



OrderFlow Introduction

OrderFlow Ltd.

Document Version:

Document Built: 2025-06-04

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 a high-level overview of the OrderFlow system and the key business functions it supports.

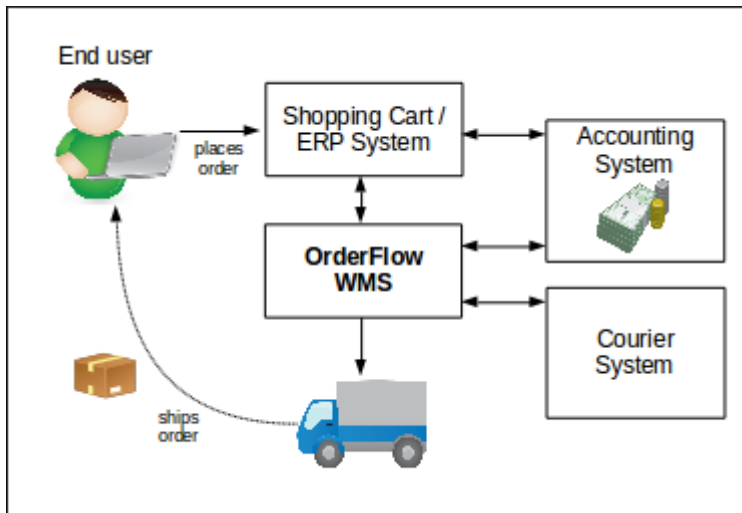
The intended *audience* of this document is:

- business leaders who are considering OrderFlow as a solution to the needs of their organisation
- potential users and administrators of the OrderFlow system

The OrderFlow System

OrderFlow is built specifically around the needs of eCommerce fulfilment. It has been designed to help retailers and fulfilment specialists to run their operations with ease, efficiency and control, while providing the optimal level of visibility of these operations.

The diagram below illustrates where OrderFlow is placed in a typical eCommerce operation.



We are all familiar with the process of buying goods online. Through a web-based store, we place orders for products selected from a catalogue, enter our payment details, then click on the 'Buy' button. All this takes place via a Shopping Cart web application.

For small stores taking only a handful of orders, the process of despatching the orders can be done manually. However, for online retailers taking hundreds or thousands of orders per day, this is not an option. For large operations, a dedicated order processing and warehouse management system is essential if this function is to be performed effectively.

OrderFlow manages all the processes that ensure the ordered items are despatched to customers as quickly and efficiently as possible.

Some of the key functions include:

- receiving deliveries of stock into the warehouse
- managing the putaway of stock into storage locations
- directing the processes that optimise the storage of stock in the warehouse
- identifying the stock that can be used to fulfil specific orders, and directing strategies to pick stock efficiently and accurately
- providing facilities for label generation and packing of shipments
- interacting with third party courier systems to retrieve labels and other documentation, and notifying these systems when shipments are despatched from the warehouse
- managing the return of stock items to the warehouse

In addition, there are many other processes that OrderFlow supports which are critical to the operations of specific customer environments.

Each of these processes may involve complex interactions within OrderFlow or with third party systems. Wherever required, OrderFlow provides the user interfaces necessary for warehouse staff, as well as the generation and printing of the appropriate paperwork and reports.

In fulfilling these functions, OrderFlow typically needs to integrate with other external systems. In addition to the Shopping Cart, it often integrates with Enterprise Resource Planning (ERP) systems such as SAP, or accounting systems such as Sage 200. In a typical installation it will also integrate with the software of several couriers, or with a dedicated Courier Management system such as MetaPack.

Desktop Interface

The desktop interface to the OrderFlow system consists of a number of tabs, each of which performs functions in distinct areas of the application. This section provides an overview of the functionality provided within each of the tabs.

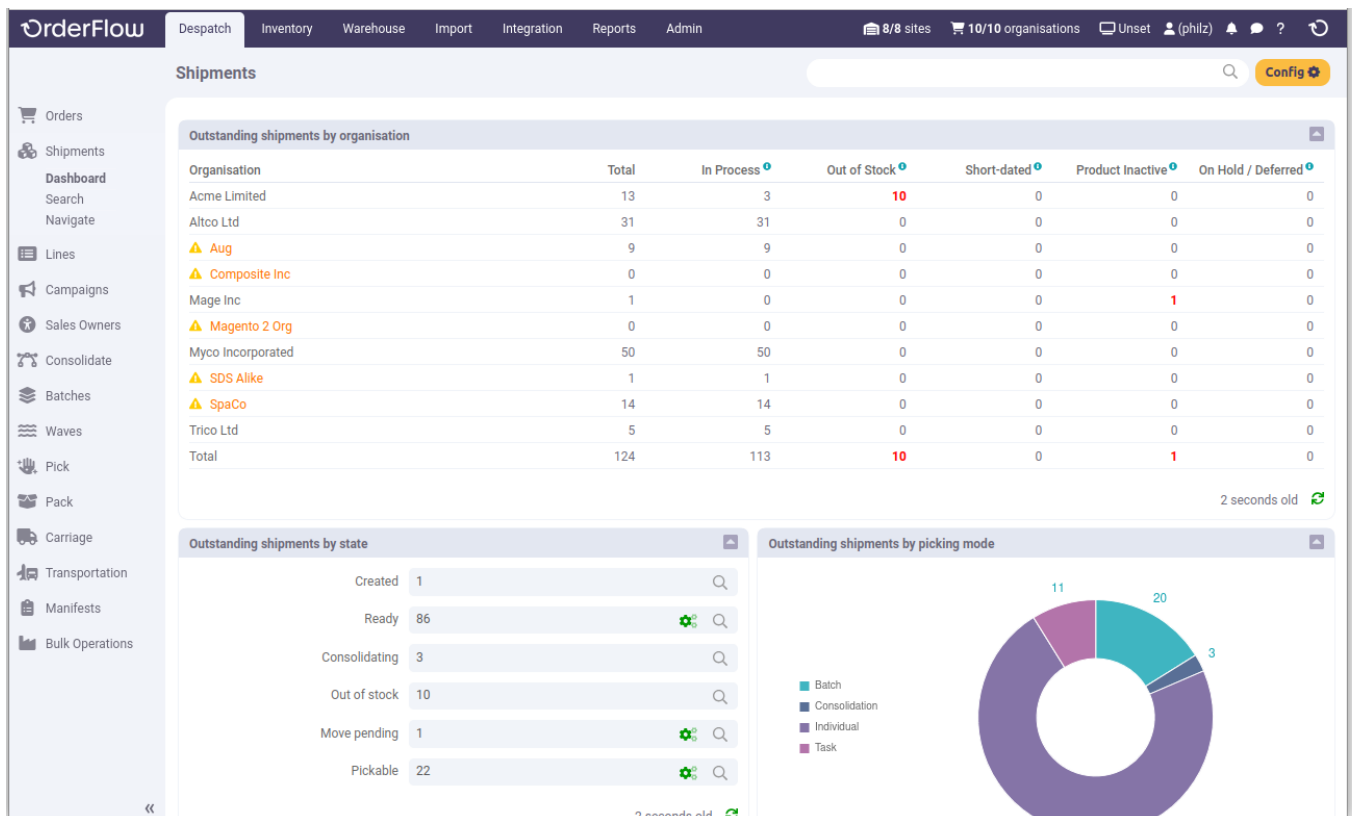
Despatch

The **Despatch** tab is concerned primarily with Order Processing activities.

Screens to search for orders received, shipments to be despatched, and their corresponding order lines are found here. In addition, there are screens for controlling batching, picking, packing and other courier-related operations.

See the [Order Processing](#) section for more details.

Inventory



Inventory

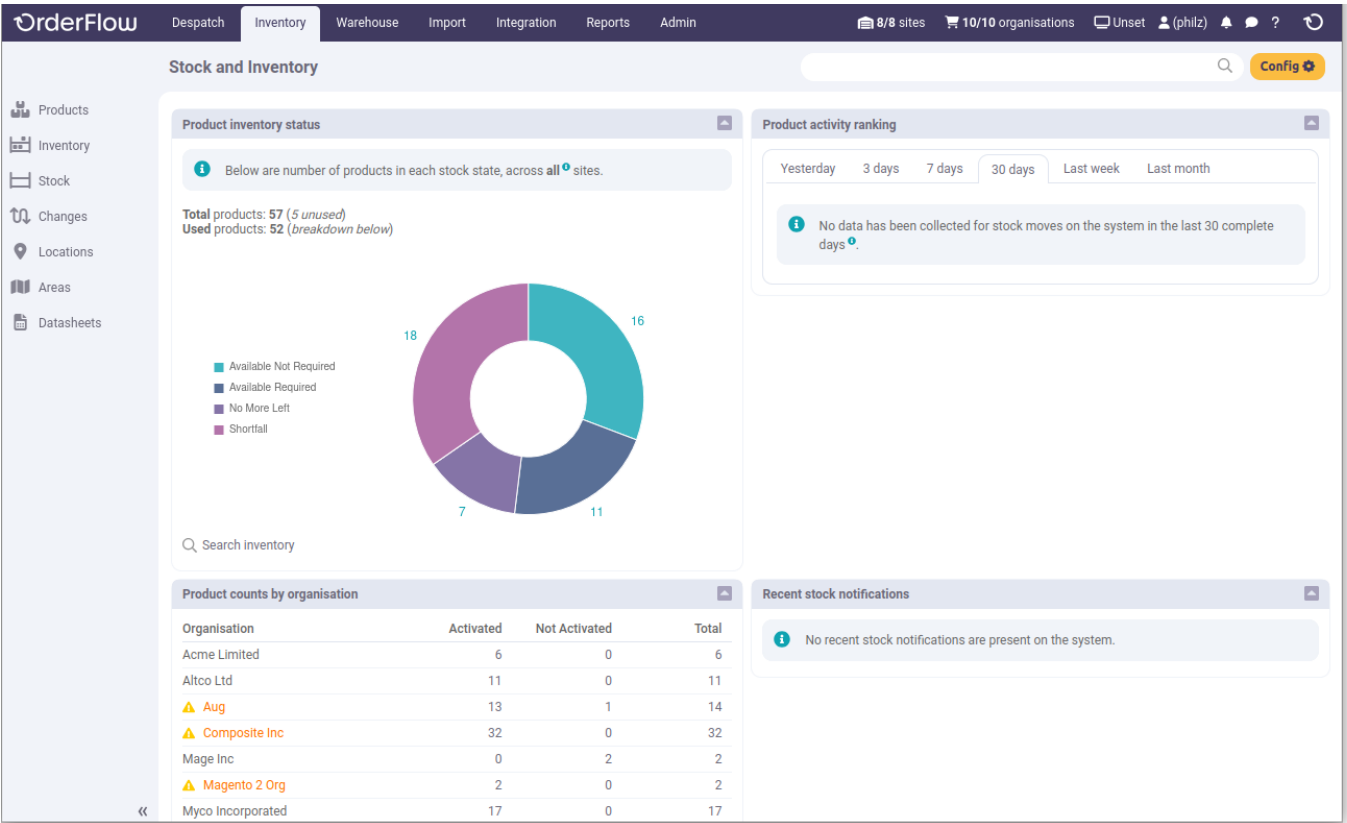
OrderFlow stock control tracks the quantity of every product SKU (Stock Keeping Unit) by location. The movement of every item within the warehouse is tracked. This capability is visible primarily through the **Inventory** tab.

This tab provides the means to search, view and administer definitions for products and locations. It shows the current stock position as well as historical changes.

It also allows for changes to be made directly to the stock position, via stock moves or stock adjustments.

See the [Warehouse Management](#) section for more details.

Warehouse



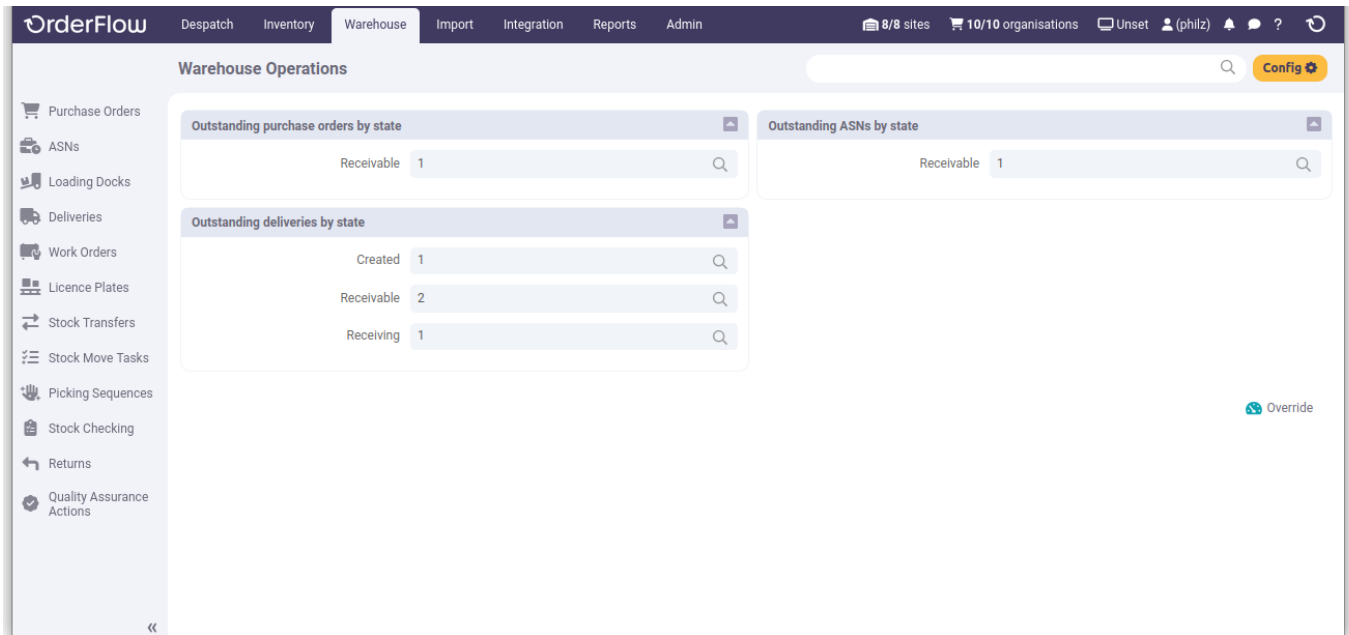
Warehouse

The **Warehouse** tab provides access to the warehouse management-related processes on the system. These include the processes for receiving incoming goods through *deliveries*, which can themselves be associated with *purchase orders* and *advanced shipping notes*.

Operations to direct stock movements in the warehouse are accessible here in the form of *stock move tasks*, as is the processing of stock *returns* into the warehouse.

See the [Warehouse Management](#) section for more details.

Import



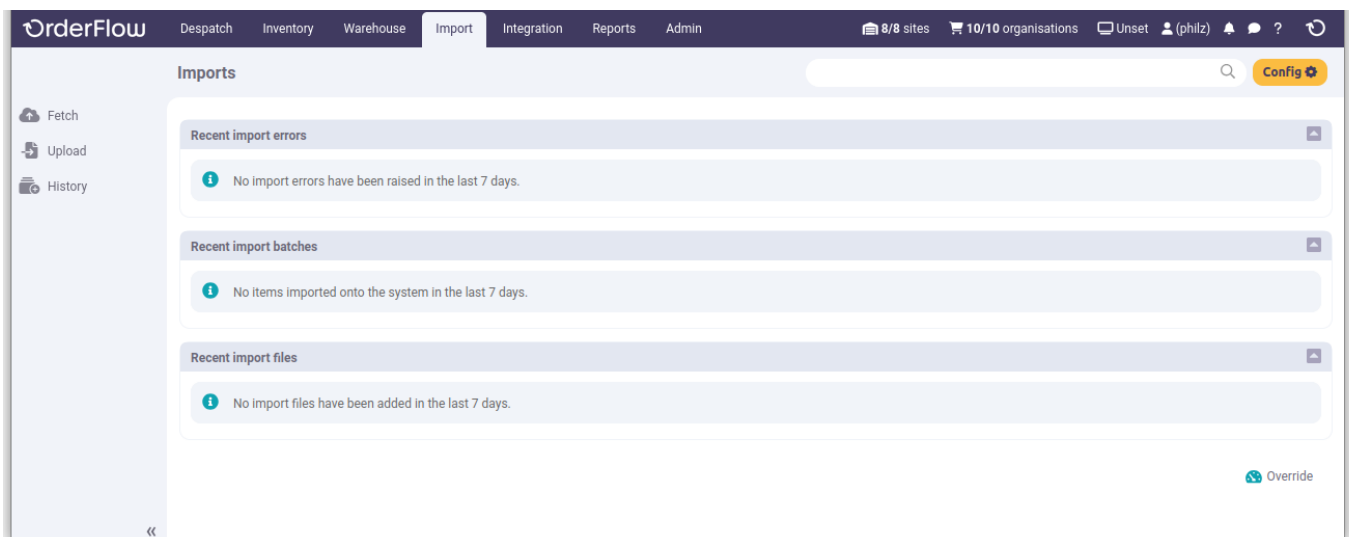
Import

OrderFlow can only perform useful functions once it has been populated with the specific details of the warehouse or warehouses it controls. This includes products, locations, users and orders.

The data can be entered manually when the system is first configured but it usually uploaded in bulk through spreadsheet files or a remote interface. In both cases, the **Import** tab is used.

This tab provides access to historical records of imported items, and any errors that may have occurred during import. This includes data imported using files (CSV, Excel, XML) as well as directly through the OrderFlow web interface.

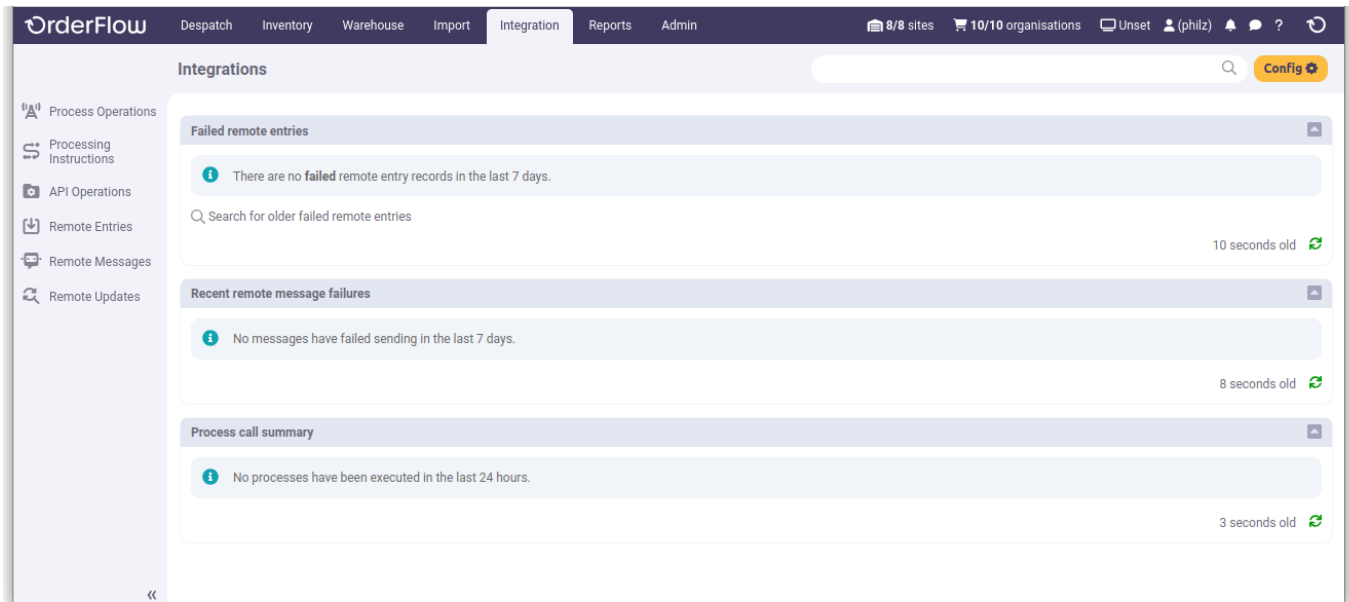
See the [Import](#) section for more details.



