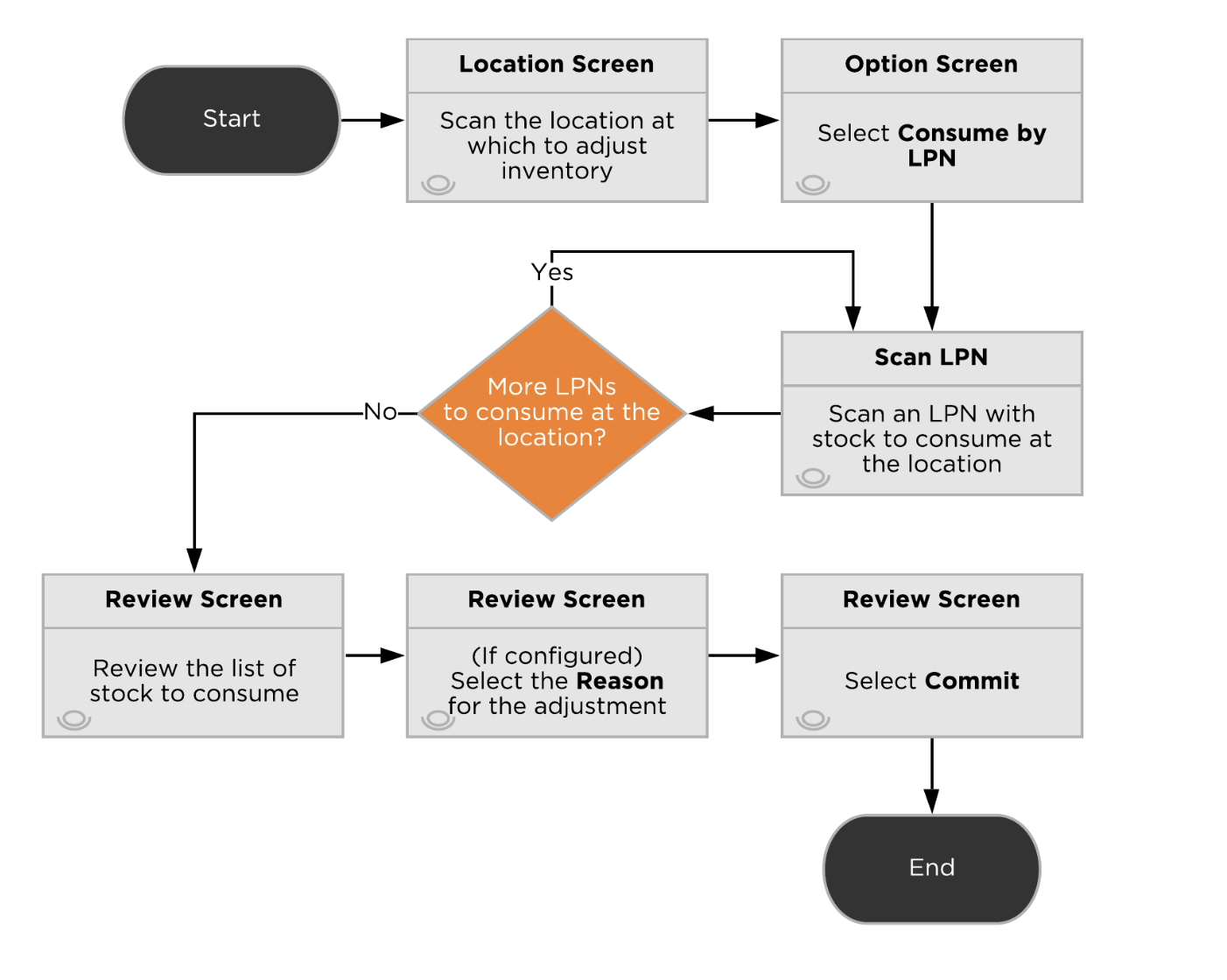
Adjust Inventory: *Consume by Item*

**

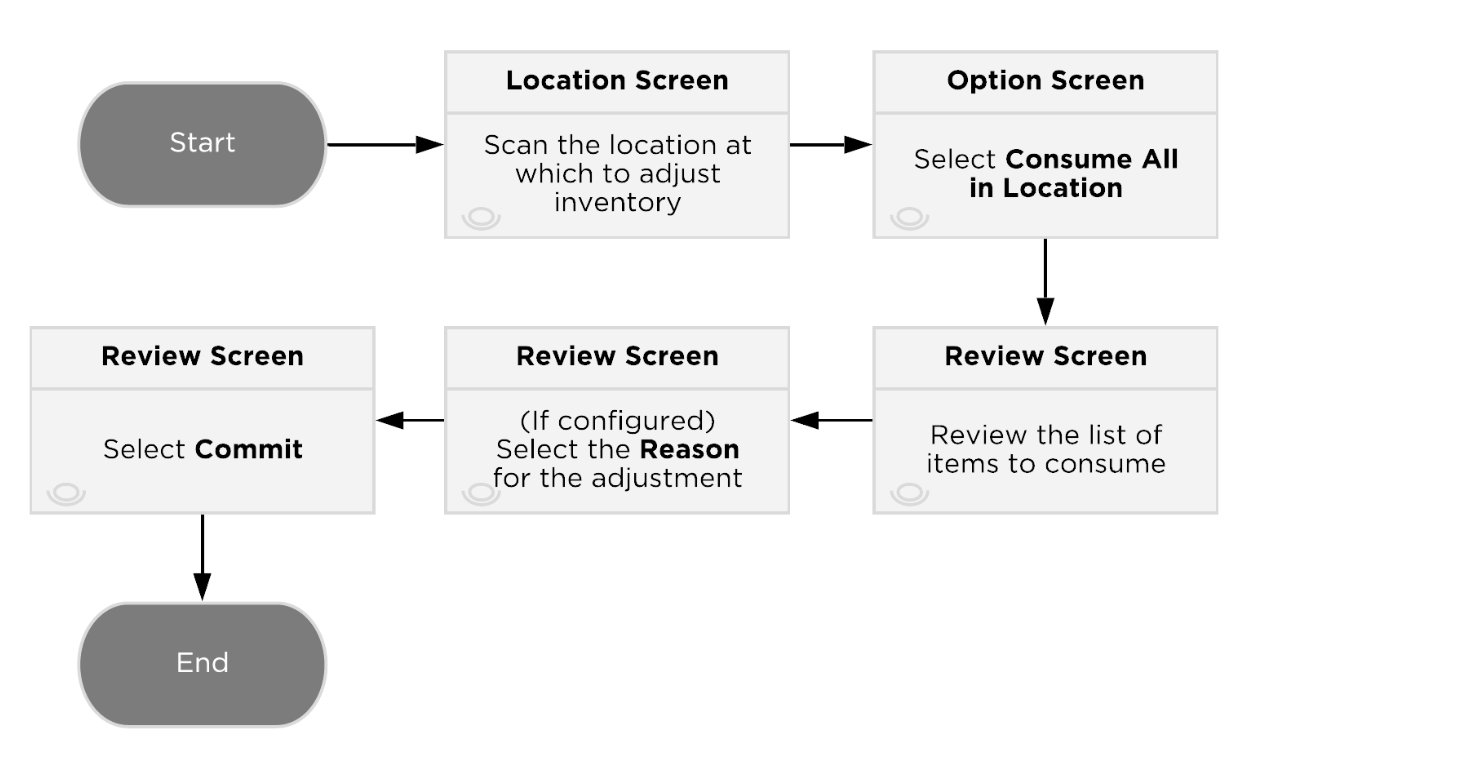
Adjust Inventory: *Create Inventory*



Adjust Inventory: *Consume by LPN*



Adjust Inventory: *Consume by Location*



### User Stories

As an Inventory Admin user, I want the ability to…

* Adjust out an entire LPN, Item, or Location.
* Adjust out/in specific stock units based on inventory attributes.
* Record adjustment codes for negative and positive adjustments.
* Select a reason code from a dropdown box for certain adjustment codes.

### Key Assumptions

* Inventory Adjustments are for Admin level users only
* Reason codes are global, across all Business Units. There are not any customer-specific adjustment codes.

#### 5.7.1. Patterson Warehouses Pack Conversion

Patterson Warehouses can convert a stock unit pack by adjusting out and adjusting in stock with the proper pack type.

## 5.8. Replenishments

**Replenishments** ([link](https://docs.deposco.com/docs/html/Content/Warehouse_management/Inventory_management/Replenishment/Replenishment.htm?tocpath=Warehouse%20management%7CInventory%20management%7CReplenishment%7C_____0)) are inventory movement tasks that are designed to move stock from reserve locations to pickable locations in order to maximize picking efficiency. They are generated periodically throughout the day on a scheduled basis.

Replenishment tasks are executed in two parts: pull and put. Users pull inventory from reserve locations in order by Replenishment → Priority then Replenishment → Pull Location → Pick Sequence. After the user builds their replenishment cart, they click ‘Cart is Full’ and are routed to complete the putaway of the Cart in order by Replenishment → Put Location → Pick Sequence. FIFO then partial pallet replenishment is used for replenishments.

All replenishment tasks created for Patterson Warehouses have a priority of 10. If a task needs to be prioritized above others, users may manually update the priority of a task to be lower than 10 so that users are prompted to complete the task before others.

### 5.9.1. Replenishment Strategy

#### Storage Based Replenishments

#### Strategy

In general, the warehouse replenishment process directs the movement of inventory from source locations to destination locations in the warehouse.

A storage record in Deposco assigns an item (specifically the pack record for an item) to a picking location and records the minimum (min) and maximum (max) inventory levels for the item in that picking location. When the quantity of stock falls below the min, a warehouse replenishment task can be created to move inventory from a reserve location to the location identified in the storage record. The task is created only if a replenishment task does not already exist for the item, and only for full pack quantities such that at least one pack is pulled and the max at the destination location is not exceeded.

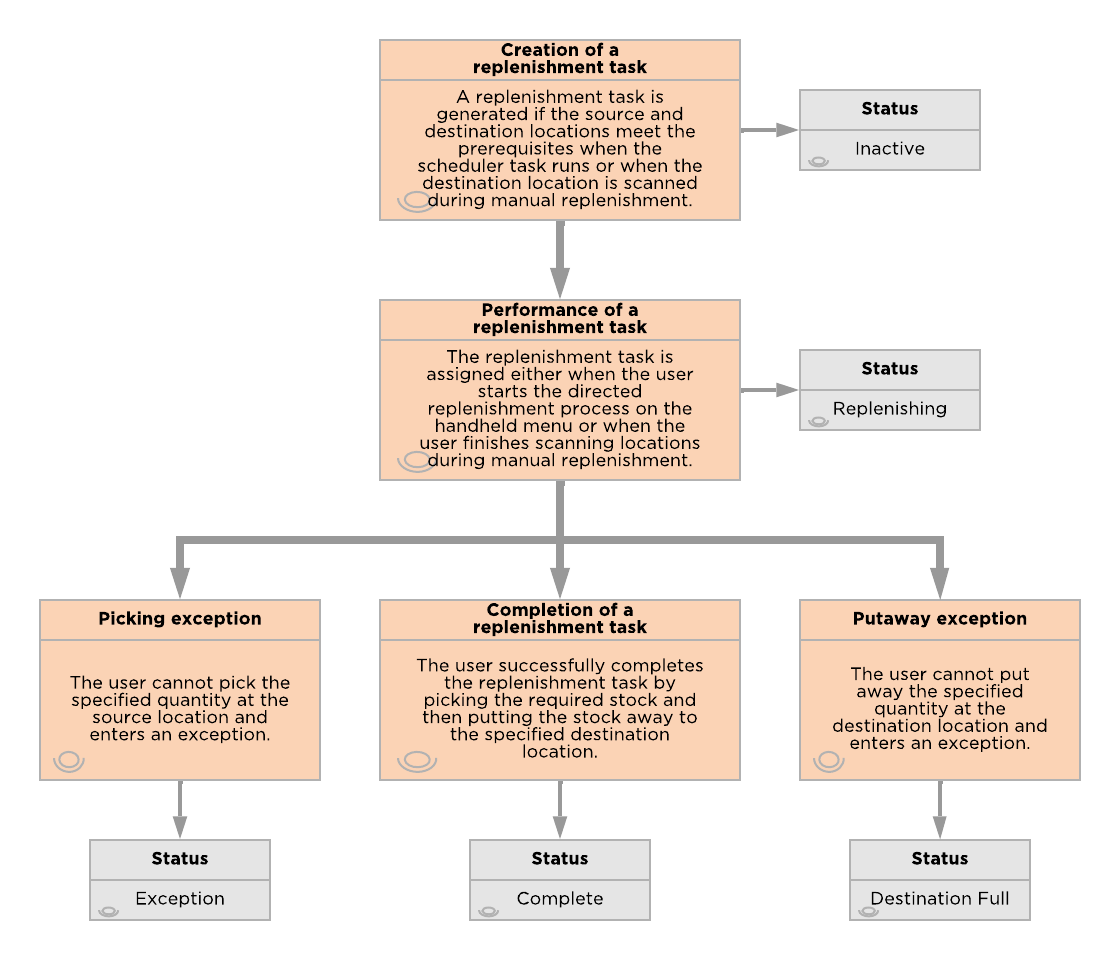
The Storage-based replenishment scheduler uses open order demand in conjunction with storage records to determine the optimal amount of inventory to be replenished to the forward-picking locations.

| **Type** | **Condition** | **Eligible Source Locations** | **Eligible Destination Locations** | **Source Location**  **Sort By** | **Destination Location**  **Sort By** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### 

#### 

#### Process Flow



### 5.9.2. Replenishment Configurations

#### 5.9.2.1. Over Replenishment

Patterson Warehouses does allow users to over-replenish when executing a replenishment task. Users may over-replenish by entering a quantity greater than the quantity prompted during the pull step. A common example is a replenishment task asks for 100 units, but 120 units are on a full pallet and you would rather not break a pallet.

#### 5.9.2.2. Replenishment Exceptions

#### Replenishment Picking Exception

If the requested number of items cannot be pulled from the reserve location during replenishment, then the user selects Exception on the Picking Screen.

**Screen Flow**

* After selecting Exception the user is prompted to select a reason code from the following list: No Stock, Stock Damaged, Stock Unreachable, Suspend Task.

**System Updates**

* The replenishment task is set to a status of Exception.
* If available stock is found in other reserve locations, then a new replenishment task is generated, and the user is prompted to pick the stock from the alternate location. (If other replenishment tasks have been assigned to the user, then the user may be directed to pick stock for those tasks before picking stock for the newly generated task.)
* The location is placed On Hold and a cycle count is generated.

#### Replenishment Putaway Exception

If the user cannot put away (replenish) the specified number of items to the destination location during replenishment, then select Exception on the Replenish Entry Screen.

**Screen Flow**

* After selecting Exception the user is prompted to select a reason code from the following list: Location Full.

**System Updates**

* If Location Full is selected as the exception reason, the replenishment task is set to a status of Destination Full.
* A new (reverse) replenishment task is created to return the stock to the source reserve location. The user is returned to the Replenish Entry Screen and is prompted to scan the source reserve location, the item, and the quantity that is being returned. (If other replenishment tasks have been assigned to the user, then the user may be directed to put away stock for those tasks before returning stock to the source reserve location for the newly generated task.)

#### 5.9.2.3. Inventory Attributes in Replenishments

Required inventory attributes are configured and maintained at the Item level by Patterson Warehouses. Serial, lot and expiration dates are **NOT** captured/confirmed during the replenishment process. If the stock being moved has serial, lot or expiration dates tied to it then they remain when moved to a pickable location.

#### 5.9.2.4. Workorder Based Replenishments

Work Order Replenishments are generated based on open work order quantity and are directed to the location defined on the Work Order.

The handheld process used to execute Work Order replenishments is housed underneath the Kitting menu option. Therefore, only users with access to the Kitting menu have the ability to execute Work Order replenishments. The Work Order Replenishments process does not filter on the user’s active zone assignment being ‘Kitting’. Instead, it automatically filters for replenishments generated by the Work Order Replenishment scheduler task in the same facility as the user’s active facility assignment.

#### User Stories

As a replenishment user, I want the ability to…

* Trigger replenishments based on a storage’s minimum quantity.
* Pull inventory from bulk locations into forward picking locations.
* Replenish multiple tasks at the same time.
* Maintain FEFO and FIFO while conducting replenishment tasks.

#### Key Assumptions

* Storage-based replenishments are only supported for static item locations and are driven based on the minimum and maximum quantities defined on the destination location’s storage record.
* System-driven replenishments are not generated if a static pickable location does not exist for the item.
* If static item location is changed, the corresponding storage record needs to be updated in parallel.
* The task is created to fill the storage location up to the maximum.
* Only locations in a ‘Ready’ status and stock units not in an ‘On Hold’ status are considered eligible for the pick/pull portion of the replenishment. Meaning, a location or stock unit with an open cycle count is not considered as its status is ‘On Hold’.
* When determining the source location, Deposco prioritizes stock in the following sequence: Oldest Expiration Date, Oldest Stock Unit Created Date, Location Pick Sequence descending.
* The priority for a replenishment task is assigned automatically starting with a value of 10, and incrementing by 10 for each eligible source location for the item.
* The putaway sequence for a batch of replenishment tasks is sorted by the location’s putaway sequence, descending.
* 10 replenishment tasks are assigned to the user when the user starts the replenishment process on the handheld menu.
* When a replenishment is driven from an LPN-Controlled location, the user is prompted to scan the Source LPN.

## 5.10. Kitting

Items in Deposco can be configured to support the bundling of one or more child items into a parent item for fulfillment. Setup for a Kit occurs at the Item Entity Level.

### Kitting Strategies

#### 5.10.1. Kit to Stock

When the warehouse is prompted to kit-to-stock, the assembled kit is stored as a completed unit. This is typically done in anticipation of demand for the assembled kit, therefore making order processing much faster.

