



OrderFlow Introduction

OrderFlow Ltd.

Document Version: 4.2.4

Document Built: 2024-02-16

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

Introduction

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

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

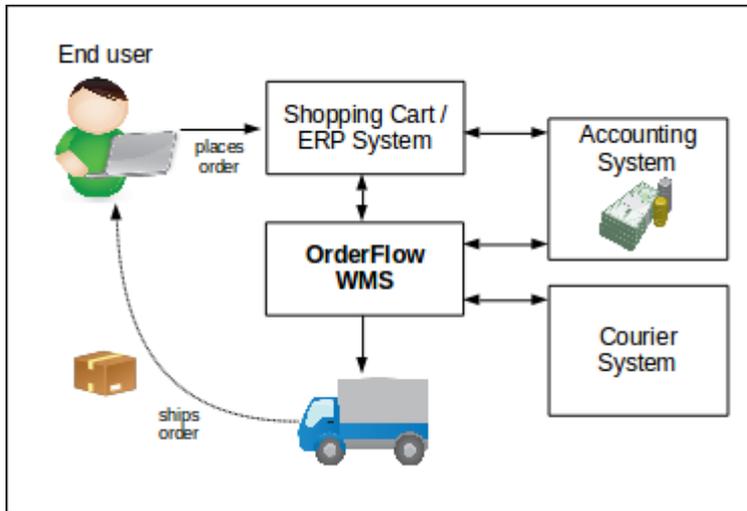
The intended *audience* of this document is:

- 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.

Desktop Interface

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

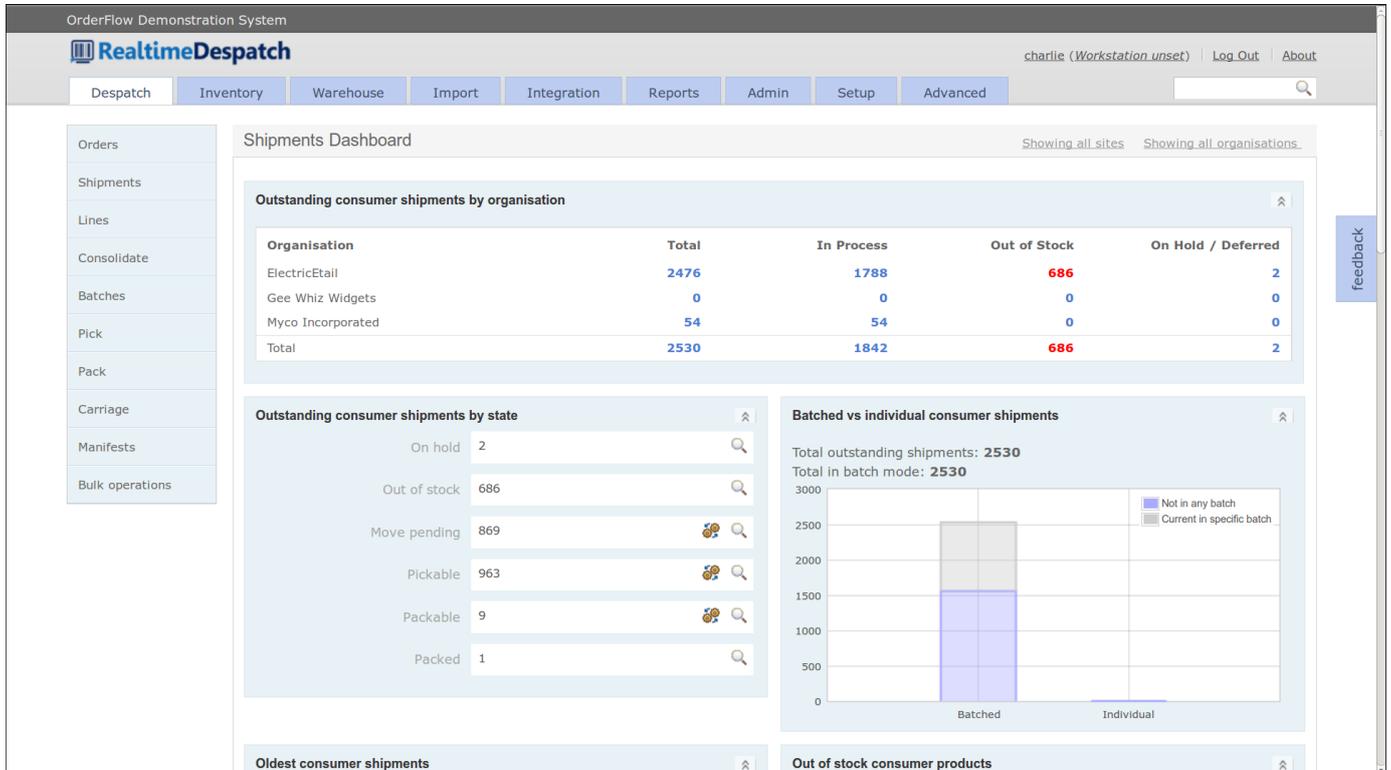
The desktop interface to the OrderFlow system consists of a number of tabs, each of which performs functions in distinct areas of the application. This section provides an overview of the functionality provided within each of the tabs.

