



OrderFlow Operational Functions

OrderFlow Ltd.

Document Version: 4.2.4

Document Built: 2024-02-16

This document and its content is copyright of OrderFlow Ltd. All rights reserved.
You may not, except with our express written permission, distribute, publish or commercially exploit the content.
Any reproduction of part or all of the contents in any form is prohibited.

Introduction

OrderFlow is an enterprise-strength order processing and warehouse management system (WMS) with a wealth of configuration options and implementation possibilities.

This document provides details of the main functions that it performs, from an operational viewpoint.

Audience

The intended audience of this document is:

- business leaders who are considering OrderFlow as a solution to the needs of their organisation
- potential users and administrators of the OrderFlow system
- designers and developers of the OrderFlow system
- those responsible for supporting and deploying OrderFlow

OrderFlow Operations

OrderFlow is built specifically around the needs of eCommerce fulfilment. It enables an organisation to manage one or more warehouses, fulfilling customer orders accurately and efficiently. Users are able to complete all the warehouse and despatch operations necessary to support the fulfillment process in a way that has been optimised to meet the particular demands of the business in which it is being used.

This document details *how* OrderFlow manages these operations, from the point of view of the different users of the system.

It covers the following broad areas:

- Warehouse Operations:
 - Goods-in
 - Putaway
 - Replenishment
 - Stock Checking
 - Returns
 - Manufacturing
- Despatch Operations
 - Order Import
 - Stock Assignment
 - Picking
 - Packing
 - End of Day
- Management Operations
 - Monitoring
 - Reporting

Stock Management

Adding Stock

There are two ways that stock can be added into the warehouse on OrderFlow. An initial stock position can be imported or manually entered. Once a system is operational, stock is usually received into the warehouse through deliveries. The details for this are covered in the [Goods In](#) section of the [Warehouse Operations](#) chapter.

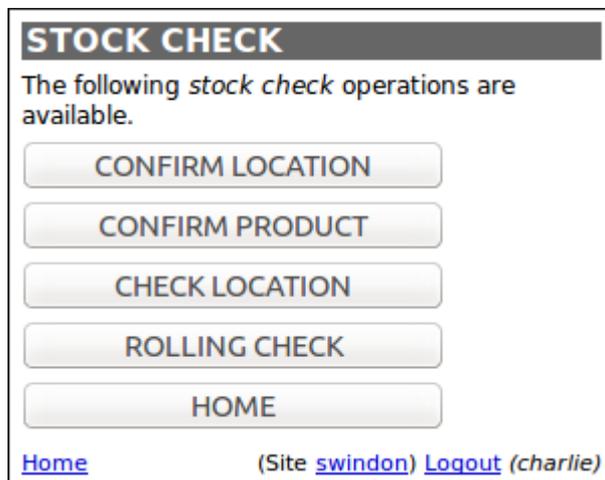
Stock Checking

Although OrderFlow keeps a tight control of the location of every unit of stock in a warehouse (or in several warehouses), it can happen that the real-world situation can fall out of step with OrderFlow's view of it. This is most likely to be because stock is physically moved without following direction from OrderFlow, or without reflecting that move in OrderFlow.

To re-align OrderFlow's picture of a warehouse's stock position, a stock check mechanism is required. This can be a one-off confirmation of the number of units of a specific product in a specific location. Or it could be assigning a specific use to a request to visit multiple locations and scan/confirm the number of units of each product found in those locations.

Handheld Stock Checking

By its nature, stock checking is best performed on a handheld device, so OrderFlow provides most of its stock-checking functionality through its handheld interface:



The following sections detail each of these stock-checking functions.

CONFIRM LOCATION

The **Confirm Location** handheld function allows a user to scan a location, after which they are presented with the stock position in that location:

STOCK CHECK

[Home](#) > *Location Confirm*

Location: **location_008**
Type: Pickable

Stock quantity (with last confirm date):

45	MicroUsbBlackberry	22/02/16 15:11
65	bluetoothhead	22/02/16 15:12

[Product not listed?](#)

Note: **2** product(s) not yet confirmed in this stock check.
Use CONFIRM LOCATION STOCK button to confirm all quantities.

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

Stock Checking

The user can then choose to adjust any stock quantities as required, or confirm any unconfirmed quantities. The ability to add products that are physically present in the location, but OrderFlow thinks aren't, is available via the *Product not listed?* link.

Once the user is confident that OrderFlow accurately reflects the stock in the select location, they can confirm this on the confirmation screen. This gives them the option to scan more products if necessary, or to confirm that the stock position presented is complete:

STOCK CHECK

[Home](#) > *Location Check*

Press **Complete** if everything in **location_008** has been confirmed.

Press **Scan More** to scan other products in this location.

Enter **Cancel** to abandon check.

Products **not confirmed** since stock check started on 22/02/2016 16:53:54 (last confirm date shown):

65	bluetoothhead	22/02/16 15:12
-----------	----------------------	-------------------

Products confirmed during the current stock check:

33	MicroUsbBlackberry	22/02/16 17:41
-----------	---------------------------	-------------------

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

OrderFlow then requests a final confirmation that any stock that OrderFlow *thought* was in the location is definitely *not* in the location:

STOCK CHECK

[Home](#) > *Location Check*

Press **Zero Unscanned Stock** to confirm none of the unscanned **location_008** stock is present.

ZERO UNSCANNED STOCK

Stock that will be **set to zero** on completion of this stock check:

65	bluetoothhead	22/02/16 15:12
-----------	----------------------	-------------------

Enter **Cancel** to abandon **location check** from location **location_008**.

CANCEL

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

Once a location has been confirmed, the user can move onto the next location of their choosing. This is very much a user-directed stock checking operation, so there is not much scope to intelligently target specific locations. If this type of stock-checking is required then a *rolling check* would be required.

CONFIRM PRODUCT

The **Confirm Product** handheld function allows a user to scan a product, after which they are presented with the stock position for that product in each of the locations where it is held:



Again, the user can then choose to adjust any stock quantities as required, or confirm any unconfirmed quantities.

Once the stock for a product has been confirmed, the user can move onto the next product of their choosing. This operation is a product-driven one, and again relies on the user deciding which product(s) to target.

CHECK LOCATION

The **Check Location** handheld function allows a user to scan a location, after which they can scan products and confirm the quantities in that location.

This is effectively a 'blind' stock check, where a user blindly scans the product(s) and enters the quantity of each product, without any prior knowledge of what OrderFlow thinks is in the location.

Again, similarly to the *Confirm Location* operation, once a location has been checked, the user can move onto the next location of their choosing. The *rolling check* can be used if the user needs to be directed to specific locations.

ROLLING CHECK

The **Rolling Check** handheld function is a stock-checking operation that is driven by OrderFlow, in that OrderFlow determines which locations need to be checked, and in which order.

The benefit of the rolling stock check is that, not only do the "oldest" locations get checked soonest, but also multiple users can perform rolling stock checks at the same time, *in the same area*.

The user can select the area in which they would like to perform a rolling stock check:

SELECT AREA

[Home](#) > *Rolling Check*

Choose **area** for stock checks.

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

They will then be presented with the next location to be checked in the selected area. This location is derived from running the "Oldest stock check location" report.

This report uses the **last_stock_check** and **location** tables to return the location that has not been checked for the longest time (or has never been checked at all), in the selected area. If more than one location satisfies this condition, then the locations are ordered by sort indicator then reference.

STOCK CHECK

[Home](#) > *Rolling Check*

Next location to be checked:

Location: **location_10**

Area: **area_1**

Most recent check: *No records present.*

Products with stock: **1** [View Products](#)

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

Stock Checking

As with the other types of stock check, the user can then scan products and confirm quantities in the location. If the quantity entered differs from the value that OrderFlow currently holds for that product/location combination, the user is required to confirm this difference:

CORRECT STOCK

[Home](#) > [Rolling Check](#) > [Location Check](#) > [Check Product](#) > [Correct](#)

The currently recorded quantity for location **location_10** and product **bluetoothhead** is **712** .
Please confirm the actual quantity is **710**.

710

[Home](#) (Site [swindon](#)) [Logout](#) (*charlie*)

External Stock Checking

When a comprehensive stock check of the entire warehouse is required, such as an annual stock check, some organisations prefer to perform this "off-platform", i.e. without using OrderFlow. The reasons for this may be historical, in that the warehouse staff know exactly how to do it the way they have always done it. Or it may be because they don't have enough handheld devices for all the staff needed to complete the stock check in the time available.

OrderFlow supports this activity by allowing the results of the off-platform stock check to be imported, using a **product location** import. This contains every product / location combination in the warehouse, alongside the quantity of stock for that combination, e.g.:

	A	B	C	D	E
1	externalReference	site	product	location	quantity
2		1 DEFAULT	DVD-RPMK	Location_4	50
3		2 DEFAULT	DVD-RPMK	location_2	100
4		3 DEFAULT	DVD-BLDRNDC	location_1	20
5		4 DEFAULT	DVD-BLDRNDC	Location_5	10

Once a product location spreadsheet is imported (using the *merge* operation), each new stock position is persisted, and any difference between the old and new positions is recorded as a stock change on OrderFlow, with a contextual note to make it clear that the change was as a result of an imported stock position.

Stock Move Tasks

OrderFlow features a generalised framework for managing stock moves within the warehouse. Stock Move Tasks can be set up for a variety of purposes.

An important application of this is in Replenishment, described below.

Replenishment

Replenishment is the term used to describe the process of moving stock from *storage* locations to *pickable* locations within a warehouse.

Storage locations will typically be locations that are less accessible than pickable locations, for example pallet racking that requires a fork-lift or an *aerial work platform* (or "man-up") to reach. Retrieving stock from these locations takes more time and can be more difficult. Additionally, specialist qualifications may be required to use the equipment to reach them.

Pickable locations, on the other hand, will typically be more easily accessible to picking staff.

The timing of when to replenish stock is influenced by several factors, including the capacity of the pickable locations, the 'box size' of the stock, the time it takes to access the replenishment locations, and most of all, the outstanding order line requirement.

OrderFlow supports 'priority' replenishment tasks and 'background' replenishment tasks.

Priority Replenishment will move whatever stock is required by existing shipments from bulk storage locations into picking locations. The scope of priority replenishment tasks is restricted to the items needed to by existing shipments with the state 'pending move'. It allows shipments that already exist within OrderFlow to be progressed

Background Replenishment is used to replenish the picking faces by moving stock out of bulk storage locations. An OrderFlow product definition can be given a 'Picking Threshold' value and a 'Picking Location Max' value. When the stock available in the picking face falls below the 'Picking Threshold' the background quantity available in the pick face up to the value defined in 'Picking Location Max'.

OrderFlow uses its task framework to drive the replenishment process. To manually create a 'background' replenishment task, navigate to *Warehouse --> Tasks --> New* and select the 'Default Background Replenishment' task type. There is a short-cut link to this page on the 'Background replenishment' dashboard fragment, which is present by default on the Tasks dashboard:

Create a background replenishment task to replenish products from *non-pickable storage locations*.

The following screenshot shows a background replenishment that was created in this way. Note the three tabs ('Line Detail' / 'By Product' / 'By List'), that show the constituents of the task from different perspectives.

Stock Move Tasks

Despatch
Inventory
Warehouse
Import
Integration
Reports
Admin
Setup
Advanced

- Purchase Orders
- ASNs
- Deliveries
- Work Orders
- Licence Plates
- Picking
- Tasks
- Dashboard
- Search
- New
- Navigate
- Stock Check
- Returns
- Suppliers

Background Replenishment Task Detail Showing all sites Showing all organisations (inc. test)

⚠ This stock move task is related to an organisation that is in **test** mode, therefore it **may be deleted** in the future.