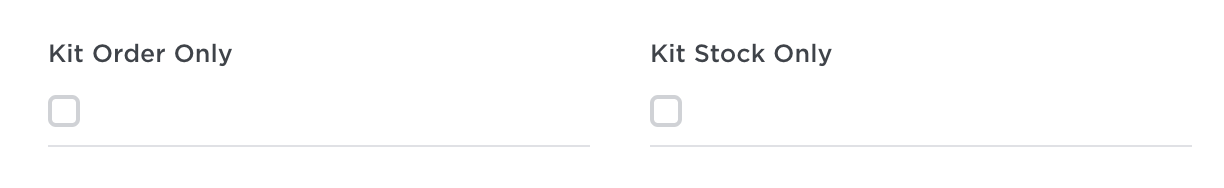
The build quantity is communicated with the Operations team using a Work Order. Work Orders are created via the WorkOrder API or Data Exchange. A Work Order uses Kit Header and Kit Detail data to determine which component items need to be included in the final assembly. Kit Headers and Kit Details can be added via Data Exchange or manually created in the UI.

The Kit To Stock process requires all sub component items in a location to be assembled. The operations team moves sub component items to the Kitting Workcell location. Replenishment tasks are generated for the component SKUs using the Work Order Replenishment scheduler. Work Order Replenishments are executed by the Kitting team. Therefore, the corresponding handheld button resides within the Kitting menu.

The Kit to Stock process performs the assembly of the kit item. The Kitting Workcell location is scanned as well as the kit/parent item. Next, the user enters the quantity of the kit item to be assembled. The Kitting Summary screen displays the sub component items to be consumed. Next, the kit is assembled.

#### 5.10.2. Kit to Order

For bundled products that are composed of multiple individual items that are not pre-bundled prior to outbound processing and instead are picked as individual components through picking processes. Example here is we have the components for a bicycle where the front tire is in pickloc1, frame is in pickloc2 and backtire is in pickloc5.



### Kitting Master Data

#### 5.10.3. Item Components / Assemblies

* This is referenced at the item level and is used in kit to order scenarios. You must define the assembly/components that should be picked to make up the parent sku.

#### 5.10.4. Kit Headers / Kit Details

* This is used in conjunction with product conversion with workorder. Kit header and kit details is how you define the kit to be built for the team.

#### 5.10.5. Workorders

* A workorder is how the team executes a task/builds the kits to then put them into stock. Every workorder has a kitheader associated with it so the team know what to build and the workorder tells the team where to build and how much quantity to create of the output.

### Kitting Processes

##### **5.10.6. Product Conversion With Workorder**

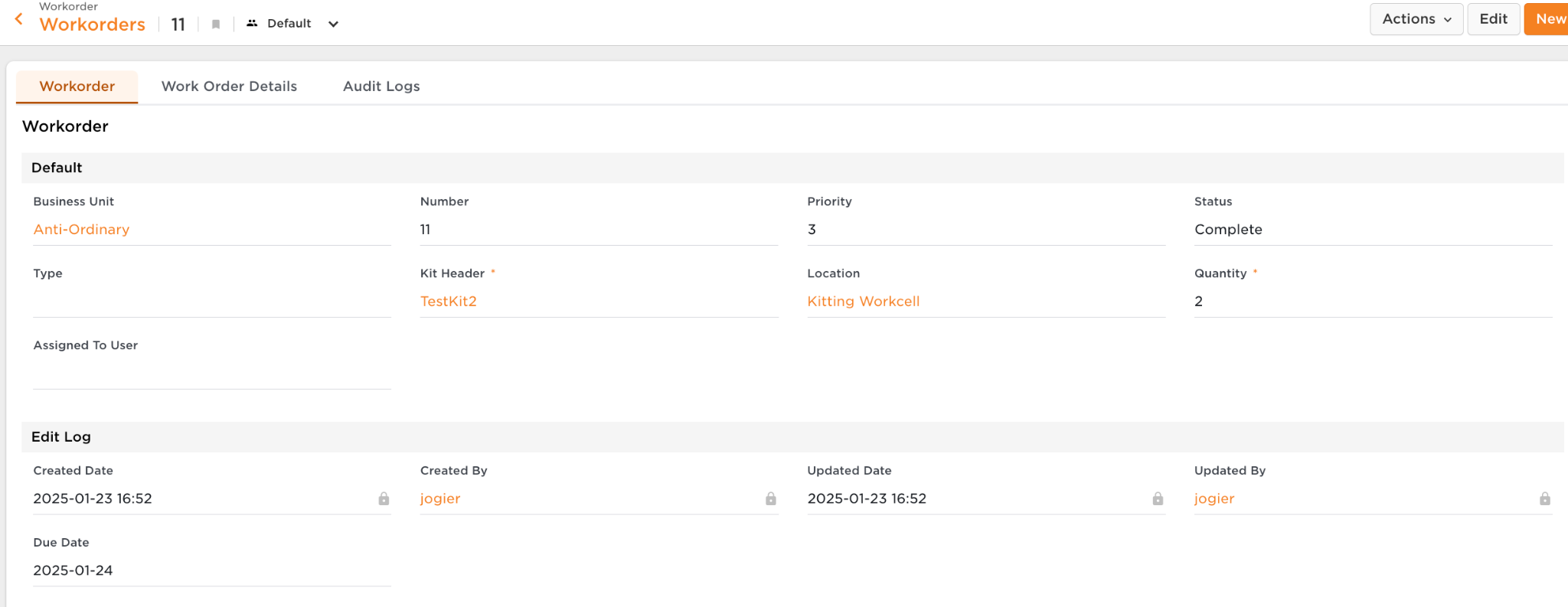
Patterson Warehouses utilizes the **ProductConversionWithWorkorder** process. This process is used to convert input items into their respective output kit items. Kits and Workorders are a prerequisite to the **ProductConversionWithWorkorder** process.

Kits define the input and output quantity of the product conversion process. Kits are created via Data Exchange upload or via the UI.

##### 

Work Orders are created for users to execute product conversion against. Work Orders are created via Data Exchange upload or via the UI. The following information is required when creating a Workorder:

* Number - Work Order Number
* Kit Header - Desired output item to be built
* Quantity - Quantity of the output item to build
* Location - Kitting location where output item is built



After a Workorder has been created, **ProductConversionWithWorkorder** can begin. Users are displayed a list of Kit Details then asked to enter the quantity of the output item built. A user may not build more than the quantity defined on the Work Order. After the stock has been built into the kitting location, users utilize the **Move Item** process to put stock away.

If not enough child item stock is not available in the kitting location an error is thrown. Adequate stock should then be moved into the kitting location to complete the work order.

### Kitting Configurations

### *5.10.7. Inventory Attributes in Kitting*

Patterson Warehouses does NOT capture serial, lot or expiration date attributes when building work orders for phase 1.

## 5.11. Inventory Calculations

Deposco calculates different types of inventory classifications. Starting with On Hand Inventory, Deposco always displays the total on hand inventory for items. Other calculations like Available to Promise and Available to Release are done in Deposco; adjustments to these calculation queries can be done and require specific approval. These inventory calculations are typically viewed on the Item UI.

| **Field** | **Description** | **Calculation** |
| --- | --- | --- |
| *Total On Hand Qty API* | * The Total On Hand calculates all inventory that is physically in the Patterson warehouses. | Sum of stock unit quantity times pack quantity across all locations |
| *On Hold Qty* | * The On Hold Qty includes any inventory with a stock unit status of ‘On Hold’ or resides in the ‘Damages’ zone. | Sum of stock unit quantity times pack quantity for any stock unit with status = ‘On Hold’ or zone = ‘Damages’ |
| *Total ATP Qty API* | * Total Available to Promise (ATP) * This value is passed to any integrated system. * This value is also what is used when releasing orders to the WMS from the OMS. | ATP = On-Hand - Allocated ( Open OrderLine Qty) - Damaged (includes all locations in Damages zone) - Receiving - On Hold Qty |
| *Total ATR Qty API* | * Available to Release (ATR) is the inventory that is available to release and fulfill Sales Orders. * This value gives insight into what items may need to be replenished to pickable locations. | ATR = On-Hand - Allocated - Unpickable - Damaged |
| *Whse On Back Order API* | * Total quantity on Sales Orders which are backordered due to lack of availability (non-pickable) or another reason. | orderline quantity where line status is ‘Back Ordered’ |
| *Reserve Loc Qty API* | * This is the total inventory of an item residing in Reserve Locations. | Sum of stock unit quantity times pack quantity where reserved=true or storage type = ‘reserve storage’ |
| *Pickable Loc Qty API* | * This is the total inventory of an item residing in Pickable Locations. | Sum of stock unit quantity times pack quantity where pickable=true and stock unit is not ‘on hold’ |
| *Open Pick Task Qty API* | * Total quantity of an item that is allocated to open/active pick tasks and therefore unavailable to allocate to Sales Orders. | sum of pick task quantity times pack quantity where pick task status not equal to ‘Complete’ or ‘Exception’ |
| *Open Order Line Qty API* | * Total quantity of an item that is required to fulfill New/Back Ordered order lines and does not include already fulfilled/canceled order line quantities. | orderline quantity where line status is ‘New’ or ‘Back Ordered’ |
| *Qty On Purchase Order API* | * Total quantity of an item that is on Purchase Orders and yet to be received and does not include already received/canceled order line quantities. | orderline quantity where line status not equal to Closed, ‘Received’, or Canceled’ |
| *Open Replen Qty* | * Total quantity of an item that is allocated to open/active replen tasks. | replen task quantity where status is “Inactive” or “In Progress” |

# 6. Outbound Processes

This section outlines all outbound related processes. This includes Customer Order and Sales Order management, wave planning, allocation, picking, packing and shipping.

**Bright Order**

## 6.1. Customer Order Management

### 6.1.1. Strategy

Order management features provide visibility and life cycle management capabilities for orders from customers to purchase goods from the time they are created through the picking, packing, and shipping process. Sales orders are used for fulfillment processing through waving, picking, packing, and shipping. As a result, sales orders are also referred to as *fulfillment orders*.

It is assumed that all customer orders are imported in Eaches. From an integration standpoint, this means each CoLine in the API request payload has a packQuantity of 1. For example, if a customer orders 10 eaches, the following values are expected in the Customer Order API request:

"orderQuantity": 10,

"packQuantity": 1,



It is assumed that the Trading Partner is provided on the Customer Order upon import.

* If the Trading Partner is equal to the Business Unit Code, the ShipTo address must be provided on the Customer Order.
* If the Trading Partner is not equal to the Business Unit the ShipTo address can be provided on the Customer Order. If the ShipTo address is not provided on the Customer Order, the ShipTo address is pulled from the Trading Partner.

The Shipping Service (ShipVia) is not evaluated as a part of CoPreprocessing logic. Instead, it is determined during Post Order Import in the WMS.

The Freight Terms Type is not evaluated as a part of CoPreprocessing logic. Instead, it is determined during Post Order Import in the WMS.

The following validations occur during pre-processing of Customer Orders:

* External order number must be unique within a single Business Unit.
  + Void and Recreate is utilized for HKC
  + Add reporting for voided orders for visibility
* A customer order must have at least one customer order line with an item pack specified on the customer order line. This validation only occurs when the packQuantity specified in the Customer Order API request does not match an existing pack quantity for the given CoLine item.
* The Ship To contact name and Ship To address must be populated for the order. The validation is dependent on the Trading Partner code provided on the order. This logic is detailed above.
* If Ship To Country is not ‘US’ then the following fields must be populated on all CoLines:
  + HTS code: CoLine.item.harmonizedCode (this validation is evaluating the Item master data)
  + Customs value: coLine.unitPrice
  + When values are blank on the CoLine then set these values based on item master data. If data is still unavailable then order is set to “Review”
* Users are able to view the customer orders in review on their dashboard for visibility and timely resolutions.

These validations are to ensure that all necessary information is included for each order. If any of the validations fail, then the order moves to the Review status, and an order management user or client user must manually review the exceptions to determine if the order should be either fixed and reprocessed or canceled. Reviewing Customer Orders is designated using the Order Management group permissions. CoPreprocessing validations record a Notice with a reason for why the customer order is in Review status.

**6.1.2. Decision Service**

Decision Service can be used to translate ship vias from an external system into a normalized Deposco ship via. Translations that occur within Decision Service are logged within the Deposco UI. Therefore, it is recommended that orders are imported into the Deposco OMS with the raw ship via sent by the customer. Once the order is sent to the WMS the shipVia is translated to a Deposco ship via. This design allows the original shipVia to be referenced on the Customer Order.

**6.1.3 CoManagement**

An order management profile (CoManagementProfile) is assigned to each Customer Order (CO).The order management profile contains parameters that direct the order through the necessary order orchestration steps, such as the Fulfillment Mode, Acknowledgement Type, and order revision settings.

* Fulfillment Mode: Release by Line, Single-Order
* Acknowledgement Type = No Acknowledgement
* Co Revision Mode = Void and create

### 6.1.3.1. Key Assumptions

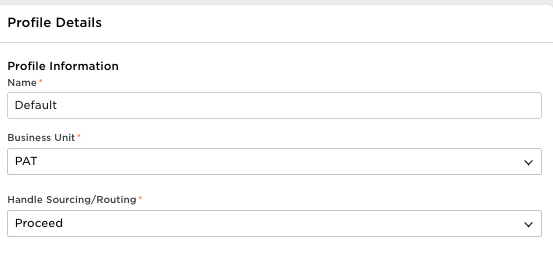
* PAT is using Co Revision Mode = Void and Create, which allows updates to the existing customer order and fulfillment order without voiding the original.

A table within the [Cancellations and Updates](#_okbnrdsxhe3e) section goes into more detail about the different Co Revision Modes and their expected behavior.

A CO Activity (CoActivity) record is a key activity or important milestone during order processing that is tracked so that you can view the steps that were taken for the CO and when they were taken. These activity records are helpful when inquiring on the status of a CO and troubleshooting an order exception scenario.

**6.1.4. Sourcing**

One default sourcing profile is configured, which considers all facilities that are in scope for the initial implementation.



**6.1.5. Routing**

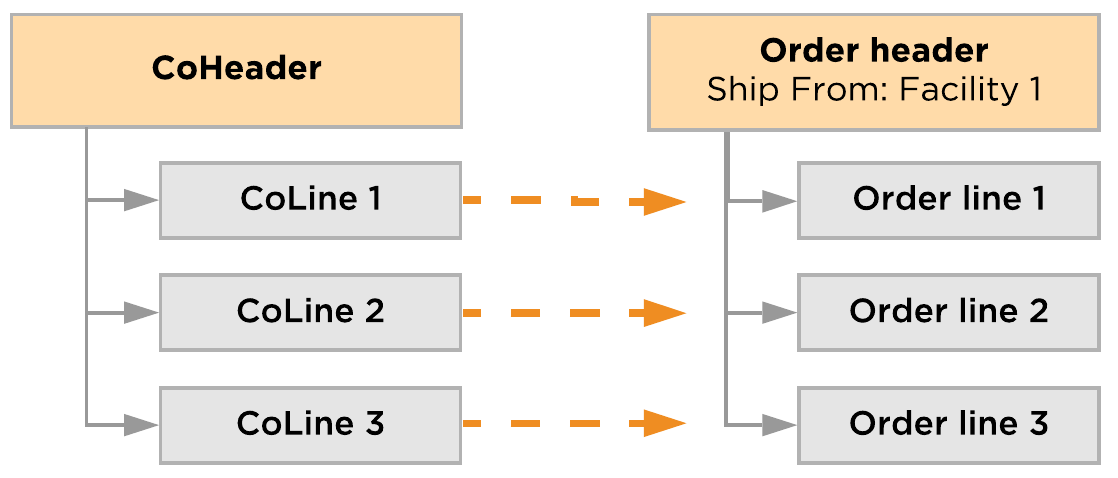
Customer Orders are created via edi/api and Deposco routes orders based on inventory availability in each warehouse. Each customer is fulfilled out of a single warehouse so customer orders are routed to a single facility only. If the customer order cannot be fulfilled from one facility then it is backordered in the oms. If the customer order has items without stock in the facility then it is also backordered and is either forced allocated to a facility or reprocessed once stock has been made available.

### 6.1.5.1. Process Flow

*Customer Order Life Cycle*

### 

*Allocation Strategy*

**

### 6.1.6. Customer Order Statuses

Customer Orders are updated throughout the duration of order fulfillment. The table below provides the possible statuses that can be applied to a customer order.

| **In Use** | **Status** | **Description** |
| --- | --- | --- |
| ✓ | Draft | A draft or quote order that was created but has not yet been submitted. Draft orders are not processed through fulfillment until they are submitted and moved to a status of New. |
| ✓ | New | The order is either a new order that was received from outside sales channels or that was manually created and submitted. |
| **✕** | Accepted | In B2B EDI scenarios, the order was reviewed and accepted, and can now move to a status of Open. |
| ✓ | Review | An error or other issue that requires manual intervention occurred during order pre-processing. A user must review the order, and then either fix and reprocess the order or cancel the order. |
| **✕** | Rejected | In B2B EDI scenarios, the order was reviewed and rejected. |
| **✕** | Future | An order is placed by a trading partner, but with the expectation that the order is to be released for fulfillment sometime in the future, based on an agreed-upon planned release date. The order remains in Future status until its planned release date. |
| ✓ | Open | The order is ready for fulfillment, either through sourcing and routing processing or by pass-through directly to release processing. Some orders, such as Contract orders, remain in an Open status while their related release orders are processed. |
| ✓ | Sourced | The order is going through the sourcing process to determine all possible fulfillment sources. |
| ✓ | Allocated | The order was processed through routing, and specific fulfillment sources were found to fulfill the order, which resulted in soft allocation of inventory. |
| **✕** | Back Ordered | The order was processed through routing, and one or more order lines were back-ordered because they could not be successfully routed and allocated after being evaluated against all of the routing rules in the routing profile that was assigned to the order. The Unallocated Order Line Policy setting for the routing profile must be set to Backorder order line for an order line to be back-ordered. |
| ✓ | Released | One or more sales (fulfillment) orders were created for the customer order and released to fulfillment locations. |
| ✓ | In Fulfillment | The sales (fulfillment) orders for the customer order have started processing in the fulfillment locations. For example, pick tasks to pick the stock for the sales order were created in Deposco when a pick wave that includes the order was released. |
| ✓ | Canceled | A process or a user canceled the order, and the order will not be processed for fulfillment. |
| ✓ | Complete | Shipping of the order is complete, and the customer order is now complete. |
| **✕** | Archived | The order has reached the archive data retention period and has been moved to long-term storage. |

### 6.1.7. Exception Management

Bright Order directs COs through automated steps to fulfill COs with the least amount of manual processing. However, validations and actions that are performed during the CO life cycle might direct an order to Review status so that it can be evaluated.