Integration

OrderFlow integrates with many third party applications. The most important of these are eCommerce systems (e.g. Shopping Carts), as these systems are the source of orders fulfilled by OrderFlow.

The **Integration** tab supports integration with third party systems. See the [Integration](#) section for more details.



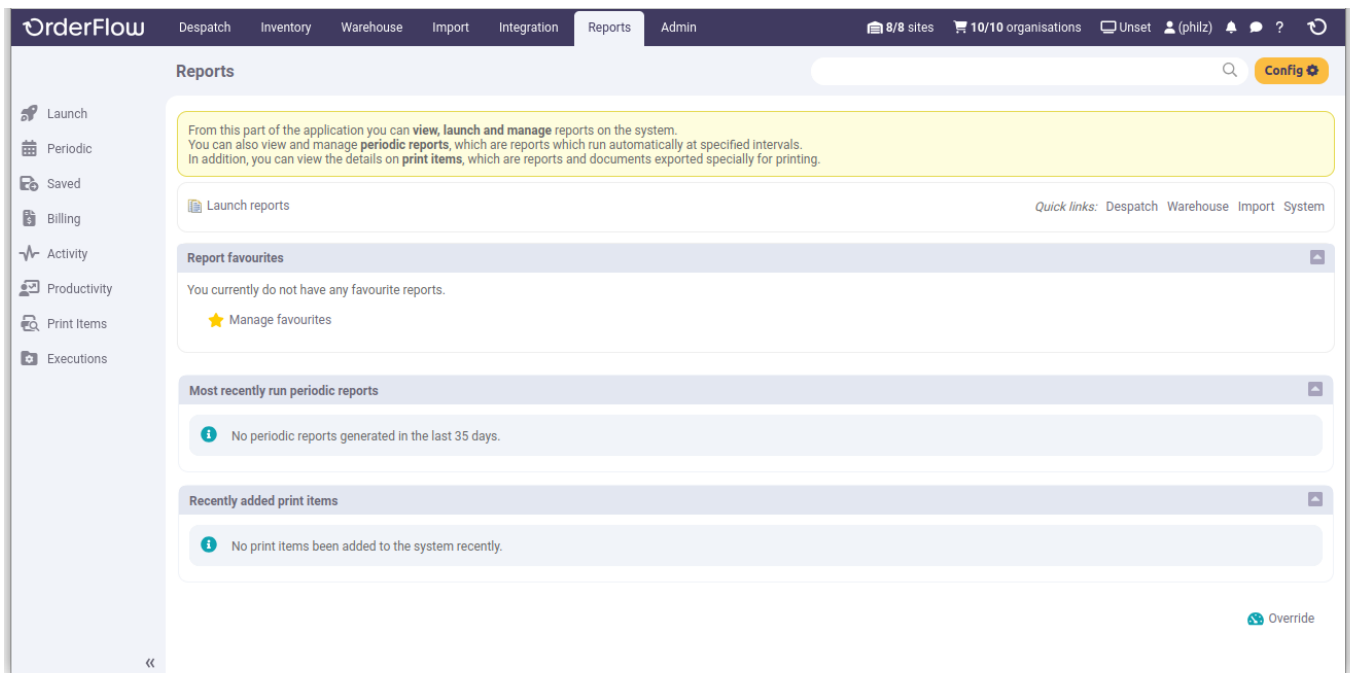
Reports

Reporting is hugely important in OrderFlow. The same reporting framework is used in areas as diverse as the production of customer paperwork (despatch notes and labels), management reports and the dashboards that are part of the user interface. Many of the system processes (such as *stock move tasks*) are driven by reports.

The **Reports** module includes the user interface for running built-in reports, as well as for creating new reports, automated reports and dashboards.

See the [Reports](#) section for more details.

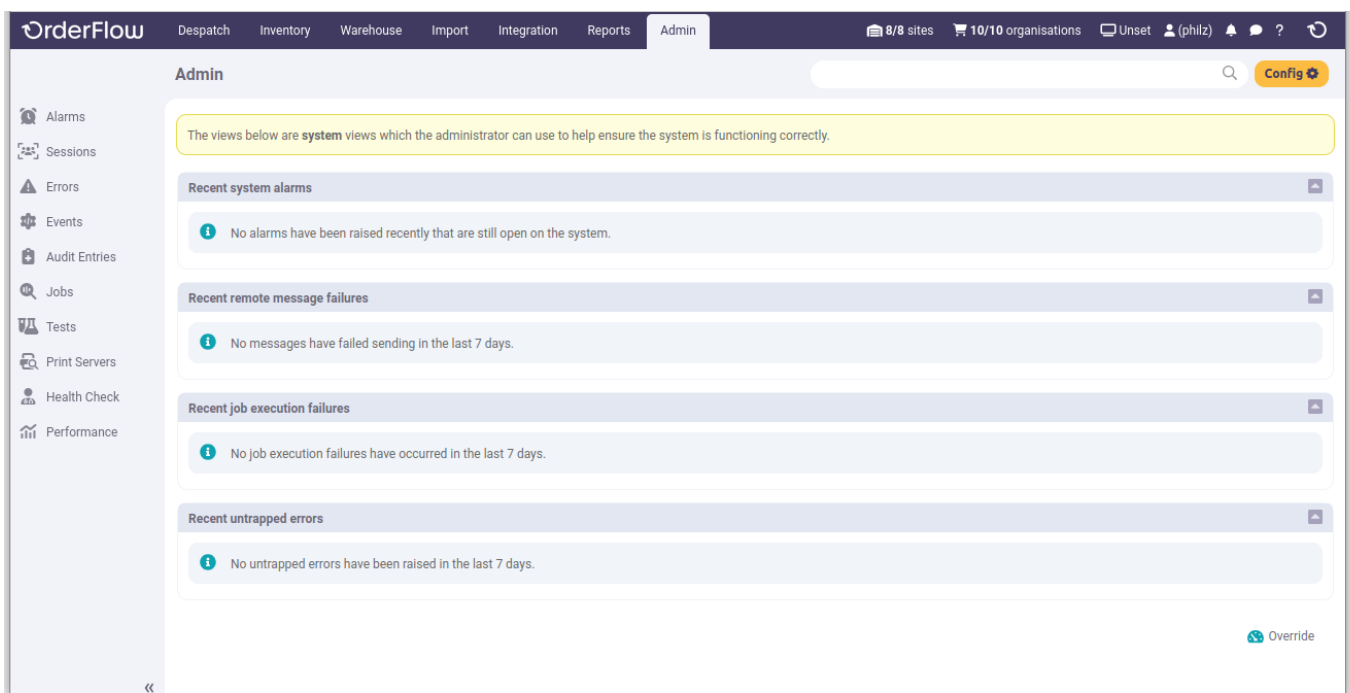
Admin



Admin

The **Admin** tab provides access to functionality that is not necessarily used by most operational staff, but is still important to the day to day running of the system. User access and permissions, logging, error and performance monitoring are all considered administration functions.

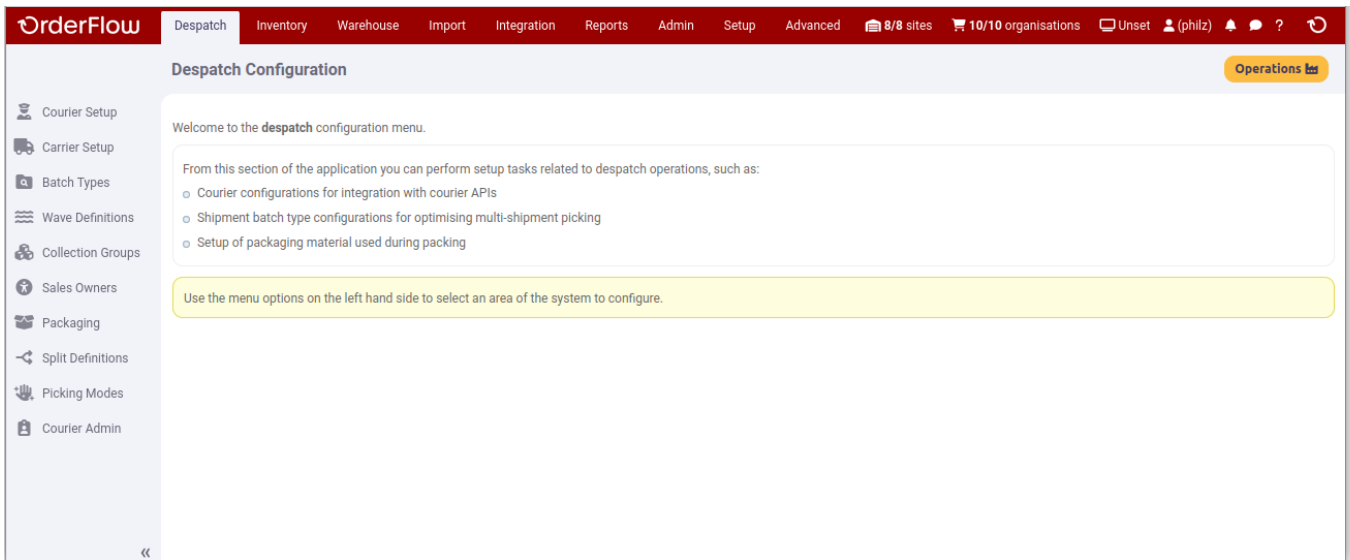
See the [Admin](#) section for more details.



Config

The **Config** tabs deal with functionality that require technical knowledge beyond the level of most users. Many of the scripting and field mapping options are accessible within these tabs, as are configurations for complex areas of the system such as *stock move tasks*. It is also typically used when new functionality is released, for example when a new courier is being introduced into the system.

See the [Config](#) section for more details.



Handheld Interface

The **Handheld** interface to OrderFlow is designed to run on small-screened handheld terminals built specifically for the heavy demands of a warehouse. Almost all of OrderFlow's warehouse management and picking-related functionality is handheld-enabled, although businesses can also choose to use paper-based processing.

The Handheld interface can be accessed from OrderFlow's desktop interface via the **Advanced --> System** menu.

CORRECT STOCK

OrderFlow

[Home](#) > [Product Confirm](#) > [Correct](#)

The currently recorded quantity for location **location_059** and product **COMP683016** is **5000 pieces** .

Please confirm the actual quantity is:

5000

pieces

CONFIRM

CANCEL

[Home](#) (Site [Swindon](#)) (Organisation [ElectricEtail](#))

Log out ([support](#))

Mobile Interface

OrderFlow features a mobile application optimised for display on small handheld devices such as smartphones. This application is designed primarily to display Key Performance Indicator (KPI) reports and other dashboard reports.

This feature is at an early stage of development.

The next two chapters provide an overview of OrderFlow's Order Processing and Warehouse Management capabilities, with illustrated examples.

Order Processing

One of OrderFlow's most fundamental functions is Order Processing, which includes the pick, pack and despatch of orders.

OrderFlow can process orders from multiple organisations and multiple sales channels within organisations. It can restrict a user's access to certain organisations and channels, so that users can only see the orders that are relevant to them.

Additionally, OrderFlow is a multi-site system, which means that it supports the fulfilment of orders across multiple warehouses and all that this entails.

This chapter tracks a typical order as it is processed through the OrderFlow System. **This is a simple example for illustration purposes only.** See the Further Reading section, below, to find out more about the depth of OrderFlow's capabilities.

Terminology

Orders and Shipments

Throughout this chapter, reference will be made to **orders**, **order lines**, **shipments** and **items**. The relationship between these is as follows:

- An order is made up of one or more order lines, to be despatched to an end user at a residential or business address.
- An order line is a product being ordered, in the quantity required.
- Order lines from the same order can be *despatched* in one or more shipments. The usual case is that an order will be despatched in just one shipment.
(Shipments are despatched via a **courier**.)
- A shipment with only one order line is called a **single-line** shipment. A shipment with more than one order line is called a **multi-line** shipment.
- Items are the individual products within an order line.

The diagram below shows a multi-shipment order *123456780*, which contains three shipments. Shipment *123456780-2* is a single-line shipment, whereas the others are multi-line shipments.

Order: 123456780

Shipment: 123456780-1

	<i>product</i>	<i>quantity</i>
Order line:	WIDGET-AAA	12
Order line:	WIDGET-BBB	5
Order line:	WIDGET-CCC	100

Shipment: 123456780-2

	<i>product</i>	<i>quantity</i>
Order line:	WIDGET-DDD	1000

Shipment: 123456780-3

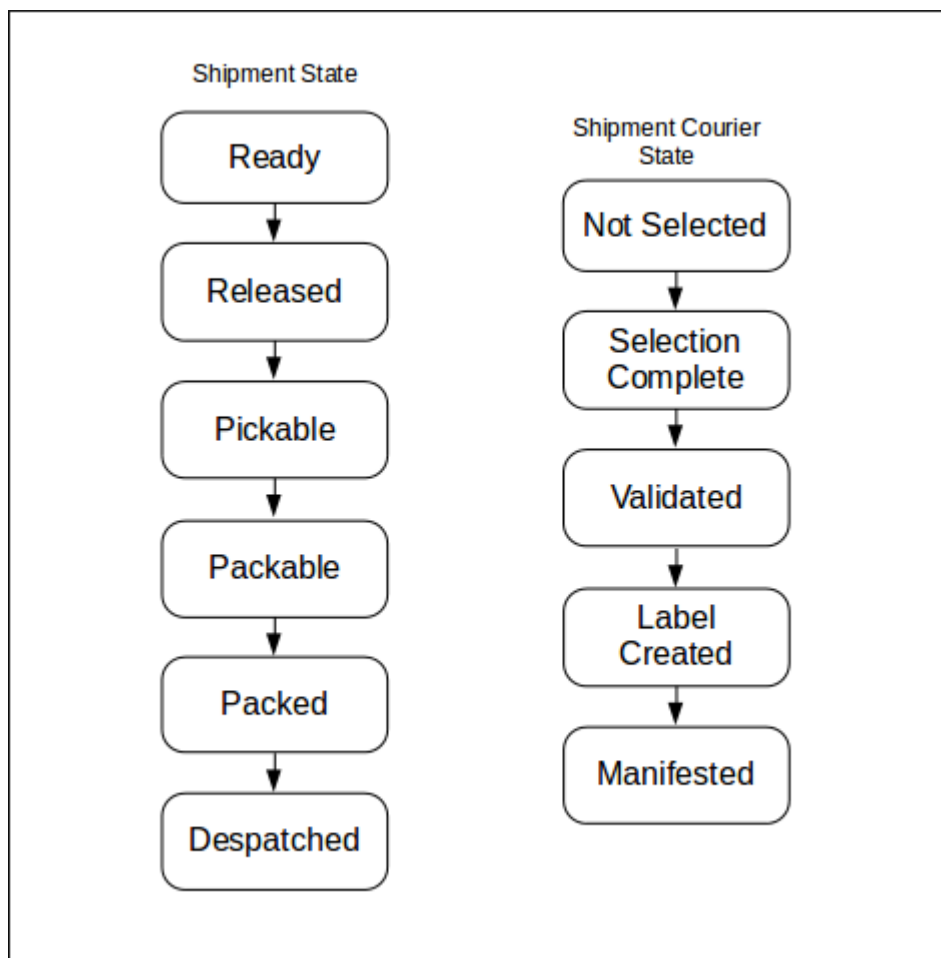
	<i>product</i>	<i>quantity</i>
Order line:	WIDGET-EEE	1
Order line:	WIDGET-FFF	3

States

OrderFlow uses **states** to monitor and control an order's progress through the system workflow. Orders, shipments and order lines can have various states, and a shipment can have additional states that signify its progression with its courier.

As an example, when an order is imported into the system, by default the order is *validated*, the order lines have a state of *created* and the shipment is *ready*. When all items in an order have been shipped, the order lines have a state of *packed*, while the shipment(s) and order are *despatched*.

There are many states that can be used for shipments as they progress through OrderFlow. Some environments may also use custom states. The transition between states is configurable, so there is no definitive state transition diagram that can be presented. However, a typical shipment state transition is shown below, alongside a typical *shipment courier* state transition:



Note that some states of the states are *terminal* states. For example, the *despatched* and *cancelled* shipment states are terminal, as there is typically no further work to do on a shipment when it reaches this state. Shipments, orders and other entities that are not in a terminal state are considered *open*.

An Order Processing Example

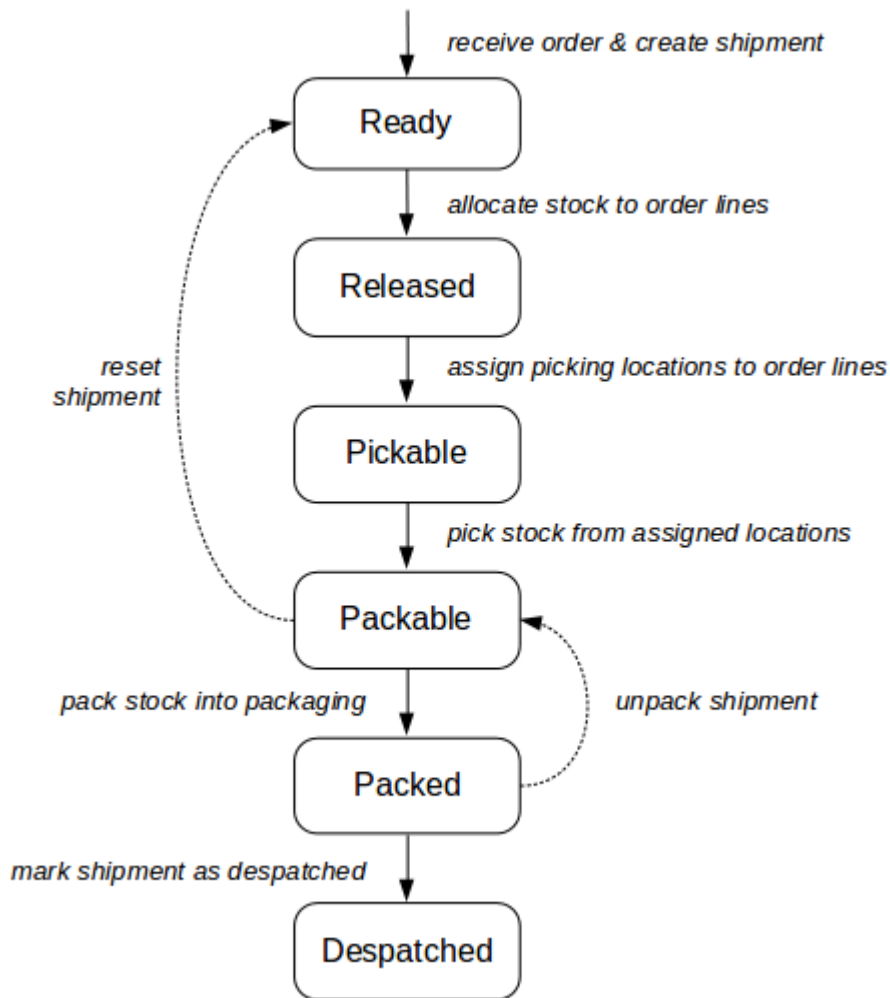
For an order to progress through a typical despatch workflow, various things need to happen to it. Firstly, the required stock needs to be allocated to its order lines, then the *specific* items in *specific* locations need to be assigned to these order lines.