Task Summary - view summary of this background replenishment task

Print the source picking and target putaway lists for this stock move task. ⓘ

Accept this stock move task and move the stock as described. ⓘ

Line Detail
By Product
By List

Product	Source	Quantity	Target	Quantity
FM1	premium_store_1	30	premium_pick_1	30
FM4	bulk_single_4	30	premium_pick_4	30
FM3	bulk_single_3	30	premium_pick_3	30
FM5	premium_store_5	15	premium_pick_5	15
LT1	bulk_multi_1	20	sd_multipick_1	20
LT2	bulk_multi_2	10	sd_multipick_2	10

© Realtime Despatch Software Ltd 2011-2016
[Return to the top](#)

The task definition that backs this task contains the configuration that dictates *how* OrderFlow creates the background replenishment task. In this case, the task uses the "Incremental Target-centric" strategy. This uses a "background picking replenishment strategy" report to populate a map of products to quantities of each product that requires replenishment. This map is then used to create the required stock move lines in the task, based on available stock in the warehouse.

The difference between *background* and *priority* replenishment tasks is encapsulated in the report that each task definition uses. The background replenishment uses "background picking replenishment strategy" report, the priority replenishment task uses the "priority location picking replenishment strategy" report.

Task configuration is a complex area - more details on this subject can be found in the [Scripting Guide](#).

Task Completion

Task *completion* is the process of carrying out the stock moves identified by the task lines. It can be used for *replenishment* and other tasks.

OrderFlow supports two modes of completion for tasks: *paper-driven* and via a *handheld terminal*.

Desktop-based task completion is achieved through a combination of a paper picking and putaway report, as well as a set of user operations via the desktop user interface to accept the task, print the paperwork, and to record modifications after picking and putaway. It is useful in environments where handheld terminals are not enabled or supported.

The typical workflow used would be as follows:

- the task is created on the system, either manually or on a schedule. Once creation is complete and the task is ready for processing, the task state will move to *ready*.
- an operator accepts the task, moving the state from *ready* to *accepted*.
- the operator will then print paperwork for the task. The printout will include:
 - a list of the source lines, which identify the source locations and products to be picked, in picking order.
 - a list of the corresponding target lines, identifying the putaway locations for the product.
- the operator will then physically carry out the stock moves on the system as identified by the lines of the task. Any lines that cannot be completed, or quantity changes that need to be made, are recorded at this point on the paperwork. At this point, no changes have been reflected on the system.
- The user then *applies* the task. Prior to applying the task, the user can record any modifications from the originally suggested locations and quantities. At this point, the stock changes associated with the task are reflected on the system.

TODO - add screenshot

Handheld terminal-based task completion has the advantage that each move is verified using a physical scan.

TODO - add instructions and screenshot

Warehouse Operations

This chapter describes how incoming stock is received into the warehouse and how customer orders are processed. It also covers other supporting processes usually found in an eCommerce warehouse, some of which are crucial to the smooth running of the fulfilment operation.

Goods-In

The 'Goods-in' operation involves a **Delivery** of stock into the warehouse.

Deliveries are usually planned and expected. An organisation will place a **Purchase Order** with a supplier, detailing exactly what items and quantities are required. Supplier purchase orders can be imported into OrderFlow and used to check-in the associated delivery.

If no purchase order has been imported into OrderFlow, it is still possible to describe the contents of a delivery before it is received by using an **Advanced Shipping Notification (ASN)**. An ASN can be used to specify what is thought to be in a delivery before it arrives.

The difference between a purchase order and an advanced shipping note within OrderFlow is that a supplier purchase order can be associated with multiple deliveries while an advanced shipping note is always used to describe a single delivery.

The OrderFlow Goods-In process will reconcile the details of what received with the associated purchase order or ASN and produce a discrepancy report where there are differences between what was *expected* and what was *received*.

It is also possible to create a delivery on OrderFlow 'on spec', without it be related to either a purchase order or an advanced shipping note. Where this is done no discrepancy report can be produced.

This section details how OrderFlow supports the Goods-in process.

Purchase Order Creation

Purchase orders cannot be created within OrderFlow but are typically created within the accounts environment or by dedicated purchasing systems. Supplier purchase orders can be *imported* into OrderFlow using its import framework. This framework allows supplier purchase order documents, typically in XML, spreadsheet or comma-separated value (CSV) format, to be imported into OrderFlow. (More details of this framework can be found in the **Importing Guide**.)

The import of purchase orders can be driven by a file-based integration with an accounting or Enterprise Resource Planning (ERP) system, or they can be transferred via FTP to a file location readable by OrderFlow. Additionally, the desktop interface to OrderFlow allows a user to submit a file (or the contents of a file) as a purchase order.

In OrderFlow, the result will be a **Purchase Order** entity, attached to which will be one or more **Purchase Order Line** entities. These can be seen under *Warehouse* --> *Purchase Orders* in the desktop interface. The Purchase Order Detail page shows various attributes of the purchase order, such as the supplier and the reference they use to identify the purchase order. It also shows the lines within the purchase order, any associated deliveries that have already been received against it, and the delivery receipts at a line-level:

The screenshot displays the 'Purchase Order Detail' page in the OrderFlow software. The interface includes a top navigation bar with tabs for 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. A left-hand navigation menu lists various modules like 'Purchase Orders', 'Dashboard', 'Shipments', 'ASNs', etc. The main content area is titled 'Purchase Order Detail' and shows details for a purchase order with reference 'po_acme_1'. It includes fields for Site, Organisation, Type, Lines, Supplier, and Supplier Reference. Below this, a table lists the 'Purchase Order Lines' with columns for Reference, Product, Description, Expected, and Outstanding. The table contains three rows: 'acme1_1' for 'ipod5', 'acme1_2' for 'woodworm_zoom', and 'acme1_3' for 'cyclepro_cape'. A section for 'Associated Deliveries' shows one delivery with reference 'acmedel_1'. At the bottom, there is a button to 'New Delivery from Purchase Order ...' and a copyright notice for Realtime Despatch Software Ltd 2011-2016.

A *Purchase Order* can be of the following types:

- **Supplier** - placed against one of the organisation's suppliers.
- **Bulk Transfer** - transfer of stock from another warehouse, within the same organisation.

A *Purchase Order Line* holds an *expected* quantity against a referenced product, plus the *outstanding* quantity of units not yet delivered. It can additionally hold an external reference, typically supplied by an external system.

Advanced Shipping Note Creation

As with purchase orders, Advanced Shipping Notes (ASNs) can be imported into OrderFlow using the import framework, in XML, spreadsheet or CSV format.

Additionally, ASNs can be created using the desktop user interface (under *Warehouse* --> *ASNs* --> *New*).

In OrderFlow, the result will be an **Advanced Shipping Note** entity, containing one or more **Advanced Shipping Note Line** entities.

These can be seen under *Warehouse* → *ASNs* in the desktop interface. The Advanced Shipping Note Detail page shows various attributes of the ASN, such as the supplier and the number of lines it contains. It also shows the lines within the ASN, any associated delivery, and any discrepancies between the ASN and the actual delivery:

The screenshot displays the 'Advanced Shipping Note Detail' page. The top navigation bar includes 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. A sidebar on the left lists various menu items like 'Purchase Orders', 'ASNs', 'Dashboard', 'Search', 'New', 'Deliveries', 'Work Orders', 'Licence Plates', 'Picking', 'Tasks', 'Stock Check', 'Returns', and 'Suppliers'. The main content area shows the following details:

- ASN Details:** Reference: abcd; Supplier: Connect Distribution Services; Actual Delivery: abcd, delivered on 11/02/2016 12:34:23; Site: Default; Organisation: Altco Ltd; Lines: 3; Created: 9 November, 2010 20:49:24; Type: Stock; State: Accepted with discrepancies; Licence Plate(s): Yes.
- ASN Lines:** This advanced shipping note contains the following lines.

Product	Description	Quantity
DVD-RPMK	The Replacement Killers	10
DVD-REDC	Red Corner	5
DVD-BLDRNDC	Black	20
- Discrepancies:** Below is a list of differences between the actual delivery quantity entered and the expected quantity from the originating advanced shipping note.

Product	Expected	Actual	Difference
DVD-BLDRNDC	20	0	Quantity short by (20)
DVD-RPMK	10	0	Quantity short by (10)
DVD-REDC	5	0	Quantity short by (5)
DVD-LTWP	None expected for this product	45	Quantity exceeded by (45)

© Realtime Despatch Software Ltd 2011-2016

An *Advanced Shipping Note* can be of the following types:

- **Stock** - all items in the ASN must be stock items, i.e. when delivered they will be put away into stock locations.
- **Container-to-stock** - items are delivered against the ASN in containers, whose contents are not confirmed, so will need to be scanned before being put away into stock locations.
- **Order** - all items in this ASN are intended to be matched with pending orders as part of the Goods-In process, the incoming items will be packed and despatched rather than being put away into stock locations.
- **Unrestricted** - the delivery for this type of ASN is not restricted to any particular type.

An *Advanced Shipping Note Line* holds a quantity against a referenced product, and may also hold a (container) location reference or an order reference, depending upon the ASN type.

Delivery Creation

When a lorry makes a delivery at the warehouse door, OrderFlow expects a user to create a **Delivery** entity. How they go about creating a Delivery entity depends on whether the user can identify the associated supplier purchase order or ASN.

If the delivery driver supplies paperwork with a *purchase order reference* with the delivery, then the user should find the *purchase order* on the system, and create a delivery from this (by using a link on that purchase order's detail page).

Similarly, if the delivery driver supplies paperwork with an *advanced shipping note reference* with the delivery, then the user should find that ASN on the system, and create a delivery from this (by using a link on that ASN's detail page).

Alternatively the user can search the pending supplier POs and ASNs by supplier and product to find the associated supplier purchase order or ASN.

These actions automatically associate the new delivery with the PO or ASN accordingly. (A user can create a delivery that is independent of any PO or ASN under *Warehouse --> Deliveries --> New.*)

In OrderFlow, the result will be a **Delivery** entity, attached to which will be one or more **Delivery Line** entities. These can be seen under *Warehouse --> Deliveries* in the desktop interface. The Delivery Detail page shows various attributes of the delivery, such as the supplier and any associated purchase order or ASN. It also shows the lines within the delivery:

The screenshot shows the 'Delivery Detail' page in the OrderFlow system. The page is titled 'Delivery Detail' and includes a sidebar with navigation options such as 'Purchase Orders', 'ASNs', 'Deliveries', 'Dashboard', 'Search', 'New', 'Work Orders', 'Licence Plates', 'Picking', 'Tasks', 'Stock Check', 'Returns', and 'Suppliers'. The main content area displays the details for a specific delivery (ID 4). The 'Delivery Details' section includes fields for ID, Site, Organisation, Purchase Order, Purchase Order State, Supplier, and Default Location. The 'Delivery Lines' section shows a table of lines with columns for Product, Description, State, Location, Date, and Quantity.