When an order is directed to Review status during order processing, where a pre-processing validation could have failed, a user can review the order data to determine if the order should be either fixed and reprocessed or canceled.

An order management team or client team member can monitor the system for order exceptions that need attention. For each order in review status, CoNotice records provide information about the validation issues that set the order to be reviewed.

After the user clears up any issues with the order data, click the Reprocess Customer Order process action link (from the CoHeader Entity) to have the order pass through the same previous steps. Reprocessing COs in review can be automated as well via the OMS Retry Scheduler.



If the order is in a bad state and should not be processed, the cancel CO process action can be used. All process actions are driven by group permissions to ensure only trained users can cancel and reprocess COs.

#### 6.1.8. Cancellations and Updates

Order Management users have the ability to reprocess or cancel COs directly from the Deposco User Interface. After Fulfillment Orders are created, users must cancel from the Fulfillment Order as opposed to the CO. Fulfillment Orders are created automatically once the CO is created because routing is skipped.

| **Co Revision Type** | **Description** |
| --- | --- |
| VoidAndCreate | This mode voids existing CoHeaders and creates a new CoHeader.  It throws a validation error if the CoHeader is not in New, Accepted, Open, Released. |
| ConditionalUpdate | This mode checks Entity/Attribute configurations to update/reject the change.   * If the attribute is ReadOnly then the update fails with the comment 'Failed: This field is read-only'. * If the attribute is inactive then the update fails with the comment 'Failed: This field is in-active' |
| Passthrough | With the below conditions met, Passthrough revision mode allows to update the existing customer order and Depending on the following conditions, this passthrough revision mode |
| Advanced  (Default) | The following changes are made for the order if supported:   * Fields are updated. * New order lines are added. * Quantities for order lines are increased or decreased. * Order lines are canceled. This is a partial cancellation. * The entire order is canceled. This is a full cancellation. * When a customer order is either partially or completely canceled, then fulfillment orders that are associated with the customer order are also canceled.   A CoRevision and associated CoRevisionDetail entities are created to track the changes for updates. |

Cancellations and updates received from a Socket or API interface can be controlled using the CoRevisionMode on the CoManagementProfile. When set to ‘Advanced’, the following changes are made for the order:

* A corevision record is attached to the customer order with the expected updates.
* Customer order shows reflected updates and records the activity in the coactivity table, also attached to the customer order.

When a customer order is either partially or completely canceled, then fulfillment orders that are associated with the customer order are also canceled if the fulfillment orders are in an eligible status of “New” or “Hold”. Otherwise, the cancellations do not persist to the fulfillment order. Updates do not persist to the fulfillment order and are, therefore, only allowed while the customer order is in the OMS.

A CoRevision and associated CoRevisionDetail entities are created to track the changes for updates. The source data file is stored on a table associated with this entity and is available for reference.

### 6.1.8. User Stories

*As an Order Management user, I want the ability to…*

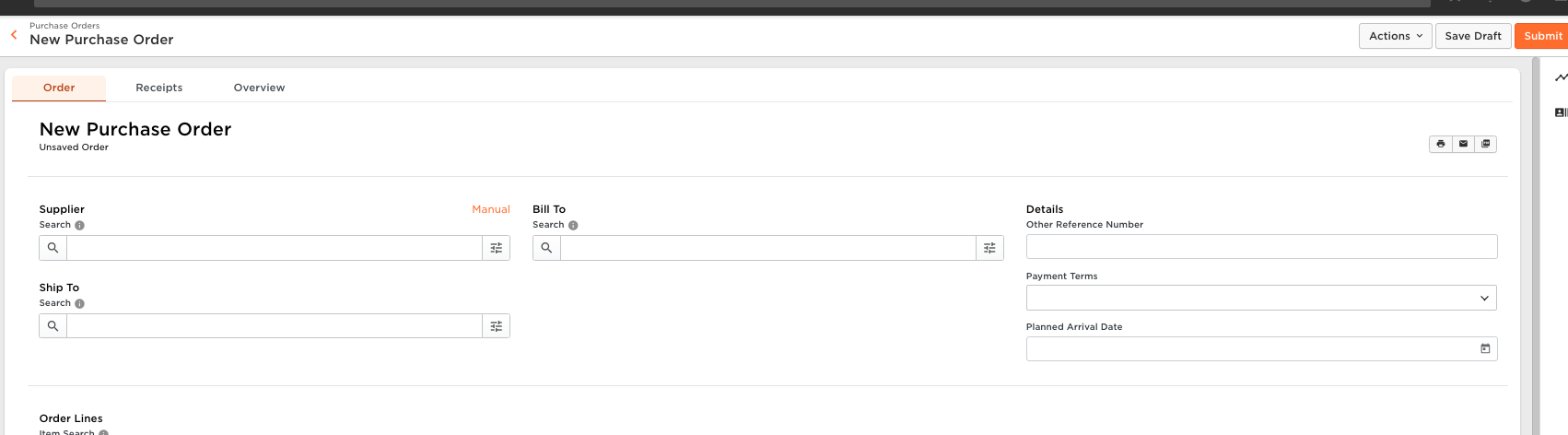
* Update header-level information on an existing Customer Order.
* Update line-level information on an existing Customer Order.
* Fully cancel an existing Customer Order.
* Cancel an existing Customer Order Line.
* Manage orders with invalid data in the OMS for customer review (e.g. missing items, invalid address information).
* Have a dashboard widget showing customer orders in review for the PAT team to review and release the customer order for fulfillment

### 6.1.8.1 Key Assumptions

* Customer orders (COs) are created via the Customer Order API.
* The external order number must be unique within a single Business Unit.
* A customer order must have at least one customer order line with an item pack specified on the customer order line.
* Trading partner/consignee must exist on the customer order
* The Shipping Service (ShipVia) is sent on the customer order and is used to fulfill the associated sales/fulfillment order
* The Freight Terms Type is not evaluated as a part of CoPreprocessing logic. Instead, it is determined during Post Order Import in the WMS.
* If Ship To Country is not ‘US’ then the following fields must be populated on all CoLines:
  + HTS code: item.harmonizedCode
  + Customs value: coLine.unitPrice
  + When values are blank on the CoLine then set these values based on item master data. If data is still unavailable then order is set to “Review”
* Incoterms - For orders, if the Duty Paid By is not set at the Customer Order level, then the value specified for the shipping service in the associated Ship Via field is used.
* There is a Single, default CoManagement Profile.
* One default sourcing profile is configured, which considers all facilities in scope for the initial implementation.
* Customer orders are routed based on available inventory and backorder management occurs at the oms level.
* Order updates are allowed if the CO is created and in Review status. Customer order updates come onto the order using a CoRevisionMode of ‘VoidandCreate’.
* Order updates are not allowed once a warehouse activity has started for a sales order/fulfillment order such as build and release wave.
* Cancellations are handled in OMS which would cancel the customer order and associated sales order/fulfillment order.
* COs in Review status are reprocessed in the OMS every 30 minutes via a scheduler task labeled, OMS Retry Scheduler. They are sorted by the customer orders created date.
* The SLA for an order starts when the order is released from the OMS to the WMS.
* A dashboard widget is created to show all customer orders in a review status.
* Deposco OMS flags duplicate orders in Review status
* Incoterms - For orders, if the Duty Paid By is not set at the Customer Order level, then the value specified on the Shipping Service ‘Duties Paid’ checkbox (true/false) is used – Select the checkbox to ship DDP, or clear the checkbox to ship DDU. Outside of DDU and DDP, this field is intended to be informational on outbound documents (i.e. Commercial Invoice). The Incoterms is expected to be mapped to CoHeader.incoTermsType.
* CoPreprocessing validations record a Notice with a reason for why the customer order is in Review status.

**6.2. Purchase Experience**

The purchase experience module allows users to create purchase orders using the po entry tool. Users can create purchase orders within the Deposco ui without requiring a csv file, such as data exchange.

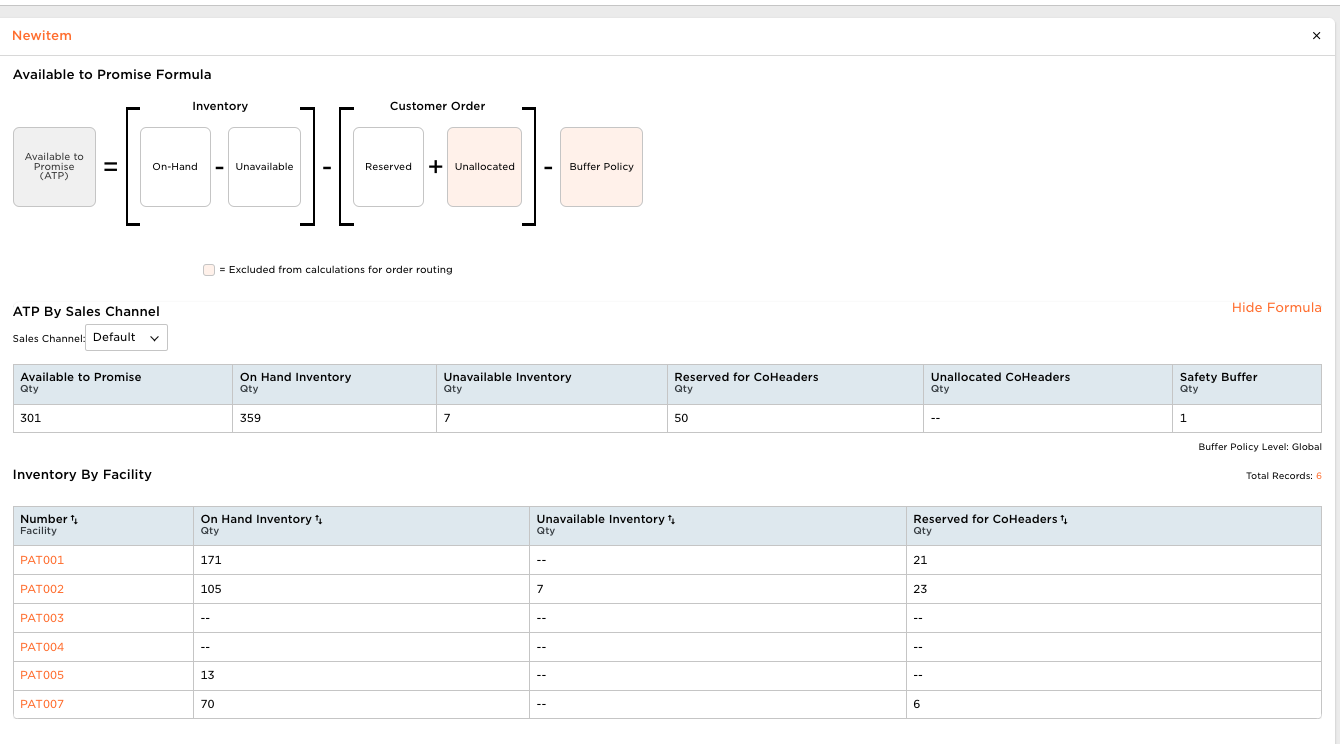


**6.3. Enterprise Inventory**

The Enterprise Inventory user interface enables you to view inventory quantities across the enterprise. For example, you can view the following quantity values for each item:

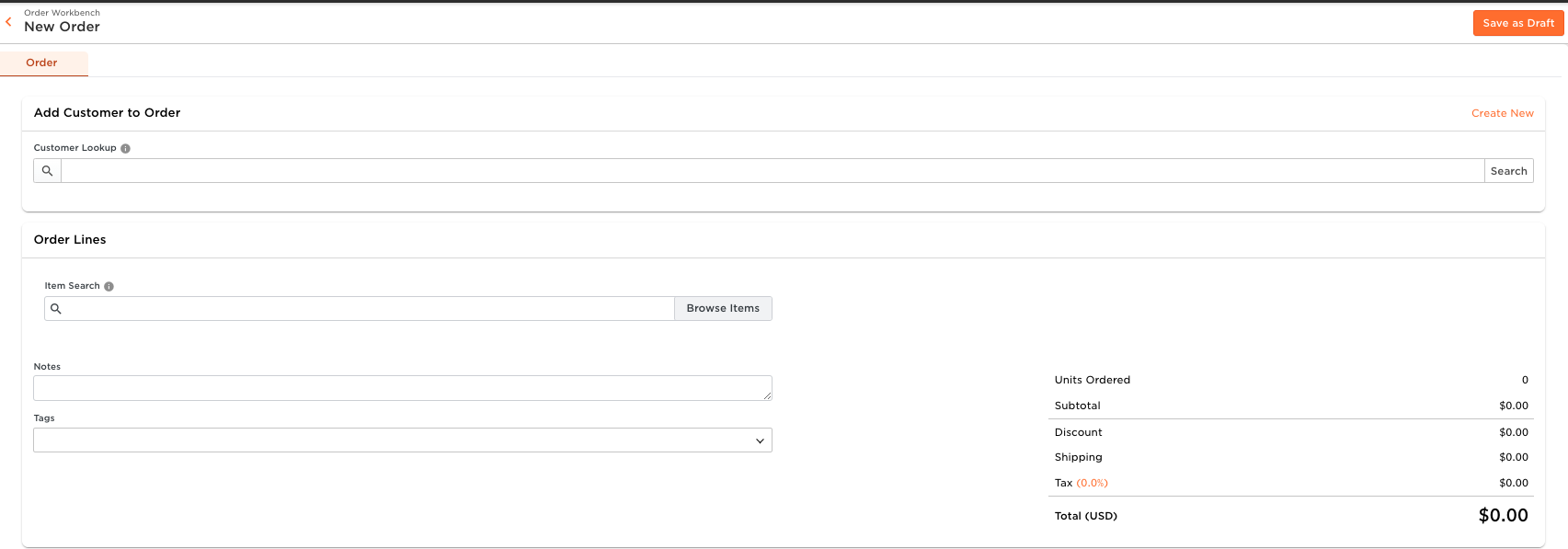
* Available to promise (ATP) quantity
* On-hand quantity
* Unavailable inventory quantity
* Reserved inventory quantity
* Safety buffer quantity

The ATP quantity is based on a configured formula which has definitions of each portion of the equation at the top of the page when you are evaluating an items inventory.



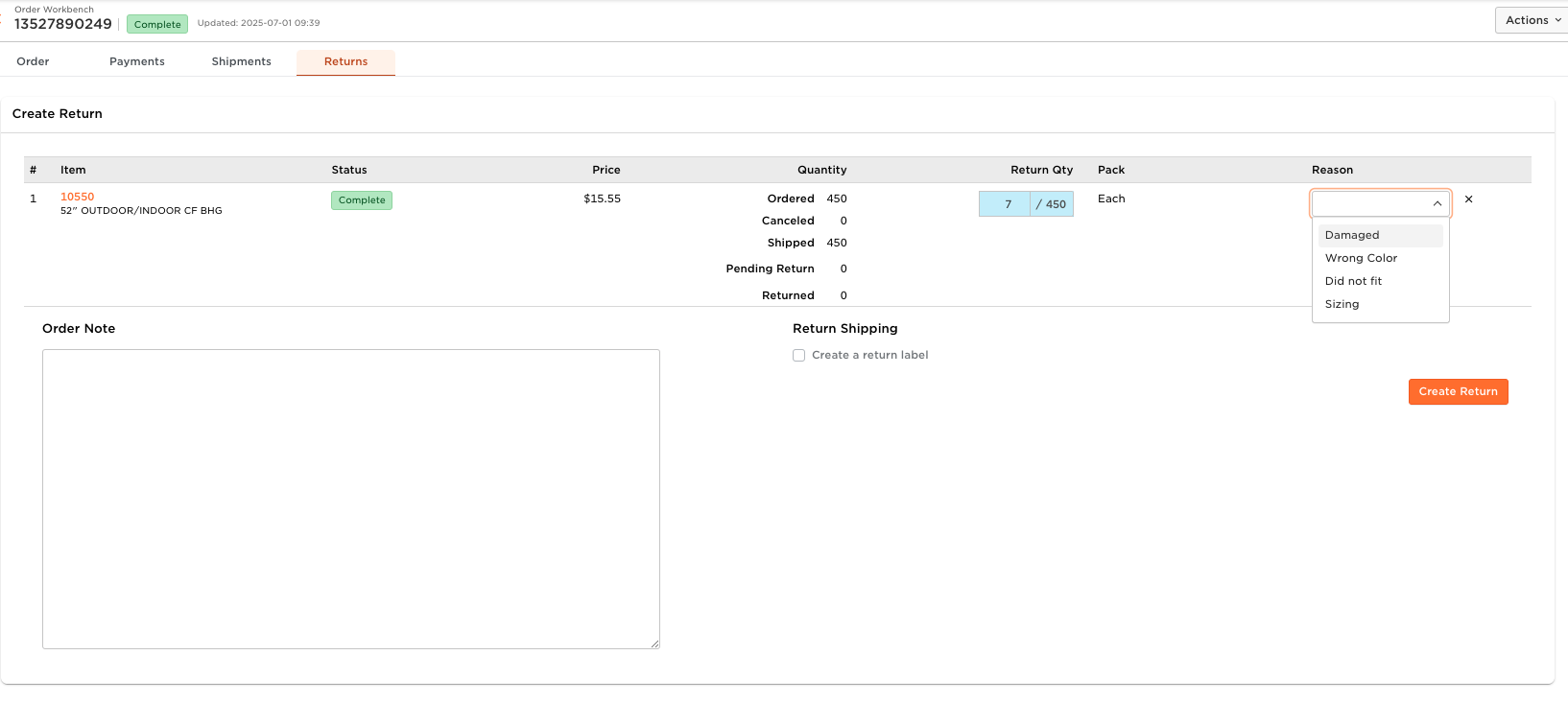
**6.4. CO Workbench (Customer Order Workbench)**

The Customer Order Workbench provides a dedicated user interface for manually creating and managing customer orders. Users can determine flat shipping rates, reason codes and pdf reports that should print at the time of customer order creation. Users may also create returns as well as rateshop and generate return labels within this process.