Once specific stock has been assigned, that stock can be picked by a warehouse operator. This can be directed by a paper report, generated by OrderFlow, or by a handheld device that the warehouse operator (called a 'picker' in this case) carries around with them.

Once a picker has picked all the items for the order's shipment(s) (or more typically, a batch of shipments at a time), they can present the stock they have picked to a packer. The packer's responsibility is to check the items are correct, in the right quantities, and pack them using the required packaging, ready for despatch. The packer will also usually include a *despatch note*, which details the items in the shipment, for the recipient to read and check.

At some point during this process, OrderFlow will determine which courier should be used to despatch the shipment, based on scripted configuration. It will also print a courier-specific despatch label to attach to the shipment. It may interact with the courier's system at this point, to inform it of the details of the shipment, and to get a despatch reference in real-time.

Once the courier turns up and takes all the packed shipments away for delivery, they can be marked as *despatched* in OrderFlow.



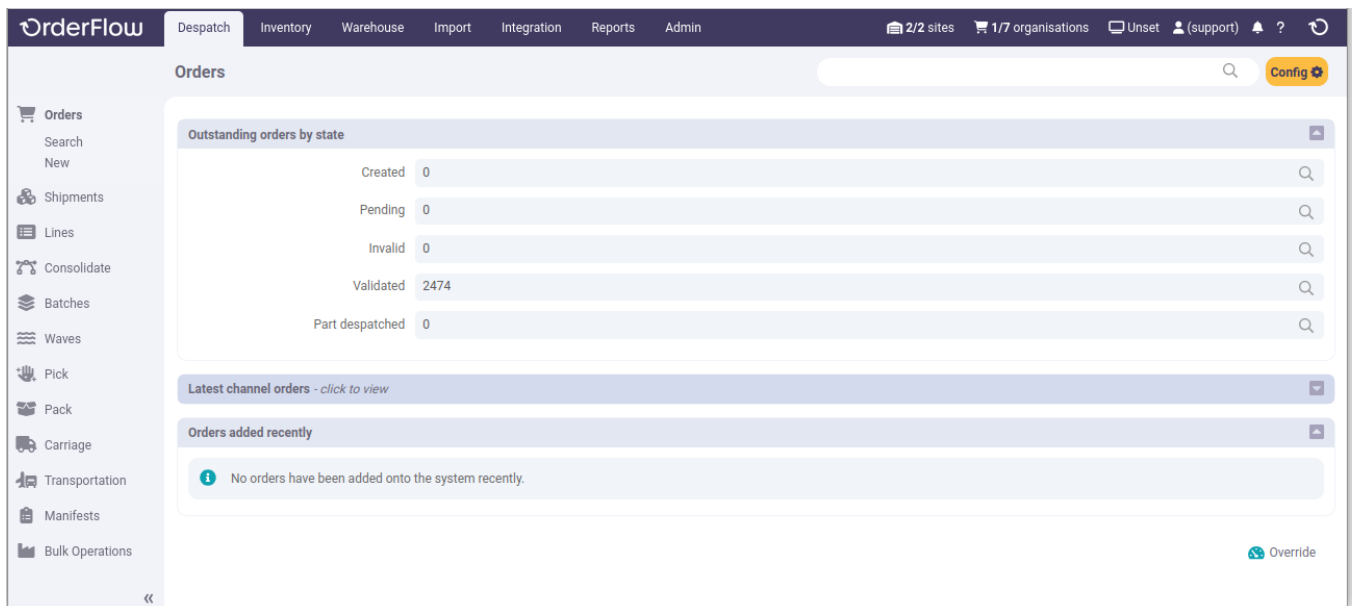
In the next sections, some of these steps are described in a bit more detail.

Order Receipt

Orders which have been imported (i.e. have been *transformed* and *validated* - see the [Advanced](#) section for more details), can be viewed on the Orders Dashboard which is accessed from the 'Despatch' tab.

For most users, the Orders Dashboard will be used only to view and search for Orders, as most orders are automatically imported and validated by OrderFlow. However there are some cases which may require manual validation, for example, orders for different countries, orders which exceed a certain value, etc.

Stock Allocation and Assignment



OrderFlow has to decide how many shipments will be required to fulfil the order. This decision will be based upon various factors, including where (in a multi-site environment) the stock resides for the order lines, since some products may only be stocked in some warehouses. Other factors when deciding whether to split an order into multiple shipments may include product lead time, product weight, or any special carriage requirements (e.g. an out-sized product).

The normal case is for an order to have just one shipment.

The next stage is to allocate the order lines with stock. Only order lines that belong to an order in the *validated* state can be allocated stock.

Stock Allocation and Assignment

OrderFlow is capable of using two types of fulfilment model - the **Stock-based Fulfilment** model or the **Just-in-time (JIT)** model.

In the majority of cases, OrderFlow automatically allocates stock using the the stock-based fulfilment model, which is described here. For details of the Just-in Time (JIT) fulfilment model, see the [Just-in Time Fulfilment](#) section, below.

When using the **Stock-based Fulfilment** model, OrderFlow employs a two-stage process for identifying the stock that can be used to despatch orders.

Stock allocation is the process for identifying *whether* there is stock present in the warehouse to fulfil the order lines in a shipment. *Stock assignment* is the mechanism by which stock in *specific locations* are reserved for picking the stock required for these order lines.

These are discussed in turn below.

Stock Allocation

Stock allocation is performed on a line by line basis for orders that are waiting to be despatched. 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.

If there is sufficient stock, the order line is given a state of *allocated*. If there is a shortfall, the order line is given a state of *out of stock*.

The state of each order line can be viewed from the 'Lines' dashboard, which is accessed from the 'Despatch' tab.

The screenshot shows the 'OrderLines' dashboard in the OrderFlow application. The top navigation bar includes tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, and Admin. The left sidebar contains a menu with options like Orders, Shipments, Lines, Search, Allocate, Assign, Consolidate, Batches, Waves, Pick, Pack, Carriage, Transportation, Manifests, and Bulk Operations. The main content area is titled 'Order Lines' and features a search bar and a 'Config' button. Below this, there is a section 'Order lines by state' with a table showing the count of lines in various states: Created (0), Allocated (2), Out of stock (329), Pre-assigned (0), Pickable (2379), Move pending (2), Short dated (0), and Picked (0). Each row has a search icon and a settings icon. Below this, there are two sections: 'Most required products' and 'Out of stock products'. The 'Most required products' section includes a message: 'The following products are required by the most incomplete order lines.' and a table with columns 'Reference', 'Lines', and 'Quantity'. The 'Out of stock products' section includes a message: 'The following products have the most out of stock order lines. New stock is required to fulfil these lines.' and a table with columns 'Reference', 'Out of Stock Lines', and 'Quantity'.

Reference	Lines	Quantity
TRAN805432	776	776
TRAN805422	283	283
TRAN805408	220	220
TRAN805410	128	128
TRAN805423	94	94
TRAN805415	94	94

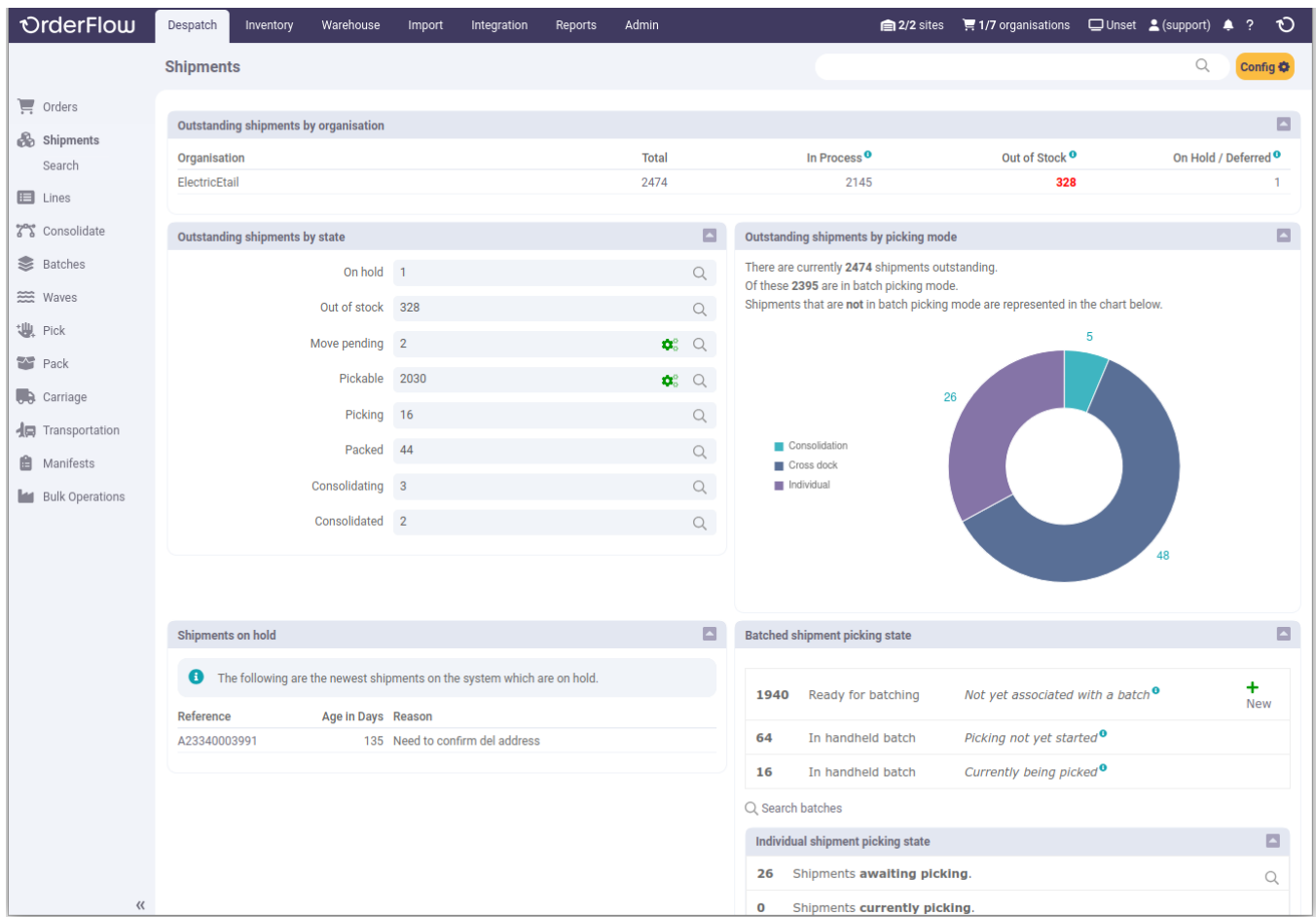
Reference	Out of Stock Lines	Quantity
TRAN805410	128	128
TRAN805414	64	64
TRAN805416	64	64
TRAN805400	32	32
TRAN805403	32	32
TRAN805425	9	9

A shipment's state can be automatically updated, based on the state of its order lines. For example, if all order lines for a shipment have a state of *allocated*, the shipment is progressed to a state of *released*.

If one or more order lines for a shipment have a state of *out of stock*, the shipment is given a state of *out of stock*.

The state of each shipment can be viewed from the 'shipments' dashboard, which is accessed from the 'Despatch' tab.

Stock Allocation and Assignment



The next stage is to assign specific picking locations for a shipment's order lines. Only shipments that have a status of *released* can have their order lines assigned.

Stock Assignment

In the majority of cases, the OrderFlow system automatically assigns stock.

Stock assignment is the process of reserving specific stock in the warehouse to fulfil the order lines in a shipment. OrderFlow attempts to identify picking locations and quantities that match the stock requirement for the order lines concerned. If a picking location can be found for an order line, the order line is given a state of *pickable*.

If all order lines for a shipment have a state of *pickable*, the shipment is given a state of *pickable*.

In some cases, picking locations cannot be found for certain order lines. Typically, this means that additional warehouse operations are required before certain order lines can be picked. For example, some of the stock required may only be present in non-pickable locations, such as bulk storage or incoming locations. In this case, it must be moved to the correct location before it can be picked. In these cases, order lines and shipments are given a state of *move pending*.

The next stage is to physically pick the stock. Only shipments that have a state of *pickable* can be picked.

Shipment Picking

OrderFlow provides many options for physically picking stock. These options can be broadly categorised into paper-based picking and handheld-driven picking.

Paper-based picking involves grouping pickable shipments into *batches*, and printing out a *picking list* (for each batch) that contains all the information needed for a 'picker' to go to the correct warehouse locations and pick the correct quantities of stock for all the shipments in the batch. For more details of batch picking, see the [Batches](#) section, below.

Handheld-driven picking requires OrderFlow to direct a 'picker' to each picking location via the handheld interface. At each location, OrderFlow instructs the picker how many units of which product to pick. One of the advantages of this method of picking is that the picking can be dynamically controlled, with the most optimal picking route determined 'on the fly'. Another advantage of this method is that the real stock position is reflected in real time in OrderFlow.

The picking state and picking locations of each shipment can be viewed from the Picking dashboard which is accessed from the 'Despatch' tab.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin 2/2 sites 1/7 organisations Unset (support) ?

Picking

Batched shipment picking state

Batch ID	State	Details	Action
1940	Ready for batching	Not yet associated with a batch	+ New
64	In hand held batch	Picking not yet started	
16	In hand held batch	Currently being picked	

Search batches

Individual shipment picking state

Count	State	Action
26	Shipments awaiting picking.	
0	Shipments currently picking.	
0	Shipments awaiting packing.	

Used batch picking locations

Used mobile batch picking locations (e.g. carts) for are shown below, together with associated handheld picking batches. The current stock quantity held in the location is also shown.

Reference	Associated Batch	Associated Task	Stock Qty	Allocation Qty
cart1			16 (check location)	16
cart2			4 (check location)	4
cart3			15 (check location)	15
cart6			6 (check location)	6
cart7			50 (check location)	48

Mobile assignments

The table below shows mobile locations which hold picking assignments for shipments. Shipments picked and in transit to the packing desk appear in this list.

Site	Shipment	State	Location	Product	Quantity	Allocated
Swindon	A233400031108	Cross docking	packingtote3	TRAN805418	1	1
Swindon	A233400031150	Cross docking	packingtote3	TRAN805418	1	1
Swindon	A233400031257	Cross docking	packingtote2	TRAN805425	1	1
Swindon	A233400031264	Cross docking	packingtote2	TRAN805418	1	1
Swindon	A233400031299	Cross docking	packingtote2	TRAN805425	1	1

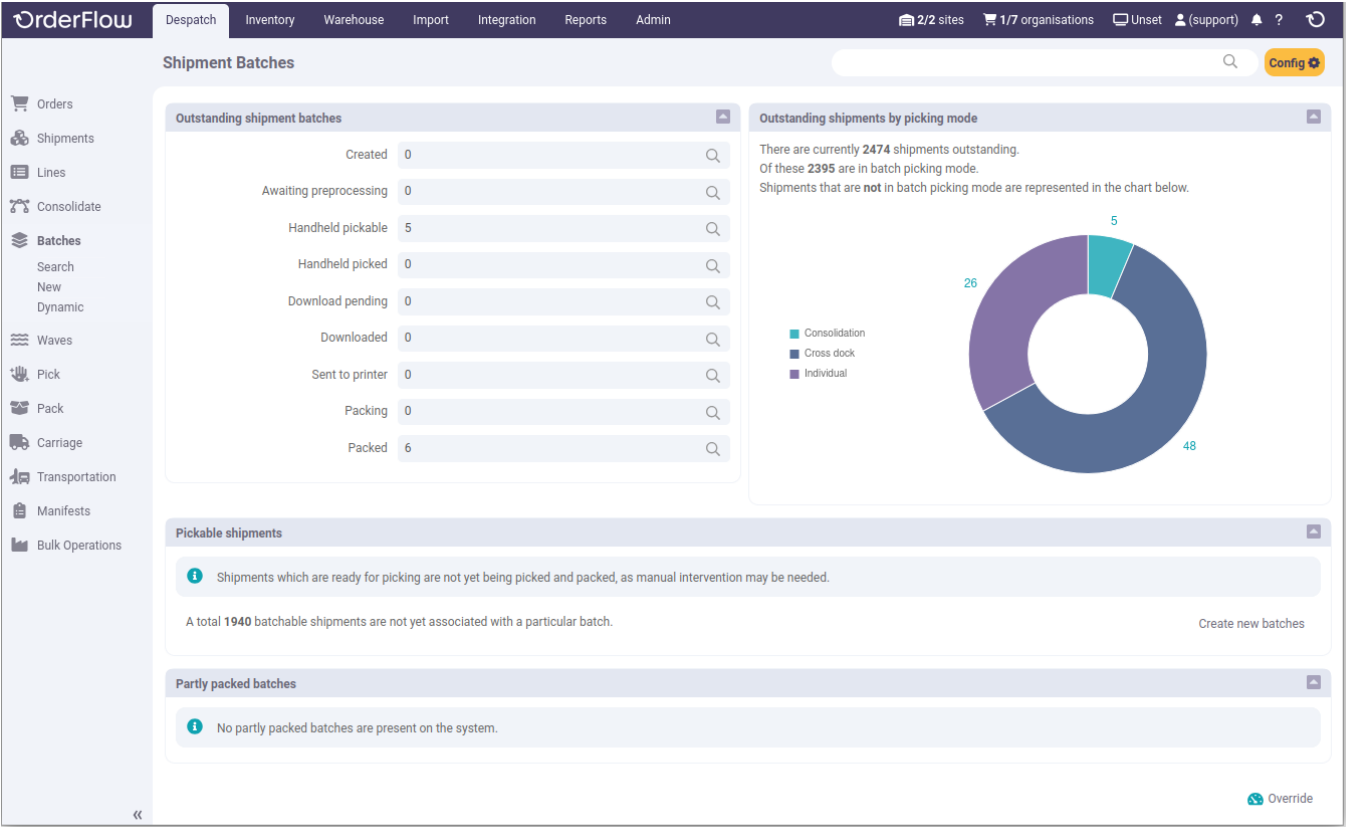
Once a shipment is picked using whatever option OrderFlow is configured to use, it is usually given a state of *packable*.

The next stage is to pack the stock. Only shipments that have a state of *packable* can be packed.

Batches

A **Batch** is a grouping of shipments that are picked together. Organising shipments into batches can fulfil a number of purposes.

- related shipments can be grouped together to support a simpler or more efficient picking operation. For example, single-line shipments can be picked more efficiently than multi-line shipments, so it makes sense to pick them together. Also, it can often be more efficient to pick together shipments that share the same priority or courier.
- for paper-based picking processes, a consolidated picking list can be printed for all of the shipments in the batch. It also makes sense in some cases to also print all of the customer paperwork (despatch notes or invoices) as part of a batch print operation. This can greatly save on time that would otherwise be required for per-shipment print jobs. This requires the use of high-end, fast printers, but allows for the preparation of paperwork in bulk as part of a background task.
- batches can also be used to support particular workflow processes. The obvious example is picking, either with the help of paper reports or handheld terminals. In some cases, other batch operations can be supported, such as marking all shipments in a batch as packed or despatched.



Shipment Packing

OrderFlow supports many different ways to pack shipments.

Shipment Packing

The most simple way is to simply mark all shipments in a batch as 'packed'.