Product	Description	State	Location	Date	Quantity
DVD-SPEED	Speed	Applied	incoming_1	05/11/2010 17:25:05	10
DVD-FRAN	Frantic	Applied	incoming_1	05/11/2010 17:25:05	5

In OrderFlow, a delivery can be one of the following 'standard' types:

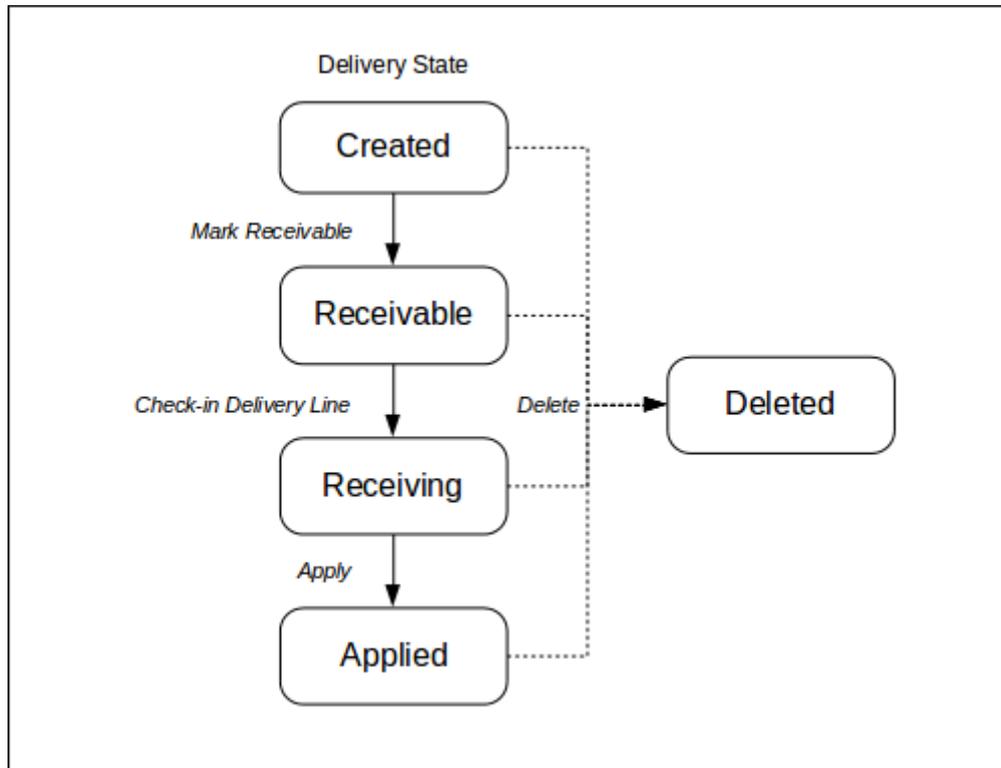
- **Stock (with deferred save)** - the default delivery type, where all products and quantities have to be supplied (to create *delivery lines*) and the delivery *applied* before stock is credited to the warehouse.
- **Stock (with incremental save)** - similar to deferred save, but stock can be credited to the warehouse during the check-in process, instead of waiting until *all* product/quantity combinations have been supplied.
- **Quarantine** - similar to deferred save, but stock *must* be credited to a *quarantine* location; that is, a location whose *logical type* has the 'quarantine' flag set. This will credit the stock into the warehouse but will not increase the 'available' stock figure for the items received.

- **Container to stock** - stock is initially credited as 'unconfirmed', then a verification process must be followed before the delivery can be applied.
- **From licence plates** - creates one or more temporary *licence plate* locations that can then be used to route the incoming stock into several different warehouse workflows, typically via the handheld user interface.

Or it can be one of the following 'cross-dock' types. ('Cross-docking' is the process of using delivered stock to fulfil orders directly, without putting that stock away to warehouse storage locations and then subsequently picking it.)

- **ASN cross-dock** - stock is immediately *cross-docked* from a delivery that was created from an **Order**-typed Advanced Shipping Note.
- **Purchase Order cross-dock** - stock is immediately *cross-docked* to a shipment that has been associated with the same supplier purchase order that the delivery has been associated with. (Shipments can be associated with purchase orders by other means - see the [Despatch Operations](#) section for more details.)
- **Licence Plate Purchase Order cross-dock** - similar to the *Purchase Order cross-dock* type, but uses a licence plate mechanism to process the stock.
- **Out-of-stock cross-dock** - incoming stock is immediately *cross-docked* with existing customer shipments that have the state 'out of stock'. The delivery does not need to be backed by an ASN or a purchase order. Instead, 'out of stock' that are out of stock are automatically matched to use the delivered stock most effectively. Incoming stock that is matched with single line shipments can be packed immediately, incoming stock that is matched with multi-line orders will be moved to a consolidation location.

A *Delivery* has a lifecycle, defined by the following state transition:



Delivery Line entities will ultimately be created and attached to the *Delivery* entity, but exactly *when* these are created depends on the *Delivery* type (as described above).

Processing Delivered Stock

A *Delivery Line* holds the received quantity against a referenced product, and also references the location in which the stock will be credited. Instead of a 'type', a delivery line has a **variation**, that determines how it will be processed. Delivery Line 'variations' are set by choices made during delivery creation, and can take the following values:

- **Just-in-time (JIT)** - the delivered units will either be cross-docked, or routed to a *consolidation* location, in order to fulfil an order whose stock has been ordered from a supplier *just-in-time* (rather than already held in the warehouse).
- **Damaged** - the delivered units will be routed to a location reserved to hold *damaged* goods.
- **Bulk** - the delivered units will be routed to bulk storage locations, using a licence plate mechanism.
- **Incoming to stock** - the delivered units will be put away to stock, after first being credited to an *incoming* location; that is, a location whose *logical type* has the 'incoming' flag set.
- **Direct to stock** - the delivered units will be put away to stock directly, i.e. without first being credited to any other location.

Reconciliation and Discrepancies

Deliveries that reference purchase orders or advanced shipping notes can be *reconciled* against them, after the delivery has been received.

Reconciliation is the process of checking what was *expected* against what was *actually received*. It is an essential process if the discovery of any discrepancies between these values requires further action, such as pursuing a refund from the supplier that sent the delivery.

When a delivery is *applied*, OrderFlow will automatically reconcile it against any associated ASN, and prominently display any discrepancies to the user. It will do the same with any associated PO, but hold these discrepancies against the PO once it has been applied. (This is because a Purchase Order may be delivered across many deliveries.)

Discrepancies and other delivery metrics can be extracted from the system for management purposes using the *reporting framework*.

Processing Delivered Stock

Stock that has been checked-in from a delivery will typically be credited to an *incoming* location, generally within the 'Goods_in' area of the warehouse. In order to efficiently use that stock, it then needs to be moved to somewhere more accessible, and separated according to product. This allows other warehouse staff to pick stock more quickly and easily, without getting in the way of delivery staff (and each other).

Put-away

OrderFlow supports several different strategies for processing delivered stock. The simplest of these is to immediately put the stock away to separate (usually known) **stock locations** when creating the incoming delivery. This may be appropriate for small deliveries but becomes very inefficient as the number of SKUs in a delivery increases.

As described in the previous section larger deliveries are typically moved to the 'Goods_in' area of the warehouse to await put-away.

Put-away is the process of moving incoming stock from an 'incoming' location to a storage location. The system can create a **stock move task** to direct a warehouse staff member to move the correct quantities of stock from the delivery's incoming

Processing Delivered Stock

location to the appropriate target locations for each product in the delivery. If there is no known stock location for a product then OrderFlow will automatically select a new one, based on its knowledge of the locations on the system. Such a task can either be created manually, or on a scheduled basis.

To manually create a putaway task, navigate to *Warehouse* --> *Tasks* --> *New* and select the 'Default Putaway' task type. There is a short-cut link to this page on the delivery detail page, once a delivery has been applied:



When a delivery does not use an *incoming* location, the stock needs to be put away to stock locations *while processing the delivery*. This can be less efficient as it will not necessarily take advantage of a shortest route through the warehouse, i.e. several trips between the delivery area and the rest of the warehouse may need to be taken.

Sometimes, a delivery may need to be put-away into a *quarantine* location, so that it needs to undergo a manual check before being transferred to other warehouse locations.

Cross-docking

OrderFlow supports more sophisticated delivery processing options, with the aim of reducing the need to *double-handle* stock. "Double-handling" of stock means moving it twice, for example moving it once to put away the stock to a stock location, then again to immediately pick the stock to pack a customer shipment. Instead, the stock could be moved straight to the packing desk, if it is known that it will fulfil outgoing shipments. This process is called **cross-docking**.

Cross-docking is most effective for single-line shipments, the handheld process for receiving the incoming delivery can be configured to match incoming stock with out-of-stock shipments, causing the required items to be passed to a packer instead of being put-away to stock.

Consolidation

For multi-line shipments, and single-line shipments whose order line has a quantity greater than the available stock quantity, if double-handling of stock is to be avoided, then delivered stock can be routed to **consolidation** locations.

A consolidation location is used for temporary storage of some of the stock for a shipment, until the rest of the stock is sourced. Note that the rest of the stock could already be in the warehouse, so the consolidation process does not always have to involve recently-delivered stock.

OrderFlow supports routing delivered stock to consolidation locations by matching multi-line shipments that have outstanding lines for the delivered stock, and either using their existing consolidation locations, or using new consolidation locations (in the case where the delivered stock is the first stock available in the multi-line shipment).

If the delivery is of type 'From licence plates', OrderFlow routes the correct stock to consolidation locations using its **workflow processing** function, described in the [Workflow Processing](#) section below.

To process such a delivery, a user should navigate to *Home* --> *Receipt* --> *LP to Incoming* on the handheld interface, select the location in which they are working and scan the licence plate that was created as part of the delivery. Selecting the *Just In Time* option (then scanning the consolidation area or location in which they want to start), they can then start scanning products from the incoming delivery.

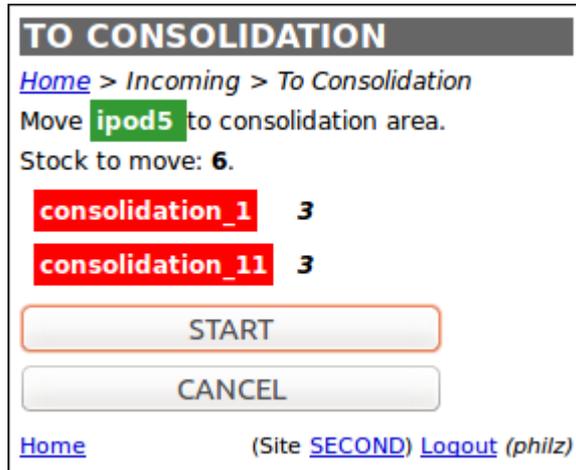
If OrderFlow matches products to consolidating shipments, it will direct the user to move the required number of units to the correct consolidation location, as illustrated in the following screenshot (which is suggesting that 3 units of product *ipod5* should

Processing Delivered Stock

be moved to consolidation location *consolidation_1*, and another 3 units should be moved to consolidation location *consolidation_11*):

Note that the consolidation process can also be used to bring together stock from different area of the warehouse or from different picking processes before all the items required for a shipment are presented to the packer.

OrderFlow drives the cross-docking and consolidation processes using its **workflow processing** function, described in the [Workflow Processing](#) section below.



Workflow Processing

It is quite probable that not all of the delivered stock would fall into one of the pre-defined categories described above, so OrderFlow supports a configurable delivery process that allows the user to route different products to different **workflows**, i.e. some stock to consolidation, some to packing (cross-docking), some to storage, some to quarantine, and some to a location to hold damaged stock.

After dealing with the stock for consolidation, to reduce the distance that the delivery-processing user has to travel, OrderFlow directs them to place stock for packing, stock for storage, damaged stock and unwanted stock in different **footprint** locations. These are locations local to the user that can be subsequently processed independently, allowing the user to process the delivery quickly and efficiently. Dealing with the stock needed for consolidation and cross-docking first means that shipments will be despatched sooner.

As described above, to invoke workflow processing, a user should navigate to *Home* --> *Receipt* --> *LP to Incoming* on the handheld interface, select the location in which they are working and scan the licence plate that was created as part of the delivery. Selecting the *Just In Time* option (then scanning the consolidation area or location in which they want to start), they can then start scanning products from the incoming delivery.

The following screenshot shows a handheld user being directed to move 3 units of product *ipod5* to a *packing* footprint location, and another single unit to a *stock* (i.e. storage) footprint location:

WORKFLOW ROUTING

[Home](#) > [Incoming](#) > [Workflow Selection](#)

Move **ipod5** to workflow *footprint* locations.

PACK	3
STOCK	1

[Home](#) (Site [SECOND](#)) [Logout](#) (*philz*)

 **Note**

If the warehouse has many consolidation locations spread over a large area, a user can select where in this area they want to work. This helps reduce congestions when many warehouse staff are accessing consolidation locations concurrently.

Returns

Customers buying goods via e-commerce will usually have the option to return those goods if they are not happy with them for any reason. A returns address will usually be present on the despatch note sent with an outgoing shipment, or if not, customers can find out where to return items to from an organisation's website. Either way, that return address may end up being a warehouse under OrderFlow's control, and as such, OrderFlow provides functionality that supports the receipt of returned goods into the warehouse.