**6.4.1. Generating Returns**

Users generate a return by selecting the customer order that was shipped then navigating to the returns tab above the customer order number. The user selects the return qty as well as the reason for the return, which is populated via lookup table, and then generates the return label during this process. If the user does not know the sales order the return is from they can receive/create a blind rma.

****

**6.4.2. Key Assumptions**

* All customer orders created via customer order workbench begin with the letter “D”.
* The majority of customer orders are be created through edi/api.
* CO workbench cannot create returns for individual components of kit to order scenarios today

## 6.5. Sales Order Management

Sales orders are used for fulfillment processing through waving, picking, packing, and shipping. As a result, sales orders are also referred to as *fulfillment orders*. Sales Order (SO) management outlines how SOs are defined, managed, and monitored in Deposco.

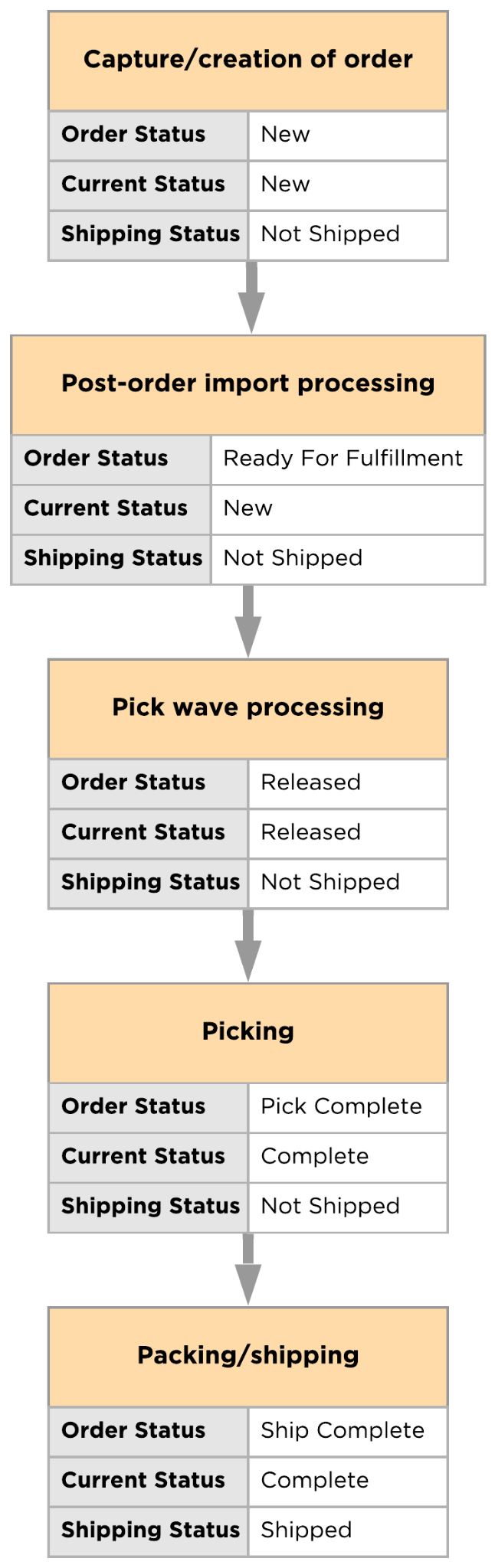
1. Orders are created in Deposco.
2. Orders are classified to determine how they should be processed (via Automations logic).
3. Multiple orders (or sometimes individual orders) are grouped into a Pick Wave to determine which orders are processed together;
4. Pick Waves are released/allocated, and orders are directly linked to Pick Tasks that specify the item, quantity, and location where the stock resides. At this point the inventory is ‘hard allocated’ and cannot be used for other orders.
5. Pick Waves are executed by users on handheld, where pickers scan the inventory into pick bins/pallets/carts, then stage to a location or packing area.
6. Packing is performed as users scan the pick bins/pallets, and packing documentation and shipping labels are automatically printed.
7. The packed boxes are either closed automatically and considered shipped OR moved to another staging location until pickup. When pickup occurs (if not auto closed) users scan to confirm what has been picked up.

### Process Flow

*Structure of a Sales Order*



*Life Cycle of a Sales Order*

**

### Sales Order Statuses

There are three fields used on the Sales Order Header to provide information on the status of the order. Each status field is listed below with a table providing the different status values and their definition.

#### Order Status

The Order status is the overall status of the order based on order activity tracking as the order progresses through the fulfillment process. It provides a more granular view than the Current Status field. Order activity tracking must be enabled for this field to be populated.

| **Order Status** | **Description** |
| --- | --- |
| New | The order is a new order that was either received from an external system or manually created and submitted. |
| Hold | The order has been processed through post-order import, failed one or more validation checks, and requires review. |
| Ready For Fulfillment | The order has been processed through post-order import and is ready for fulfillment. |
| Planned | The order has been added to a pick wave as the pick wave was built. |
| Partially Released | When the pick wave for the order was released, stock was allocated to partially or completely fulfill at least one but not all order lines for the order. |
| Released | When the pick wave for the order was released, stock was allocated to fulfill the order. |
| Back Ordered | When the pick wave for the order was released, there was not enough stock to fulfill one or more of the order lines for the order. |
| Picking | A work group that includes pick tasks that are associated with the order has been assigned to a user, and the user has picked at least one stock unit to fulfill the order. |
| Pick Complete | All pick tasks to fulfill the order have been completed. |
| Canceled | A process or a user canceled the order, and the order will not be fulfilled. |
| Partially Shipped | Some of the stock to fulfill the order has shipped, and the shipment (or the trip, for freight shipping scenarios) has been closed. Other parts of the order are either being picked or are back-ordered. |
| Ship Complete | All order lines have been fulfilled and shipped, and the shipment (or the trip, for freight shipping scenarios) has been closed. An order may also move to this status if one or more order lines were canceled but the rest were successfully fulfilled. |

#### Current Status

The Current status is the overall status of the order as the order progresses through the fulfillment process.

| **Current Status** | **Description** |
| --- | --- |
| New | The order is a new order that was either received from an external system or manually created and submitted. |
| Hold | The order has been processed through post-order import, failed one or more validation checks, and requires review. |
| Released | The order was added to a pick wave, and stock was allocated to fulfill the order when the pick wave was released. |
| Back Ordered | The order was added to a pick wave when the pick wave was built. However, when the wave was released, there was not enough stock to fulfill one or more of the order lines for the order. |
| Picking | A work group that includes pick tasks that are associated with the order has been assigned to a user so that the user can pick stock to fulfill the order. |
| Canceled | A process or a user canceled the order, and the order will not be fulfilled. |
| Complete | All pick tasks to fulfill the order have been completed. |
| Closed | All pick tasks to fulfill the order have been completed, but some or all of the order was returned. |

#### Shipping Status

The Shipping status indicates whether the order has not yet shipped, part of the order has shipped, or all of the order has shipped.

| **Shipping Status** | **Description** |
| --- | --- |
| Not Shipped | No shipments have been created for fulfillment of the order. |
| Partially Shipped | At least one shipment has been created for fulfillment of the order, but one or more order lines have not yet been fulfilled. |
| Shipped | All order lines have either been fulfilled and shipped or canceled. |

### 6.5.1. Sales Order Types and Origins

#### Sales Order

* Imported via EDI/API
* Created via customer order workbench
* CSV/Data Exchange for Transfer orders

**6.5.2. Key Assumptions**

* All sales orders are created from an existing customer order.
* Only outbound transfer orders and inbound transfer orders should be created manually without first processing through the oms.

### 

### 6.6. Sales Order Pre-Processing

#### 6.6.1. Automations

Automations enable users to process orders after they are pulled into Deposco through a socket integration, API integration, or Data Exchange import.

During this processing, a set of predefined actions are performed on orders if they meet certain conditions.

PAT has defined the below use cases for automations:

| **Trading Partner** | **Description** |
| --- | --- |
| Amazon Seller Central | Ship as fedex home delivery prepaid |
| Wayfair | Ship as fedex home delivery prepaid. Ship as LTL if over 201 pounds OR 20 cases |
| Walmart DSV | If address is international set shipvia to Fedex International |
| Mule-it.com | All orders changed to Fedex Home Delivery prepaid. |
| Homefront Essentials | Always change from third party to prepaid |
| HD Supply | All HD Supply should ship as UPS Ground Third party |
| Build.com | Routed on the [build.com](http://build.com) site, not through deposco. |
| Northern tool and equipment | Residential and up to 70 pounds uses fedex home delivery. Anything over 70 pounds and up to 150 pounds will ship fedex ground. Commercial will ship Fedex Ground up to 150 pounds. Always ships as third party. |
| Sherwin Williams | Commercial ships Fedex Ground prepaid. Residential ships fedex home delivery. LTL ships Estes |
| Tractor Supply | Orders less than 150 pounds use Tractor supply UPS account third party billing. Shipments over 150 pounds are shipped via XPO |
| Acme Tools | Small parcel use fedex ground third party. When routing use fedex freight collect |
| Ace Hardware | Small Package Fedex Ground prepaid. LTL ships Estes prepaid |
| Orgill | Fedex Ground prepaid for commercial. Fedex Home Delivery prepaid if shipping to someones home |
| Farrell Calhoun | Fedex Ground prepaid |
| Tamalpais Paint & Color Inc | Fedex Ground prepaid |

PAT is responsible for maintaining changes to Automation logic after initial configuration by the Deposco team.

#### 6.6.2. Post Order Import

Post Order Import (POI) logic is applied upon Sales Order creation in a similar fashion to Automations. Post Order Import logic is used in the event that complex conditions or behaviors are needed that are not available via Automations.

* Set “POI Complete” (orderHeader.complete) = “True” at the end of Post Order Import processing to confirm POI is executed properly on all orders.
* OrderLines are evaluated to determine if the order is Ready to Ship (RTS) or Not Ready To Ship (NRTS) The Value RTS or NRTS is set on the orderHeader.customAttribute1 (Order Type)
* Ship Vias are evaluated to determine if the shippingService.freightType == “Parcel”. If this is true orderHeader.customAttribute2 (Fulfillment Type) is set to “Parcel”.
* Trading Partner = “Parts Orders”
  + Orders come down with SCAC code= AM24 and if order exceeds 5 pounds and is a ceiling fan then ship Fedex Home Delivery
  + If commercial ship as Fedex Ground, if residential ship Fedex Home Delivery.
* Residential check using fedex address validation

POI rules are configured by Deposco consultants and may be altered or updated by PAT once configured. **Any new poi rules the team wishes to create must be created and tested first in ua/sandbox before moving to production.**

### 6.7. Sales Order Updates and Cancellations

#### 6.7.1. Sales Order Updates

Outbound orders and outbound order lines may be updated by one of the following methods:

* Imported updates via API/EDI integration

The outbound order or line must be in one of the below eligible statuses in order to perform an update.

* New

If users only make manual updates in the Deposco ui these updates don’t not match what is expected in the ERP and may cause downstream issues.

**6.7.1.1 Key Assumptions**

* All updates for a customer order/sales order come from api.
* Once warehouse activity has been initiated updates will not be pulled down

#### 6.7.2. Sales Order Cancellation

Users should only cancel customer orders on the customer order entity. If a user cancels an order or orderlines on a sales order the system does not attempt to reroute the cancelled portion back through the oms.

### 6.8. PAT Order Classifications

Order classification encompasses the pre-fulfillment activities that occur when a Sales Order is created during POI. These actions are performed automatically, by the system, as orders are created in the WMS. This includes data cleansing, and order profiling, which are all detailed in the below sections.fx

#### 6.8.1. Order Release Profiles

Deposco has multiple Order Release Profiles working in tandem (Singles Orders,Batchpicking, etc,).

The default ORP is defined with a custom pick strategy of Biggest pack first then Location Pick Sequence pick strategy. PAT uses the order close policy of “Close if backordered (Fill or Kill No Zero SLs) but does not create 0 qty shipment lines” and a backorder policy of “All stock on order to fulfill”. This means Deposco cancels any quantities or orderlines that could not be fulfilled after the order has been shipped and Deposco does not build and release the order onto a pickwave unless ALL stock on the order can be fulfilled.

Location Pick Sequence - Allocate product in order of Pick Sequence added through DE uploads. Lower the number = better sequence. i.e if pick sequence = 1 for pickloc1 → product is always allocated from pickloc1, pending On Hold statuses.

**Batch Picking ORP**

**Single Order Picking ORP**

**Single Bulk ORP**

#### 6.8.2. Allocation/Wave Release

PAT Orders are manually waved and released, but DEP can create waving schedulers with the expectation they do not run automatically and only be used for waving large amounts of orders.

* **Option 1:** User manually selects orders to be waved and build and releases the orders from a PAL on the sales order
* **Option 2:** Manually select “run” on the waving scheduler to wave and release orders in bulk

Waving rules are defined using any order-level attributes (tradingPartner, shipVia, custom attributes, priority, hazmat etc.).

Waves are Undone, or Rolled Back by another PAL on the PickWave entity. There are 2 options here:

* **Option 1:** Undo Release: Undo entire pickwave - and all orders on it.
* **Option 2:** Undo Release - Order: Undo only a specific order from the PickWave.

When Undo Release is performed - the pickwave is changed to “Planned” status from “Released” and the Order(s) are changed to “New” current status.

#### 6.8.3. Stock Roll-up Functionality

The stock roll-up functionality is used during the waving and picking processes to optimize picking and to “roll-up” larger picks into the largest pack type that exists in relation to the order.

Example: My customer order is created with one orderline that requires 125 eaches of item A. I currently have pack types of Each(1), Case(12) and Pallet(1000) within Deposco. When the user builds and releases this order, assuming there is enough available stock to fulfill it, we would expect to roll up the picktasks and have the user pick 10 cases of 12 in one pick task (120 eaches) then 5 eaches in another picktask. If you had 150 eaches in the location the picktask is directed to then the stock would now be 12 cases and 6 eaches before picking.

### 6.8.3.1 Key Assumptions

* PAT uploads all associated pack types for an item prior to wave release
* Users will scan pack upc/gtin during picking to ensure correct pack is picked
  + PAT will assign pack upc’s to remaining 25% of items missing

#### 6.8.4. Pick Wave Grouping

Deposco groups orders into different Pick Waves - these Pick Waves are used for greater efficiency in picking alike or similar orders.  
  
The grouping policies (or “rules”) to be used for PAT in Deposco are:

1. Freight Orders - Orders where the customerOrder.customAttribute2 = “Freight” and a limit of one per wave.

2. RTS Singles - Orders where the orderHeader.customAttribute1= = “RTS” and totalOrderQuantity = 1

3. RTS Multis - Orders where the orderHeader.customAttribute1= = “RTS” and totalOrderQuantity > 1

4. NRTS Multis - Orders where the orderHeader.customAttribute1= = “NRTS” and totalOrderQuantity > 1

5. NRTS Singles - Orders where the orderHeader.customAttribute1= = “RTS” and totalOrderQuantity = 1

#### 6.9. BackOrder Management

The status “Back Order” can mean one of the following:

1. “Back Order” in the OMS means there is not enough stock to fulfill the entire CoHeader in its entirety. Deposco’s Bright Order Module only releases small parcel orders on a scheduled basis if the entire order can be fulfilled. PAT team will force allocate freight orders.
2. Stock awaiting replenishment, meaning there is enough inventory inside the four walls of the warehouse to release the order from the OMS to the WMS, but the inventory is not in an eligible location based on the waving logic.

To aid the user in determining which orderLine is the cause of a back order in the WMS, an orderLine level field exists, called orderLine.backOrdered, which is a boolean field.

* True means the line was the cause of a back order and a replenishment is needed to fulfill the back order
* False means the line was not the cause of a back order and there is enough stock to release this line at a pickable location.

Orders are only sent to Deposco if they should be considered for fulfillment, meaning stock is available for release in the WMS.

With this backorder policy, picking cannot start unless the entire order can be fulfilled - this will only apply to small parcel. Large LTL/Freight orders will go through OFP and trigger replenishments before picking is complete. Normal LTL/Freight orders will use some stock on order to fulfill order release profile. Stock must be in a pickable location for allocation to occur. Back ordered orders are removed from the pick wave upon wave release. This strategy is used to mitigate multiple-package orders.

An order goes into a “Back Ordered” status when a pick task is exceptioned and there is not any eligible stock available to fulfill the exceptioned pick task. It is up to the discretion of the operations team to determine next steps – cancel the back ordered items/quantities or await a replenishment and re-release the order for picking.

If any sales order line is identified as a true back order, either the entire sales order is canceled or the sales order is partially canceled in the WMS by the operations team. By default, the OMS attempts to re-route the order to another facility. If the order cannot be re-routed, the back order is then managed in the OMS by the client’s customer service team. If re-routing is not desired, the canceled quantities need to be canceled on the Customer Order (OMS) prior to initiating the cancellation on the Sales Order (WMS).

### User Stories

*As an Order Management user, I want the ability to…*

* Fully cancel an existing Sales Order.
* Partially cancel an existing Sales Orderline’s quantity.
* Manage back orders in the OMS if inventory is not available in the WMS due to a stockout.
* Validate a sales order’s address using Address Validation service.
* View Back Orders and determine which lines caused the back order.

### Key Assumptions

* All outbound orders are associated with a Customer Order, meaning they originate from the OMS.
* If an order is canceled in the WMS the OMS attempts to re-route this order
* Cancel the CoHeader/CoLines in the OMS and not the WMS if orders are not to be re-routed
* Back Ordered Lines are marked as orderHeader.backorderd = “true”

#### 6.10. Waving

Pick waves organize the daily flow of outbound order fulfillment work within a warehouse or distribution center. Waving is an application of short-interval scheduling that selects a set of orders based on properties like planned release or ship time, then releases the work needed for those orders in an organized fashion, so that the work can be performed efficiently by a set of users within a work shift.

Waving enables management to monitor and manage performance throughput throughout the day so that problems can be addressed quickly and staffing can be more effectively utilized throughout the shift.