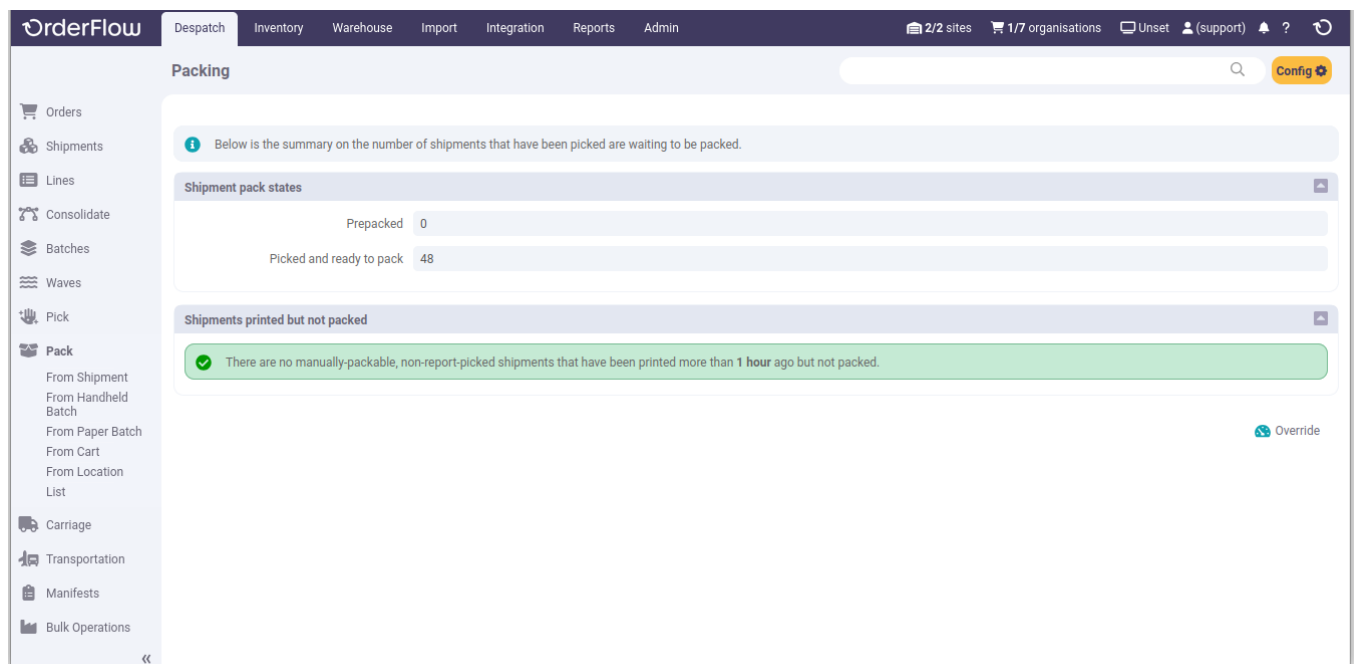
This is effective if the batch is paper-based and the shipment paperwork (including integrated courier labels) was already generated and printed before the batch is picked. In this case, a 'packer' would simply place the correct stock in the correct shipment packaging, and inform OrderFlow that all the batch's shipments have been packed. OrderFlow would have to trust the packer to put the correct stock in the correct packaging. (Note that this may constitute several packages.)

Some organisations want tighter control over packing, i.e. they want to be sure that no mistakes are made during the packing process, which would result in the wrong goods being despatched. This may be the case if they stock many different products, or if products are very similar and can't easily be distinguished by eye.

In this case, OrderFlow can be configured to require the packer to scan the items in a shipment before it can be marked as packed. (This process is known in OrderFlow as *packing consolidation*.) This unambiguously identifies the products and immediately flags any unexpected products (and indeed any incorrect quantities) to the packer. Only when all the correct products in the correct quantities have been scanned can a shipment be marked as packed. This is achieved from the 'packing screen' in OrderFlow.

Another reason that OrderFlow may require the packer to access the 'packing screen' for a shipment is that the courier label may need to be printed separately from any customer paperwork. (This is especially the case if a desktop courier system is being used - see the **Courier Integration Guide** for more details.) In this case, the courier label would typically be printed on a thermal printer local to the packer's 'packing station' (i.e. PC).

The packing operations on OrderFlow can all be controlled from the menus under the 'Packing' dashboard:



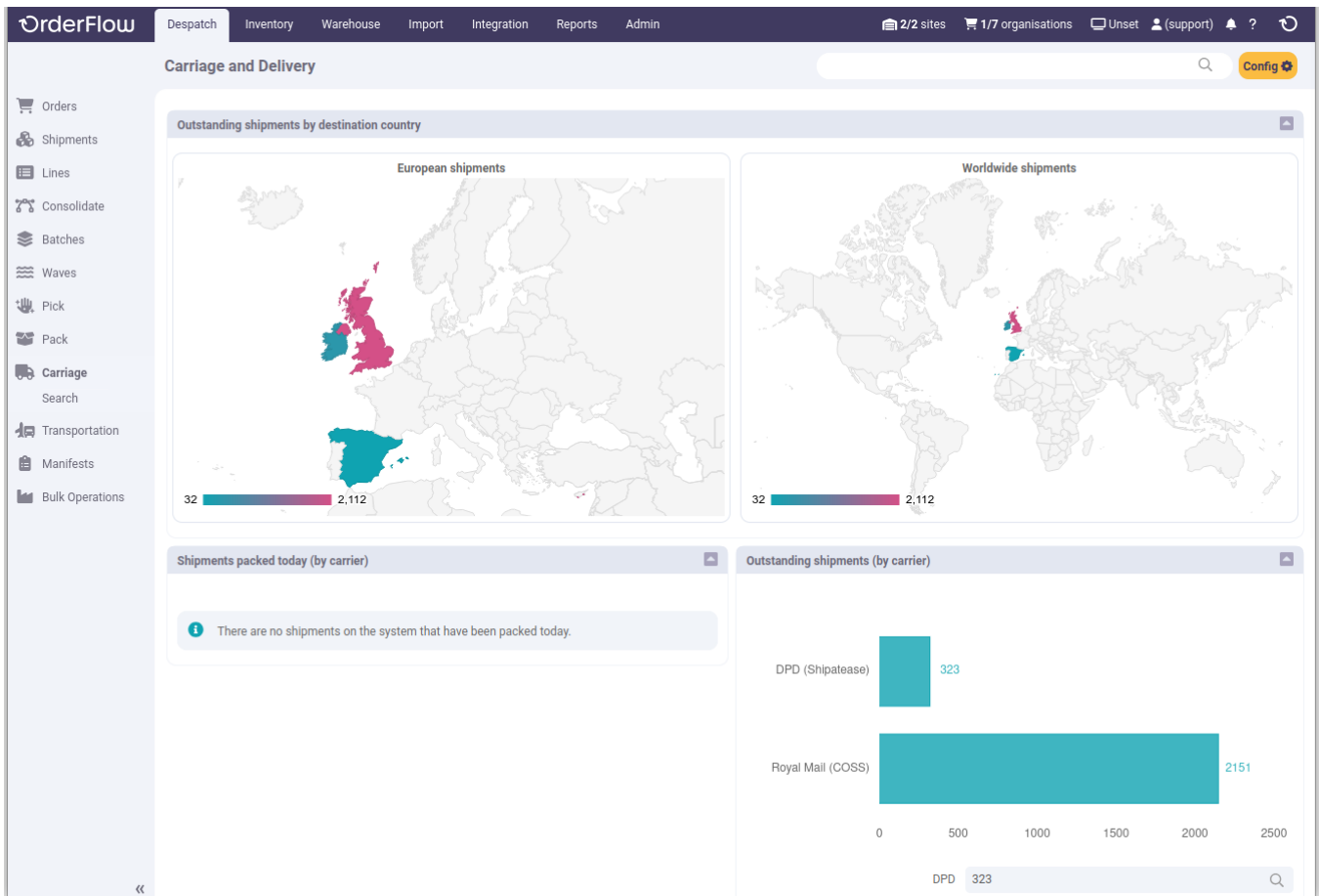
As part of the 'packing consolidation' process, OrderFlow supports other refinements, such as requiring the packer to scan a product's batch code, or to supply a quantity instead of scanning every product.

Typically, once a shipment is packed, the stock is debited from the warehouse. OrderFlow can be configured to perform this action later on, at shipment *despatch* time, if required.

Once a shipment is packed it is given a state of *packed*.

Courier Selection

In parallel with getting the stock ready for each shipment's despatch, OrderFlow progresses the selection of an appropriate courier for the shipment, to facilitate the actual shipping of ordered items directly to an end user's address. The 'Carriage and Delivery Dashboard' provides information relating to this:



A courier can be selected in two ways:

- **manually:** the user explicitly chooses a carrier and service from one of the available choices, or
- **automatically:** via the application of a 'Courier Selection' script at some point in the shipment's progression through the workflow.

A courier for a shipment can be selected at any time from the Receive Order to Pick shipment stages. When a courier can be selected in the process is configurable.

Automatic selection via script can use any of the shipment or order's attributes to determine which courier, service and options to use. For example, the shipment's destination and weight, and the priority of the order may influence this selection.

Courier Selection

If no courier selection has been made at the point of packing, the user will typically be redirected to a manual courier selection screen before being allowed to proceed to the packing screen. An example of this screen is shown below. Depending upon the courier and service selected, there may be additional validation that needs to be performed, which may involve real-time communication with a courier's system.

The 'courier state' of a shipment is progressed separately to the shipment state.

Courier Selection

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset(support)?

OrdersShipmentsSearchLinesConsolidateBatchesWavesPickPackCarriageTransportationManifestsBulk Operations

Shipment Courier Selection

Please specify your courier selection here. The shipment courier state is **selection complete**.

Before moving onto the packing screen, it needs to be **validated**.

Courier Details

Shipment

A23340003999 to JOHN B MOORE

Detail view

Delivery Type

Courier to home

To Address

WILLOW COTTAGE, FRONT STREET, HUTTON HENRY, HARTLEPOOL, CLEVELAND, SN15 5ND, GB

Priority

Normal (1)

Weight

100 gram

Courier

Royal Mail

Royal Mail (COSS)

Carrier

Royal Mail

Service

Next Day

Modify courier selection

☐

Bulk Transport

☐

DHL

☐

DPD (Shiptease)

☒

Royal Mail (COSS)

☐

Royal Mail (PPI)

☐

Scurri

☐

UPS

☐

USPS (Endicia)

Please select from one of the available courier services.

☐

High Volume

☒

Next Day

☐

Standard

Please select from each of the available courier service options.

Signature required

☒

Signature

☐

No signature

Apply selection

Back to Shipment

Just-in Time Fulfilment

The objective that underpins JIT fulfilment is to minimise the necessity to hold and double-handle stock and, where possible, to fulfil outstanding orders directly from stock received through incoming deliveries.

There are significant differences between the way that OrderFlow applies JIT fulfilment for single-line versus multi-line shipments.

- for single-line shipments, it is normally possible to send received stock directly to the packing area as soon as they come in. So no storage of these products is required at all.
- for multi-line shipments, temporary storage of the incoming stock is required. This is because the stock for the different order lines will be received at different times, possibly through separate deliveries and from separate suppliers.

Therefore for multi-line shipments, a pure JIT approach is not possible, because order lines for multiple products must be *consolidated* before the shipment can be shipped. OrderFlow achieves this through a JIT consolidation process. Items received are held in shipment-specific temporary storage locations in a consolidation area. These are normally close to the incoming and packing areas of the warehouse. Once all of the stock for all of the lines in a multi-line shipment have been consolidated in the single location, the shipment can be picked and packed. For more details of the consolidation process, see the [Consolidation](#) section, below.

One variation of the JIT model for multi-line shipments involves fulfilling items partly from stored stock. For example, if some stock for out-of-stock multi-line shipments is present in the warehouse, it is possible to consolidate the shipment order lines partly from incoming stock via the JIT process, and partly from stored stock.

Consolidation

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.

Typically, an OrderFlow warehouse that supports this process will have a 'consolidation area', which will contain many *consolidation locations*.

Depending upon the size of the products, a consolidation location may be a pallet, a shelf or a 'pigeon hole'. Each consolidation location will be associated with at most one shipment at a time.

Stock for a multi-line shipment will get directed to the specific consolidation location for that shipment, as it becomes available. (This could be as a result of an incoming delivery, or because of a stock move from bulk storage.)

Once *all* the stock for a shipment is present in the consolidation location, the shipment is considered to be 'consolidated'. OrderFlow then marks it as *pickable*, which means it can then be picked and presented to a 'packer'.

When a consolidated shipment is picked from a consolidation location, the location is empty and can then be re-used for another consolidating shipment.

The 'Consolidation' dashboard shows the state of consolidating shipments in OrderFlow. The 'Consolidation Locations' screen below shows the consolidation locations in use.

Just-in Time Fulfilment

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset(support)?

OrdersShipmentsLinesConsolidateLocationsOperationsAdminBatchesWavesPickPackCarriageTransportationManifestsBulk Operations

Locations

Please select a site: Swindon

Below are the locations being used for consolidation in site Swindon.

Used consolidation locations

The following view shows consolidation locations currently being used.

Location	Shipment	Priority	State	Product	Required	Present	Shortfall
consol_01	A233400032398	Not set	Consolidated	TRAN805418	1	1	0
				TRAN805422	1	1	0
consol_02	A233400032037	Not set	Consolidating	TRAN805408	1	0	1
				TRAN805425	1	1	0
consol_03	A233400032356	Not set	Consolidated	TRAN805418	1	1	0
				TRAN805422	1	1	0
consol_04	A233400032235	Not set	Consolidating	TRAN805408	1	0	1
				TRAN805425	1	1	0
consol_05	A233400032349	Not set	Consolidating	TRAN805408	1	0	1
				TRAN805425	1	1	0
consol_06	#1015	Not set	Consolidating	3607681850145	1	0	1
				3607681850152	1	0	1
				632059936483	1	1	0
consol_07	#1024	Normal	Consolidating	12356011005	1	0	1
				3607681832035	1	1	0

View free consolidation locations in this site

View orphaned consolidation locations in this site

Warehouse Management

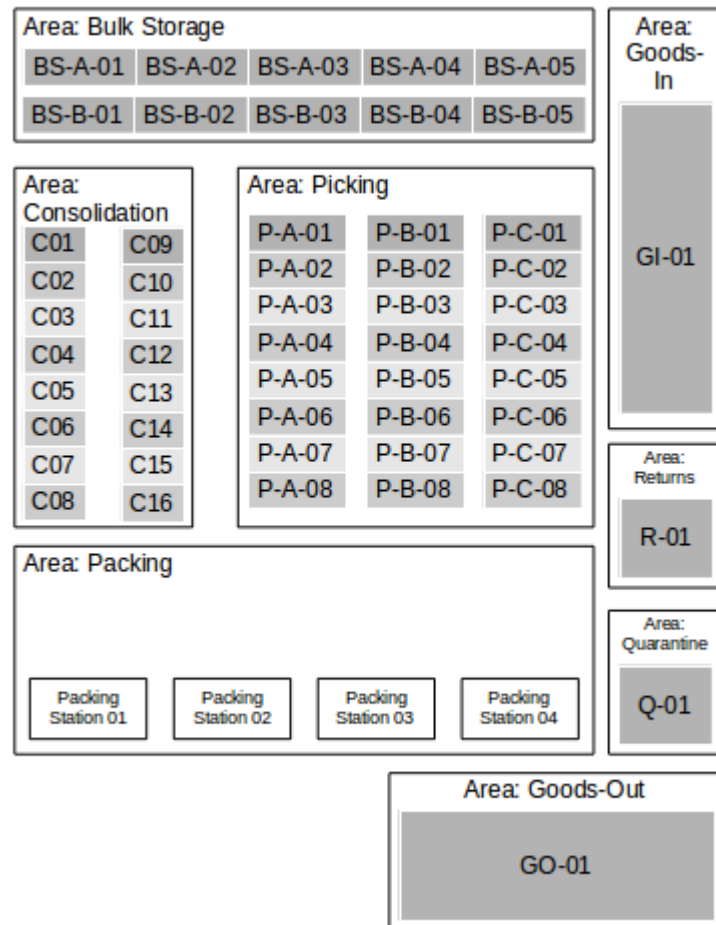
OrderFlow has user-friendly Inventory and Warehouse functions which make it easy to manage stock in an efficient and cost-effective way.

Stock can be placed in any number of different **locations** in a warehouse. In OrderFlow, these locations can be configured to be of various types that allow for fine-grained usage and control. For example, locations such as shelves, pallets, picking carts, totes, fridges, consolidation 'pigeon holes' (and so on) can all be created.

Locations can be assigned to **areas** within a warehouse, to allow for easy compartmentalisation of the available warehouse space. For example, locations can be defined as picking locations, bulk storage locations or quarantine locations.

This chapter describes how OrderFlow manages the stock in locations, and how it makes such stock available for various warehouse operations.

The diagram below shows a simple example warehouse layout, using OrderFlow terminology. The shaded areas denote locations.

Warehouse: **Swindon**

Terminology

Throughout this chapter, reference will be made to **sites**, **areas** and **locations**. The relationship between these is as follows:

- An OrderFlow customer will have one or more sites; a site typically being an individual warehouse.
- A site is made up of one or more areas.
- An area is made up of one or more locations.

Locations

A location is simply a defined place in a site where stock can reside.

Locations can be configured to hold stock for just a single product, or they can be configured to contain units of multiple products. OrderFlow enforces such restrictions throughout.

Locations can be configured to have a number of different attributes, such as aisle, bay, level, position, side, dimensions. They can also be given various indicators to allow them to be ordered in different ways for different purposes, such as picking, replenishment and stock checking. (More on this later.)

Locations are defined to be of specific *Location Types*.

Location Types

OrderFlow categorises locations into different types, to help it manage the flow of stock more easily. **Location Types** are configurable entities that can have any combination of various defined attributes. This allows for fine-grained control of how locations are used, particular to the specific customer needs.

There are many attributes of a location type - some commonly-used ones are detailed here:

- *pickable*: Determines whether this location can be used as a source location for picking operations. If stock is only in non-pickable locations, then *replenishment* of picking locations must occur before picking can proceed.
- *storage*: Indicates that the location can be used for storage. Only *storage* locations will typically be considered as target locations for *putaway* operations.
- *multiProduct*: Indicates that a location is designed to contain multiple products.
- *incoming*: Used specifically for locations for receiving incoming deliveries.
- *damaged*: Used to hold stock that is considered to be damaged.
- *mobile*: Used for locations which do not have a fixed physical position in the warehouse. Physically, may be a cart, cage, trolley, tote or some other mobile container. Used extensively in handheld scanner-based operations.

Products

Products are the shippable goods that are received into the warehouse via deliveries, stored in locations, and sent out in shipments following the receipt of orders into OrderFlow.

Products are usually counted in *units* and generally referred to as **stock** (when talking about physical instances of a product).

Products can be categorised according to physical attributes and also logically typed. For example, an 'instruction manual' product could be categorised as a 'book' of 'default' type, whereas a 'tool kit' product could be categorised as a 'tool' of type 'bundle'.

Typical products types are:

- *default*: A sellable stock-based product, eligible for stock notifications
- *virtual*: A virtual product, for which no stock is held, but is sellable. No fulfilment for a product of this type is required.
- *bundle*: A bundle is sellable product which is composed of underlying (typically) stock based products. It is also a virtual product in the sense that there is no physical representation of the bundle, only the underlying constituent products.

Inventory

The term 'inventory' is used in OrderFlow to mean the overall view of the stock position, without going into detail as to exactly *where* that stock resides.

For example, if a product is held in differing quantities in many different locations in a warehouse, the *inventory* for that product would include the *total* number of units across all those locations.

The inventory would also detail how much of that total quantity was *usable*, (i.e. excluding any damaged or quarantined stock), and how many units of that product were *required* by pending order lines.