Authorisation

A return can be 'pre-authorised' or not. 'Pre-authorised' means that the return has been planned, usually by a customer services business function, *before* it is received by the warehouse. The authorisation process often involves the customer being given a "returned merchandise authorisation" or "RMA" which they are asked to quote on paperwork included with the returned items.

A returns authorisation can be created through the OrderFlow desktop interface or by an external system (typically the customer service platform) through the OrderFlow API.

Unplanned returns arrive at the warehouse without warning, possibly because the courier has been unable to find the delivery address or the recipient refused to accept delivery, or simply because the returns policy does not require the customer to notify the business before returning unwanted items.

Organisations may choose to treat pre-authorised and unauthorised returns differently, i.e. they may scrutinise the goods more closely for unauthorised returns. OrderFlow presents two menu options to create pre-authorised and unauthorised returns separately:

Returns
Search
Authorisation
New

Return Creation

To create both pre-authorised and unauthorised returns, the user is required to complete a form that details the following:

- Return type, i.e. authorised / unauthorised / unknown
- Organisation
- Channel (if known)
- Order reference
- Return date (if known)
- Any notes, e.g. detailing any discussion with the customer

For pre-authorised returns, a customer service user of OrderFlow will be able to establish these details directly from the customer conversation. Otherwise, a warehouse operator should be able to establish these details from any paperwork accompanying the returned goods.

Once a return has been created, its detail page will look something like this:

Returns

The screenshot displays the 'Return Item Detail' page. The top navigation bar includes tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. A sidebar on the left lists various system functions, with 'Returns' highlighted. The main content area is divided into several sections:

- Return Details:** A form containing fields for RMA (2), Site (Swindon), Organisation (Myco Incorporated), Channel (Magento), Type (Authorised), Pre-authorized (Yes), Authorisation (100000146), Note (Cust. said Micro USB Blackberry charger doesn't work.), and Comments (No comments have been left for this Return.). It also shows metadata like Created (23 February, 2016 12:20:32), Created by (charlie), State (Created), Return Date (Not set), and Last Updated (23 February, 2016 12:20:32).
- Default return locations:** A list showing Stock (quarantine_3), Quarantine (quarantine_1), and Destroy (quarantine_2).
- Original Order Lines:** A table with columns for Product, Description, and Quantity. It lists three items: Blackberry 9700 (Blackberry Bold 9700b, quantity 1), MicroUsbBlackberry (Micro USB Blackberry charger, quantity 1), and UniCharge35 (Universal charge adapter, quantity 1). Each item has a magnifying glass icon and a link to 'Add Return For Order Line'.
- Return Lines:** A section stating 'This return item does not contain any lines.'

At the bottom of the page, there is a copyright notice: © Realtime Despatch Software Ltd 2011-2016 and a link to 'Return to the top'.

Note that there are no return *lines* at present - these only get created when stock is checked into OrderFlow following actual receipt of the returned goods.

Return Check-in

When the goods arrive at the warehouse door, the user that processes the returned parcel will identify it as a return based on the paperwork that accompanies it. They can then search the existing returns, using the order reference on the paperwork. If this information is missing then the product can be used to search for a pre-authorized return. (If no return is found on OrderFlow, the warehouse user can create one as previously described.)

From the return item detail page, return lines can be added to the return, either by using the links alongside each order line, or by selecting to 'Add' a return line. (This behaviour is controlled by the "Return Lines Linked to Original Order" property.)

While creating a return line, the user can specify the product, quantity, the *type* of return line (i.e. stock / quarantine / destroy / to supplier), the *reason* for return (if available from the paperwork), and the *condition* of the returned item(s). (The return reasons and conditions available are both controlled by application properties.) Additionally, the user may be required to specify whether to proceed with a refund or not (based on configuration), and any notes can be added by the user at this point:

Returns

Despatch Inventory Warehouse Import Integration Reports Admin Setup Advanced

Purchase Orders ASNs Deliveries Work Orders Licence Plates Picking Tasks Stock Check Returns Search Authorisation > New Suppliers

New Return Line

Showing all sites Showing all organisations

Add new line for return 2.

Product lookup - click to input product code.

Selected product details:

Reference: **MicroUsbBlackberry**
Description: Micro USB Blackberry charger tester
Organisation: Myco Incorporated

Please enter a quantity for the return line and confirm to create the new line entry. Also select a return type, and optionally use the note field to record any additional information against this return line.

Quantity: 1
Type: Quarantine
Reason: E - Quality/Manufacturing fault
Condition:
Note: As new, Packaging damaged, Product damaged (refurbishable), Product damaged (irreparable)

Create Cancel

[Back to return](#)

© Realtime Despatch Software Ltd 2011-2016 [Return to the top](#)

Additionally, an image can be added to a return line, in order to capture the state of the returned item, for example.

Once all the required lines have been added to a return, it can be *applied*. This credits the returned stock to the appropriate locations in the warehouse. (The default locations for each return type are held in application properties, or supplier-specific locations may be configured.) The stock can then be returned to the correct location as part of a putaway task (see the Put-away section), or returned to the original supplier of the goods.

Return Entities

The result of an applied return process will be a **Return Item** entity, attached to which will be one or more **Return Line** entities. These can be seen under *Warehouse* -> *Returns* in the desktop interface.

Despatch Operations

This chapter details how OrderFlow handles incoming orders, including how it decides which orders should get which stock, and the processes involved in getting that stock out the door in the right package, with the right paperwork and the right courier label on it.

It also covers any 'end of day' processes that may be required, to report on the day's activity to couriers and/or management.

Order Entities

TODO - would be useful to have a bit more detail on the relationships between the entities, or at least a link to relevant section of INTRODUCTION document.

Order Import

Orders can be imported into OrderFlow via an XML over HTTP API, by importing files from the server file system or by manually uploading files through the OrderFlow desktop interface. For full details, see the "Import Configuration" section in the [Introduction to OrderFlow](#) document.

If the process of importing orders has completed successfully, the import mapping will set the state of the newly created orders and shipments. Orders are typically set to the **validated** state. These orders will contain shipments that have been set to the **ready** state.

Stock Assignment

Stock Allocation

As detailed in the "Stock Allocation" section of the [Introduction to OrderFlow](#) document, stock allocation is the process of identifying *whether* there is stock present in the warehouse to fulfil the order lines in a shipment. It is performed on a line by line basis. Based on the priority of the outstanding shipments, and other criteria, OrderFlow determines, for each order line, whether there is sufficient stock in the warehouse to meet the order line's stock requirement.

In detail, OrderFlow is typically configured to perform stock allocation automatically on a schedule, although it is possible to view the current allocation status (for a site), and to invoke the allocation function, from the *Despatch --> Lines --> Allocate* menu on the desktop interface. The following screenshot of this page shows that a few organisations have order lines that are *pending* allocation (in the 'Default' site):

Stock Assignment

Despatch
Inventory
Warehouse
Import
Integration
Reports
Admin
Setup
Advanced
🔍

- Orders
- Shipments
- Transfers
- Lines
- Dashboard
- Search
- > Allocate
- Assign
- Consolidate
- Batches
- Pick
- Pack
- Carriage
- Manifests
- Bulk Operations

Order Line Allocation

Showing all sites Showing all organisations (inc. test)

Order lines awaiting allocation.

i This screen shows order lines awaiting allocation (for site **Default**).

Pending order lines are in the *created* state with no previous attempt made to allocate stock. If allocation fails due to insufficient stock, the order line is move to the *out of stock* state. If one or more order lines is out of stock, the shipment is also moved to the out of stock state.

Among shipments in a *ready* or *out of stock* state, still awaiting allocation.

	Channel	Pending	Out of stock	Allocated
<input type="checkbox"/>	Acme Web Site	0	0	0
<input type="checkbox"/>	AllCo.com	42	0	0
<input type="checkbox"/>	Composite.com ⚠️	0	0	0
<input type="checkbox"/>	Myco.com	53	0	0
<input type="checkbox"/>	SDS ⚠️	3	0	0
<input type="checkbox"/>	Trico Channel 1	8	0	0
<input type="checkbox"/>	Trico Channel 2	0	0	0
<input type="checkbox"/>	Trico Channel 3	0	0	0

🔍 [Verify allocations](#)

© Realtime Despatch Software Ltd 2011-2016
[Return to the top](#)

ALLOCATION PROCESS

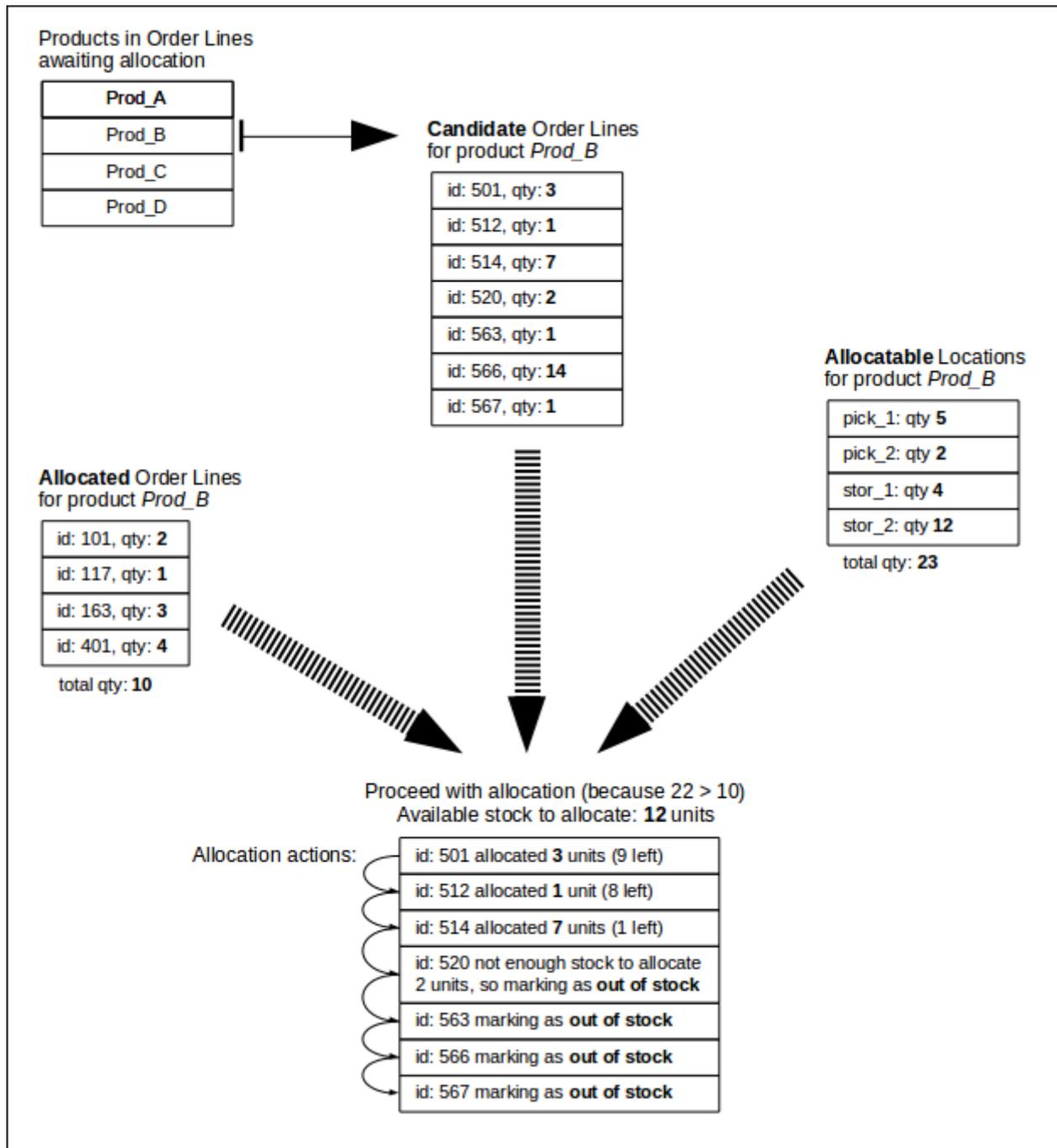
The manual and scheduled invocations of the allocation process result in the same function being performed:

1. For the specified channel, all products in order lines awaiting allocation are extracted from the database.
2. For each product extracted:
 - 'Candidate' *created* or *out of stock* order lines for the product (that are in *ready*, *out of stock* or *reset* shipments) are extracted from the database, in the order defined by the "Order Line Allocation Sort Expression" application property. (If the "Allocation Should Consider Consolidation Stock" property is set, then *consolidating* order lines and shipments are also extracted.)
 - Additionally, the quantities of stock (for the product) in **allocatable** locations is extracted. (These locations are ordered according to the "Assignment Location Sort Expression" application property.)
 - The 'allocation quantities', i.e. the quantities of all order lines in *allocated*, *move pending*, *pickable* or *picked* states (for the product) are also extracted.
 - If the total of the allocation quantities is equal to (or exceeds) the total quantities in allocatable locations, then all the candidate order lines for the product are marked out of stock immediately, and the allocation process ends for that product. (I.e. there is no spare stock for the product.)
 - If, on the other hand, there is more stock in allocatable locations than the stock already allocated (i.e. there is available stock), then each candidate order line is allocated its required quantity of the product in turn. The order line's state updated to **allocated**.
 - If an order line requires more stock than the remaining amount, then it *and all subsequent lines* are marked as **out of stock**. (This is to prevent a situation where a high-quantity line never gets allocated its stock, because it is always given to lower quantity lines.)
3. Finally, the parent shipments of those order lines that have had allocation attempted will have their state updated, according to the following conditions:
 - Shipments that have at least one order line out of stock are marked as **out of stock**.
 - Shipments that are either in the **ready** or **out of stock** state, but whose order lines have progressed to a state at least as far down the process as the shipment 'allocated' state, will be marked as **allocated**. (This uses the 'progress indication' mechanism, which is expanded upon in the [Progress Indication](#) section below.)

Note that this means that allocation may be attempted for the same product across different channels. However, this does not cause a problem as the process will effectively be skipped for a product that has already undergone the allocation process.

Stock Assignment

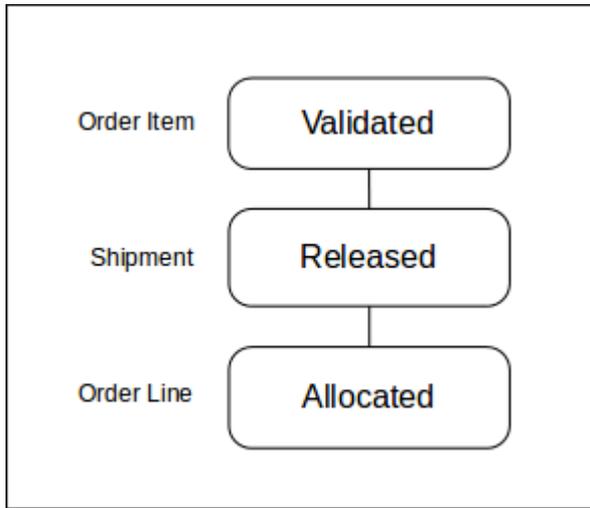
The following diagram depicts an example allocation for product *Prod_B*:



Stock Assignment

Note that no **Order Line Location** entries are created during the allocation process.

If there is sufficient stock for a shipment's order lines in the warehouse, then the states of the entities after allocation will typically be as shown below.



PROGRESS INDICATION

OrderFlow already has a flexible state transition model (for orders, shipments and order lines), using the **State Definition** and **Operation Definition** configurable entities. The scriptable nature of the target states in operation definitions allows the transitions between states to be carefully controlled, and to vary based on any aspect of the order / shipment / order lines involved.

However, sometimes workflows can become so complex that maintaining such scripts to define state transitions can become unwieldy. In some cases, certain states in the workflow may need to be skipped, depending upon the shipment context.

To provide an alternative, each **State Definition** entity can be assigned a **Progress Indication** value. This is a number between 0 and 100, which gives an indication of the position of the state in the workflow. The default progress indication values are shown in the following diagram:

		Order Item State	Shipment State	Order Line State
Progress Indicator	0	created	created	created cross-dockable
	10	pending	.	.
	20	validated	ready reset	.
	30	.	allocated released address not validated awaiting payment request	allocated
	40	.	held before picking	.
	50	.	pickable cross-docking courier not validated consolidated	pickable cross-docking
	60	.	packable picked	picked
	70	part-despatched	packed	.
	80	.	manifested	packed
	90	despatched cancelled	despatched cancelled	.
	100	.	.	bypassed

Progress indication values are used during allocation and assignment to determine whether a shipment can be moved to the next state, based on the progress indication of the states of the shipment's order lines. For example, a shipment whose order lines are all in the **allocated** state can be progressed to the **allocated** state, as they both have the same progress indication value.

Stock Assignment

Additionally, if the "Progress Indication State Control" application property is set, the progress indication values are used to determine whether a parent entity's state should be transitioned, based on the progress indication values of all its children, for *any* state operation.

Stock Assignment

The allocation process assigns the correct states to order lines and shipments depending upon whether the required stock is present in the warehouse.

The stock assignment process goes one step further, and assigns stock from specific locations to allocated order lines. It is possible that an order line could be fulfilled by stock residing in multiple locations, so OrderFlow holds this association in a separate **Order Line Location** entity.

These assignments can be viewed in the 'Stock Location Allocations' section on the order line detail page, which is accessible from the *Despatch --> Lines --> Search* menu (or via a shipment):

The screenshot displays the 'Order Line Detail' page. The main content area is titled 'Order Line Details' and contains the following information:

- Shipment: multishipment1
- Product Reference: PHO-SONY-XP-MINIPRO
- Product Description: Sony Ericsson Xperia Mini Pro
- Quantity: 2
- Volume: 50 cm³
- State: Pickable
- Shipment State: Move pending
- Complete: No
- Operations: 2

Below this is the 'User Defined Fields' section, which shows 'User Defined Values' as 'No values set'.

The 'Stock Location Allocations' section contains a message: 'Stock for this order line is assigned from the location(s) shown below.' Below this message is a table with the following data:

Product	Location Reference	Location Type	Area	Quantity	State	Created
PHO-SONY-XP-MINIPRO	location_4	Pickable	area_1	2	Created	26/02/2016 17:37:54

At the bottom of the page, there is an 'Available Operations' section with the text 'available operations for this order line'.

The assignment process is very similar to the allocation process, but it applies to order lines & shipments in different states, and it creates **Order Line Location** entities if successful. Again, it can be performed manually from the *Despatch --> Lines --> Assign* menu, or on a schedule. Both invocations of the assignment process result in the same function being performed:

1. For the specified channel, all products in order lines awaiting assignment are extracted from the database.
2. For each product extracted:
 - 'Candidate' allocated- or move pending- order lines for the product (that are in released- or move pending- shipments) are extracted from the database, again in the order defined by the "Order Line Allocation Sort Expression" application property.

Stock Assignment

- Additionally, the quantities of stock (for the product) in pickable locations is extracted. (These locations are ordered according to the "Assignment Location Sort Expression" application property.)
- The 'assigned quantities' i.e. the quantities of all order lines that have an existing Order Line Location- record (for the product) are also extracted. (These will be in pickable locations.)
- If the total of the assigned quantities is equal to (or exceeds) the total quantities in pickable locations, then all the candidate order lines for the product are marked as **move pending-* (if they are not already in this state), and the assignment process ends for that product.
- If, on the other hand, there is more stock in pickable locations than the stock already assigned (i.e. there is available pickable stock), then each candidate order line is assigned its required quantity of the product **from one or more of the pickable locations in turn.
The order line's state is updated to pickable.
- If an order line requires more stock than the remaining amount available in pickable location, then it and all subsequent lines- are marked as move pending. (This is to prevent a situation where a high-quantity line never gets assigned its stock, because it is always given to lower quantity lines.)

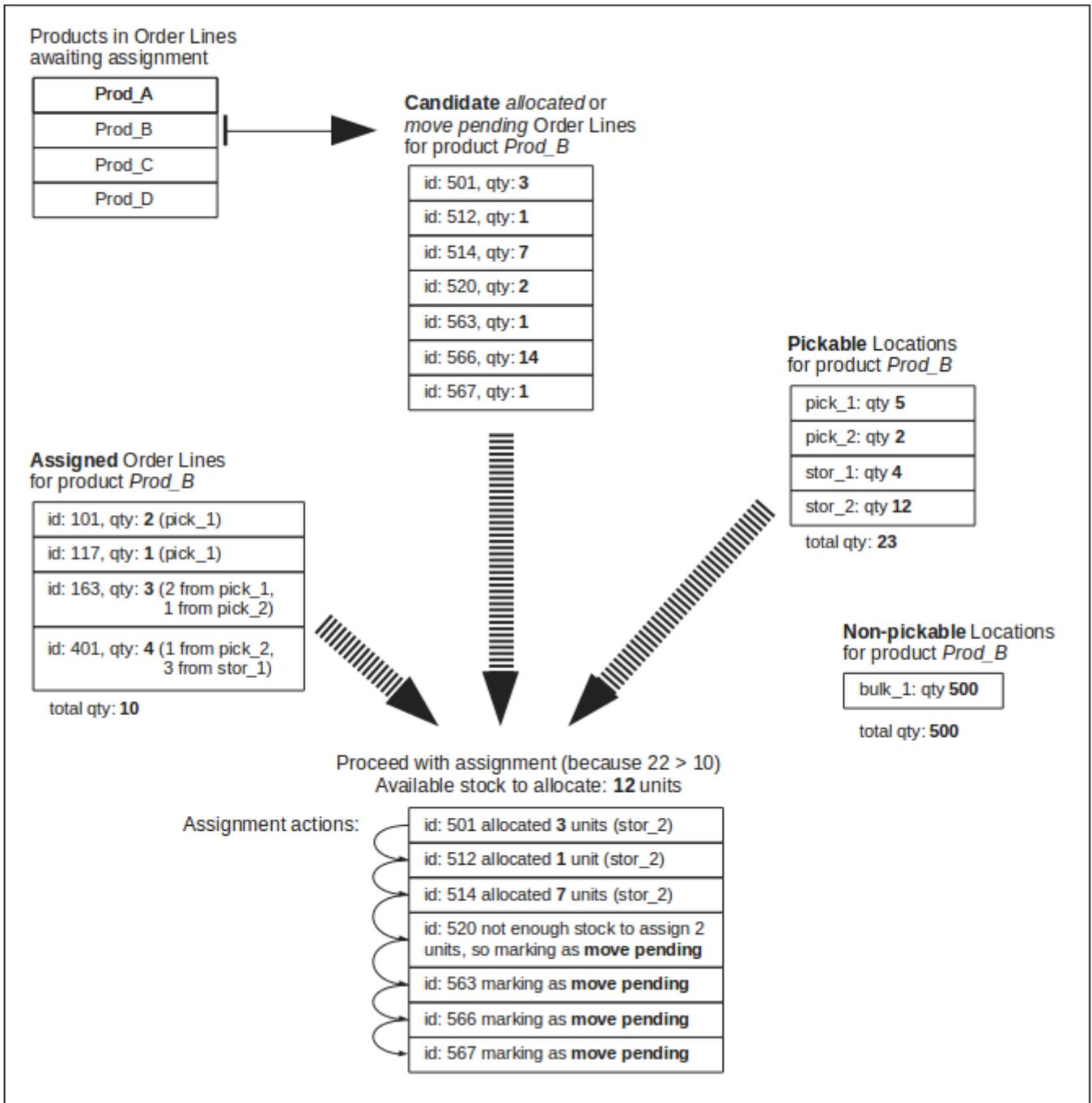
3. Finally, the parent shipments of those order lines that have had allocation attempted will have their state updated, according to the following conditions:

- Shipments that have at least one order line in a 'move pending' state are marked as move pending.
- Shipments that are either in the released or move pending state, but whose order lines have progressed to a state at least as far down the process as the shipment 'pickable' state, will be marked as assigned. This will progress the shipment to its pre-pick state, which is affected by several factors, including its courier state, the result of any configured 'can pick' script, and whether it is a consolidating shipment. If there is no reason not to, the shipment will be progressed to the pickable state.