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

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.

OrderFlow Demonstration System

RealtimeDespatch charlie (Workstation unset) | Log Out | About

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

Products | Inventory | Stock | Changes | Locations | Areas | Categories | Datasheets | Admin

Stock and Inventory Dashboard Showing all sites Showing all organisations

Below are **stock and inventory**-related views of the system.

Product inventory status

The following shows the number of distinct products in each stock state.

Available	17
Required	36
Shortfall	19

Existing products by state

Created 38

Products requiring reorder

The following are the "most required" products that require reordering, either because they have negative availability, or because the stock level has fallen below the product's reorder trigger.

Reference	Available	Reorder threshold	Target quantity	On-order quantity	Suggested reorder qty
TRAN805423	-94	-	-	-	94
TRAN805424	-64	-	-	-	64
TRAN805406	-64	-	-	-	64
TRAN805416	-63	-	-	-	63
TRAN805407	-45	-	-	-	45

feedback

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

OrderFlow Demonstration System

RealtimeDespatch

charlie (Workstation unset) | Log Out | About

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

Purchase orders
ASNs
Deliveries
Work orders
Licence plates
Picking
Tasks
Returns
Suppliers

Warehouse Operations Dashboard

Showing all sites Showing all organisations

Below are **warehouse operations**-related views of the system.

Outstanding Supplier Purchase Orders by state

Completed	2
Partially applied	1

Outstanding ASNs by state

Receivable	2
------------	---

Outstanding deliveries by state

Created	5
---------	---

Override

© Realtime Despatch Software Ltd 2011-2015

Return to the top

feedback

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

The screenshot displays the 'Import Dashboard' in the RealtimeDespatch system. The dashboard includes a navigation menu with options like 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. The 'Import' tab is active. The dashboard content is organized into several sections:

- Recent import errors:** A message states 'No import errors have been raised in the last 7 days.'
- Recent import batches:** A message states 'Below are the most recent import batches added to the system in the last 7.' Below this is a table with the following data:

Entity	Scope	Type	Successes	Failures	Duplicates	Added
Product	myco	xml	1	0	0	12/2/15 2:26:44 PM
Order	Amazon UK	csv	2476	0	0	12/1/15 6:58:22 PM
Order	Magento	xml	54	0	0	12/1/15 6:51:17 PM

Below the table is a search bar labeled 'Search import batches'. The dashboard also features a 'Recent import files' section with a message: 'No import files have been added in the last 7 days.' A 'Feedback' button is visible on the right side of the dashboard.

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

OrderFlow Demonstration System

RealtimeDespatch

charlie (Workstation.unset) | Log Out | About

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

Showing all sites Showing all organisations

Message Definitions

Message type definitions on the system are shown below.

Category	Name	Client	Description	
Api integration	api_magento_rest	HTTP-OAUTH	Magento REST API base message type	✘
Api integration	api_magento_soap	HTTP-XML-SESSION	Magento SOAP API base message type	✘
Api integration	delivery_notification	HTTP	Delivery Notification message	✘
Api integration	inventory_notification	HTTP	Inventory notification message	✘
Api integration	landmark_parcel_request	HTTP	Landmark	✘
Api integration	order_notification	HTTP	Order notification messages	✘
Api integration	purchase_order_notification	HTTP	Purchase Order Notification message	✘
Api integration	return_notification	HTTP	Return Notifications	✘
Api integration	stock_change_notification	EMAIL	Stock Change Notification	✘
Courier integration	hypaship_order_request	HTTP	Hypaship	✘
Courier integration	royalmail_tracked_despatch_notifica ...	FTP	Royal Mail Tracked Despatch Notifications (Pre-advice files)	✘
Courier integration	ups_api_shipment_accept	HTTP	UPS API Shipment Accept	✘
Courier integration	ups_api_shipment_confirm	HTTP	UPS API Shipment Confirm	✘
Courier integration	ups_api_shipment_void	HTTP	UPS API Shipment Void	✘