The *available* quantity can be derived as the usable quantity less the required quantity. Note that this can be negative if there is insufficient stock in the warehouse to fulfil all order lines for that product. (Such a negative amount is called the *shortfall* of a product.)

Tasks

Stock move **tasks** are used by OrderFlow to manage many warehouse processes. These are essentially instructions to move explicit quantities of products from one location to another. They are also used internally to drive certain processes.

There are many different types of tasks - some of these are listed here:

- *putaway tasks*: where stock is moved from incoming delivery bays to storage locations.
- *replenishment tasks*: where stock is moved from bulk non-pickable storage locations to pickable locations.
- *picking tasks*: the process of picking stock to fulfill orders, especially when using a handheld terminal.
- *consolidation tasks*: used when multiline shipments are routed into a consolidation area prior to packing.

When a stock move task is executed and applied, **stock changes** are created in OrderFlow. These are immutable records whose purpose is to preserve an audit trail of "what happened when".

Warehouse Management Features

This chapter describes the most commonly used stock management features provided by the OrderFlow system.

Goods In

In day-to-day warehouse operations, stock is added to the warehouse through the receipt of a **delivery**. A *delivery* is the arrival of physical goods at the warehouse door.

OrderFlow represents this operation as a 'Delivery' object, which may or may not have 'Delivery Lines' attached, depending upon whether the contents of the delivery are known in advance.

Warehouse Management Features

The contents of a delivery may be known in advance if OrderFlow is informed of an *Advanced Shipping Note*, or indeed if the delivery is in direct response to a *Purchase Order*. More details about these concepts can be found in the **Warehouse Processes Guide**.

If the contents of a delivery are not known, then the items in the delivery are scanned and quantities confirmed. OrderFlow then creates the delivery lines from the supplied information.

Once a delivery is *applied* (either incrementally or in one go), the stock in OrderFlow is credited. The delivery detail page (shown below) shows all delivery-related information.

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset (support) ?

Delivery Detail: DMS676565026-210609

Back to list

Purchase Orders

ASNs

Loading Docks

Deliveries

Search New

Work Orders

Licence Plates

Stock Move Tasks

Picking Sequences

Stock Checking

Returns

Quality Assurance Actions

Delivery details

ID246

SiteSwindon

OrganisationElectricEtail

TypeHandheld from Licence Plates

StateReceiving

Purchase OrderDMS676565026

Purchase Order StatePartially applied

CommentsNo comments have been left for this Delivery.

Licence Plates4 in total
2 of which are open awaiting receipt

SupplierDMS Imports

Supplier ReferenceDMS676565026-210609

Created22-Aug-2024 13:28:58 (by charlie)

Last Updated22-Aug-2024 09:03:03

PrintedNo

+ Add a comment to this Delivery

No more stock to add to licence plates? Click to view option

Delivery lines

So far, this delivery contains 4 lines.

Product	Description	State	Location	Quantity	Date Added
TRAN805422	8GB ULTRA-SPEED SAMSUNG USB FLASH ...	Applied	incoming_001	12	22/08/24 09:03:03
TRAN805412	500GB PORTABLE EXTERNAL ANTI-SHOCK ...	Applied	incoming_001	7	22/08/24 09:05:16
TRAN805404	250GB PORTABLE EXTERNAL ANTI-SHOCK ...	Applied	incoming_001	1	22/08/24 09:05:45
TRAN805422	8GB ULTRA-SPEED SAMSUNG USB FLASH ...	Applied	incoming_001	5	21/08/24 13:30:58

The stock credited to OrderFlow can be handled in various different ways. The most simple is for it to be credited to an *incoming* location and then *put away* to pickable or storage locations. This is achieved via a *stock move task*, detailed in the following section.

More complex methods of processing delivered stock include routing straight to packing (*cross-docking*), to *consolidation*, or a combination of these (and more). More details on this can be found in the **Warehouse Processes Guide**.

Stock Move Tasks

At their most basic level, stock move tasks are simply instructions to warehouse operators to move one or more products from one or more *source* locations to one or more *target* locations.

Warehouse Management Features

They can be used for all sorts of processes - the simplest being *putaway* or *replenishment* processes, where stock is routed to or from storage locations (respectively). (Putaway processes can also route stock to pickable locations.)

Stock move tasks are also be used to drive the picking process - even dynamically when handheld devices are in use. Other common uses of stock move tasks include routing stock from storage to consolidation locations.

The screenshot displays the OrderFlow Warehouse Management interface. The top navigation bar includes 'OrderFlow', 'Despatch', 'Inventory', 'Warehouse' (selected), 'Import', 'Integration', 'Reports', and 'Admin'. The right side of the bar shows '2/2 sites', '1/7 organisations', 'Unset', 'support', and a search icon. The left sidebar lists various warehouse management functions: Purchase Orders, ASNs, Loading Docks, Deliveries, Work Orders, Licence Plates, Stock Move Tasks (selected), Picking Sequences, Stock Checking, Returns, and Quality Assurance Actions. The main content area is titled 'Stock Move Tasks' and contains three sections: 1. 'Open stock move tasks': A table listing recent tasks with columns for ID, Site, Organisation, Type, State, Source lines, and Target lines. 2. 'Products to put away': A table listing products with columns for Reference and Quantity. 3. 'Background replenishment (from storage)': A message stating there are no products to background replenish. Below these sections is a 'Stock move tasks applied today' section with a message stating there are no tasks applied today. An 'Override' button is visible in the bottom right corner.

ID	Site	Organisation	Type	State	Source lines	Target lines
746	Swindon	ElectricEtail	Pick for manufacturing	Ready	1	1
703	Swindon	ElectricEtail	Pick for manufacturing	Ready	3	3
735	Swindon	ElectricEtail	Pick for manufacturing	Ready	1	1
761	Swindon	ElectricEtail	Batch pick	In progress	4	4
771	Swindon	ElectricEtail	Route from incoming	In progress	1	1
748	Swindon	ElectricEtail	Batch pick	In progress	2	2

Reference	Quantity
TRAN805418	43
TRAN805422	13
TRAN805412	7
TRAN805404	1

Storage, Replenishment and Reordering

Products in OrderFlow can be configured to have specific attributes, that allow their movement and behaviour to be finely controlled. These setting can be applied per *site*, so different behaviour can be applied to different warehouses.

For example, a *storage area* or a more specific *storage location* can be defined for a product, which directs OrderFlow where to put away stock for this product.

Similarly, a *primary picking location* can be set, which restricts the typical picking processes to that location. In this case, *replenishment* of stock from storage location(s) to the primary picking location must occur before this product can be picked.

Also, a *reorder threshold* can be set for a product, which defines the minimum number of product units that are held in a warehouse before it is included in a *reorder report*. This will ultimately result in the product being included in a purchase order which, when imported into OrderFlow, will show the product as being 'on order'.

The following screenshot shows the 'Warehouse Product' configuration details that can be applied:

Warehouse Management Features

Warehouse product details

The following details have been configured for product **MURRANGRYMANPA330ML** for warehouse **Swindon**.

Strategy and Locations

Picking Strategy

Not set

Primary Picking Location

Not set

Primary Storage Location

Not set

Replenishment Area

Not set

Storage Area

Not set

Physical Storage Type

Not set

Licence Plate Storage

Default Container Type

Not set

Licence Plate Quantity Max

Not set

Replenishment

Pick Location Max

Not set

Pick Replenish Threshold

Not set

Primary Storage Max

Not set

Storage Replenish Threshold

Not set

Reorder

Reorder Threshold

50

Target Quantity

2000

Miscellaneous

Excess Receipt Approval (%)

Not set

Use the **Escape** key to exit.

Edit warehouse product details

Returns

When the recipient of an order returns a product (or products) for any reason, they usually send those products back to the warehouse from where they were despatched.

OrderFlow supports processes to allow these goods to be checked back into (typically) a quarantine location, ready for whatever reconditioning or repair needs to take place before returning to stock. Sometimes the goods have to be sent back to the original supplier, so will not be returned to the normal storage / pickable stock for re-use.

This 'returns' process allows the user to record the condition of the returned goods, the customer's reason(s) for returning the goods, and (possibly) whether they believe a refund should be paid. The following screen shot shows the detail of a return line:

© OrderFlow Ltd. 2025

Page 34

Further Reading

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset(support)?

Purchase Orders

ASNs

Loading Docks

Deliveries

Work Orders

Licence Plates

Stock Move Tasks

Picking Sequences

Stock Checking

Returns

Search

New Authorised

New Non-authorised

Quality Assurance

Actions

«

Return Item Detail

Config

Back to list

Return Details

ID8

Created02-Sep-2024 12:58:03 (by charlie)

SiteSwindon

Last Updated02-Sep-2024 12:58:03 (by charlie)

OrganisationElectricEtail

StateCreated

ChannelNone

Return DateNot set

TypeWrong item

Pre-authorizedYes

AuthorisationRMA00437

Order ReferenceA233400032423

NoteCustomer will post back today

CommentsNo comments have been left for this Return.

+ Add a comment to this Return

EditDelete

Default return locations

Stock: quarantine_3

Quarantine: quarantine_1

Destroy: quarantine_2

Return Lines

This return item does not contain any lines.

Original Order Lines

This return is for the order A233400032423. The original lines for that order are shown below.

Product	Description	Quantity	
TRAN805422	8GB ULTRA-SPEED SAMSUNG USB FLASH DRIVE MEMORY STICK	1	+ Add Return For Order Line
TRAN805431	NEW 8GB CLASS 6 FLASH MEMORY READYBOOST SDHC SD-HC CARD	1	+ Add Return For Order Line

Further Reading

Warehouse Processes Guide (to be published)

Advanced Features

OrderFlow has many other features that have not yet been covered in this document. This section aims to introduce a few of them, guided by their position in the desktop user interface.

Import

As we have already seen, OrderFlow has the potential to require a large amount of configuration data in order to support the more complex warehouse layouts and processes. Keying-in this configuration data would be time-consuming and prone to error, so it is made possible for such setup / configuration data to be *imported* from spreadsheets etc.

In addition to configuration data, transactional data such as orders needs to be imported, again to avoid having to key-in such data. OrderFlow supports many different data import mechanisms and formats, with the aim of being flexible enough to integrate with other systems with the minimum of effort.

For example, it is possible to import orders from comma-separated value (CSV) files, spreadsheets, or richer data formats such as XML or JSON. Imported data can be automatically transformed into more-expected formats so that OrderFlow can process it more easily. OrderFlow can be *pushed* orders from a client that uses OrderFlow's application programming interface (API), or that places files on its FTP server, for example. Alternatively, OrderFlow can be configured to *pull* orders from a remote system (e.g. a shopping cart).

The history of all import attempts, both successful and failing, are recorded in OrderFlow's **import history**, so that it is always possible to view how data arrived on the system. The screenshot below gives an example of such a history.

Import

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset(support)?

Import Batches

FetchUploadHistoryBatchesErrorsFiles

Search criteria

EntityReference

AreaAsnBarcodeCampaignCarrierCountryCountry PropertyCourier EventCourier ServiceCustomerDeliveryGrouped ProductLicence PlateLocationLocation AttributeMenuOrderPackagingProductProduct AttributeProduct CategoryProduct DatasheetProduct ImageProduct LocationProduct LotProduct Relationship

Site

More search fields ...

ResetSearch

Import batch search results

ID	Type	
117304	csv	Mager
117303	csv	Sedge
117302	csv	Sedge
117301	csv	Sedge
117300	csv	Sedge
117299	csv	Sedge
117298	csv	Sedge
117297	csv	Sedge
117296	csv	Sedge
117295	csv	Sedge

	Succeeded	Failed	Duplicates	Time Taken	Date	
1	0	0	287 ms	27/02/25 11:40:10		
0	1	0	149 ms	27/02/25 11:26:07		
0	1	0	78 ms	27/02/25 11:16:28		
0	1	0	91 ms	27/02/25 11:15:57		
0	1	0	68 ms	27/02/25 11:14:40		
0	1	0	96 ms	27/02/25 11:12:51		
0	1	0	73 ms	27/02/25 11:09:53		
0	1	0	87 ms	27/02/25 11:09:11		
0	1	0	26 ms	27/02/25 11:08:36		
0	1	0	77 ms	27/02/25 11:08:18		

Page 1 of 1Viewing 1 - 10 of 10

Note: the current search is limited to data up to 14 days old.

Integration

The **Integration** area of OrderFlow primarily deals with communications with external systems, both at the message level and at the application programming interface (API) level. It also holds the configuration for some of OrderFlow's internal processes, allowing them to be changed without the need for a new version of the codebase.

Remote Messages

The **Remote Messages** area of OrderFlow deals with communications with external systems. It shows all messages sent to external systems, and their corresponding responses.

Each remote message is of a certain *message type*, which is a configurable object that provides flexibility to have fine-grained control over *how* these messages are sent, and what the responses mean.

For example, an 'inventory notification' message type can be configured to be sent over HTTP, with a maximum of 5 retries, and that a successful message is one with HTTP response codes 200 or 401. Additionally, the response body must not contain the text ".

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsUnset(support)

Search Outgoing Messages

Search criteria - click to view criteria applied for this search

Remote message search results

ID	Scope	Purpose	Type	State	Reference	Retries	Created	Sent	Duration
1869038	Sedgebrook Beer and Wine	Delivery notification	Delivery notification	Created	305	0	30/10/24 11:25:37		
1869037	Global	Shipment validation	Ups api shipment confirm	Sent	#rtd-store-1081	0	30/04/24 15:03:10	15:03:12	1853 ms
1869036	Global	Fetch orders	Api shopify rest	Sent		0	27/03/24 00:02:55	00:02:56	782 ms
1869035	Global	Fetch orders	Api shopify rest	Sent		0	26/03/24 00:02:55	00:02:55	335 ms
1869034	Global	Fetch orders	Api shopify rest	Sent		0	25/03/24 00:02:55	00:02:55	283 ms
1869033	Global	Fetch orders	Api shopify rest	Sent		0	24/03/24 00:02:55	00:02:55	348 ms
1869032	Global	Fetch orders	Api shopify rest	Sent		0	23/03/24 00:02:55	00:02:55	366 ms
1869031	Global	Fetch orders	Api shopify rest	Sent		0	22/03/24 00:02:55	00:02:55	350 ms
1869030	Global	Inventory	Api shopify rest	Sent	9009518598628	0	21/03/24 10:40:06	10:40:06	312 ms
1869029	Global	Inventory	Api shopify rest	Sent	889212070793	0	21/03/24 10:40:05	10:40:06	449 ms
1869028	Global	Fetch orders	Api shopify rest	Sent		0	21/03/24 00:02:55	00:02:55	362 ms
1869027	Global	Fetch orders	Api shopify rest	Sent		0	20/03/24 00:02:55	00:02:55	300 ms
1869026	Global	Fetch orders	Api shopify rest	Sent		0	19/03/24 00:02:55	00:02:55	344 ms
1869025	Global	Fetch orders	Api shopify rest	Sent		0	18/03/24 00:02:55	00:02:55	288 ms
1869024	Global	Fetch orders	Api shopify rest	Sent		0	17/03/24 00:02:55	00:02:55	325 ms

Page 1 of 3

Viewing 1 - 15 of 41

Note: the current search is limited to data up to 730 days old.

API Definitions

This area of OrderFlow allows an OrderFlow user to configure entire external APIs without code changes. This allows the specifics of external interfaces to be kept separate from OrderFlow's internal processes and logic. For example, a process to fetch a product catalogue from a remote system may consist of one API request, or it may consist of several requests and responses. OrderFlow's API configurations keep this detail separate from its core processing.

It also gives a high degree of power and flexibility to change how OrderFlow interacts with remote systems 'on the fly'. An example of this would be if a particular API to 'pull' orders from a remote shopping cart system was configured, and the remote system changed its status codes, then those changes could be reflected in OrderFlow immediately.

OrderFlow
Despatch Inventory Warehouse Import Integration Reports Admin Setup Advanced
2/2 sites 1/7 organisations Unset (support) ?

API Definition
Operations

Message Types
API Definitions
Process Definitions
Processing Instructions
Connection Test

Summary Info

Reference magento2
Name Magento 2
Type Shopping cart
Module rtd2-integration-magento2
Default Message Type api_magento2_soap
Options This API does support inventory notification

Edit API definition

Versions - click to view versions for this API

Operation definitions

Operations for this API definition are shown below.

Reference	Name	Entity	Synchronous	
fetch_order_soap	Fetch order (SOAP)		✓	✗
fetch_product_soap	Fetch product (SOAP)		✓	✗
inventory_notification_soap	Inventory Notification (SOAP)		✗	✗
notify_sales_owner_allowance_changes ...	Notify sales owner allowance changes (Magento 2 SOAP)		✗	✗
shipment_despatch_notification_soap	Shipment despatch notification (SOAP)		✗	✗

Process configurations

Process configurations linked to this API are shown below.

Configuration	Definition	Version	Detail	Activated	
Q	Notify sales owner allowance changes		(Built-in)	No	✗
Q	Refetch failed products		(Built-in)	Yes	✗
Q	Remote order update		(Built-in)	Yes	✗
Q	Remote product update		(Built-in)	Yes	✗
Q	Embedded inventory notification		inventory_notification_invoke	Yes	✗
Q	Shipment event notification		(Built-in)	Yes	✗
Q	Fetch single order		(Built-in)	Yes	✗
Q	Fetch single product		(Built-in)	Yes	✗
Q	Fetch updated products		(Built-in)	Yes	✗

Process Definitions

OrderFlow's 'Process Framework' provides a powerful framework to control many of its internal processes, and to define and configure new ones.

The motivation behind this is to enable OrderFlow to be more dynamic in terms of being able to handle changing requirements *without* the need to develop, test, build and deploy new code. The framework allows logic to be encapsulated in scripts, reports, API operations and other constructs that control the process. Changing the processes has immediate effect.

For example, a process to re-fetch all failed orders might invoke a report to query *which* orders have failed to import successfully, then it might invoke an API operation to explicitly fetch details of those orders. Depending upon the scope of the process that is being run, a specific API will be invoked. This may be different for different scopes (e.g. sales channels).

Another example is a more simple 'shipment despatched notification' process.

This would be configured to invoke an API operation to notify a remote system that a shipment has been despatched. The variation here would be in the API operation - one or more remote messages may end up being sent in this case.

Integration

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support) ?

Message Types

API Definitions

Process Definitions

Test

Processing Instructions

Connection Test

Process definitions

Operations

Filter by type:

Show inactive processes

Below is a list of process definitions for the current application.