Notes

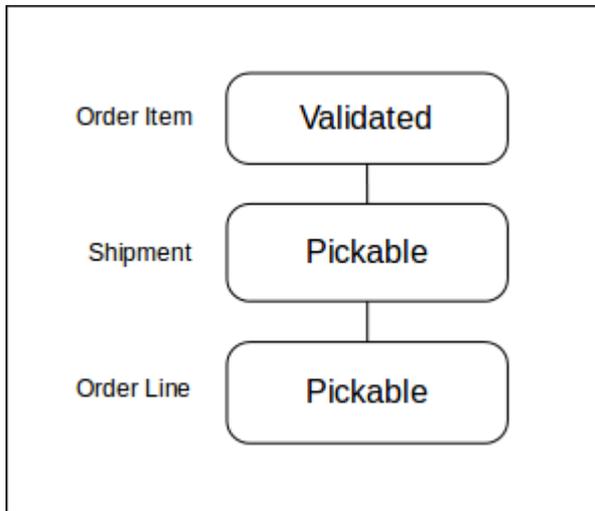
If the product is configured to have the 'primary only' picking strategy then just the primary picking location is used to assign stock from.

The following diagram depicts an example assignment for product *Prod_B*. Note that this is not intended to follow on from the allocation example, as in this case there are additional stock in non-pickable locations, and the 7 candidate lines are all allocated:



Stock Assignment

If there is sufficient stock for a shipment's order lines in pickable locations (and there are no reasons for the shipment to enter another pre-pick state), then the states of the entities after assignment will typically be as shown below.



Despatch Workflows

TODO describe the despatch workflows in a bit more details

Shipments Fulfilled Directly from Stock

Single Line Cross Docked Shipments

Shipments Fulfilled from Consolidation

Shipments Fulfilled from Stock via Consolidation

Manual Shipment Expediting

This section will be expanded upon in a future version.

Picking

While Picking is very much a despatch-related operation, it is a big area, with many aspects to consider. For this reason it is treated separately within the [Picking](#) chapter.

Courier Management

Courier Management is a primarily despatch-related function. However, as with Picking, it is a substantial area, so is covered separately in the [Courier Management](#) chapter.

Packing

This section will be expanded upon in a future version.

End of Day

This section will be expanded upon in a future version.

Picking

Picking describes the processes by which stock is retrieved from locations in the warehouse to be made available for despatch, typically via a Packing process.

Once it has been established that there is sufficient stock in the warehouse to fulfil the orders, and that specific order lines have been assigned stock from specific pickable locations, it is possible for warehouse staff to actually pick the stock from those locations.

As detailed in the "Shipment Picking" section in the [Introduction to OrderFlow](#) document, OrderFlow provides many options for physically picking stock. These options can be broadly categorised into paper-based picking and handheld-driven picking.

Most picking is controlled by the configured **Shipment Batch Types**, which can be found under the *Setup -> Batch Types* menu on the desktop interface.

Batch Selection

Module Requirement

The functionality described in this section requires the module *rtd2-batch*.

The pick and pack process followed by a shipment is determined by the "Batch type" it is associated with. "Batch type" definitions can be used to define the optimal despatch process followed by different types of shipment:

- single-line shipments might be picked using a paper based picking process that allows the picker to make one sweep around the picking face and present the packer with a mixed tote of barcoded items
- multi-line shipments might be picked into a sub-divided cart, the handheld can prompt the picker to put the required number of each item into the correct subdivisions, making it easy for the packer to find the items required for each shipment when packing.
- multi-line orders that need to be picked from more than one area of the warehouse might be picked in a handheld process that requires the items to be moved into a consolidation location.

Shipments are associated with the appropriate batch type by the application property 'batch.selection.script'. The batch selection script is used to automatically assign shipments to the appropriate batch type and then put into a specific batch with other shipments that have the same profile.

Paper-based Picking

A paper-based picking approach requires the product / location quantities for all order lines to be picked to be printed out and given to a warehouse user to perform the pick.

It is usual for the shipments (that contain the order lines to be picked) to be grouped together in *batches*, as it would be inefficient to print one shipment at a time, go and pick it, then pack it and repeat for the next shipment. The number of shipments grouped together (the "batch size") depends upon the 'profile' of the shipments, i.e. how many order lines and the typical quantities of those lines. It can also depend on the products themselves - for example small products such as batteries can be picked in greater numbers than larger products such as storage boxes.

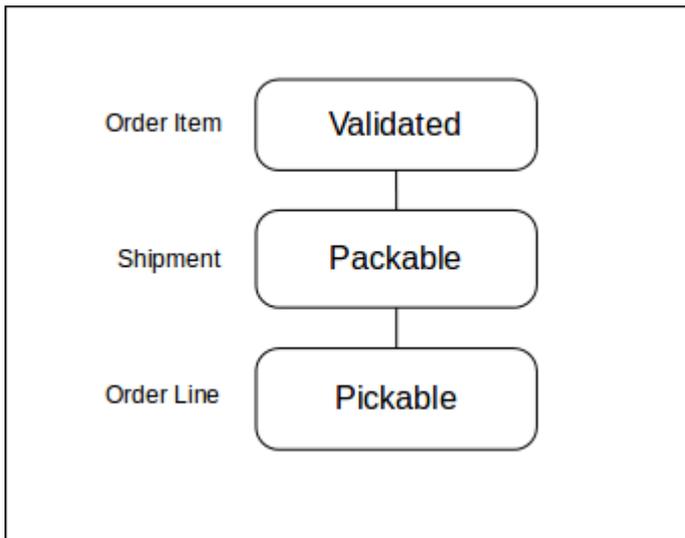
The most useful delineation is between single-line and multi-line shipments. Single-line shipments can usually be picked into a single picking cart, whereas multi-line shipments have to be picked into sub-divided picking carts, or picking trays or similar. This distinction is driven by how the picked items will be handled at the packing desk, in that when a product is picked out of the cart, a single-line shipment will only ever require more units of the same product to be picked out of the cart. A multi-line shipment will require the packer to pick additional (different) products out of the cart - products which may be buried under other products in a single cart. If the products for a multi-line shipment are all picked into the same sub-division of a sub-divided picking cart, it will be easy for the packer to find all the products for the shipment.

Built-in picking reports available include:

- **Picking List (A4)** - the default picking list, that details the quantity of each product to pick from each location. This is ordered by the batch type's *Report Picking Sort* value, which will typically take into account the location in some way, to ensure an efficient route through the warehouse. Note that shipments are not separated in any way in this report, so if two shipments require the same product to be picked, then there will be a single line on the report with the combined quantity.
- **Barcoded Picking List (A4)** - similar to the default picking list, but with product barcodes displayed for each line.
- **Picking List with Full Description (A4)** - similar to the default picking list, but with full product description displayed for each line.
- **Single-line Picking List (A4)** - similar to the default picking list, but each product / location combination is broken down into the quantities required for each shipment.
- **Indexed Multi-line List (A4)** - similar to the single-line picking list, i.e. each product / location combination is broken down into the quantities required for each shipment. Additionally, each product / location / shipment combination is assigned an **index**, which can be used to direct the picked stock to a specific sub-division of the picking cart. The index will be the same for different lines in the same shipment.
- **Multi-line List (A4)** - similar to the default picking list, but this report groups the product / location quantities by shipment, so that all lines for a single shipment are displayed alongside each other. This list relies on Order Line Locations being created for the shipments concerned, so it makes no sense for this report to be run before all shipments are pickable.
- **Multi-line Picking Note (A4)** - similar to the multi-line list, but each shipment is printed on a separate page.
- **Barcoded Multi-line Picking Note (A4)** - similar to the multi-line note, but with product barcodes displayed for each line.

Once printed, a picker can follow the picking list to pick the required items into a cart of their choice. During this process, OrderFlow does not know the exact location of each of the stock items, so it assumes that the stock is still in the source location until the user informs it otherwise.

Depending upon the batch type configuration, the result of the pick operation can be stock in the picking cart, associated with lines and shipments in the following states:



Handheld-based Picking

Module Requirement

The functionality described in this section requires the modules *rtd2-batch*, *rtd2-batch-taskpick* and *rtd2-handheld*.

The alternative to paper-based picking is for the warehouse users to be directed through the picking process by using a handheld device. OrderFlow can be configured to direct picking operations via its Handheld interface.

Handheld-based picking can be used to pick stock for shipments in batches, but it can also be used to pick stock in other processes, such as picking from storage locations into picking or consolidation locations.

Handheld Batch Picking

Shipments can be routed to handheld-based picking by being assigned to shipment batches of a type that has a *picking type* of **handheld**. Batches of this type (in the *handheld_pickable* state) will be visible on the *PICK --> PICK BATCH* menu on the handheld interface, e.g.:

SELECT BATCH

[Home](#) > [Pick Batch](#)

Multiline.bulk (5)

Singleline.bulk (20)

Multiline.ground (5)

Singleline (10)

Singleline.bulk (20)

Mixed.bulk (5)

Singleline (10)

Singleline.bulk (20)

These batches are ordered by the *sort indicator* of the batch type, then the batch id. This allows batches of certain types to be prioritised over others.

When one of the available batches is selected, the user scans the picking cart that they are using, then confirms that they want to proceed with creating a 'handheld_picking' **Stock Move Task** to drive the batch pick. The first open line (by id) of that stock move task is then presented to the user:

PICK LINE

[Home](#) > [Pick Batch](#) > [Pick Line](#)

The next pick is from **trico1**.

9 order line allocations await picking for batch **79**.

Handheld-based Picking

For each line, the user then scans the recommended source location (i.e. the pickable location where the stock is currently residing), then the product(s) from that location:

SCAN PRODUCT
[Home](#) > [Pick Batch](#) > [Pick Line](#)

Product: **TRI3** (2)
From: **trico1**

Please scan the product for the **pick line**:
Remaining items: **2**

Once picked, the user is directed to pick the next line (which could actually be from the same picking location):

PICK LINE
[Home](#) > [Pick Batch](#) > [Pick Line](#)

The next pick is from the **same location**: **trico1**.