The table below outlines the definitions of each Pick Wave statuses.

table below outlines the definitions of each Pick Wave statuses.

| **Pick Wave Status** | **Description** |
| --- | --- |
| Planned | Pick Wave has been created but not Released. Orders have been grouped into a Pick Wave but the Work Group and Pick Tasks have not been created yet. Only available in Build and Release Wave (legacy) |
| Released | A work group has been created with pick tasks. |
| Pending Replenishment | If replenishment tasks were generated for the pick wave, the pick wave was generated as part of a preview run for the Order Fulfillment Profile, and the Wave Demand Replenishment Policy for the profile is set to Replenish Then Pick, then the pick wave is created during the preview run with a status of Pending Release, and the work groups and pick tasks are created with a status of Pending Replenishment. The orders on the pick wave move to a status of Pending Replenishment. |
| Partially Pending Replenishment | If replenishment tasks were generated for the pick wave, the pick wave was generated during a manual or scheduled run for the Order Fulfillment Profile, and the Wave Demand Replenishment Policy for the profile is set to Replenish And Pick, then the pick wave is created during the processing run with a status of Partially Pending Replenishment |
| Pick Complete | When all pick tasks for a work group have been completed, the work group moves to a status of Complete. The user brings the bin(s) with the picked stock to a staging location such as a packing location. If all pick tasks for all orders on the pick wave have been completed but the orders have not yet been packed or shipped, then the status of the pick wave moves to Pick Complete |
| In Packing | After the first order on the pick wave has been packed into a shipping container and shipped, the pick wave moves to a status of In Packing. The pick wave then remains in a status of In Packing until all orders have been packed and shipped. |
| Ship Complete | All sales orders in a pick wave have been shipped. |

### Order Fulfillment Profiles

Order Fulfillment Profiles are available for the Patterson team to configure. There is no capability to leverage customization such as Wave Release rules logic in Order Fulfillment Profiles. Order Fulfillment Profiles during Phase 1 of the implementation are used for orders in which Patterson requires an order to be released when there is not enough pickable stock to release the order.

The scenario in which Patterson utilizes OFP is this:

A single order is released into the WMS that is so large that many pallets must be replenished to the pick front location. Deposco releases what is available and generates a Wave Based Replenishment for these items. Once the stock is picked the Order’s status is updated to “Pending Replenishment”. Once the replenishment is complete an event triggers the order to “re-release”. This process repeats itself until the order is fulfilled.

### Build And Release Pick Waves (Legacy)

Pick waves organize the daily flow of outbound order fulfillment work within a warehouse or distribution center. Waving is an application of short-interval scheduling that selects a set of orders based on properties like planned release or ship time, then releases the work needed for those orders in an organized fashion, so that the work can be performed efficiently by a set of users within a work shift.

Waving enables management to monitor and manage performance throughput throughout the day so that problems can be addressed quickly and staffing can be more effectively utilized throughout the shift.

The creation of pick waves is two steps. Pick Waves are Built by the system, either by a scheduler or manually using the Build Pick Wave PAL. Pick Waves are then released manually by a waving clerk using the Release Pick Wave PAL. The capability to Build and Release a pick wave in one step is supported by the Build And Release Wave PAL.

#### Wave Profiles

The configuration used to determine the different wave profiles is stored within a delegator as part of the Pick Wave Build Profile application. A delegator contains different wave profiles.

Each wave profile allows for a unique rule set to be applied to the orders being processed. Based on the above order classifications, certain expressions in the rule set categorize orders into the wave profile. The expressions are defined within each wave profile further below. Pick Waves are grouped by key value pairs in each rule set.

Based on the order classifications, waves are grouped by the following attributes:

* Order Type
  + Ready To Ship
  + Not Ready to Ship
  + Freight
* Business Unit
* Facility

Example wave numbers:

| **Wave Type** | **Format** | **Example** |
| --- | --- | --- |
| RTS, Singles | [facility]-[business unit]-RTS-[date]-[build time]-[counter] | *PAT007-HKC*-RTS*–251112\_08:33:35-1* |
| NRTS | [facility]-[business unit]-NRTS–[date]-[build time]-[counter] | *PAT007-HKC-*NRTS*–251112\_08:33:35-1* |
| RTS, Multis | [facility]-[business unit]-Mixed–[date]-[build time]-[counter] | *PAT007-HKC-*Mixed*–251112\_08:33:35-1* |
| Freight | [facility]-[business unit]-[tradingPartner]-LTL–[date]-[build time]-[counter] | *PAT007-HKC-*COSTCO-LTL*–251112\_08:33:35-1* |

The configuration used to determine the different wave types is stored within the Pick Wave Build Profile application. Each profile allows for a unique rule set to be applied to the orders being processed.

The wave types for each Pick Wave Build Profile are outlined below:

**Standard Build Pick Wave Profile**

| **Seq** | **Wave Type** | **Criteria** | **Order Release Profile** | **Min / Max # of Orders** | **Cartonization** | **Zones** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | RTS Singles | orderHeader.customAttribute1 = “RTS” && orderHeader.pickWave == null  && totalOrderQuantity = 1 | Batch Picking | 1 / 100 | True | TBD |
| 2 | RTS Multis | orderHeader.customAttribute1 = “RTS” && orderHeader.pickWave == null  && totalOrderQuantity > 1 | Batch Picking | 1 / 50 | True | TBD |
| 3 | NRTS Singles | orderHeader.customAttribute1 = “NRTS” && orderHeader.pickWave == null  && totalOrderQuantity = 1 | Single Scan Picking | 1 / 10 | False | TBD |
| 4 | NRTS Multis | orderHeader.customAttribute1 = “NRTS” && orderHeader.pickWave == null  && totalOrderQuantity > 1 | Batch Picking | 1 / 1 | False | TBD |
| 5 | Freight | orderHeader.customAttribute2 = “Freight” && orderHeader.pickWave == null  && totalOrderQuantity > 1 | Single Scan Picking | 1 | False | TBD |
| 6 | Exception, Re-Pick | total order quantity > 1 && fulfillment type == parcel && pickWave != null | Batch Picking | 1/12 | False | TBD |

Each wave type is described in detail below.

#### Ready To Ship (RTS)

Ready To Ship (RTS) orders are processed together on a pick wave; this allows for increased picking efficiency since pickers can pick multiple orders/items onto the picking LPN while traveling, without the need to separate the inventory by order.

* Fulfillment Type is set to Parcel
* POI Complete is true
* Order is RTS
* Order is in either a New or Back Ordered status
* Order is not already associated with a Pickwave
* Min number of orders is N
* Max number of orders is N
* Order Release Profile is Batch Picking
* Order is Cartonized
* Shipping label is generated

#### Not Ready To Ship (NRTS)

NRTS orders that are grouped into multis waves. These orders are picked into discrete bins.

* Fulfillment Type is set to Parcel
* POI Complete is true
* Order is NRTS
* Order is in a New or Back Ordered status
* Order is not already associated with a Pickwave
* Min number of orders is 1
* Max number of orders is N
* Order Release Profile is BatchPicking
* Order is not Cartonized
* Shipping label is not generated

#### 

#### Mixed

Orders that have a mix of RTS and NRTS orderLines. These orders are picked into discrete bins.

* Fulfillment Type is set to Parcel
* POI Complete is true
* Order is Mixed
* Order is in Back Ordered status
* Order is in a New or Back Ordered status
* Min number of orders is 1
* Max number of orders is N
* Order Release Profile is BatchPicking
* Order is not Cartonized
* Shipping label is not generated

#### Freight

B2B orders that are shipping LTL/FTL.

* Fulfillment Type is set to Freight
* POI Complete is true
* Order is in a New or Back Ordered Status
* Order is not already associated with a Pickwave
* Min number of orders is 1
* Max number of orders is 1
* Order Release Profile is SingleOrderPicking

#### Exception Re-Pick

Orders that have been exceptioned during picking. These orders are picked into discrete bins.

* Fulfillment Type is set to Parcel
* POI Complete is true
* Order was exceptioned during picking
* Order is in Back Ordered status
* Order is already associated with a Pickwave
* Min number of orders is 1
* Max number of orders is 12
* Order Release Profile is BatchPicking
* Order is not Cartonized
* Shipping label is not generated at picking

#### Exception Orders

In any of the above RF picking processes, a picker may exception a pick task if they are unable to complete the task. If stock for that item exists in a secondary location, they are rerouted to the location with the next highest pick sequence. The newly created pick task is picked in order of pick sequence.

When a pick task is exceptioned, and a new pick task is not generated, the SO and Order Line is placed on Back Order and a cycle count task is generated to ensure that the stock unit quantity is correct. If there are other orders that also have pick tasks for the same stock unit, they are reallocated during the picking of that specific order. The stock unit within the picking location where the exception is performed is put in a On Hold status until the cycle count is completed. When orders have an exceptioned task, they cannot ship until the remainder of the order is picked or canceled. Orders that are incomplete are manually moved to an Exception staging location after picking. If an order is not picked complete a validation is in place to prevent the packing of this order.

Exception orders are re-waved onto an Exception Re-Pick pick wave. This allows all exceptioned orders to be re-picked together once enough available inventory has been replenished to the picking locations. Have message in packing after container scan that the order is in backordered and to reach out to admin team. User will stage to end of their aisle.

**Order Fulfillment Profiles**

| **Seq** | **Wave Type** | **Criteria** | **Order Release Profile** | **Min / Max # of Orders** | **Cartonization** | **Zones** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Reset | Resset Orders | Single Scan Picking | 1 / 100 | False | TBD |

#### 6.11. Pre-Manifest/Shipping Label Generation at the Wave

With cartonization at the wave, the process of determining the optimal shipping containers for each order occurs when a pick wave is released.

Cartonization at wave release is used for the following wave types:

* Ready To Ship

There are many flavors of cartonization at wave release based on the specific requirements of the operations. The following settings are enabled for this implementation:

* Cartonization Provider: Deposco
* Shipping Label Generation
  + Shipping Labels are generated, but not printed during wave release. They are stored in the database to be printed at the pack station.

The Single Container Packing process is used to print the shipping label and update the shipping status of the order. After the Single Container Packing process is complete, the shipping container can then be sealed and moved to the next step in the fulfillment process.

#### Data Requirements

* Pack Dimensions - all items must have valid pack dimensions (Length, Width, Height, Weight, WeightUom, DimensionUom) in order to be eligible for cartonization.
* Pack Additional Attributes - consider setting the Ready to Ship flag, Carton Group attributes. These attributes are considered when evaluating the optimal carton during cartonization.
* Units of Measure - all pack dimension UOMs must be in the same measurement system for a single pack.
* Carton Type Dimensions - all carton types must have valid dimensions (Length, Width, Height, Weight, WeightUom, DimensionUom) in order to be eligible for cartonization.
* Carton Type Additional Attributes - consider setting a maximum weight limit and defining a business unit on the carton type entity when applicable. These attributes are considered when evaluating the optimal carton during cartonization.

### Order Release Profiles

An order release profile defines how orders are processed when a pick wave is released. This includes:

* How to allocate stock for order fulfillment.
* How to manage orders and order lines when there is an inventory shortage for an order (backorder policy).
* How to create pick tasks and their associated pick lines for allocated stock.
* How to group pick tasks into work groups.
* Which picking process to use for picking the orders on the pick wave.
* The default staging location that is assigned after the orders in the pick wave are picked.

Multiple order release profiles can be created to accommodate different picking scenarios. The following table details the different order release profiles:

| **Name** | **Backorder Policy** | **Pick Option** | **Pick Strategy** | **Pick Task Grouping Policy** |
| --- | --- | --- | --- | --- |
| Batch Picking | All stock on order to pick | Zone Picking | Pick Sequence, FEFO, FIFO | Bulk Item Pick |
| Single Order Picking | All stock on order to pick | No Zone Pick | Biggest Pack First, Biggest Quantity First, FEFO, FIFO | Bulk Item Pick |

### 

## Parcel Order Fulfillment

The picking processes in Deposco are used to select and validate the stock units being allocated to Sales Orders. Users are prompted to scan and confirm each Location, Item, quantity and picking container.

### Parcel Overview

Once orders are grouped into pick waves, they are allocated and ready to begin fulfillment. The following table details the process for which each wave type is picked.

| **Wave Type** | **Order Release Profile** | **Picking Process** | **Packing Process** |
| --- | --- | --- | --- |
| RTS | Batch Picking | Batch Picking | Single Container Packing |
| NRTS Multis | Batch Picking | Batch Picking | Single Scan Packing |
| NRTS Singles | SingleBulkPick | Single Scan Picking | Single Scan Packing |
| Freight | Single Order Picking | Single Scan Picking | Assign Trip To Container |
| Exception / Re-pick | Batch Picking | Batch Picking | Single Scan Packing |

### 

#### 6.12. Ready To Ship Picking (Batch Picking)

Ready To Ship Picking utilizes the Batch Picking process. This process is used to pick ready to ship items into their own shipping container. Once the item is picked, and labeled, it is added physically to a picking vehicle (most commonly a pallet). Once the picker gets to the end of the aisle they stage this picking vehicle at the pack station.

### Key Assumptions

* Each item is Ready To Ship
* Each item is picked into its “own” LPN
* LPN Labels are printed after each item is picked
* The LPN Number of each item is its tracking number
* Orders are staged at the end of the aisle to the aisles pack station
* Items are picked to a pallet/conveyor physically but not systematically

#### 6.13 Not Ready To Ship Sinlges (Single Scan Picking)

Not Ready to Ship Picking utilizes the Single Scan Picking Process. With Not Ready to Ship Picking, a group of orders is picked at the same time and sorted into one bin or similar equipment..

### Key Assumptions

* No item on the order is ready to ship.
* Items are picked into reusable transient bins
* LPN Labels are not printed during picking
* One picker will pick the entire wave across all eligible zones.
* The wave is staged at the end of picking and NOT at the end of each aisle.

#### 6.15 Not Ready To Ship Multis (Batch Picking)

Not Ready To Ship Multis Picking utilizes the Batch Picking Process. With Mixed Picking, a group of orders is picked at the same time and sorted into reusable or transient bins on carts or similar equipment, which provides preliminary staging for packing and shipping of each order. These orders can contain a mixture of RTS and NRTS items.

### Key Assumptions

* Orders has more than one item and is not fully RTS
* Items are picked into reusable transient bins
* LPN Labels are not printed during picking
* One picker will pick the entire wave across all eligible zones.
* The wave is staged at the end of picking and NOT at the end of each aisle.

#### 6.16 Exceptions During Picking

PAT has the option to “exception” any pickTask using the “exception” button on the ItemEntryScreen. The Exception menu contains four buttons: “Exception”, “Suspend Task”, “Suspend Workgroup” and “Stop Workgroup”.

* **Exception** - User is be prompted to select a reason code (NO STOCK, OTHER, STOCK DAMAGED). Once selected, the associated stockUnit is put “On Hold” and a cycle count task is added to the cycle count queue (if one did not already exist).
  + Reallocation - Deposco is configured to automatically reallocate the pickTask to another stockUnit location, if available. Reallocation is set to true for all picking processes. If the pickTask cannot be reallocated the order goes into a “Back Ordered” status.
* **Suspend Task** - User skips the current pickTask. This pickTask is added to the end of the workGroup for the user to complete later.
* **Suspend Workgroup** - User Suspends the workGroup. The workGroup is in a “suspended” state and the original user is still assigned. This is commonly used when the user stops for a lunch break and does not want another picker to take their place.
* **Stop Workgroup** - Users Suspend the workGroup. The workGroup is unassigned from this user so that it is picked up by another user. This is commonly used when the user leaves for the day in the middle of their pick.

**Key Assumptions**

* The picking screens are split to where there is one scan per screen for the location, item and quantity entry.
* As a wave is in progress, the inventory is systematically staged in the default WHSE–Floor location.
* After picking is completed, the user scans the picking container to a staging location of their choice. The user must scan a location of type ‘Staging’.
* Orders that have been exceptioned are waved separately.
* Cycle counts are created during an exception and only the stockUnit is placed “On Hold”.

## Packing

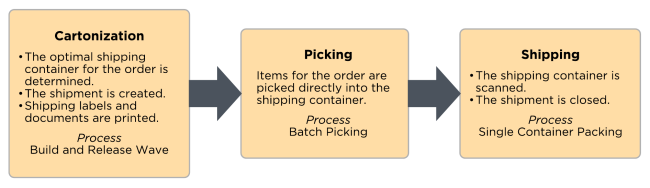
#### 6.17.1 Ready To Ship Packing (Single Scan Packing)

Ready To Ship Packing utilizes the Single Container Packing process. This process is used in scenarios where orders are cartonized during the building of the wave.

With cartonization at the wave, the optimal shipping container for an order is determined when a pick wave is released. The order is picked directly into the shipping container, and a packing process is therefore unnecessary.

If the shipment is created during the cartonization process, then the user scans the shipping container in the Single Container Packing process to identify the order and shipment. The user verifies the container dimensions and weight and the shipping label and packing slip are printed. Once the user selects the Submit button, the shipping container is sealed and staged.

#### Process Flow



#### Key Assumptions

* A packing slip specified on the tradingPartner.packingSlipReport is printed during this process.
* The pre-manifested shipping label is printed after each container is systematically packed and shipped
* API Calls are not made to the carriers during this packing process. Shipping Label Generation is performed during wave release.
* Shipments are closed after packing all containers
* Deposco can print a commercial invoice for international shipments
  + This is sent electronically for UPS and FedEx

#### 6.17.1 Over Packing (Single Scan Packing)

NRTS and Mixed orders are packed using the **SingleScanPacking** process. This process allows packers to pack the stock from one container into multiple bags/boxes by packing stock into a shipping container. This is done via grocery scanning into a shipping container or using the New Bin button. This process can be repeated until the order is fully packed.

#### Shipping Error Handling

If a shipping error occurs at the pack station, a separate error label with the container barcode is printed once the user stages the container. The Operations team applies the container label and uses the Move Container process to systematically move the container to the Hospital location for triage. This process mitigates the chance of a single order creating a bottleneck at a pack station due to a shipping error.