Type	Reference	Name	Site Global	Activated
API Other	ezrentout_fetch_stock_updates	EZRentout Fetch Stock Updates	✓	Yes ✗
API Other	ezrentout_inventory_notification	EZRentout inventory notification	✓	Yes ✗
API Robotics	export_updated_containers	Export updated containers	✓	Yes ✗
API Robotics	export_updated_products	Export updated products	✓	Yes ✗
API Shopping Cart	delivery_applied_notification	Delivery applied notification		Yes ✗
API Shopping Cart	delivery_line_applied_notification	Delivery line applied notification		Yes ✗
API Shopping Cart	embedded_inventory_notification	Embedded inventory notification	✓	Yes ✗
API Shopping Cart	fetch_catalogue	Fetch catalogue	✓	Yes ✗
API Shopping Cart	fetch_new_orders	Fetch new orders	✓	Yes ✗
API Shopping Cart	fetch_new_purchase_orders	Fetch new purchase orders		Yes ✗
API Shopping Cart	fetch_open_orders	Fetch open orders	✓	Yes ✗
API Shopping Cart	fetch_open_purchase_orders	Fetch open purchase orders		Yes ✗
API Shopping Cart	fetch_outstanding_products	Fetch outstanding products	✓	Yes ✗
API Shopping Cart	fetch_product_barcodes	Fetch product barcodes	✓	Yes ✗
API Shopping Cart	fetch_product_images	Fetch product images	✓	Yes ✗
API Shopping Cart	fetch_single_order	Fetch single order	✓	Yes ✗
API Shopping Cart	fetch_single_product	Fetch single product	✓	Yes ✗
API Shopping Cart	fetch_updated_products	Fetch updated products	✓	Yes ✗
API Shopping Cart	order_event_notification	Order event notification		Yes ✗
API Shopping Cart	order_imported_notification	Order imported notification	✓	Yes ✗
API Shopping Cart	product_activated_notification	Product activated notification	✓	Yes ✗
API Shopping Cart	product_imported	Product imported	✓	Yes ✗
API Shopping Cart	purchase_order_completed_notification	Purchase order completed notification		Yes ✗
API Shopping Cart	recent_despatched_shipments_notification	Recent despatched shipments notification		Yes ✗
API Shopping Cart	recent_inventory_notification	Recent inventory notification	✓	Yes ✗
API Shopping Cart	refetch_failed_orders	Refetch failed orders	✓	Yes ✗
API Shopping Cart	refetch_failed_products	Refetch failed products	✓	Yes ✗
API Shopping Cart	refetch_failed_purchase_orders	Refetch failed purchase orders		Yes ✗
API Shopping Cart	refetch_product_barcodes	Refetch failed product barcodes	✓	Yes ✗
API Shopping Cart	remote_order_update	Remote order update	✓	Yes ✗
API Shopping Cart	remote_product_update	Remote product update	✓	Yes ✗
API Shopping Cart	return_notification	Return notification		Yes ✗
API Shopping Cart	shipment_event_notification	Shipment event notification		Yes ✗
API Shopping Cart	shipment_split_notification	Shipment split notification		Yes ✗

Reports

The **Reports** area of OrderFlow is where all the reports used by the system are configured. Some of them can also be run from the user interface, for display or download.

A 'report' is simply the extraction of data from OrderFlow's underlying database, presented in a particular format. Example formats could be comma-separated values (CSV), Excel spreadsheet (XLS), Portable Document Format (PDF), text or Jasper Reports XML format, which allows for more presentable output, e.g. despatch notes. More details on writing reports are available in the [OrderFlow Report Writer's Guide](#).

Reports are used both externally, where the data is sent or downloaded to an external system, or internally to OrderFlow, for example to drive certain processes, or to present 'dashboard fragments' on the user interface.

The following screenshot shows some of the *Admin* and *Alarm* reports configured.

The screenshot displays the OrderFlow Reports interface. The top navigation bar includes links for Despatch, Inventory, Warehouse, Import, Integration, Reports, and Admin. The Reports section is active, showing a 'Launch Reports' page. On the left, a sidebar lists various report categories: Launch, Favourites, Periodic, Saved, Billing, Activity, Productivity, Print Items, and Executions. The main content area shows a list of reports grouped by type. Below the 'Launch Reports' header, there is a search bar and a 'Config' button. The reports are organized into two main sections: 'Admin reports' and 'Alarm reports'. Each section contains a table with columns for Report Name, Description, Type, and Formats.

Report Name	Description	Type	Formats
Anonymisable Orders	Identifies orders that can be anonymised	Admin	text
Application Property List	Report showing list of application properties.	Admin	xls xlsx
Inventory Update	Inventory Update - To reconcile with Keylinks LV system for total stock quantities	Admin	csv
List of Country Names	List of country names on the system (to be used as a report filter on other reports)	Admin	text
Oldest Anonymised Order	Used to identify the oldest date for which there is an order that has been completed but not anonymised. Applies optionally for a channel.	Admin	text
Orders Full	Identifies all orders for a specified customer	Admin	csv xls xlsx
Product Stock Holding Snapshot	Product Stock Holding Snapshot.	Admin	csv

Report Name	Description	Type	Formats
Channel Import Error Alarm	Report to generate channel import error alarm records.	Alarm	csv
Configuration Exception Alarm	Report to generate configuration exception alarm contents.	Alarm	csv
Failed API Remote Entries Alarm	Identifies failed API remote entries and generates an application alarm	Alarm	csv
Organisation Import Error Alarm	Report to generate organisation import error alarm records.	Alarm	csv
Purge Error Alarm	Report to generate purge error alarm contents. ref #19596 (for testing, set to report successful)	Alarm ⓘ	csv
Remote Error Alarm	Report to generate remote error alarm contents.	Alarm	csv
Remote Error Alarm Mobile	Report to generate remote error alarm contents.	Alarm ⓘ	csv text
Remote Message Error Alarm	Identifies remote messages in an Error state after 3 retries and generates an application alarm.	Alarm	csv

The *Reports* section also contains information on user activity and performance, plus billing configuration and metrics that third-party fulfilment houses can use to bill their clients.

Reports

Finally, the **Print Items** sub-menu contains details of everything that OrderFlow has been requested to print, plus details of which *print servers* are connected. An example print item detail page is displayed below.

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdmin

2/2 sites1/7 organisationsPacking Station 1support

LaunchPeriodicSavedBillingActivityProductivityPrint ItemsSearchExecutions

Print Item 169

Details for the printed item 169 are shown below.

Printed Item

ID	169	Type	Despatch note
Site	Swindon	Entity	Shipment 0003
State	Ready	File Name	despatch_note_300095.xml
Created	27-Feb-2025 14:15:04	Format	jasper_report_xml
Batched	Yes	Number of Pages	1
		File Mime Type	application/xml
Queue	DEFAULT		
Report Key	rtd2-paperwork:despatchnote_simple		

View configuration

Download

File content

File length is 9267 bytes. The file is not stored in compressed form.

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperPrint xmlns="http://jasperreports.sourceforge.net/jasperreports/print" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports/print http://jasperreports.sourceforge.net/xsd/jasperprint.xsd" name="defaultDespatchNote" pageWidth="595" pageHeight="842" topMargin="0" leftMargin="20" bottomMargin="0" rightMargin="20" locale="en_GB" timezone="Europe/London">
  <property name="net.sf.jasperreports.export.xml.start.page.index" value="0"/>
  <property name="net.sf.jasperreports.export.xml.end.page.index" value="0"/>
  <property name="net.sf.jasperreports.export.xml.page.count" value="1"/>
  <origin band="pageHeader"/>
  <origin band="columnHeader"/>
  <origin band="detail"/>
  <origin band="columnFooter"/>
  <origin band="lastPageFooter"/>
  <page>
    <text textAlignment="Left" verticalAlignment="Middle" textHeight="10.937988" lineSpacingFactor="1.215332" leadingOffset="-1.8896484">
      <reportElement uuid="7d8c0c59-0c87-40a8-b854-193cb2927209" mode="Transparent" x="457" y="32" width="110" height="14" forecolor="#000000" origin="0" srcId="1" printId="1"/>
      <font fontName="Verdana" size="9" isBold="false"/>
      <textContent><![CDATA[0003]]></textContent>
    </text>
    <text textAlignment="Left" verticalAlignment="Top" textHeight="10.937988" lineSpacingFactor="1.215332" leadingOffset="-1.8896484">
```

Admin

The **Administration** area of OrderFlow deals with the day-to-day running of the system, from the set-up of user permissions and role definitions, through logging and error management, to displaying which scheduled jobs are running and have run.

It is expected that the system manager would use this menu to keep abreast of any problems encountered in a running system.

This area also shows system performance information and exposes configuration of the *archiving* and *housekeeping* operations within OrderFlow. These functions ensure that the system does not have to handle an ever-increasing amount of data, which would itself cause performance problems.

The following screenshot shows the *Memory Dashboard*, which details the memory utilisation of the OrderFlow instance's Java Virtual Machine (JVM).

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Memory

Views relating to memory use on the system.

Memory utilisation

Note that this view only shows the memory utilisation of the JVM to which the application is connected.

Name	Type	Used	Max	Percentage
CodeHeap 'non-nmethods'	Non-heap memory	1,887.4	5,836.8	32.3
Metaspace	Non-heap memory	191,490.9	-0.0	-19149115200.0
CodeHeap 'profiled nmethods'	Non-heap memory	50,774.9	122,908.7	41.3
Compressed Class Space	Non-heap memory	20,903.9	1,073,741.8	1.9
G1 Eden Space	Heap memory	6,291.5	-0.0	-629145600.0
G1 Old Gen	Heap memory	229,661.8	268,435.5	85.6
G1 Survivor Space	Heap memory	2,097.2	-0.0	-209715200.0
CodeHeap 'non-profiled nmethods'	Non-heap memory	20,154.2	122,912.8	16.4

Memory management

Setup

The **Setup** area of OrderFlow is where the bulk of the configuration relating to business processes is made. This configuration includes what **organisations**, **channels** and **sites** are defined and active in the OrderFlow instance. It also includes all the configurable **properties** in the system, which are instrumental in controlling how OrderFlow behaves.

There are also sub-menus to configure couriers, batch types, print queues, workstations and other entities. More details are given in the following sections.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin **Setup** Advanced 2/2 sites 1/7 organisations Unset (support) ?

System Setup

Welcome to the **setup** area of OrderFlow. The current system state is: **operational**

Currently on the system there is/are:

- 11 active couriers
- 6 batch types
- 4 print queues
- 3 workstations

Organisations

Organisation	Organisation State	Channel(s)	Channel State	Products
Sedgebrook Beer and Wine	Live	Manual Magento Trade	Test Live	31

Property Groups

Quick property group search links:

Activity	Handheld	Remote Order
Address	Import	Reports
Advanced Stock Check	Intersoft Sapiant	Returns
API2Cart API	Inventory	Royal Mail - Click & Drop
Apollo	Kardex API	Royal Mail - DMO
ASN	Licence Plate	Royal Mail - Tracked
AWS	Magento 2 API	S3
Batch Processing	Magento 2 Native API	Salesforce Commerce Cloud API
Cherwatch API	Magento API	Scout24
Consolidation	Messaging	Seller
Country Groups	Notification	Ship@Ease
Courier	Order Entry	Shopify API
Courier (Generic)	Pack	Shopify GraphQL API
Courier (Shared)	Packages	SmartFreight Web Service API
Customs	Packaging	State
Dashboard	Paperwork	System
Data Protection	Postal Sort	Transportation
Delivery	Pricing	Units of Measure
Desktop	Printing	Unset
Despatch	Product	UPS
Despatch note	Productivity	User Preferences
EZRentout API	Purchase Order	User-Defined
FedEx Ship API	Quality Assurance	Warehouse
FTP	Remote Import	WooCommerce API

Override

Sites, Organisations and Channels

As specified in the [Order Processing](#) section, OrderFlow can handle orders from multiple organisations, and multiple sales channels within organisations. It is also a multi-site system, supporting fulfilment of orders across multiple warehouses.

Whereas the relationship between organisations and channels is an inclusive one, OrderFlow considers sites to be orthogonal to these, i.e. it is possible for a shipment for a particular channel to be fulfilled in any site.

All these entities can be configured under the *Channels*, *Organisations* and *Sites* menus. The following screenshot shows an organisation detail page.

Setup

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin **Setup** Advanced 2/2 sites 1/7 organisations Unset (support) ?

System Setup

Welcome to the **setup** area of OrderFlow. The current system state is: **operational**

Currently on the system there is/are:
11 active couriers
6 batch types
4 print queues
3 workstations

Organisations

Organisation	Organisation State	Channel(s)	Channel State	Products
Sedgebrook Beer and Wine	Live	Manual Magento Trade	Test Live	31

Property Groups

Quick property group search links:

Activity	Handheld	Remote Order
Address	Import	Reports
Advanced Stock Check	Intersoft Sapient	Returns
API2Cart API	Inventory	Royal Mail - Click & Drop
Apollo	Kardex API	Royal Mail - DMO
ASN	Licence Plate	Royal Mail - Tracked
AWS	Magento 2 API	S3
Batch Processing	Magento 2 Native API	Salesforce Commerce Cloud API
Chemwatch API	Magento API	Scuri
Consolidation	Messaging	Seller
Country Groups	Notification	Ship@Ease
Courier	Order Entry	Shopify API
Courier (Generic)	Pack	Shopify GraphQL API
Courier (Shared)	Packages	SmartFreight Web Service API
Customs	Packaging	State
Dashboard	Paperwork	System
Data Protection	Postal Sort	Transportation
Delivery	Pricing	Units of Measure
Desktop	Printing	Unset
Despatch	Product	UPS
Despatch note	Productivity	User Preferences
EZRentout API	Purchase Order	User-Defined
FedEx Ship API	Quality Assurance	Warehouse
FTP	Remote Import	WooCommerce API

« Override

Application Properties

Application Properties within OrderFlow enable multi-layered configuration to be applied dynamically to a deployed instance. Typically, when the value of an application property is updated, that will have an immediate effect on the application, without needing to restart it.

Properties can control all sorts of aspects of OrderFlow, from despatch processing, through printing, to external system configuration and much more.

One very powerful aspect of application properties within OrderFlow is that they can be *scoped* and *site-specific*. This means that there can be several property *values* for a single property *definition*. The values would apply to different scopes (i.e. channels/organisations) or sites, thus allowing system behaviour to differ, depending upon the scope of the processing concerned, and to what site it is applicable.

For example, a shipment might have its courier assigned by a 'courier selection script' property. This property could have different values for different *warehouses*, because only certain couriers actually pick up from each warehouse. It may also differ based on the shipment's *organisation* (i.e. the company selling the goods). This would usually be the case, as it is typical that it is organisations who set up the contracts with courier companies. Finally, the script may also differ at the *channel* level, because only certain channels may attract certain courier services or options (e.g. eBay orders may be prevented from being despatched on a next day service).

The following screenshot shows a scriptable property with different scopes.

Setup

The screenshot displays the OrderFlow Setup interface. The top navigation bar includes tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. The Setup tab is active, showing a sidebar with various system components and a main area for 'Property Detail'. The 'Courier Selection Script' property is selected, showing its details: Title 'Courier Selection Script', Reference 'courier.selection.script', and Description 'Script to set a shipment's courier and delivery method fields. Should return null.' Below this, a table lists values for the property across different scopes and sites. An 'Edit Property Value' modal is open, allowing the user to edit the script value. The modal includes fields for Title, Description, Applicability (set to Channel (Magento Trade)), and a large text area for the script value.

Scope	Site	Active	Value	Last updated
Shared		Yes	return;	25/03/15 10:25:18
Magento Trade				
Manual	Swindon			

```
def delivery=values.courierDeliveryItem
def countrycode = value.implicitAddress.countryCode;
def puCountries =
['BE','BG','CZ','DK','DE','EE','IE','EL','ES','FR','IT','CY','LV','LT','LU','HU','MT','NL','AT','PL','PT','RO','SI','SK','FI','SE'];
def rmDomestic = ['GB','UK'];