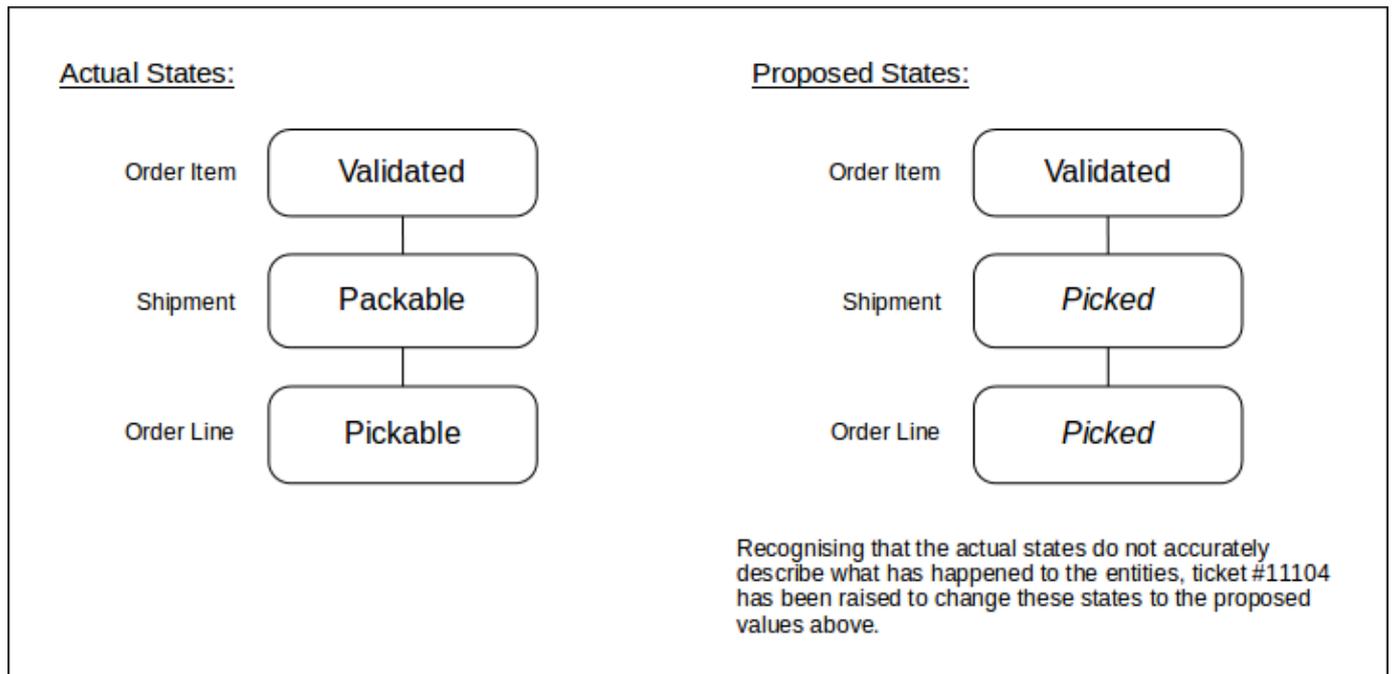
8 order line allocations await picking for batch **79**.
Last pick: **2** of **TRI3** from **trico1**.

Once all the stock move task lines (which equate to Order Line Location entries) have been picked, the user can then confirm that they have completed the batch pick:

COMPLETE PICK
[Home](#) > [Pick Batch](#)

No more lines to pick for batch **79**.
Confirm to complete pick.

The result of a handheld batch pick will be stock in the picking cart, associated with lines and shipments in the following states:



The picked stock can be destined for one of two purposes, defined by the *picking target* of the batch type. If this is set to **Packing** then the picked stock can be taken to a packing station, to be physically packed into the required packaging (see the [Packing](#) section). This is the normal case, especially for single line shipments.

If the batch type's picking target is set to be **Consolidation** then the picked stock is destined for the shipment consolidation area. Once a full batch pick has taken place, a separate task will be initiated to move the items to the individual locations in the consolidation area (see the [Consolidation](#) section below). This is typical for large batches of multi-line shipments.

Handheld Picking by Sequence

Module Requirement

The functionality described in this section requires the modules *rtd2-order-consolidation* and *rtd2-order-pick*

OrderFlow support another mechanism to pick stock; that is by using its concept of **picking sequences**.

This section will be expanded upon in a future version.

Consolidation

As described in the [Introduction to OrderFlow](#) document, **consolidation** in OrderFlow refers to the process of the temporary storage of *some* of the stock for a multi-line shipment, usually in the absence of *all* of the stock.

A shipment that is assigned a consolidation location is considered to be "in consolidation", regardless of whether yet there is any stock in that location. If a shipment is in consolidation, its consolidation location can be seen on the shipment detail screen:

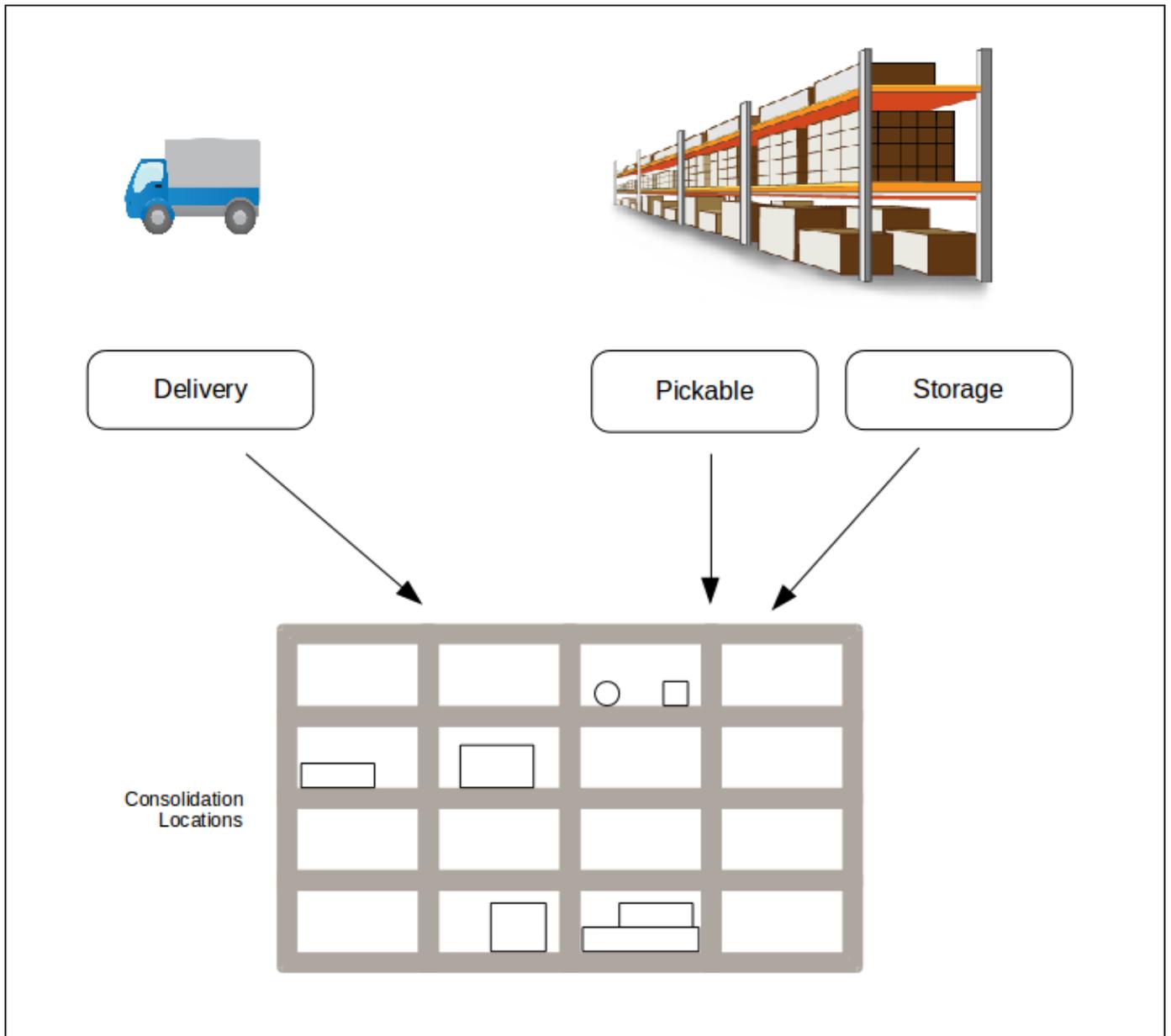


Picking To Consolidation

Stock can be routed to consolidation locations via the following mechanisms:

- From **Delivery** - some organisations order stock from their suppliers only after they have received orders (called *just-in-time* fulfilment). Such stock will be received in deliveries, some of which will be destined for shipments in consolidation. The [Warehouse Consolidation](#) section covers this process in more detail.
- From **Storage** locations - some consolidating multi-line shipments can have lines that can be fulfilled from stock that is stored in non-pickable *storage* locations in the warehouse. Moving such stock will be driven by the creation of a **Pick batch to consolidation** stock move task.
- From **Pickable** locations - similar to from storage locations, but where the source location is *pickable*.

These mechanisms are loosely depicted in the following diagram:



Picking From Consolidation

Once all the stock for a shipment is present in its consolidation location, it can be picked from consolidation to take to a packing desk. OrderFlow supports this process using its handheld interface to direct a user to the next completed shipment's consolidation location, so that they can then scan the products into a picking cart. The picking cart can then be taken to a packer.

The following application properties are used to sort the completed shipments into priority or efficient picking order (respectively):

- `shipment.consolidation.priority.sort.expression`
- `shipment.consolidation.encyency.sort.expression`

This process is accessible from the *PICK --> FROM CONSOLIDATION* menu on the handheld interface:

PICK CONSOLIDATED SHIPMENT

There are **22** shipments in consolidation ready for picking.

PICK NEXT - PRIORITY

PICK NEXT - EFFICIENT

CANCEL

[Home](#)

Courier Management

Carriers and Services

OrderFlow integrates with a host of systems to support courier integration. On the system, a courier integration is represented using the *courier* entity.

In some cases, the integration is directly with the *carrier* itself. In other cases, the integration is with a third party integration service, such as MetaPack or NetDespatch. When integrating with a third party integration service, this can be done either with a particular carrier and service in mind. However, some integration services, notably MetaPack, will select a carrier and service automatically.

OrderFlow supports a range of couriers to enable shipment of items directly to end user addresses. These courier integrations are both directly through integration with the individual carriers systems, and indirectly through specialist integration services such as [MetaPack](#).

See the Advanced section for more details on Integration.

OrderFlow supports three types of integration with third party couriers. Which of these three types of integrations a particular courier uses does have significant implications for order management workflow as well as on the ground support requirements.

- **Web Services API integration.** OrderFlow integrates with the courier via a web services API. OrderFlow uses the courier's API to submit details of shipments to be sent out. Usually, the courier returns one or more labels to be printed, either as a PDF document, some image format (such as PNG), or in raw printable text using a printing language such as ZPL. It also normally provides a tracking reference for the customer's use.

The API also often exposes other operations, for cancelling shipments, generating end of day manifests, etc.

Where possible, we favour using a web services API to integrate with couriers in preference to the other methods, described next.

- **EDI-based integration.** A second type of courier integration involves the use of EDI (Electronic Data Interchange). With this type of integration, OrderFlow manages the assignment of tracking references to shipments internally, typically allocating tracking references from a range provided by the courier. OrderFlow also generates the label internally.

Shipment details are uploaded in bulk to the courier via an end of day manifest, typically either via FTP or HTTP.

This approach has the advantage of allowing the courier integration to easily fit within the order processing workflow. The main weakness is that there is no mechanism for validating the selection of services directly with the courier prior to shipping parcels.

- **Desktop software integration.** Many courier, particularly those that do not provide a web services API, provide a desktop software application for capturing shipment details and generating labels. These software packages are typically installed on desktop PCs on the packing desk in the warehouse. With the help of the Print Server, OrderFlow normally submits shipment data by writing to a file in a folder monitored by the courier desktop application. It then reads files in that folder to obtain tracking references.

The main advantage of the desktop software integration approach is that less work is required initially for the typical integration. However, these integrations typically require much more support effort to set up and maintain, and are consequently our least favoured type of courier integration.

Courier Properties

Courier Stages

For each courier, an association is made between the *Shipment Courier State* and the relevant courier stage.

The key courier stages are:

- **import**: indicates the courier state which the shipment should reach at the point where import is completed. If the *import courier state* is on or after `partially_selected`, this may indicate that the courier selection script should be run directly following import.
- **pick**: indicates the courier state which should be reached prior to picking. Mostly commonly used to indicate that courier selection or validation needs to take place before the shipment can be marked as `pickable`.
- **pack**: indicates the courier state the shipment needs to before the shipment can be moved to packing. Typically used to force a courier selection or validation prior to displaying the packing screen.
- **label print**: typically used to indicate that the shipment needs to be validated or accepted before any attempt can be used to print the shipment label.
- **despatch**: currently not used. However, may in the future be used to indicate that a shipment needs to appear on a manifest before it can be marked as despatched.