#### Key Assumptions

* This process is used to pack NRTS and Mixed orders
* A packing slip specified on the tradingPartner.packingSlipReport is printed during this process.
* Orders with a current status of “Complete” are eligible to be packed
* Items are grocery scanned into each “shipping” container
* Items are systematically moved to a new container once the ‘New Bin’ button is selected.
* By creating multiple containers via the ‘New Bin’ button, a master tracking number is generated with select carrier integrations.
* API calls are made to the carriers during the packing process in order to generate and print shipping labels.
* Shipments are created and closed once packing has been completed
* Deposco can print a commercial invoice for international shipments
  + This is sent electronically for UPS and FedEx

### 6.32. Shipping Requirements

PAT uses a variety of shipping methods for domestic and international shipments. Below is an outline of the shipping carriers, whether they are directly integrated and which documents and shipping options they require:

* UPS Parcel - integrated
  + Services:
    - UPS Ground
    - UPS Next Day
    - UPS 2nd Day Air
    - UPS 3 Day select
    - UPS Next Day Air Saver
    - UPS Next Day Air AM
    - UPS Next Day Air
    - UPS Standard
    - UPS Worldwide ExpeditedSM
    - UPS Worldwide ExpressSM
    - UPS Worldwide Saver
  + Documents:
    - Shipping Label
    - Commercial Invoice
* USPS Parcel (Stamps) - integrated
  + USPS Ground Advantage
  + USPS Priority Mail
  + Documents:
    - Manifest
* Fedex Parcel
  + Fedex Ground
  + Fedex Home Delivery
* Ship Outside System - non-integrated
  + Services:
    - Ship Outside System
  + Documents:
    - Packing Slip

##### ***6.32.5. Manifest***

Both UPS and USPS shipments are automatically manifested upon shipping label generation. This requires no end of day manifest process to be performed or printed.

#### 6.32.8. Freight Terms Type

PAT utilizes multiple shipping accounts per carrier. However, there are scenarios in which parties other than PAT are responsible for paying for shipping costs. The account that is responsible for payment of shipping costs for a shipment is determined based on the Freight Terms Type field on the associated order header for the shipment. PAT sends Deposco which freight terms type to utilize for the orders via edi/api.

##### ***6.32.9. Prepaid***

Set the Freight Terms Type field to a value of Prepaid on the order header if the shipping carrier account that created the shipment is responsible for payment of shipping costs.

##### ***6.32.10 Third Party***

Set the Freight Terms Type field to a value of Third Party on the order header to set up payment by a trading partner for a shipment.

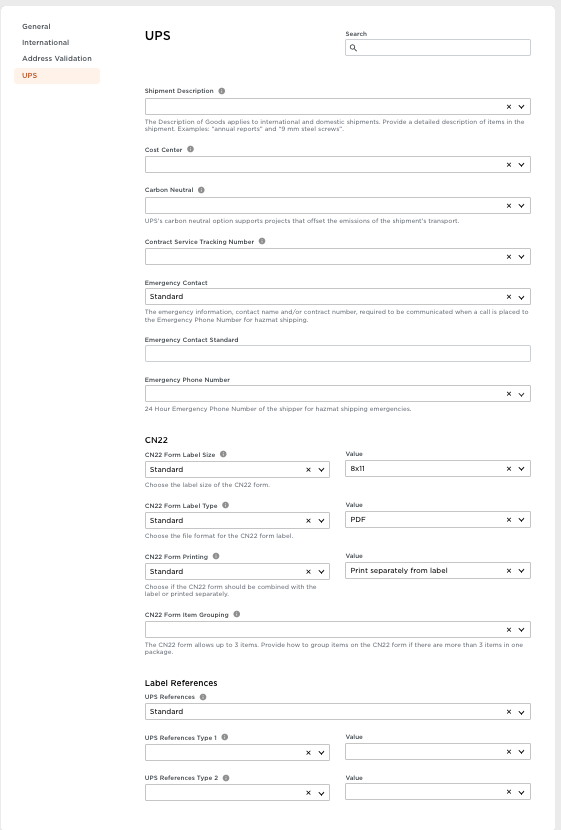
Provide the carrier account information for the customer by using one of the following options:

* On the order header, enter the shipping carrier account number in the Freight Bill To Account field, the name on the account in either the Freight Bill To Name or Freight Bill To Attention field, and the country in the Freight Bill To Country field.
* Use automations to set these fields

This method is only used by PAT in the scenario that their customers want to use their own shipping accounts.

#### 6.32.11. Shipping Options

This provides PAT users the ability to make changes to the shipping labels for UPS such as ship from address, return address and reference numbers on the shipping labels.



##### ***6.33. Ship From Address***

The ship from address used for PAT in phase 1 is be:

295 Marathon Way

Southaven, MS 38671

5555555555

##### ***6.34 Reference Field Mappings***

PAT has the external order number mapped to the transaction reference number field and the Deposco order number mapped to the purchase order field on UPS labels. Deposco maps the deposco order number to the bottom of the usps labels.

#### 6.35 International Shipping Options

PAT requires the commercial invoice to be sent electronically for international shipments and uses the following shipping services:

* UPS
* FedEx

##### ***6.36 VAT***

No VAT numbers have been provided to Deposco from PAT so this is not configured at this time.

##### ***6.37 Duties Paid By***

PAT orders utilize Delivered Duty Paid (DDP) on all orders where the seller is responsible for all costs and risks associated with delivering goods to a specific location.

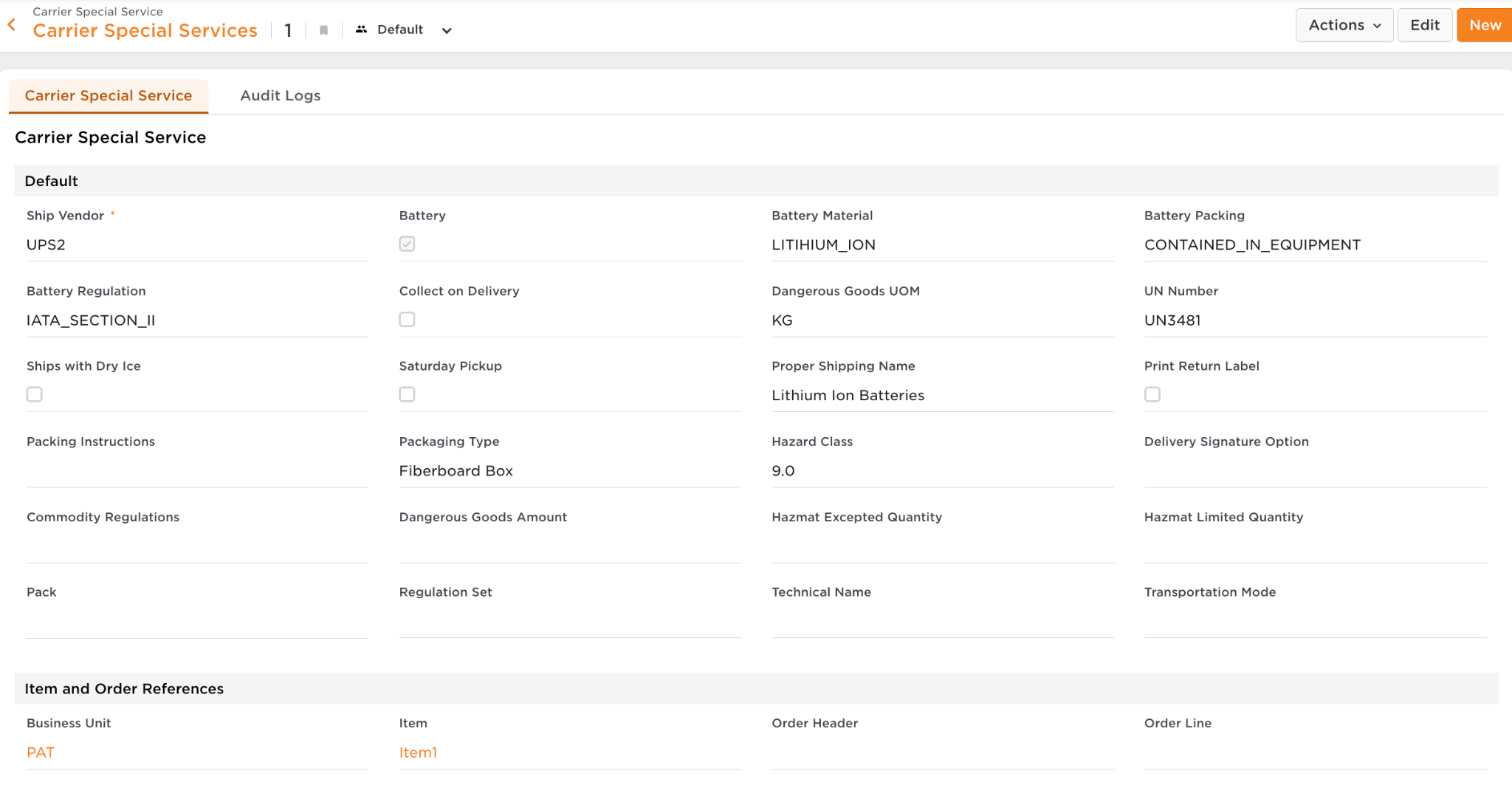
#### Special Services

##### ***6.38 Hazmat***

PAT ships hazmat items with lithium ion batteries and lithium metal batteries that require carrier special service records to be created. These hazmat items are shipped through UPS and Fedex and require hazmat shipping labels for UN3480, UN3481 and UN3091 UN Numbers. PAT only needs a hazmat shipping label and UN3091 and UN3481 stickers. The PAT team utilizes the carrier special services table for any hazmat items prior to shipping. Labels will be class 9, only label required on outside of box. Will need to print SDS sheet in case of damaged items.

**6.38.1. Key Assumptions**

* PAT has OP950 paper for Fedex and Deposco makes api calls to Fedex to print off necessary information
* PAT has 49CFR document paper for UPS and Deposco generates this report using rules in the packing process
* PAT does not require hazmat information to be stamped onto the UPS shipping labels, only the hazmat pdf documents are printed
* Items will need hazmat flag checked and the UNcode set on item and carrier special service record



##### 

This section outlines outbound order fulfillment for B2B, Retail, and Wholesale orders imported via API or manually uploaded via Data Exchange. These orders are shipping via LTL/FTL carriers.

# 7. Freight/Retailer Processes

### Warehouse Process Flow Overview

### Allocation Overview

Once orders are grouped into pick waves, they are allocated and ready to begin fulfillment. The following table details the process for which each wave type is picked.

| **Wave Type** | **Order Release Profile** | **Picking Process** | **Packing Process** | **Shipping Process** |
| --- | --- | --- | --- | --- |
| Freight | Single Order Picking | Single Scan Picking | Single Scan Packing | Assign Trip to Container |

### Wave Release

One freight order is assigned to each pick wave unless delivering to the same DC. Freight orders are released using the SingleOrderPicking or other defined Freight Order Release Profiles (ORP).

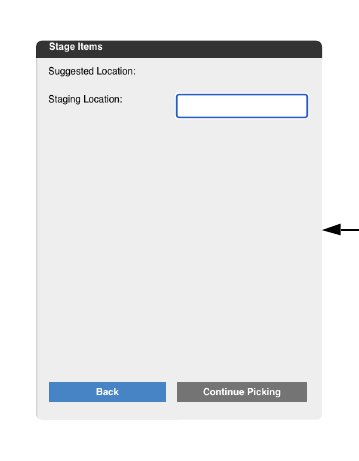
### Freight Picking (non-reset)

Freight orders are picked directed to a pallet using a pallet jack or forklift as the picking vehicle. This picking process is commonly referred to as “Pick-to-Pallet”. Each user has a roll of pre-printed, unique, LPN labels to use for picking. The pallet barcode is applied to the pallet by the picking user at the beginning of the picking process. The user scans this pallet barcode to begin picking, which assigns a work group from the pick wave to their user. To note, if the pick wave has multiple work groups, multiple users are eligible to pick the wave at the same time.

Similar to the parcel picking processes, users are prompted to verify the lot and/or expiration date for all pick tasks.

Depending on the capacity of the pallet, the user can stage inventory as needed by selecting the Stage button on the picking screen. Different staging locations are used for freight orders. However, users are allowed to scan any location when staging inventory.

Once all inventory has been picked, the user completes picking by staging the remaining items to the appropriate staging location. Orders for different trading partners may need to be staged at different staging locations. This decision is user-driven for the first pallet for a given order. As the remaining pallets are picked, the system provides a suggested staging location based on the staging location of the first pallet.

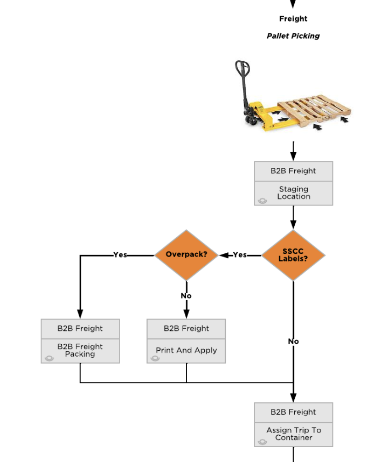


### Freight Picking (reset Orders)

Freight reset orders come in a few times a year and are imported and shipped by the thousands. Each order is waved onto its own pickwave. These orders are picked to a systematic LPN, but are placed on a physical aisle conveyor as they are picked. As the items associated with the order the “Repalletize” process is used to move them from the picking LPN onto the final shipping LPN.

Once these orders have been picked, repalletized they will be staged, and prepped for Trip Loading.

### Freight Packing

Certain freight orders may require overboxing, which can be performed using the Freight Packing process. This process mimics the Multis Packing process, which allows a user to scan a picking container and assign particular items/quantities to an outbound carton.

### Trip Management

Freight orders have trips at the sales order level. A trip/load can consist of multiple orders and a sales order can have multiple trips/loads. This is the primary distinction between LTL/TL and Parcel workflows.

Calculated fields exist on the OrderHeader to estimate the number of cases, number of pallets, and total weight associated with the order. The following assumptions are included in these calculations:

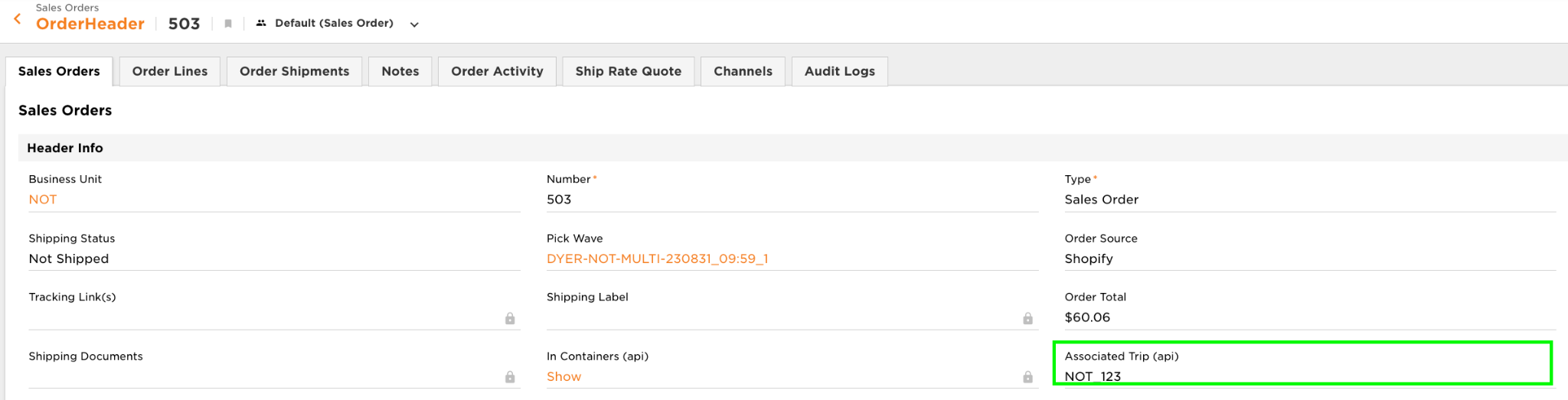
* Pallet LxWxH: Based off the Item Pallet Pack Type
  + (API SQL)
* Pallet Weight: Based off the Item Pallet Pack Type
  + (API SQL)
* Pallet Density: Total Weight(pallet) / Total Volume (pallet)
  + (API SQL)
* Total Order Cube: SUM of the cubic inches of the entire order.
  + (API SQL)

Trips/Loads for freight orders are built sometimes multiple days in advance of routing. In collect environments, many times these are built “Just-In-Time” (JIT).

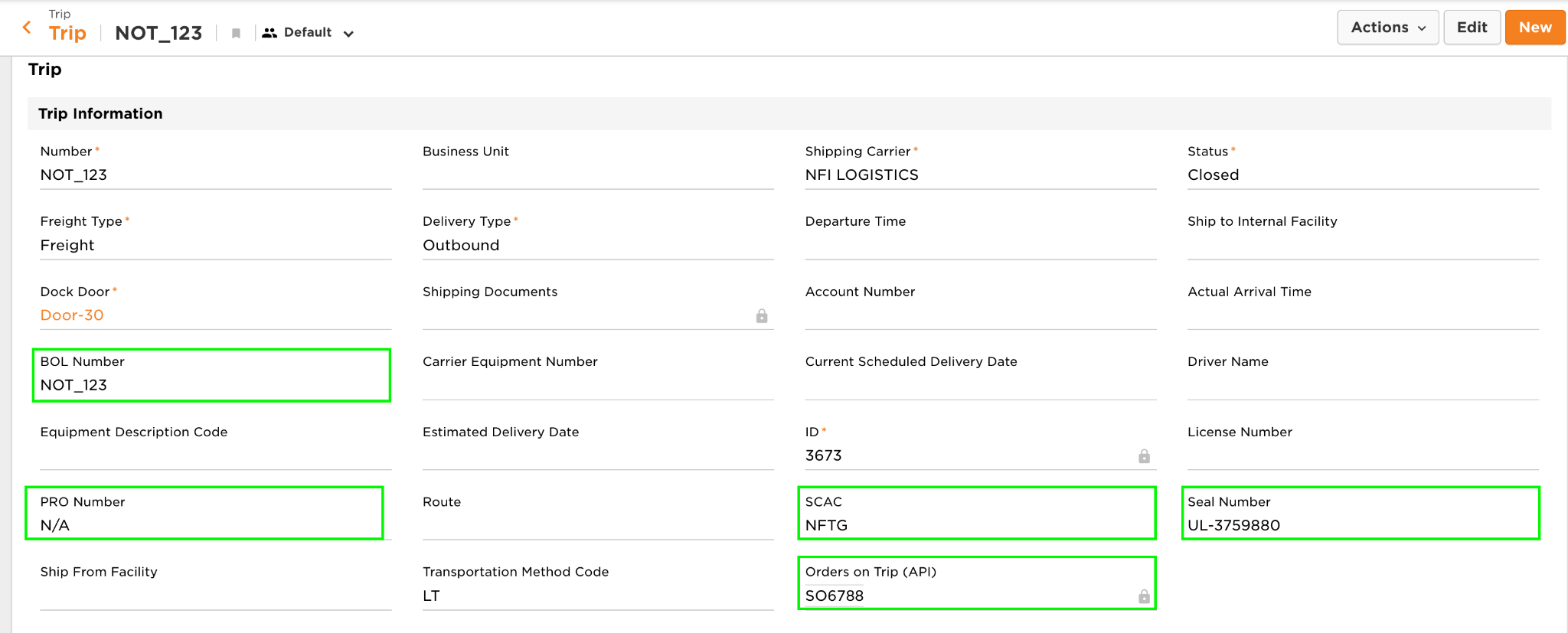
A trip can be generated as shown below.