def courierReference = null;
def serviceCode = null;
def courierOptions = null;
```

Couriers

Couriers are integral to OrderFlow - shipments will not go anywhere unless despatched via a courier. Under the **Couriers** tab, OrderFlow exposes configuration that allows couriers of many different types to be set up.

The configuration required for couriers can range from simple to complex; from a simple 'printed postage impression' (PPI) courier, that just requires a label to be printed for a shipment, to a fully-integrated, real-time label-producing courier that has an up-to-date picture of the shipments it needs to collect at each scheduled pick-up.

Couriers can be configured to require despatch references, which can be managed by OrderFlow, or by the courier's own external system. A courier's external system can be remotely-accessible as a web service, for example, or it could be locally-installed on a packing station. In this case, communication to the courier's desktop system is via a print server instance.

Naturally within OrderFlow, couriers can be restricted to be applicable to only certain sites and certain channels and/or organisations (via a script).

A courier can also be configured to have many *services* (e.g. 'next day', '3-day' etc.) and *options* (e.g. 'signed for'), to which shipments within OrderFlow can be automatically assigned.

The following screenshot shows an example courier configuration.

Setup

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Courier Setup

ListNew

Carrier Setup

Batch Types

Wave Definitions

Collection Groups

Packaging

Split Definitions

Picking Modes

Courier Detail

Operations

PreviousBack to listNext

Details for the current courier **royalmail_ppi** are shown below. Use the 'Edit courier' link to set advanced options.

Edit courier


Summary

Reference

royalmail_ppi

Name

Royal Mail (PPI)



Activated

Yes

Module

rtd2-courier-royalmailppi (royalMailPpi)

Default Carrier

Royal Mail

Property Group

Royalmail ppi

Options

Separate label printing is disabled
Label reprinting support is disabled
Shipment-level package label printing is disabled
Customs document printing support is disabled
Courier validation is not required
Courier validation and preparation is not separated
Courier cancellation is not required
Pack scanning does not need to be courier specific
Despatch notes are not dependent on this courier
Service code selection is not required
Display of the service reference is disabled
Selection of a service option(s) is not required
Cancellation of the courier will not clear shipment packages
Cancellation of the courier will not clear the selected service
Cancellation of the courier will clear the intermediate reference
Changing the courier does not depend upon the output of the 'Shipment Change Courier Service' scriptable property
Generation of a courier manifest is not required

Courier Services

The active services that apply for this courier are shown below. Courier services are ordered by 'item order' value (if set), then name.

Show inactive couriers services

Reference	Name	Carrier	Carrier Code	International	Active
packetpost_daily_1	Packetpost Daily Rate 1st Class	Royal Mail	packetpost_daily_rate_1st_class	x	Yes

New Service

Shipment Courier States

Import State

Not set

Before Pack State

Selection complete

Picking State

Not set

After Pack State

Selection complete

Despatch Options

Despatch Ref Required

Not set

Despatch Ref Source

Input

Batch Types

Shipment Batches (or just plain 'Batches') are defined in OrderFlow as groupings of shipments that it makes sense to process at the same time. By 'process', this typically means pick and pack. For example, high priority shipments might need to be processed separately before other standard priority shipments. Or shipments destined for a particular courier pick-up may need to be processed before other shipments destined for a later pick-up.

Like many things in OrderFlow, the behaviour of batches can be controlled by configuration, allowing warehouse processes to be changed without recourse to developing and deploying new code. This configuration is held in **Batch Types**, which allow aspects such as minimum and maximum batch size, sorting, picking type, packing type etc. to be set.

More information on batch types can be found in the **Warehouse Processes Guide**.

The following screenshot shows some of the configurable aspects of a batch type.

Setup

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Courier Setup

Carrier Setup

Batch Types

List

New

Wave Definitions

Collection Groups

Packaging

Split Definitions

Picking Modes

Edit Batch Type

Operations

PreviousBack to list

Edit details for current batch type **specialoffers**.

Batch type details

ReferenceSpecialoffers

NameSpecialoffers

DescriptionSingle line orders for special offer (high-volume) items. Individual shipments within a batch will all be for the same high-volume SKU.

Sort Indicator1

Despatch Note Print QueueDefaultDespatch Note Print Operation

Courier Label Print Queue

Packing TypeScan to Pack

Completion StateSent to printer

Active☒

Is Default☐

Courier Dependent☐ (If true, then cannot change courier while associated with a batch)

Bulk Despatch Note Printing☐ (If false, then despatch note will be printed just prior to packing)

Bulk Courier Label Printing☐ (If false, then courier label will be printed just prior to packing)

Batch type scope - click to view or edit batch type scope

Batch size

Min Batch Size8

Max Batch Size15

Allow Max Batch Size Override☐

Set channel-specific thresholds

Set site-specific thresholds

Scriptable batch sizes - click to view or edit scriptable batch sizes

Picking detail

Picking TypeReport

Picking TargetPacking

Picking Report TypeList(Leave empty to omit printing of picking report)

Picking Report Keystrd2-batch;picking_list_a4_barcoded

Handheld Task Report Keys

Report Picking Sortshipment.externalReference, product.externalReference

Handheld Picking Sortlocation.externalReference, product.externalReference

Batch creation

Batch Creation StrategyuniqueSKUCreationStrategy

Batch Creation Base URI

Workstations and Print Queues

The set-up of different Workstations in OrderFlow effectively allows it to communicate locally when accessed from a particular workstation. For example, configuring a workstation-specific print queue name allows OrderFlow to route print jobs to that queue, and only the print server on that workstation will be configured to poll that queue. The result is that print jobs will be printed on the printer attached to that workstation.

© OrderFlow Ltd. 2025

Page 49

Setup

Print queues are used to route different kinds of 'printable' items to the correct place. This is typically a physical printer, but could also be a file system location, for example, when communicating with courier desktop systems. A separate application, called the **Print Server**, polls print queues to perform the *actual* printing.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin **Setup** Advanced 2/2 sites 1/7 organisations Unset (support) ?

Edit Workstation Search Operations

Edit workstation entry **packing1**. Back to list | Next

Workstation details

Reference

Name

Description

Site

Area

Marshalling Location

Cancel Clone Update

Workstation properties

Name	Value		
despatch.note.print.queue	DEFAULT		
crossdock.print.queue	DEFAULT		
shipment.batch.print.queue	DEFAULT		
collection.label.print.queue	DEFAULT		
package.label.print.queue	DEFAULT		
label.print.queue	DEFAULT		
licence.plate.print.queue	DEFAULT		
stock.move.task.print.queue	DEFAULT		
document.print.queue	DEFAULT		
product.label.print.queue	DEFAULT		

New Property

Other Setup Configuration

There are other elements available to configure under the 'Setup' tab, that have not already been covered.

Packaging Types can be defined, which can be assigned to shipments during the despatch process to record how a shipment was packaged. This is useful for some organisations that want to report on how much packaging they use, to make it easier to order supplies.

It is also possible to define **file resources** in OrderFlow. These are images, documents or message bundles that can be used in various places in the system. For example, images are used on despatch notes, documents may back *product datasheets*, and message bundles can assist in producing internationalised paperwork from a single report definition.

Finally, the 'Setup' tab is where **countries** are configured. These are used in despatch operations to determine country-specific properties, usually when communicating with courier systems. For example, one courier may require the ISO 3166-1 alpha-2 (two-letter) code, whereas another may require the ISO 3166-1 numeric code. Both properties can be attached to the country object, making lookup easier.

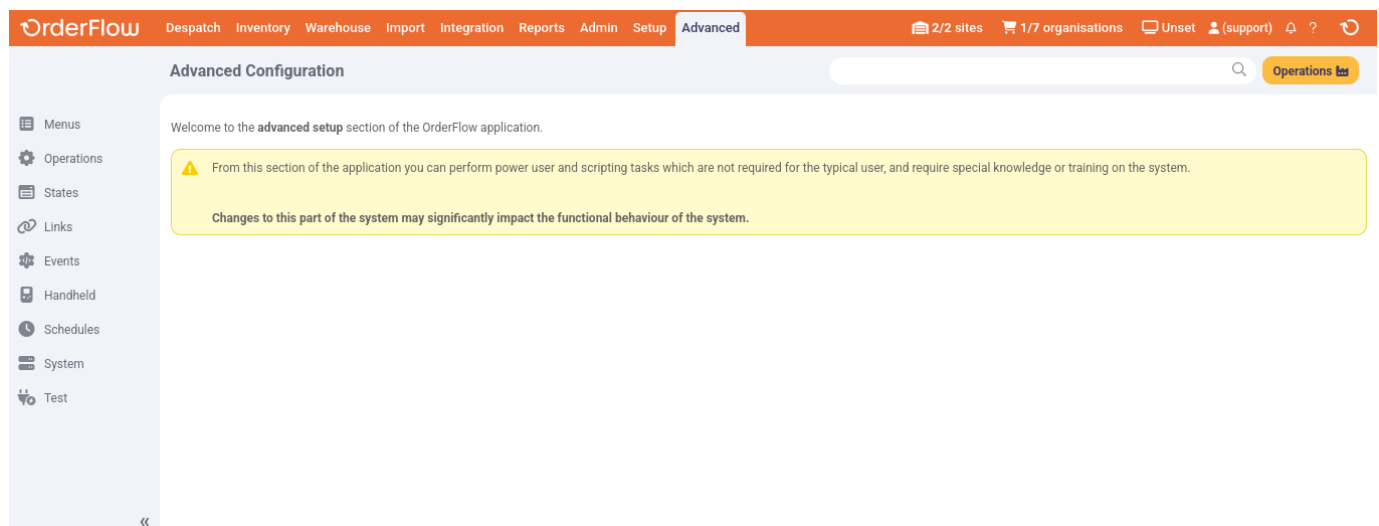
Advanced

The **Advanced** section of OrderFlow is where the more advanced configuration is set. This configuration controls the fundamental behaviour of the application, and as such should only be changed by experienced system administrators or those with special knowledge or training in this area.

The advanced configuration covers the behaviour of data **imports**, which define how various data is imported onto OrderFlow. It defines the various **states** that entities on the system can pass through, and the **operations** that move them between states.

It includes configuration for **events** and their listeners, **schedules** and their handlers, and also when certain **links** are displayable. In addition to some advanced **warehouse** configuration, there is the possibility to change certain **system** features, such as **menus**, or even which software modules are loaded.

This section provides an overview of the advanced features of OrderFlow in more detail.



Import Configuration

Data can be imported into OrderFlow using a variety of mechanisms. The most simple is via OrderFlow's API, an XML over HTTP interface.

Alternatively, data files can be placed on file system locations accessible to OrderFlow, which in this case would periodically poll those directories and read the files. The configuration to define these locations, and *how* the files should be read and processed, is held under the **Import Files** sub-menu.

OrderFlow Despatch Inventory Warehouse **Import** Integration Reports Admin Setup Advanced 2/2 sites 1/7 organisations Unset (support) ? ↻

Setup Post Import Mappings Import Handlers **Import Files** List New

Import File Definition Detail: asn/import/xml/file Operations

Details for current import file definition `asn/import/xml/file`.

Definition details

Reference	asn/import/xml/file
Name	ASN XML file import
Type	Advanced Shipping Note
Description	Advanced shipping note via XML import.
Scope	Organisation
Input/Output Handler	file
Folder Path	Not set - uses convention
File Pattern	Not set - matches all readable files
Import Operation	Insert (default)
Process Handler	product/import/xml
Activated	Yes

Folder locations

Below are the folder locations in use for this import file. The table also indicates for each of these whether the folder is readable and writable, and whether it exists.

Note: No folder path set. The import folder is based on convention involving the external reference and organisation.

Scope	Import Path	Exists	Readable	Writable
Geewhiz Widgets organisation	/organisation-geewhiz/asn/import/xml/file	×	×	×
ElectricEtail organisation	/organisation-electricetail/asn/import/xml/file	×	×	×
Sedgebrook Beer and Wine organisation	/organisation-sedgebrook/asn/import/xml/file	×	×	×
OrderFlow Demo organisation	/organisation-rtdemo/asn/import/xml/file	×	×	×
Tyrell Trading organisation	/organisation-tyrell/asn/import/xml/file	×	×	×
clone.of.Realtime Despatch Demo organisation	/organisation-test1/asn/import/xml/file	×	×	×
Six By Six organisation	/organisation-six_by_six/asn/import/xml/file	×	×	×
Shore Group organisation	/organisation-shore/asn/import/xml/file	×	×	×

Prepare Folders Click to create the directories listed in the Import File Definition 'Import Path' field.

[Edit definition](#)

To define how a file should be processed, the *Import File* definitions reference **Import Handler** definitions. These reference specific functions within OrderFlow that define how the data is processed. They also define the expected object type, the format and any encoding that the data has.

Additionally, a data transformation can be applied at this stage, to potentially label the imported data fields to expected values.

OrderFlow Despatch Inventory Warehouse **Import** Integration Reports Admin Setup Advanced 2/2 sites 1/7 organisations Unset (support) ?

Edit Import Handler Definition: order/import/rtd1_csv Operations

Previous | Back to list | Next

Enter details for current import handler definition order/import/rtd1_csv.

Definition details

Reference

Name

Description

Entity

Handler

Format

Delimiter Quote Character

Encoding

Transformation