feedback

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

The screenshot displays the 'Admin' section of the RealtimeDespatch system. The main heading is 'Warehouse Reports'. Below this, there is a search bar with a 'Type' dropdown menu and a 'Text' input field. The page is divided into two main sections: 'Warehouse reports' and 'Periodic reports'. Each section contains a table with columns for 'Report Name', 'Description', 'Type', and 'Formats'. The 'Warehouse reports' table lists reports such as 'Barcodes', 'Locations (multiple) A4', 'Product Inventory by product', 'Product Reorder', 'Product Reorder (Ungrouped)', and 'Retruns 2'. The 'Periodic reports' table lists 'Deliveries - end of day' and 'Stock - end of day'. A sidebar on the left contains navigation links for various system functions, and a 'feedback' button is located on the right side.

OrderFlow Demonstration System

RealtimeDespatch

charlie (Workstation unset) | Log Out | About

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

Showing all sites Showing all organisations

Warehouse Reports

Below is a list of the warehouse reports that are available to run.

Type [] Text []

Warehouse reports

Report Name	Description	Type	Formats
Barcodes	Basecodes report, giving list of barcode for products and category matches	Warehouse	jrxml
Locations (multiple) A4	Barcode and location references on A4	Warehouse	jrxml
Product Inventory by product	Product Inventory as spreadsheet	Warehouse	csv xls
Product Reorder	Product reorder report, giving list of products whose available quantity has fallen below the target reorder threshold	Warehouse	jrxml
Product Reorder (Ungrouped)	Product reorder report, giving list of products whose available quantity has fallen below the target reorder threshold. No grouping of results.	Warehouse	jrxml
Retruns 2	Returns recorded between two dates, with a status of 'applied'	Warehouse	jrxml

Periodic reports

Report Name	Description	Type	Formats
Deliveries - end of day	Deliveries over date range, suitable as end of day report	Periodic	csv
Stock - end of day	Stock report, suitable for use as end of day report	Periodic	csv

feedback

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.

Setup

RealtimeDespatch charlie (Packing Station 1) | [Log Out](#) | [About](#)

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

Users | Roles | Logging | Errors | Events | Audit entries | Jobs | Housekeeping | Health check | Performance | Couriers

Admin Dashboard

Showing all sites | Showing all organisations

The views below are **system** views which the administrator can use to help ensure the system is functioning correctly.

Recent system alarms

No alarms have been raised recently that are still open on the system.

Recent remote message failures - [click to view](#)

Recent job execution failures

No job execution failures have occurred in the last 7 days.

Recent untrapped errors

Below are the most recent untrapped errors. Repeated occurrence of the same exception may indicate a programming error.

User	Type	Message	Created
charlie	IllegalArgumentException	Parameter 'type' is required.	1/14/16 7:05:35 PM
charlie	DataIntegrityViolationException	could not insert: [rtd.domain.Location]; ...	1/14/16 5:04:25 PM

[Search errors](#)

Override

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

Setup

The **Setup** tab deals with functionality that requires technical knowledge beyond the level of most users. It is normally used when new functionality is released, for example when a new courier is being introduced into the system.

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

Advanced

OrderFlow Demonstration System

RealtimeDespatch

charlie (Workstation unset) | Log Out | About

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

Showing all sites Showing all organisations

Setup Dashboard

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

Currently on the system there is/are:

- 5 active **couriers**
- 4 **batch types**
- 5 **print queues**
- 3 **workstations**

Organisations

Organisation	Organisation State	Channel(s)	Channel State	Products
ElectricEtail	Live	ElectricEtail Amazon UK ⚠️ Ebay	Test Live	29
Gee Whiz Widgets	Live	Amazon NL	Live	0
Myco Incorporated	Live	Magento ⚠️	Test	9
Total				38

Property Groups

Quick property group search links:

- Address
- Archive
- ASN
- Batch Processing
- Consolidation
- Inventory
- Licence Plate
- Magento API
- Messaging
- MetaPack
- Remote Import
- Reports
- Returns
- Royal Mail - DMO
- Royal Mail - Tracked

Advanced

The **Advanced** tab deals with functionality that requires technical knowledge beyond the level of most users. Many of the scripting and field mapping options are in this tab, as are configurations for complex areas of the system such as *stock move tasks*.

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

Handheld Interface

OrderFlow Demonstration System

RealtimeDespatch

charlie (Workstation unset) | Log Out | About

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

Showing all sites | Showing all organisations

Input handler definitions

Listing of input handler definitions for current application.