*Select Order(s) in the OrderHeader view > PAL: Create Add To Trip > Done*

*The Trip is then displayed on the field named “Associated Trip (api)”*



*In the Trip Management Application Menu > Trip.*



At this point the Trip management team can start entering Trip-specific information:

* PRO Number
* BOL Number
* SCAC Code
* Estimated Pick Up Date

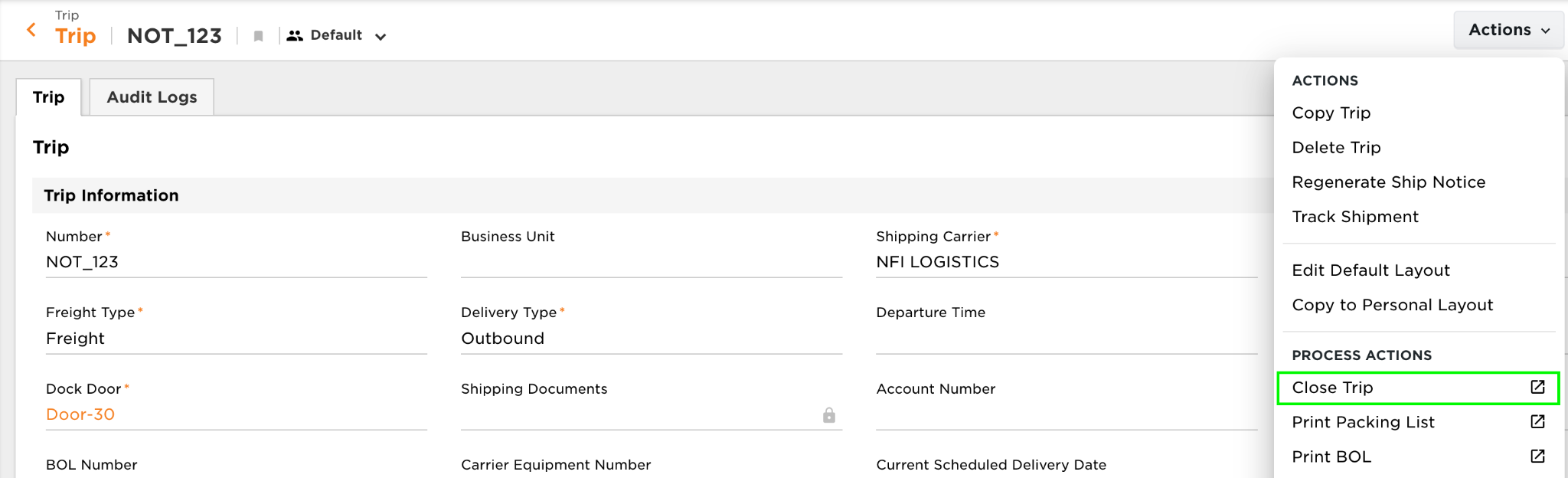
Before the floor users begin loading the pallets onto the trip, the trip management team must update the trip status from ‘Planned’ to ‘Open’ and assign the trip to a shipping door. To note, only one open trip can be assigned to a dock door at a time.

The floor user is prompted to first scan the staging location where their pallets reside. If a trip has pallets in multiple locations, the operations team needs to consolidate the pallets to a single location prior to beginning the AssignTripToContainer process. There is a validation to force the user to scan pallets assigned to the order to the trip. If a user attempts to scan a pallet that is not associated with one of the orders on the trip, an error message is displayed.

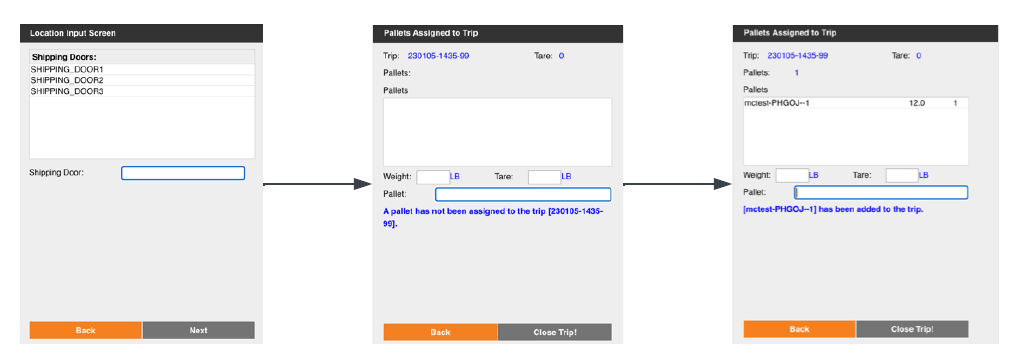
The user selects the shipping door to be used for the trip and begins scanning outbound pallets to assign each of them to the trip. After all pallets have been scanned onto the trip, the user is not taken to the Custom Details Input Screen to enter routing information such as the SCAC code and PRO number. This is done by the back office team.

The trip is closed using a soft close process within the handheld. A soft close leaves the trip open and the shipment is not marked as shipped. If the BOL requires updated information, the respective fields can be updated on the trip or shipment entity within the UI. At this point, a packing slip can be generated by the floor team based on the pallets assigned to the trip. This packing slip is available via a Trip-level PAL. An additional packing slip per order is printed at the PAL on the OrderHeader.

Once all outbound pallets have been physically loaded onto the truck, a PAL on the Trip is used to hard close the trip. At this point, the trip is marked as closed and the shipment is marked as shipped.



#### Screen Flow



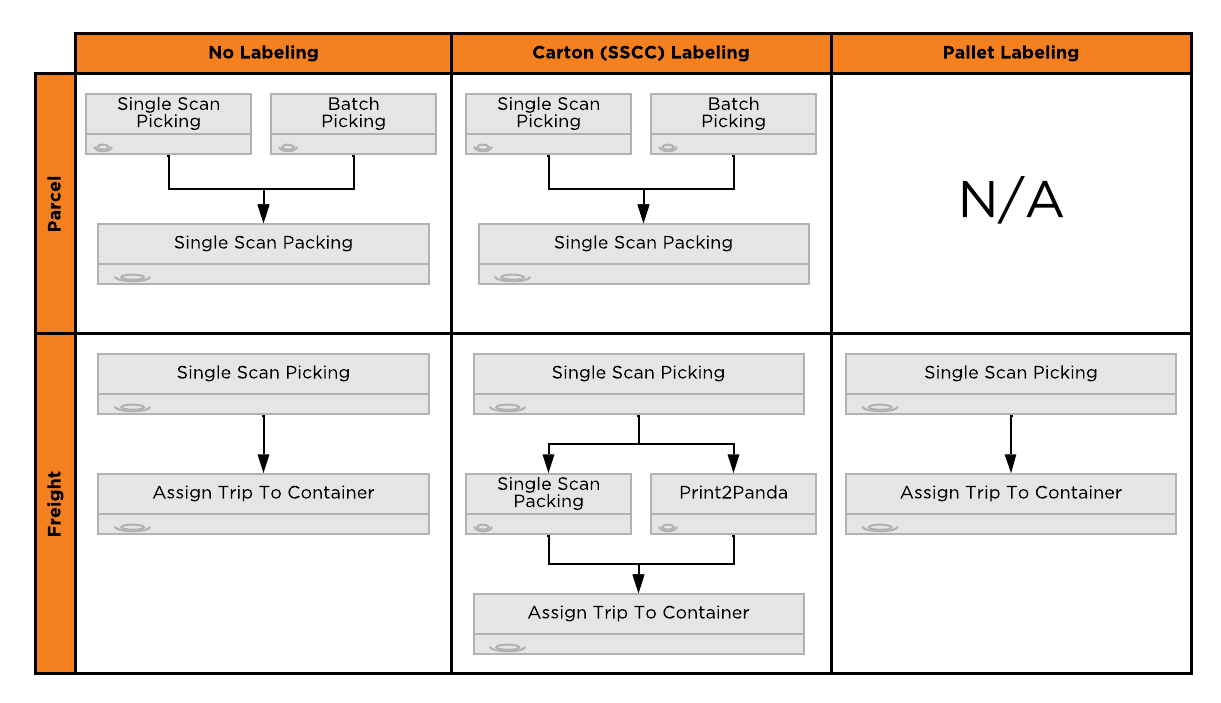
### Key Assumptions

* Freight orders are picked directly onto an outbound pallet.
* Trips can be generated upfront, prior to the order releasing onto the floor, or after picking.
* Each Order must have a trip
* Multiple orders can be assigned to the same trip.
* A validation exists to prevent users from scanning a container onto a trip that is not associated with an order assigned to the trip.
* Multiple orders can be waved and picked together if they are going to the same DC. The DC Number field is TBD by the Patterson Team.
* Freight orders capture serial numbers during picking
* Freight orders do not capture serial numbers during packing.
* BOLs are printed at the Order level and/or the Trip level
  + Notes are set on the trading partner that are to print on the generic BOL
* BOLs can be printed before picking (estimations), after picking, and after Trip Closure
* After Soft Closure the Trip’s status is updated to “Loaded”
* Trips cannot be closed unless the following information is populated:
  + PRO Number
  + BOL Number
  + SCAC Code
* BOLs and Packing Slips are printed by CSR’s via PAL’s

## Retail Requirements

Retail requirements are based on the trading partner assigned to the sales order. Specific fields on the trading partner control the printing of carton labels and packing slips.

The below matrix shows the different workflow processes required based on the sale order’s ship method (y axis) and trading partner’s labeling requirements (x axis).



### Carton Labels

A generic pallet and carton label is configured by default.

The operations team is responsible for providing sample carton labels for each retailer and verifying the format in the WMS is retailer-compliant.

### Pallet Placards

A generic pallet placard will be printed during the Print and Apply process. This report is defined on the Trading Partner.

### Print and Apply

The Print2Panda process is used to print retailer-compliant carton labels. This process is only needed if the retailer for an order has specific carton labeling requirements, indicated by the tradingPartner.ssccNumbersRequiredForTrip field.

This process is initiated by the user scanning the pallet LPN. The contents of the pallet LPN is evaluated and then divided into inner containers based on the quantity of the case pack for each item. For example, if the case pack quantity for an item is 50, and 100 units exist on the pallet, the 100 units are split into two cartons (inner containers) of 50 units each. The inner container numbers are formatted based on SSCC requirements using the tradingPartner.ucceanCompanyNumber.

# 8. Billing

Third-party logistics (3PL) providers capture billable activities that are performed for tenant companies in the Deposco system, calculate the cost for those activities, and then generate invoices to send to the tenant companies.

Each tenant is defined as a company, or business unit (BU), in the Deposco system. To capture billing activity for a tenant, Patterson Warehouses creates a billing profile for the tenant. Within the billing profile multiple billing config records are added, each representing a unique billable activity. Billing Configs include the cost or charge for each activity and the frequency at which data is captured for that activity.

Billing schedulers run daily to gather data for activities that are charged to tenants based on the billing config records in the billing profiles for tenants. The data, which includes the quantity and cost of the activities that are performed, is stored as a set of billable transactions in the Deposco system.

In some cases, Patterson Warehouses performs **AdHocBilling** to generate billable transactions for work that is manually tracked at completion so that it can be billed to a tenant.

After billable transactions are created, the transactions can be grouped and used to generate invoices that are sent to tenants.

**Key Assumptions**

* Billing Invoices will be generated within Deposco and manually exported to Netsuite
* PAT will add all prices and determine frequency of transaction creation
* Below configs are only what is in scope for Phase One of implementation.

## 8.2. Scheduler-Based Billing Configs

### Billing Charges

Patterson Warehouses will be utilizing the following billing configs:

| **Name** | **Description** | **Logic** | **Type** | **Frequency** | **Generated By** |
| --- | --- | --- | --- | --- | --- |
| Pallet Putaway | Cost per LPN received/created on the receipt line and putaway | Deposco will count the total number of LPN’s received/created per Purchase order/blind receipt and create a billable transaction per purchase order/blind receipt | Pallet Putaway | Daily | Scheduler |
| Container Inbound - 20’ | Cost per 20 foot container received | User will type in the footage of the container (20) on as a text field during receiving process which will be stamped onto orderheader and Deposco will bill per purchase received off the 20 foot container | Container Inbound - 20’ | Daily | Scheduler |
| Container Inbound - 40’ | Cost per 40 foot container received. | User will type in the footage of the container (40) on as a text field during receiving process which will be stamped onto orderheader and Deposco will bill per purchase received off the 40 foot container | 40 Ft Container Inbound Fee | Daily | Scheduler |
| Container Inbound - 3,001 CS+ | Cost per container where the receiving user has received >= 3,001 cases. Charge per container received and not by purchase order | If the container received exceeds 3,000 cases then Deposco will create a billing transaction based on the container received, not by purchase order | 3001 Plus Case Fee | Daily | Scheduler |
| Full Pallet Out | Billing per full pallet picked/shipped | Checking shipment line pack type that was shipped to determine if the pack type picked/shipped was indeed a pallet | Full Pallet Out | Daily | Scheduler |
| Loose Case Out | Billing per loose case picked | Checking shipment line pack type that was shipped to determine if the pack type picked/shipped was a loose case | Loose Case Out | Daily | Scheduler |
| Inner Pack Out | Billing per inner pack picked | Checking shipment line pack type that was shipped to determine if the pack type picked/shipped was an inner pack | Inner Pack Out | Daily | Scheduler |
| Each Out | Billing per each picked | Checking shipment line pack type that was shipped to determine if the pack type picked/shipped was an each and [pack.name](http://pack.name) != “Case” | Each Out | Daily | Scheduler |
| Kit Assembly | Cost per kit to order kit picked | If customer orders 2 bicycles which require picking components then Deposco will bill based on price per kit \* orderline.shippedPackQuantity. | Kit Assembly | Daily | Scheduler |
| Small Parcel Order Management | Price per small parcel order shipped | If the shipping service tied to the ship via on the order has a ship vendor code that is NOT “Ship Outside System” then we will consider this as small parcel and bill per order shipped. | Small Parcel Order Management | Daily | Scheduler |
| Returns Processing Per Return | Cost per customer return or blind rma received | Will bill based on customer return or blind rma received at the order level. | Returns Processing Per Return | Daily | Scheduler |
| Labels | Billing per UCC Label Printed for an order | If an order requires 17 ucc labels printed we will check the “Printed” flag on the associated containers, sum up the number of containers per order and create a billing transaction on a per order basis. | Outbound Label Fee | Daily | Scheduler |
| Shrink Wrap | Charge per full pallets picked and built pallets | Checking shipment lines if the pack type = “Pallet” and checking shipped container type for “Pallet” for built pallets before shipping. | Shrink Wrap Outbound Fee | Daily | Scheduler |

## 8.3. AdHoc Billing Billing Configs

### AdHoc Billing Charges

PAT team will manage the creation and deduction of ad hoc charges in Deposco. There are no adhoc billing charges currently expected for Phase one of go-live.

# 9.0 Integrations

## UPS

PAT will use base Deposco UPS integration. Any small UPS request changes will be handled using the Deposco Rules Engine or shipping options.

## FedEx

PAT will use base Deposco UPS integration. Any small Fedex request changes will be handled using the Deposco Rules Engine or shipping options.

## Stamps.com

PAT will use base Deposco Stamps.com integration. Any small stamps request changes will be handled using the Deposco Rules Engine.

## Rest API Integrations

Deposco commonly integrates with homegrown applications as well as integration platform (iPaaS) solutions to transmit customer data with Deposco.

Each RESTful web service call for 2.0 uses HTTP OAuth2 authentication to authorize the transaction. There is no session management or state maintenance on the server.

Create a Deposco user account that can be used to access the API. The account must have company permissions to the business units (companies) that are associated with the data they are accessing and managing through the API.

An application must be created and approved in the Deposco Developer Portal and installed in the Deposco customer environment. A refresh token is then generated for the application in that environment. Using the refresh token, an access token must be requested. The access token is used as a bearer token in the 'Authorization' header of each request. The access token is valid for one hour. When it expires, use the refresh token to request a new access token. Refresh tokens are valid until the application is updated in the developer portal.

The step to request an access token is outlined in the Deposco Developer Portal under [Authentication and Authorization.](https://developer.deposco.com/docs/reference#tag/Authentication-and-authorization)

All needed information for schemas, data structures, and example payloads can be found in the [Deposco Developer Portal](https://developer.deposco.com)

URLs:

The URL to access the Deposco API 2.0 is: https://api.deposco.com/version/

Version is one of the following values:

* The API version number, such as v2.0. Only the grade number and major version number must be provided. In the example of v2.0, the grade number is 2 and the major version number is 0.
* latest, which uses the most recent stable version of the API. This prevents you from having to update the version number on an ongoing basis; however, breaking changes are introduced when the major version number is incremented.
* beta, which uses the current beta version of the API. The beta version may include recently developed API features that have not yet been fully tested.

NOTE: In previous versions of the Deposco API, different URLs were used to access the user acceptance (UA) and production environments. Starting with version 2.0 of the API, the access token in the request is used to direct the request to a specific environment.