```
<fieldmapper>
<mappings useinput = "false" indexfield="Sales Record Number">
  <mapping to = "externalReference_index">input["Sales Record Number_index"]</mapping>
  <mapping from = "Sales Record Number" to = "externalReference"/>
  <mapping to = "channel">"amazon_uk"</mapping>
  <mapping from = "Sale Date" to = "dateCreated"/>
  <mapping from = "Sale Date" to = "datePlaced"/>
  <mapping from = "Paid on Date" to = "dateAuthorised"/>
  <mapping from = "Buyer Email" to = "invoiceEmailAddress"/>
  <mapping from = "Buyer Full name" to = "invoiceContactName"/>
  <mapping from = "Buyer Phone Number" to = "invoiceDayPhoneNumber"/>
  <mapping from = "SRN Total Value" to = "totalPriceGross"/>
  <mapping from = "Postage and Packaging" to = "shippingPriceNet"/>
  <mapping from = "PayPal Transaction ID" to = "paymentTransactionInfo"/>
  <mapping from = "Buyer Full name" to = "deliveryContactName1"/>
  <mapping from = "Buyer Address 1" to = "deliveryAddressLine1"/>
  <mapping from = "Buyer Town/City" to = "deliveryAddressLine2"/>
  <mapping from = "Buyer County" to = "deliveryAddressLine4"/>
  <mapping from = "Buyer Postcode" to = "deliveryPostCode"/>
  <mapping from = "Buyer Country" to = "deliveryCountryCode"/>
</mappings>
</fieldmapper>
```

Content Script

Activated ☒

Cancel Clone Update

Finally, to ensure that imported data can be further transformed through configuration, each imported data object is passed through an **Import Mapping**. This is the case for data imported from file, or via the API.

Import Mappings can be defined for an object and scoped by **channel**, so that data received from different sales channels can be treated differently. These mappings contain a **pre-translation** script and a **post-translation** script - the difference between these is expanded upon in the **Importing Guide**. Each script can contain multiple mappings that utilise the **Groovy** scripting language, to apply logic and potentially transform individual data fields in the imported data. Additionally, the post-translation script can utilise functionality in OrderFlow to apply more logic or further change the imported entities.

An example of what import mappings can do is the following *order* import mapping configured for the 'Magento' sales channel. The pre-translation sets a few attributes of the order and shipment, and applies some logic when setting the mobile phone number. The post-translation invokes pre-defined functionality that sets the weight, courier and batch type of the shipment.

Advanced

OrderFlow Despatch Inventory Warehouse **Import** Integration Reports Admin Setup Advanced 2/2 sites 1/7 organisations Unset (support) ? ↺

Edit Import Mapping

Operations

Back to list | Next

Enter details for current order import mapping for channel 'Magento Trade'.

Applicability

Entity

Channel

Unmodifiable Fields

Activated ☒

Pre-translations

Script Detail <mapping qualifier = \"order\" to = \"deliveryMobilePhoneNumber\">
 if (input['deliveryDayPhoneNumber'] != null) return null;
 (input['deliveryDayPhoneNumber'].startsWith('07') ? input['deliveryDayPhoneNumber'] : null)</mapping>
 <mapping qualifier = \"order\" to = \"deliveryPostCode\">'SN15 5ND'</mapping>
 <mapping qualifier = \"order\" to = \"state\">'validated'</mapping>
 <mapping qualifier = \"shipment\" to = \"state\">'ready'</mapping>
 <mapping qualifier = \"shipment\" to = \"site\">'swindon'</mapping>
 <mapping qualifier = \"shipment\" to = \"priority\">'50'</mapping>
</mappings>"/>

Post-translations

Script Detail </transformation>
 <transformation qualifier = \"shipment\" handler = \"courierSetterTransformer\">
 </transformation>
 <transformation qualifier = \"shipment\" handler = \"batchSetterTransformer\">
 </transformation>
</transformations>"/>

Notes

Cancel Clone Update

Menus

The menus defined on the desktop, handheld and mobile interfaces are all defined in the **Menus** sub-menu of the Advanced tab on the desktop interface.

Each menu entry has a 'mode', which defines which of OrderFlow's interfaces it appears on. Each menu entry will define a URI, which is the (relative) URI that will be accessed when the menu is selected. Menu hierarchy is derived from each menu entry's *parent menu* - the top-level menus can be seen in the following screenshot.

Advanced

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Menu Search

Search criteria

Reference

Title

Mode

Parent Reference

Reset

Search

Menu search results

Reference	Title	Parent	Mode	Visible	Order	URI
admin	Admin		Operations	✓	11	/admin/view/dashboard.htm
admin/address	Address	admin_config	Config	✗	23	/address/admin.htm
admin/alarm	Alarms	admin	Operations	✓	1	/admin/view/alarm.htm
admin/alarm/search	Search	admin/alarm	Operations	✓	4	/admin/alarm/searchnew.htm
admin/audit	Audit Entries	admin	Operations	✓	9	/admin/audit/dashboard.htm
admin/audit/dashboard	Dashboard	admin/audit	Operations	✗	1	/admin/audit/dashboard.htm
admin/audit/search	Search	admin/audit	Operations	✓	2	/admin/audit/searchnew.htm
admin/correos	Correos	admin_config	Config	✗	22	/correos/admin.htm
admin/courier	Courier Admin	despatch_config	Config	✗	33	/admin/courier.htm
admin/courier/correos	Correos	admin/courier	Config	✗	1	/correos/admin.htm
admin/courier/endicia	USPS Endicia	admin/courier	Config	✓	2	/endicia/admin.htm
admin/courier/parcelperfect	Parcel Perfect	admin/courier	Config	✓	4	/parcelperfect/admin.htm
admin/dataremoval	Data Removal	admin_config	Config	✓	12	/admin/view/dataremoval.htm
admin/dataremoval/config	Config	admin/dataremoval	Config	✓	2	/admin/dataremoval/configuration.htm
admin/dataremoval/manage	Manage	admin/dataremoval	Config	✓	1	/admin/dataremoval/manage.htm

Page 1 of 39

Viewing 1 - 15 of 582

Menu entities themselves can be role-restricted; that is, they will only be displayed to users that have certain roles. This makes the OrderFlow interface appear different to different users - something that is quite useful as it can be used to restrict more advanced features to certain users. For example, warehouse staff whose role it is to pack shipments do not need access to the Advanced tab.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced** 2/2 sites 1/7 organisations Unset (support) ? ↺

Edit Menu Operations

Current menu item `inventory/area/list`.

Menu details Menu detail

Reference `inventory/area/list`

Pair Reference

Parent Reference `inventory/area`

Link URI `/inventory/area/list.htm` View

Title `List`

Link Title `List areas in warehouse`

Additional URIs

Requires Modules

Icon

Order `1`

Mode `Operations`

Roles

<input type="checkbox"/> activity_administrator	<input type="checkbox"/> activity_recorder	<input type="checkbox"/> advanced_administrator
<input type="checkbox"/> advanced_stock_check_administrator	<input type="checkbox"/> asn_handler	<input type="checkbox"/> batch_handler
<input type="checkbox"/> billing_administrator	<input type="checkbox"/> consolidation_printer	<input type="checkbox"/> customer_services_operator
<input type="checkbox"/> delivery_handler	<input type="checkbox"/> despatchClientView	<input type="checkbox"/> despatch_administrator
<input type="checkbox"/> despatch_printer	<input type="checkbox"/> despatch_read_only	<input type="checkbox"/> despatch_report_operator
<input type="checkbox"/> general	<input type="checkbox"/> handheld_warehouse	<input type="checkbox"/> help_scout
<input type="checkbox"/> import_administrator	<input type="checkbox"/> import_operator	<input type="checkbox"/> import_read_only

States and Operations

One of OrderFlow's more powerful features is its configurable state transition model. This means that the processing of the main transactional entities (such as orders, shipments and order lines) can be changed 'on the fly' in a deployed system, just by changing the configuration.

States are defined for orders, shipments and order lines. Each state definition references the *operations* that are available to an object while in that state. For example, a shipment in the *pending approval* state only has the 'approve' and 'defer approval' operations available to it.

The following screenshot shows part of the 'Order invalid' state definition for the order-item object.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced** 2/2 sites 1/7 organisations Unset (support) ?

Edit State Definition

Current state definition invalid for entity `rtd.domain.OrderItem`.

« Previous | Back to list | Next »

Menus
Operations
States
List
New
Links
Events
Handheld
Schedules
System
Test

State definition details

Name

Entity

Title

Description

Dashboard Base URI

Dashboard Base URI Description

Activated ☒

Workflow details

Progress Indication

Target State

Source Operation

Applicability

Operations

<input type="checkbox"/> anonymise	<input type="checkbox"/> hold
<input type="checkbox"/> approve	<input type="checkbox"/> hold for update
<input checked="" type="checkbox"/> cancel	<input checked="" type="checkbox"/> mark valid
<input checked="" type="checkbox"/> cancel temporarily	<input type="checkbox"/> release
<input type="checkbox"/> create unauthorised return	<input type="checkbox"/> submit
<input checked="" type="checkbox"/> edit order type	<input type="checkbox"/> unset hold for update

Links

<input type="checkbox"/> create unauthorised return	<input checked="" type="checkbox"/> edit invoice address
<input checked="" type="checkbox"/> edit delivery address	<input checked="" type="checkbox"/> edit invoice contact
<input checked="" type="checkbox"/> edit delivery contact	<input checked="" type="checkbox"/> edit order type

«

Cancel Update

Each **operation** referenced by a state definition defines what functionality is invoked when that operation is invoked. It does this by referencing known functionality via the *handler* attribute.

When an operation is invoked, the object is typically assigned a new *state*. The exact state to be assigned is based on fall-back logic that uses the invoked functionality and the operation definition, which can contain scripted logic. If an object has a parent object (e.g. a shipment's parent object is an order), then the parent object's state can be defined to be automatically updated if all its children have reached the operation's target state.

An operation definition can also be assigned *modes*, which determine where in the desktop user interface the operation will be made available. For example, operations with the mode 'detail' are presented to the user as buttons on the object *detail* pages.

Similarly to menus, operations can be restricted to certain roles, so that more advanced operations are not invocable by all users.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced** 2/2 sites 1/7 organisations Unset (support) ?

Edit Operation Definition Operations

Current operation definition **cancel_packages** for entity **rtd.domain.Shipment**.

Operation definition details

Name:

Title:

Entity:

Description:

Module:

Handler:

Check Handler if Allowed: ☒

Read-only Enabled: ☐

Item Order:

Activated: ☒

Initially Hidden: ☒

Requires Confirmation: ☒

Confirmation Prompt:

Input URI:

State transition details

Target State (Scriptable):

Error State (Scriptable):

Update Parent State: ☐ Results in parent set to target state once change has been applied to all children.

Target Parent State (Scriptable):

Partial Parent State (Scriptable):

Record Parent Operation: ☐ If no parent state change occurs, a parent operation will be recorded if this is checked.

Visibility

Modes: ☐ admin ☒ batch ☒ detail ☐ list ☒ pack ☐ expedite ☐ remote

States: ☐ allocated ☐ awaiting supplier export ☐ cancelled ☐ consolidated ☐ consolidating ☒ courier invalid ☐ courier not selected ☐ courier not validated ☐ created ☐ enroute dockable ☐ enroute docking ☐ deferred

Links

A **Link** definition in OrderFlow is a configurable object that gives control over which links (i.e. hyperlinks or icon links) are displayed on the desktop user interface. Each link defines a URI to where it leads, if invoked.

Each link definition applies to an object (e.g. shipment, order etc.), and defines the states in which the link should be displayed. Similarly to the operation definition, a link definition can also be assigned *modes*, which determine where in the desktop user interface the link will be made available.

The following screenshot shows a 'Change shipment priority' link definition for the shipment object.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced** 2/2 sites 1/7 organisations Unset (support) ?

Edit Link Definition Operations

Current link definition **change_priority** for entity **rtd.domain.Shipment**.

Link definition details

Name:

Title:

Entity:

Description:

Base URI:

Modes

☐ admin ☐ pack

☐ batch ☐ expedite

☒ detail ☐ remote

☐ list

States

☒ allocated ☒ packable

☐ awaiting supplier export ☐ packed

☐ cancelled ☒ packing

☐ consolidated ☐ paused

☐ consolidating ☐ pending bundle extraction

☒ courier invalid ☐ pending orderline merge

☐ courier not selected ☐ pending product activation

☒ courier not validated ☐ pending site selection

☒ created ☐ pending split

☒ cross dockable ☒ pickable

☒ cross docking ☒ picked

☐ deferred ☐ picked for consolidation

☐ despatched ☒ picking

☐ held after reset ☒ pre-assigned

☐ held before packing ☐ prepacked

☐ held before packing ☒ ready

☐ held for merge

Events

The **Events** section of OrderFlow is another area that adds to the power and flexibility of the system. As various changes to entities occur, OrderFlow can fire events, which contain details of what has just happened to the object concerned.

Event definitions and their listeners define what functionality should be invoked when certain events fire. For example, if a shipment has just been marked as *despatched*, a 'shipment despatched' event is fired, and the event definition for this event invokes each of the configured active listeners for this event. One of these listeners may ultimately notify an external system (e.g. a shopping cart) of this event.

An event definition can support a comma-separated list of *sourceData* values. This allows the associated event listeners to be fired for each of the values supplied. For example, if you want to fire the same event listener when a shipment is in the state *packed* and *despatched*, the value for *sourceData* would be *packed,despatched*.

The configurability of event definitions and their listeners allows system behaviour to be changed without needing to develop and deploy new code.

OrderFlow
Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced**
2/2 sites 1/7 organisations Unset (support) ?

Edit Event Definition
Previous Back to list Next

Current event definition **shipment_despatched**.

Event Definition Details

Name
Description
Source Type
Source Entity
Source Data
Filter Condition
Activated ☒
Persistent ☐
Target Delay (in seconds)
Trigger manually

Cancel Update

The following listeners have been set up for this event definition.

Name	Scope	Sites	Module	Handler	Activated	
shipment_despatch_listener	Globally-scoped	-	rtd2-notification	order_notification	No	
shipment_despatch_process_listener	Globally-scoped	-	rtd2-process	process_event	Yes	
shipment_despatch_courier_notifier	Globally-scoped	-	rtd2-courier	shipment_courier ...	No	
shipment_despatch_fixed_shipping_cost	Globally-scoped	-	rtd2-order	shipment_apply_f ...	No	
sales_owner_shipment_event_listener	Globally-scoped	-	rtd2-sales-owner	sales_owner_ship ...	No	
shipment_despatch_process_listener_textlocal	Globally-scoped	-	rtd2-process	process_event	No	

New listener definition

Schedules

In order that OrderFlow can present a real-time picture of warehouse activity in a performant manner, some of its processing needs to be done in the background, i.e. not in response to user or external system action. For example, it would soon fall over if it tried to calculate the overall picture of inventory levels following every single stock change.

It is for this reason (amongst others) that OrderFlow has a scheduling capability, which allows for processes to run at predefined times throughout each day. This capability is configured in **Schedule Definitions**, which are each associated with a cron expression, to define when the schedule will fire.

Schedule Handlers can be attached to schedule definitions - these reference known functionality in OrderFlow, which will be executed in order when the schedule fires. Each handler can be restricted to certain sites, organisations and channels, if required, and each can be configured to pass certain parameters to the invoked functionality. Additionally, each handler invocation can be configured to repeat for a defined duration or count.

Schedule handler invocations can be recorded in the database for later inspection. This provides useful evidence of system behaviour, if necessary.

The following screenshot shows a schedule definition, which fires every minute and invokes the active handlers.

OrderFlow Despatch Inventory Warehouse Import Integration Reports Admin Setup **Advanced** 2/2 sites 1/7 organisations Unset (support) ? ? ?

Edit Schedule Definition Operations

Current schedule definition **despatch_process**.

Schedule Definition Details

Name:

Description:

Cron Expression:

Thread priority:

Suggested Max Duration: (in minutes)

Activated: ☒

Cancel Clone Update

Handlers set up for this schedule definition are shown below. Show inactive handlers

Run	Name	Scope	Sites	Handler	Activated
	shipment_reintroduce	amazon_uk connectdirect connecttrade ebay_electricetail ebay_myco ... plus 7 more	swindon	shipmentReintroduction (rtd2-order)	Yes
	shipment_ready	amazon_uk connectdirect connecttrade ebay_electricetail ebay_myco ... plus 7 more	swindon	genericStateOperation (rtd2-state)	Yes
	consolidate_to_pickable	Globally-scoped	swindon	consolidateToPickable (rtd2-order-consolidation)	Yes
	shipment_select_courier	amazon_uk connectdirect connecttrade ebay_electricetail ebay_myco ... plus 7 more	swindon woodbridge	invokeStateOperation (rtd2-state)	Yes
	implicitly_progress_shipments	amazon_uk connectdirect connecttrade ebay_electricetail ebay_myco ... plus 7 more	swindon	shipmentImplicitProgress (rtd2-order)	Yes

New handler definition

Warehouse Configuration

The advanced **Warehouse** section allows the configuration of various warehouse-related aspects of OrderFlow. This section gives an overview of some of these aspects.

LOCATION AND PRODUCT TYPES

Location Types and **Product Types** can be defined here; these enable a high degree of flexibility for organisations to set up the warehouse in the exact way that they want to, as these types essentially add a layer of abstraction in front of the attributes that a location or product may have. This allows subtly-different types to be defined, which will influence warehouse behaviour in respect to location and product usage.

Advanced

The following screenshot shows the definition of a 'Incoming - preconsolidation' location type.

The screenshot displays the 'Edit Location Type' interface in the OrderFlow application. The top navigation bar includes 'OrderFlow' and various menu items like 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. The 'Inventory' menu is currently selected. The main header shows '2/2 sites', '1/7 organisations', 'Unset', and user support options. The left sidebar contains a navigation menu with 'Products', 'Grouped Products', 'Product Categories', 'Location Types' (selected), 'Container Types', 'Stock Status', and 'Product Admin'. The main content area is titled 'Edit Location Type' and shows details for the 'incoming_preconsolidation' location type. The 'Summary' section includes fields for 'Reference' (incoming_preconsolidation), 'Name' (Incoming - preconsolidation), and 'Description' (Incoming locations to which stock is initially credited prior to identifying a target workflow for further post goods-in operations). The 'Activated' checkbox is checked, and the 'Stock Status' is set to a dropdown menu. The 'Attributes' section is divided into four categories: 'Basic' (Allocatable, Damaged, Mobile, Pickable, Quarantined, Storage, System, Temporary, Unconfirmed, Virtual), 'Stock' (Enforces FIFO, Multi-lot, Multi-product, Multi-UOM, Negative Stock Allowed, Piece Only, Ringfenced, Stock Checkable, Tracked), 'Warehouse Usage' (Consolidation, In Transit, Inbound Staging, Incoming, Inspection, Loading Dock, Marshalling, Outbound Staging, Outgoing, Pre-receipt, Rework, Workstation), and 'Licence Plates' (Licence Plate, Licence Plate Pickable, Nested Stock Only, Nests Licence Plate, Nests Container Licence Plate, Suspends Licence Plate). The 'Incoming' checkbox under 'Warehouse Usage' is checked. The 'Nests Licence Plate' checkbox under 'Licence Plates' is also checked. At the bottom right, there are 'Cancel', 'Clone', and 'Update' buttons.

LOCATION SELECTION DEFINITIONS

Location Selection Definitions (or just 'Selection Definitions') expose a mechanism to select warehouse locations for certain tasks, in a configurable manner. These selection definitions can then be used in various places in OrderFlow, and as such can be changed or adjusted if necessary, changing the behaviour of the application without the need for any code to be changed.

A selection definition can filter locations by their attributes, either by inclusion or exclusion. It also has a filter script, which allows further flexible configuration to be applied via hooks into the search objects.

Sorting, weighting and results sizes can be configured to give fine-grained control over the location selection results.

The following screenshot shows an example of a location selection definition.

Advanced

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Suppliers

Delivery Types

Location Selection

Selection Config

Selection Definitions

Licence Plate Types

Stock Move Tasks

Stock Checking

Quality Assurance

Operations

PreviousBack to listNext

Test

Enter details for selection definition initial_putaway_target.

Summary

Referenceinitial_putaway_target

DescriptionDefault primary selection definition for putaway operations

Display TitlePrimary locations, or locations already containing product

Entityrtd.domain.ProductLocation

TypeLocation incoming

Activated

Filtering

Include Attributesstorage

Exclude Attributesincoming,temporary,damaged,quarantined,consolidation,nestedStockOnly

Report Key

Special Entries Scriptif (request?.warehouseProduct?.primaryPickingLocation) locations.add(request?.warehouseProduct?.primaryPickingLocation);
if (request?.warehouseProduct?.primaryStorageLocation) locations.add(request?.warehouseProduct?.primaryStorageLocation);

Selection Filter Strategies

Selection Filter Scriptfilter.activeLocations = true;
filter.positiveOnly = true;
filter.productLotId = null;

Result Weighting Scriptif (option.location == option.warehouseProduct?.primaryPickingLocation) option.addWeight(20);
if (option.location.logicalType.pickable) option.addWeight(15);
if (option.location == option.warehouseProduct?.primaryStorageLocation) option.addWeight(10)

Sort Expressionlocation.externalReference

Force Location Ordering

Min Results to Consider1

Max Results to Consider15

Max Results to Return15

Export

CancelCloneUpdate

TASK DEFINITIONS

Stock move tasks, as introduced in the [Stock Move Tasks](#) section, are instructions to move stock from one location to another. To determine *how much* of *which* stock to move from *which* location to *which other* location, OrderFlow uses the configurable *Task Definition* object.

Such an object facilitates the automatic creation of the right stock move tasks at the right time. The task definition defines which built-in *strategy* will be used when creating tasks of this type. This strategy may utilise a report to

© OrderFlow Ltd. 2025

Page 63

determine which lines to include in the task, or it may use a configured *line selection strategy*, which references a *Location Selection Definition* (as detailed in the previous section).

Sorting, scoping and creation and processing modes can also be defined in a task definition, as well as other options. See the **Warehouse Processes Guide** for more information.

The following screenshot shows a task definition for the 'Default background replenishment' task.

Advanced

OrderFlow

DespatchInventoryWarehouseImportIntegrationReportsAdminSetupAdvanced

2/2 sites1/7 organisationsUnset(support)?

Suppliers

Delivery Types

Location Selection

Licence Plate Types

Stock Move Tasks

Stock Checking

Quality Assurance

Edit Task Definition

Operations

PreviousBack to listNext

Enter details for task definition **background_replenishment**.

Summary

Referencebackground_replenishment

NameDefault Background Replenishment

DescriptionDefault background replenishment for products whose picking quantity has fallen below the picking replenishment level.

Printed Report Key

Transfer TypeInternal transfer

Activated☒

Wave Aware☐

Logging

Logging Enabled☐

Max Log Size

Inputs

System-created Requirement☐

User-created Requirement☐

Line Report Keystrategy_background_picking_replenishmentView

Default Product Count40

Query Boost Factor1.5

Input Parametersdesired_product_count: Number of products: optional: integer

Max Weight

Weight Units

Task Strategies

Strategy classrtd.warehouse.stockmove.strategy.impl.IncrementalSourceCentricStockMoveTypeStrategy

Source Line Strategysource_centric_source

Target Line Strategysource_centric_target

Location Selection Strategies

1st Source Selectionreplenishment_sourceView

2nd Source Selection

1st Target Selectioninitial_replenishment_targetView

2nd Target Selectionbackup_replenishment_targetView

Source Quantity Allocator

Target Quantity Allocatorpicking_replenishment

Task Target Type

Task Target Reference

Manual Target Location Selection Configuration

Alternative Target Location Selection Configuration

Line Sorting Strategies

Source Line Sort Expressionproduct.location.locationSortIndicator, location.sortIndicator

Target Line Sort Expression

Applicability

Creation Modes☒ Desktop☐ Handheld☐ Embedded☐ Imported

Processing Modes☒ Desktop☐ Handheld☐ Embedded☐ Imported☐ Licence plate embedded

ScopeOrganisation

Organisations☒ sedgebrook

© OrderFlow Ltd. 2025

Page 65

PICKING DEFINITIONS

OrderFlow's **Picking Sequence Definitions** can also be configured under the *Warehouse* sub-menu. *Picking Sequences* are a mechanism to combine different task types into the same picking operation, to improve the efficiency of picking operations in the warehouse.

A picking sequence definition can be assigned a *sort expression*, and also hook into specific logic to be invoked when the picking sequence is complete. More details can be found in the **Warehouse Processes Guide**.

The following screenshot shows an example 'Multiline pick for consolidation' picking sequence definition, which combines the task definitions 'multiline pick' and 'storage to consolidation'.

The screenshot shows the 'Edit Picking Sequence Definition' interface in OrderFlow. The top navigation bar includes 'OrderFlow', 'Despatch', 'Inventory', 'Warehouse' (selected), 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. The right side of the bar shows '2/2 sites', '1/7 organisations', 'Unset', '(support)', and icons for help, search, and refresh. A left sidebar lists navigation options: Suppliers, Delivery Types, Location Selection, Licence Plate Types, Stock Move Tasks, Picking Sequences (selected), Stock Checking, and Quality Assurance. The main content area is titled 'Edit Picking Sequence Definition' and includes an 'Operations' button. Below the title is a breadcrumb: 'Enter details for picking sequence definition multiline_pick_for_consolidation.' The form is divided into two sections: 'Summary' and 'Applicability'. The 'Summary' section contains fields for Reference (multiline_pick_for_consolidation), Name (Multiline Pick for Consolidation), Short Name (MULTILINE), Type (Task - multi), Task Definitions (multiline_pick_storage_to_consolidation), Description (Handles the multiline pick for consolidation, which includes order lines assigned to picking locations in storage, as well as order lines for shipments already in consolidation.), Sort Expression, and Action Delegate. There are also checkboxes for 'Activated' and 'Wave Aware'. The 'Applicability' section has a 'Sites' field with checkboxes for 'swindon' and 'woodbridge', both of which are checked. At the bottom right are buttons for 'Cancel', 'Clone', and 'Update'.

REASON CODES

Finally, **Warehouse Reason Codes** can be configured under the *Warehouse* sub-menu. These provide a flexible way of controlling certain aspects of the warehouse process, such as what options are presented to users when making stock adjustments or setting return reasons. As the following screenshot shows, their applicability can be restricted by site and also restricted to certain handheld operations.

OrderFlow Despatch **Inventory** Warehouse Import Integration Reports Admin Setup Advanced 2/2 sites 1/7 organisations Unset (support) ? ↺

Edit Warehouse Reason Code Operations

Back to list | Next

Enter details for warehouse reason code `stock_adjustment_damaged_in_process`.

Summary

Reference `stock_adjustment_damaged_in_process`

Title `Damaged (In Process)`

Short Title `Damaged (In proc.)`

Description `Negative stock adjustment for goods damaged in the warehouse (and as such should NOT be returned to the supplier).`

Type `Stock adjustment`

Type Applicability `Negative adjustment`

Order `2`

Activated ☒

Applicability

Desktop ☒

Handheld ☒

Handheld Operations (Scriptable) `pick_line,task_line_short,task_line_unavailable,batch_short_pick_with_correction,embedded_move_stock_to_damaged,task_product_unavailable`

Site-specific ☐

Cancel Clone Update

System, Housekeeping and Test

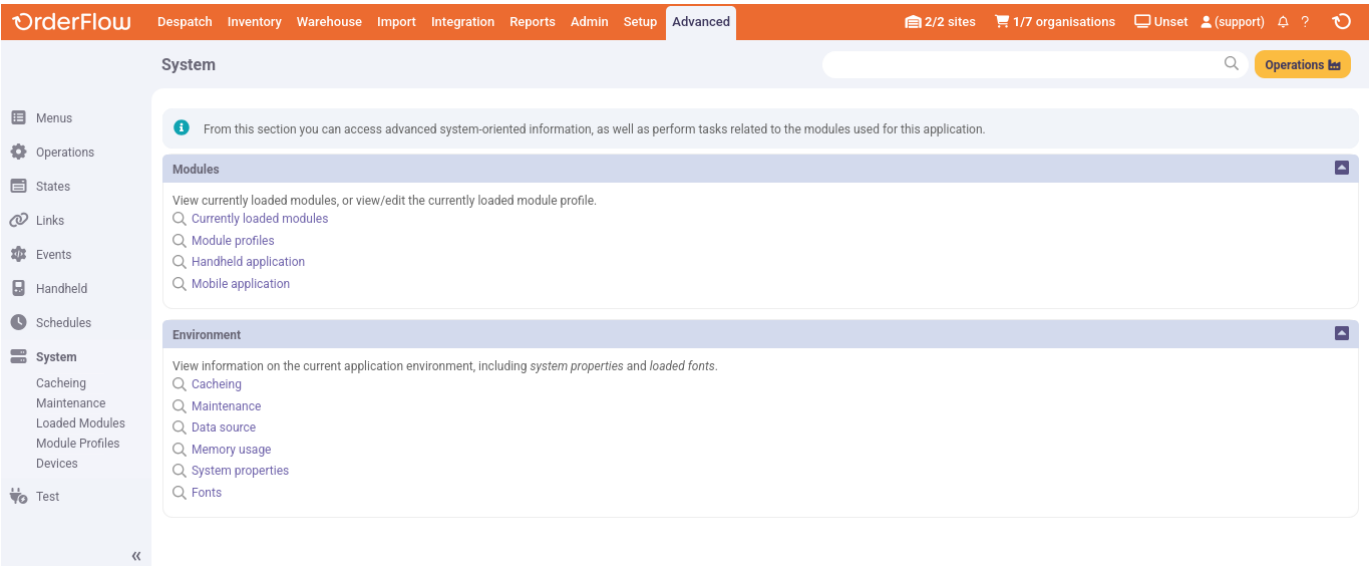
The **Housekeeping** area of OrderFlow contains configuration for dealing with **purging** and **archiving** of data from the system. *Purging* data allows non-critical event data to be permanently deleted from the system, whereas *archiving* allows business transactional data to be removed from the system, potentially to the file system or another database, for possible later business analysis.

The **System** sub-menu includes the configuration for which software *modules* are loaded on the currently-running instance. OrderFlow is a modular application, which means that, on top of a set of *core* modules, there are many *optional* modules that deliver specific functionality. The optional modules are enabled according to what functionality a customer requires. It is possible to change the module configuration, but only those modules that have been *commissioned* by OrderFlow can be enabled.

Additionally, this section includes the display of various system environment properties, memory usage and loaded fonts. It also provides some control over resetting the system cache and data source.

Finally, the **Test** sub-menu supports the testing of OrderFlow itself, by allowing certain *data sets* to be loaded, if available on the underlying file system. This feature is only useful in a development environment, i.e. it is not to be used on deployed instances of OrderFlow.

Further Reading



Further Reading