Courier Management Workflow

TODO - describe the workflow for different types of integrations

API: before shipment becomes pickable, but after stock assignment, label is fetched.

Describe shipment states:

- courier not selected
- courier not validated
- courier invalid

Cancellation

Manifesting

TODO - add this, perhaps in a courier chapter.

Manufacturing

Module Requirement

The functionality described in this section requires the modules *rtd2-product-group* and *rtd2-warehouse-bom*.

OrderFlow supports a **manufacturing** process. It does this via the following capabilities:

- It can hold definitions for raw materials.
- It can hold definitions for manufactured products.
- It holds the 'ingredients' of a manufactured product in a bill of materials, i.e. which raw materials in which quantities are required to manufacture that product.
- It provides a process that allows a user to 'check out' enough raw material stock to manufacture the required number of manufactured products.
- It provides a process that allows a user to 'check in' the number of units of the manufactured product (created from the raw materials).

The raw materials can be specified in units, by weight, or by volume.

OrderFlow uses **Work Order** terminology to support its manufacturing process, in that the process of turning raw materials into finished products is managed using a **Works Order** entity. These can be seen under *Warehouse* --> *Works Orders* in the desktop interface:

The screenshot shows the 'Work Order Dashboard' in the OrderFlow software. The interface includes a top navigation bar with tabs for Dispatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. A left sidebar contains a menu with items like Purchase Orders, ASNs, Deliveries, Work Orders, Dashboard, Search, New, Licence Plates, Picking, Tasks, Stock Check, Returns, and Suppliers. The main content area displays a 'Recent work orders' section with a table of data.

Finished product	Picking location	State	Quantity
finished_2	raw_picked_5	Picking	1
finished_2		Created	1
finished_1	raw_picked_1	Picking	3
finished_2	raw_picked_2	Picking	2
finished_1	raw_picked_3	Picking	7

At the bottom of the dashboard, there is a copyright notice: © Realtime Despatch Software Ltd 2011-2016, and a 'Return to the top' link. An 'Override' button is also visible in the bottom right corner of the dashboard area.

Raw Materials

In OrderFlow, a **raw material** for a works order is simply a product of type *raw_material*. There isn't anything special about this product type, but it is not usually sellable and the stock levels are not usually sent to any upstream systems (i.e. shopping carts.).

Raw materials can be specified as units, as weights or as volumes, e.g.:

Work Order Lines

This work order includes the following **lines**.

Product	Required Quantity	Actual Quantity	Shortfall	Excess
raw_1	1	0	1	
raw_3	2.0 g	0 g	2.0 g	
raw_5	3	0	3	

Finished Products

A **finished product** is defined as a product of a type that has the *composite* attribute, indicating that its definition depends on other products. These *constituent* products are likely to be defined as raw materials, but could also be packaging materials, or anything else that is required to create the finished product.

The following screenshot of a product detail page for a finished product shows the constituent products *raw_1*, *raw_2* and *raw_3*:

The screenshot shows the 'Product Detail' page for 'finished_2'. The page includes a navigation menu on the left with options like 'Dashboard', 'Search', 'New', 'Inventory', 'Stock', 'Changes', 'Locations', 'Areas', 'Categories', 'Datasheets', and 'Admin'. The main content area is divided into several sections:

- Details for product finished_2:** A table with fields for Reference (finished_2), Description (Finished 2), Organisation (Composite Inc), Barcode (Not set), Created (21/03/2014 14:39:57), Activated (Yes), Sellable (No), Dangerous (No), Fragile (No), Type (Composite), Category (Not set), and State (Created).
- Warehouse-specific:** A section indicating that the product currently has no warehouse-specific details, with an option to 'Add warehouse details'.
- Pricing:** A section with a link to 'view pricing details for this product'.
- Datasheets:** A section with a link to 'view datasheets for product'.
- Constituents:** A section stating that the current product is a composite product containing the following manufacturing products:

Product	Type	Quantity
raw_1	Raw component	1
raw_3	Raw component	2.0 g
raw_5	Raw component	3

At the bottom of the page, there are several action buttons: 'Edit product', 'Create label', 'Work order', 'Deactivate', and 'Remove manufacturing products'.

Creating Product Relationships

In a customer's environment, the relationships between finished products and raw materials usually have to be carefully defined, there is no facility to create these relationships on an ad-hoc basis via the desktop interface.

Instead, these relationships can be *imported* using a **grouped product** spreadsheet. This defines the quantities of raw materials required for composite products, in either units, or the default weight (or volume) storage units. (These are held in the *Product weight storage unit* and *Product volume storage unit* properties respectively.)

An example of a grouped product spreadsheet for manufacturing products is shown below:

	A	B	C	D	E	F
1	externalReference	parent	child	definition	quantity	
2	finished_1:raw_1:manufacturing	finished_1	raw_1	manufacturing	2	
3	finished_1:raw_2:manufacturing	finished_1	raw_2	manufacturing	1	
4	finished_1:raw_4:manufacturing	finished_1	raw_4	manufacturing	3	
5	finished_2:raw_1:manufacturing	finished_2	raw_1	manufacturing	1	
6	finished_2:raw_3:manufacturing	finished_2	raw_3	manufacturing	2	
7	finished_2:raw_5:manufacturing	finished_2	raw_5	manufacturing	3	
8	finished_3:raw_4:manufacturing	finished_3	raw_4	manufacturing	2	
9	finished_3:raw_6:manufacturing	finished_3	raw_6	manufacturing		
10						
11						
12						
13						
14						
15						
16						
17						
18						

This simply needs to be a unique value. OrderFlow will determine the correct unique identifier based on the other data provided

A product in the organisation used for the import. Must be a composite product.

A product in the organisation used for the import.

The type of the relationship. Points to a grouped product definition.

The *definition* column refers to a **Grouped Product Definition** entity, which defines the *type* of grouped product relationship. For works orders, the grouped product definition to use to define relationships between raw materials and finished products is the **Manufacturing** definition.

Grouped Product definitions can be managed from the *Advanced --> Warehouse --> Grouped Products* menu in the desktop interface.

Works Order Creation

Once the relationships between raw materials and finished products are created (using grouped products), an actual **Works Order** can be created via the desktop interface, from the *Warehouse --> Works Orders --> New* menu. The user finds the finished product using its reference, after which OrderFlow uses the available raw material stock to calculate the possible number of units of the finished product that could be made. This value is displayed in the 'Max Supported Quantity' field:

The screenshot displays the 'New Work Order' form. The 'Product lookup' section shows a search bar with the text 'Product lookup - click to input product code.' Below it, the 'Selected product details' section lists: Reference: finished_2, Description: Finished 2, and Organisation: Composite Inc. The 'Work order details' section includes: Picking product location: Not set, and Finished product location: Not set. The 'Quantity' section includes: Max Supported Quantity: 3, Max Configured Quantity: For the current site **Second** the recommended maximum finished product quantity is **not set.**, and Actual Quantity: 3. A 'Create' button is located at the bottom right of the form. The footer of the page contains '© Realtime Despatch Software Ltd 2011-2016' and a 'Return to the top' link.

The 'Max Configured Quantity' field holds any warehouse-specific limit to the number of units that can be manufactured in a single work order. This value can be configured in the finished product's *warehouse product* configuration.

Based on these maximums, the user can then decide how many units of the finished product to manufacture, then select the 'Create' button. The resulting work order contains *work order lines*, which detail the quantities of raw materials required to make the number of finished products requested by the user:

Works Order Creation

Despatch | Inventory | Warehouse | Import | Integration | Reports | Admin | Setup | Advanced

Purchase Orders | ASNs | Deliveries | **Work Orders** | Licence Plates | Picking | Tasks | Stock Check | Returns | Suppliers

Work order for finished_2

Showing all sites | Showing all organisations (inc. test)

Summary

Finished product	finished_2	Organisation	Composite Inc
Quantity	2	Site	Second
Picking target location	raw_picked_4	Created	25 February, 2016 10:52:45
Finished product location	Not set		
State	Created		

[Edit](#) [Delete](#)

i You can now pick the work order by creating a task.

[Create picking task](#)

Work Order Lines

This work order includes the following lines.

Product	Required Quantity	Actual Quantity	Shortfall	Excess
raw_1	2	0	2	
raw_3	4.0 g	0 g	4.0 g	
raw_5	6	0	6	

© Realtime Despatch Software Ltd 2011-2016 [Return to the top](#)

Works Order Picking

Once the picking target location is set, a **task** can be created for the work order. The purpose of such a task is to direct a warehouse user to physically pick the stock required for the manufacturing process, into the picking target location. (The task definition used is hard-coded to *pick_for_manufacturing*.)

Once the task is applied, i.e. the stock has been picked, the works order can be *submitted to manufacturing*, as shown in the following screenshot:

The screenshot displays the 'Work order for finished_2' page. The navigation menu includes: Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. The left sidebar lists: Purchase Orders, ASNs, Deliveries, Work Orders, Licence Plates, Picking, Tasks, Stock Check, Returns, and Suppliers.

Summary

Finished product: **finished_2**
 Organisation: Composite Inc
 Quantity: 2
 Site: Second
 Picking target location: **raw_picked_4**
 Created: 25 February, 2016 10:52:45
 Finished product location: Not set
 State: Pick complete

[Edit](#) [Cancel](#)

Submit for manufacture

Work Order Lines

This work order includes the following **lines**.

Product	Required Quantity	Actual Quantity	Shortfall	Excess
raw_1	2	2	0	
raw_3	4.0 g	4.0 g	0	
raw_5	6	6	0	

Work Order Picking Tasks

The following **tasks** have been created to pick this work order.

ID	State	Open Lines	Complete
20	Applied	0 of 3 source lines 0 of 3 target lines	Yes

© Realtime Despatch Software Ltd 2011-2016 [Return to the top](#)

Works Order Completion

If the finished product location is not set, the works order detail page advises the user to set this, otherwise the manufactured stock cannot be checked back into OrderFlow:

Works Order Completion

Work order for finished_2 [Showing all sites](#) [Showing all organisations \(inc. test\)](#)

Summary

Finished product	<input type="text" value="finished_2"/>	Organisation	<input type="text" value="Composite Inc"/>
Quantity	<input type="text" value="2"/>	Site	<input type="text" value="Second"/>
Picking target location	<input type="text" value="raw_picked_4"/>	Created	<input type="text" value="25 February, 2016 10:52:45"/>
Finished product location	<input type="text" value="Not set"/>		
State	<input type="text" value="Manufacturing"/>		

[Edit](#) [Cancel](#)

⚠ Once manufactured, you will need to set the finished product incoming location before checking in the manufactured stock. Use the 'Edit' link to set this location.

Once set, the works order can be completed:

Work order for finished_2 [Showing all sites](#) [Showing all organisations \(inc. test\)](#)

Summary

Finished product	<input type="text" value="finished_2"/>	Organisation	<input type="text" value="Composite Inc"/>
Quantity	<input type="text" value="2"/>	Site	<input type="text" value="Second"/>
Picking target location	<input type="text" value="raw_picked_4"/>	Created	<input type="text" value="25 February, 2016 10:52:45"/>
Finished product location	<input type="text" value="finished_1"/>		
State	<input type="text" value="Manufacturing"/>		

[Edit](#) [Cancel](#)

i Once manufactured, mark this work order as complete and check **2** of **finished_2** into the finished product incoming location **finished_1**.

[Complete work order](#)

When the work order is completed, OrderFlow credits the finished product location with the requested number of units of the finished product. This stock can then be moved to pickable locations (or picked directly if the finished product location is pickable), so that it can be picked for outgoing shipments.

The entities created in OrderFlow during the manufacturing process include a **Work Order** entity, attached to which will be one or more **Work Order Line** entities. A **Work Order Task** entity is created to manage the relationship between the works order and the underlying pick-for-manufacturing **Stock Move Task**.

Management Operations

Monitoring

This section will be expanded upon in a future version.

Reporting

This section will be expanded upon in a future version.