Example URLs

Use the endpoint below to create or update items with API version 2.0:

* https://api.deposco.com/v2.0/items

Use the endpoint below to create or update items with the most recent stable version of the API:

* https://api.deposco.com/latest/items

Use the endpoint below to create or update items with the current beta version of the API:

* https://api.deposco.com/beta/items

For more information regarding the Deposco API 2.0 and best practices please refer to the Deposco Developer Portal: <https://developer.deposco.com/>

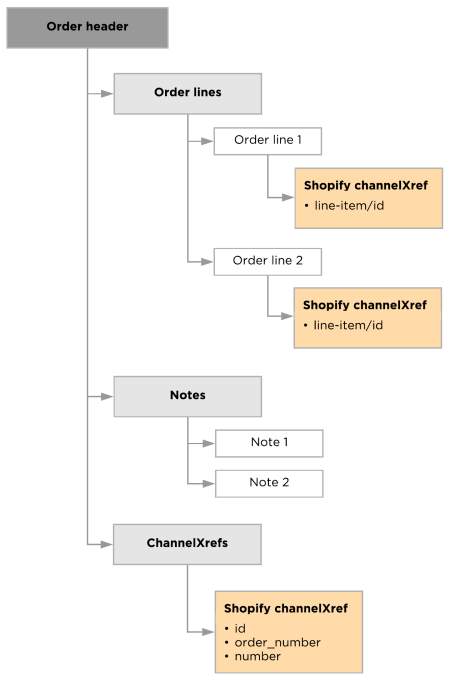
### Deposco Integration Setup

An Integration (IntegrationPoint) can be set up in the Integrations UI. An integration point record exists for each distinct external system and business unit combination. As each new business unit is onboarded, a new integration record is required.

The presence of an integration point record allows Deposco to create a relationship between the transactional data and the Integration record. This is used for traceability purposes.

### Channel Cross-References

If an entity in Deposco (other than an item) is integrated with an external system, then a channel cross-reference (channelXref) is created for the entity and associated integration.

The channelXref provides information about the data in the external system, such as the ID for an entity in the external system, and provides a link between the entity in Deposco and the integration that provides a connection to the external system. The data in a channelXref depends on the entity and integration.

The “channels” field needs to be included in all inbound messages.

More details can be found in the Deposco Developer Portal here: [Channel listings and channel cross-references](https://developer.deposco.com/guides/?page=Channel%20listings%20and%20channel%20cross-references)

Sample Payload:

| "channels": {  "channel": [  {  "feedName": "CINTAP\_HKC",  "jobName": "CINTAP\_HKC-{interface}",  "ref1": "{internalId}"  }  ]  } |
| --- |

### Channel Listings

If an item in Deposco is integrated with an external system, then a channel listing is created for the item. The channel listing provides information about the item in the external system, such as the item number and internal IDs in the external system, and provides a link between the item in Deposco and the integration that enables the connection with the external system.

In a multi-channel environment, an item may be defined in multiple systems, and therefore may have multiple channel listings in Deposco, as shown in the example illustration.

More details can be found in the Deposco Developer Portal here: [Channel listings and channel cross-references](https://developer.deposco.com/guides/?page=Channel%20listings%20and%20channel%20cross-references)

Sample Payload:

| {  "number": "S456",  "businessUnit": {  "businessKey": {  "code": "ABC"  }  },  "channels": [  {  "integration": {  "businessKey": {  "name": "CINTAP\_HKC"  }  },  "listingStatus": "Linked",  "saleable": "true",  "packQuantity": "1",  "ref1": "1234567890"  }  ] } |
| --- |

### 

### Webhook

A webhook event subscriber is available to configure subscriptions to the following Deposco events:

* shipNotice

The following table lists the key fields that are available in the webhook API. A full list of available webhook API fields is provided in the Webhook.xsd schema.

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | xs:long | Internally generated identification  number for the webhook. |
| events/event | xs:string | Name of the Deposco event that that  initiates the webhook call. A single  event is supported for each webhook. |
| callbackUrl | xs:string | Destination URL to which the  webhook subscription should send the POST request when an event  occurs. |
| secretKey | xs:string | Internally generated secret key for the  webhook. The secret key and an  HMAC header are sent in the HTTP  header on every POST request that is  sent to the specified callback URL. To  validate that the request was sent  from Deposco, the API that receives  the request should calculate its own  hex-encoded HMAC-SHA256 digest  using the shared secret and request  body. If the result matches what was  sent in the HTTP header, then the  request came from Deposco.  See Create a webhook subscription  for an example Python 3 function for  verifying an HMAC digest from a  Deposco webhook request with the  secret key for the webhook and the  HTTP request body as a string. |
| includeBusinessUnits | xs:boolean | Used in 3PL environments to  determine whether the webhook  includes events for only the primary  company (3PL), only a specific tenant  company, or for both the primary  company and all tenant companies.  When creating the webhook:   * To include events only for the primary company, set includeBusinessUnits to false and then specify the primary company as the business unit in the URL. This is the default value. * To include events only for a specific business unit, set includeBusinessUnits to true and then specify the tenant company as the business unit in the URL. * To include events for both the primary company and all tenant companies, to include events that occur for tenant companies, set includeBusinessUnits to true and then specify the primary company as the business unit in the URL. |
| createdDateTime | xs:dateTime | Date and timestamp for when the webhook was created. |
| updatedDateTime | xs:dateTime | Date and timestamp for when the webhook was last updated. |

#### Verify the webhook

An HMAC header and secret key are generated and provided in the response when you create a webhook. These values are sent in the HTTP header on every POST request that is sent to the specified callback URL. To validate that the request was sent from Deposco, the API that receives the request should calculate its own hex-encoded HMAC-SHA256 digest using the shared secret and request body. If the result matches what was sent in the HTTP header, then the request came from Deposco. The following Python 3 function provides an example for verifying an HMAC digest from a Deposco webhook request with the secret key for the webhook and the HTTP request body as a string.

Sample Script:

| **import** hmac **import** hashlib WEBHOOK\_SECRET\_KEY = '56E4BCED2648D4F54C201C3911DFE93B0BBBF181DC2C496A774D70582AEA2EA1' HMAC\_TO\_VERIFY = '5212af0f4aecb7b6fd8ed82e7b73c2f9e2ccb8160a5deaca941d12634516094f' REQUEST\_BODY = 'test' **def** **verify\_webhook**(data, hmac\_header):  digest = hmac.new( bytes.fromhex(WEBHOOK\_SECRET\_KEY), data.encode('utf-8'), digestmod=hashlib.sha256).digest()  computed\_hmac = digest.hex()  **return** hmac.compare\_digest(computed\_hmac.encode('utf-8'), hmac\_header.encode('utf-8') ) print(verify\_webhook(REQUEST\_BODY, HMAC\_TO\_VERIFY)) *# True* |
| --- |

### Master Data Interfaces

All master data interfaces leverage Deposco’s REST API 2.0. The client performs POST and GET requests to the following Deposco REST API 2.0 endpoints: Items API and Trading Partners API. The Master Data Interfaces serve primarily to create and update item/SKU information as well as create and update customer/vendor information. In addition, if the warehouse performs an update to an item’s pack unit of measure, this will be communicated to the external system.

### Item Creation

Patterson will leverage Deposco’s REST API 2.0 to send POST requests to the Item API endpoint. The payload will include all item level details for respective attributes.

* HTTP Method: POST
* Direction: Inbound to Deposco
* Authentication: Oauth2
* Token URL: https://auth.deposco.com/oauth2/token
* Request URL: <https://api.deposco.com/v2.0/items>

#### Key Assumptions

* Channel Listings and Channel Cross-References (channelXrefs) are required in all inbound messages.
* Error handling will be owned by the Patterson team as Deposco’s error response will be provided back

#### User Stories

*As an API user, I want the ability to…*

* Create a new Item.
* Update information on an existing Item.

### Item Update

Patterson will leverage Deposco’s REST API 2.0 to send GET requests to the Item API endpoint. The payload will include all item, pack, and UPC level details for respective attributes.

* HTTP Method: GET
* Direction: Outbound from Deposco
* Authentication: Oauth2
* Token URL: <https://auth.deposco.com/oauth2/token>
* Parameters
  + updatedAfter
* Request URL: <https://api.deposco.com/v2.0/items/?updatedAfter=2024-10-01T14:10:50-05:00>

#### Key Assumptions

* The Item Update interface will primarily use the Updated Date parameter to retrieve updated items.

### Trading Partner

Patterson will leverage Deposco’s REST API 2.0 to send POST requests to the Trading Partner API endpoint. The payload will include all item level details for respective attributes.

* HTTP Method: POST
* Direction: Inbound to Deposco
* Authentication: Oauth2
* Token URL: https://auth.deposco.com/oauth2/token
* Request URL: <https://api.deposco.com/v2.0/tradingPartners>

#### Key Assumptions

* Channel Listings and Channel Cross-References (channelXrefs) are required in all inbound messages.
* Error handling will be owned by the Patterson team as Deposco’s error response will be provided back.

### Inbound Process Interfaces

Inbound process interfaces will leverage Deposco’s REST API 2.0. The client performs POST requests to the following Deposco REST API 2.0 endpoints: Purchase Order API and Customer Return API. The client performs GET requests to the following Deposco REST API 2.0: Receipts API. The Inbound Process Interfaces serve primarily to communicate warehouse inbound process messages between Deposco and external systems.

### Purchase Order

Patterson will leverage Deposco’s REST API 2.0 to send POST requests to the Purchase Order API endpoint. The payload will include all purchase order level details for respective attributes.

* HTTP Method: POST
* Direction: Inbound to Deposco
* Authentication: Oauth2
* Token URL: https://auth.deposco.com/oauth2/token
* Request URL: [https://api.deposco.com/v2.0/orders/purchaseOrders](https://api.deposco.com/beta/orders/purchaseOrders)

#### Key Assumptions

* Channel Listings and Channel Cross-References (channelXrefs) are required in all inbound messages.
* Error handling will be owned by the Patterson team as Deposco’s error response will be provided back.
* There will be one order line per distinct SKU/item unless an inventory attribute such as Lot Number or LPN is specified. If an inventory attribute is specified, then there can be multiple order lines for a single SKU/item

#### User Stories

*As an API user, I want the ability to…*

* Create a new Purchase Order.
* Update header-level information on an existing Purchase Order that has not yet been received.
* Update the quantity on a Purchase Order line that has not yet been received.
* Cancel a Purchase Order line that has not yet been received.
* Cancel a Purchase Order that has not yet been received

### Receipt Confirm

Patterson will leverage Deposco’s REST API 2.0 to send GET requests to the Receipts API endpoint. The payload will include all receipt line level details for respective attributes.

* HTTP Method: GET
* Direction: Outbound from Deposco
* Authentication: Oauth2
* Token URL: <https://auth.deposco.com/oauth2/token>
* Parameters
  + businessUnit
  + updatedAfter
* Request URL: <https://api.deposco.com/v2.0/receipts/?businessUnit=KALMAR&updatedAfter=2024-10-01T14:10:50-05:00>

#### 

#### User Stories

*As an API user, I need to…*

* Retrieve receipt information, including quantity, item, container, location, damaged quantity.
* Retrieve receipt information for a specific date range

### Outbound Process Interfaces

Outbound process interfaces will leverage Deposco’s REST API 2.0 and Deposco’s Webhook Event Subscriber. The client performs POST requests to the following Deposco REST API 2.0 endpoints: Customer Order API. The client will receive a POST request from Deposco’s webhook subscription. The Outbound Process Interfaces serve primarily to communicate customer orders that require fulfillment. Once the warehouse fulfills an order, the ship notice sends an event notification to Patterson.

### Customer Order

Patterson will leverage Deposco’s REST API 2.0 to send POST requests to the Customer Order API endpoint. The payload will include all item level details for respective attributes.

* HTTP Method: POST
* Direction: Inbound to Deposco
* Authentication: Oauth2
* Token URL: <https://auth.deposco.com/oauth2/token>
* Request URL: <https://api.deposco.com/v2.0/customerOrders>

The same request URL and payload is used to create, update, and cancel a Customer Order.

1. To create a Customer Order, set the status in the payload to “New”.
2. To update a Customer Order, keep the status in the payload as “New” and update the header and line level fields. To note, a customer order can only be updated if it has not yet been released to the WMS.
3. To cancel a Customer Order, set the header and/or line status in the payload to “Canceled”, depending on the cancellation.

Customer Order requests are processed asynchronously. In order to check the status of the message, a subsequent API request is required. The initial response from the POST request of the customer order will provide asynchronous request id.

* HTTP Method: GET
* Direction: Outbound from Deposco
* Authentication: Oauth2
* Token URL: <https://auth.deposco.com/oauth2/token>
* Request URL: [https://api.deposco.com/latest/async/{requestId](https://api.deposco.com/latest/async/%7BrequestId)}

Sample payload of Customer Order Response:

| {  "request": {  "id": "1a87477a-ec90-4d74-9e20-9f31b1144271"  } } |
| --- |

Sample payload of Asynchronous request Response:

| {  "results": [  {  "status": "OK",  "id": 5,  "businessKey": {  "externalOrderNumber": "SHorder3",  "businessUnit.code": "HKC",  "tradingPartner.id": 6  }  }  ] } |
| --- |

#### Key Assumptions

* Channel Listings and Channel Cross-References (channelXrefs) are required in all inbound messages.
* Error handling will be owned by the Patterson team as Deposco’s error response will be provided back
  + If there is a data issue in the payload (e.g. an item has not been created/does not exist in the system), the Customer Order is created and put in Review status. The customer is responsible for correcting the data and order after it has been created.

#### User Stories

*As an API user, I want the ability to…*

* Create a new Customer Order.
* Update header-level information on an existing Customer Order.
* Update line-level information on an existing Customer Order.
* Fully cancel an existing Customer Order.
* Partially cancel an existing Customer Orderline’s quantity.

### Ship Notice

Patterson will leverage Deposco’s Webhook Event Subscriber to receive POST requests from Deposco’s webhook subscription. Once the warehouse fulfills an order, the Deposco application triggers a request back to the upstream application based on the event of the order’s fulfillment. Shipment messages, referred to in Deposco as ship notices, are sent using a webhook communication method. The payload will include all shipment level details for respective attributes.

* HTTP Method: POST
* Direction: Outbound from Deposco
* Authentication: HMAC

Deposco uses an HMAC shared secret. This is used to generate an HTTP header on every POST request that is sent to the callback URL so that the user knows the request originates from Deposco. The value of the header is a base64-encoded HMAC-SHA256 digest using the shared secret and body of the request. To ensure the request is sent from Deposco, the API receiving the request calculates their own base64-encoded HMAC-SHA256 digest using the shared secret and request body. If the result matches what was sent in the HTTP header, then the request came from Deposco.

ShipNotice event transactions are stored in the EventLog table. When a webhook is triggered for a shipNotice event, a record in the EventSubscriptionLog table is created. The “status” field on the EventSubscriptionLog table informs the user as to whether the webhook API request is successful or not.

Deposco has a mechanism (scheduler task, *Reprocess Failed Events*) in place to retry shipNotice EventSubscriptionLogs that meet the following criteria:

* The event is in a status of “Failed”.
* The event has been retried less than 4 times.
* The event was created within the last 2 hours.

#### Key Assumptions

* ShipNotice is the only event-based message to be used
* Error handling will be owned by the Patterson team as Deposco’s error response will be provided back
* Additional fields cannot be added to the shipNotice payload
* Functional integration mapping is outside of the scope of this document and will be handled by the customer team at a later date

#### User Stories

*As an API user, I want the ability to…*

* Retrieve customer order information.
  + The “references” object is available at the header and line-level in the shipNotice payload.
* Retrieve shipment information, including quantity, tracking number, carrier, service.
* Retrieve carton information, including item, quantity, container number, tracking number.

# Reporting

## Process Reports

Patterson Warehouses will be using Deposco base process report:

* Packing Slip
* Commercial Invoice
* Vics BOL
* UCC/SSCC
* LPN Labels
* Pallet Placard
* Pick ticket (freight orders only)

**Operational Reports**

Patterson Warehouses will use Deposco base operational reporting to replace the reports that are being used currently. Any changes to base reports needed will be scoped executed against the estimated reporting hours allocation.

Below is a list of Patterson Warehouses existing report along with Deposco’s potential template report that will fit the need (The reports will not be exactly the same but will consist of similar data records that can be tweaked to show more information if needed):

Patterson Warehouses and customers will use the following operational reports:

* [Scheduled Email](https://docs.deposco.com/help/bright-performance/report-builder/schedule-a-company-report-to-be-sent-by-email) Reports
  + On hand and ATP quantity stock reports
  + Shipped Orders per day
  + Daily item snapshot
  + Cycle count variance and adjustment reports daily
* [Report Builder](https://docs.deposco.com/help/bright-performance/report-builder/create-a-company-report) Reports
  + Consolidation report for consolidating partial pallets
  + Kit history report showing components shipped as well as kits shipped
  + Billing transaction report
  + Cube Report for routing
* Dashboard
  + Open order widget
  + Shipped order widget
  + Low inventory widget
* Pentaho/pdf Reports
  + Warehouse Receipt
  + Packing Slip
  + Pick ticket

# UI Requirements

This section outlines the UI functional requirements that have been submitted to the Deposco base product team. These UI features are descriptions of future state UI functionality. The actual functionality once delivered may differ from what is stated in this section. However, the overall theme should be similar.

## External/Internal Images & Notes

Entity images and entity notes can be marked as external/internal to designate if the entity record will be visible to the customer users.

**Restricted Entity Layouts**

Entity layouts can be restricted to control what a customer user can see.

**Hiding Audit Logs**

Patterson Warehouses customers will not be able to see audit logs. This will be a global permission for all Patterson Warehouses customers.

**Editing Orders**

If an order is not in “New” status, the 3PL tenants should not be updating information on the orders. It is recommended they have a read only permission.

# 

# Customer Agreement

Your signature is required as proof of understanding and acceptance of the design document as described above. Upon acknowledgement of your approval, Deposco will engage in development of the work stated in this document and the design specification. Please review this document in its entirety. Upon approval, please sign, date and return.

| **Deposco, Inc.** 11605 Haynes Bridge Road Suite 200 Alpharetta, GA 30009  By: | |
| --- | --- |
| Name: |  |
| Title: |  |
| Date: |  |

| **Patterson Warehouses** 5388 Airways Blvd. Memphis, TN 38116  By: | |
| --- | --- |
| Name: |  |
| Title: |  |
| Date: |  |