Reference	Name	Active	Handler	Transformation
asn/import/csv	ASN CSV import	yes	import_asn_csv	<fieldmapper> <mappings ...
import/xml	Native XML import	yes	import_xml	
order/import/rtd1_csv	Amazon order CSV import	yes	import_order_csv	<fieldmapper> <mappings ...
order/import/tightspot	TightSpot CSV import	yes	import_order_csv	<fieldmapper> <mappings ...
product/import/csv	Product CSV import	yes	import_product_csv	<fieldmapper> <mappings ...
productdatasheet/import/combined_csv	Product Datasheet Combined CSV import	yes	import_productdatasheet_combined_csv	fieldmapper> <mappings ...
purchaseorder/import/csv	Purchase Order CSV import	yes	import_purchaseorder_csv	<fieldmapper> <mappings ...
purchaseorder/import/sage_csv	Purchase Order Sage CSV import	yes	import_purchaseorder_csv	<fieldmapper> <mappings ...

© Realtime Despatch Software Ltd 2011-2015

[Return to the top](#)

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

[Home](#) > [Location Check](#) > [Confirm](#) > [Correct](#)

The currently recorded quantity for location **location_1** and product **ipod5** is **0**.

Please confirm the actual quantity is **12**.

12

CONFIRM

CANCEL

[Home](#) [Config](#) (Site [SECOND](#)) [Logout](#)

© Realtime Despatch Software Ltd 2011-2015 (logged in as **philz**)

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

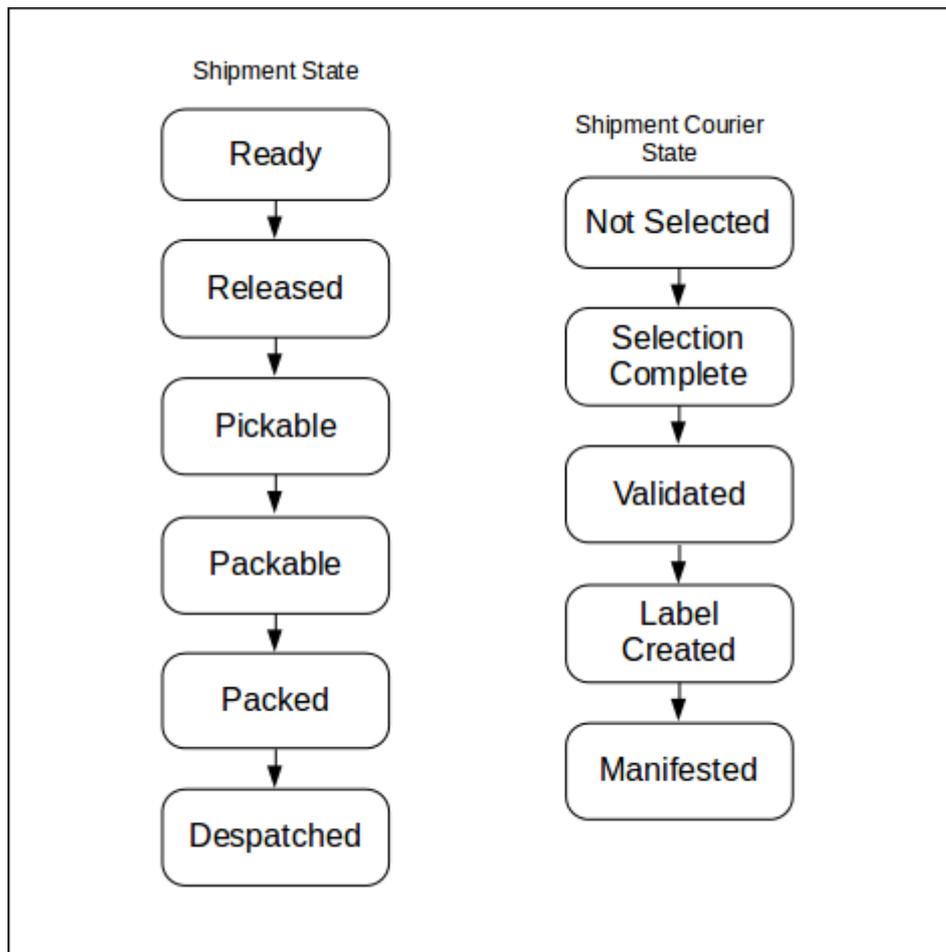
	<i>product</i>	<i>quantity</i>
Order line:	WIDGET-EEE	1
Order line:	WIDGET-FFF	3

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.

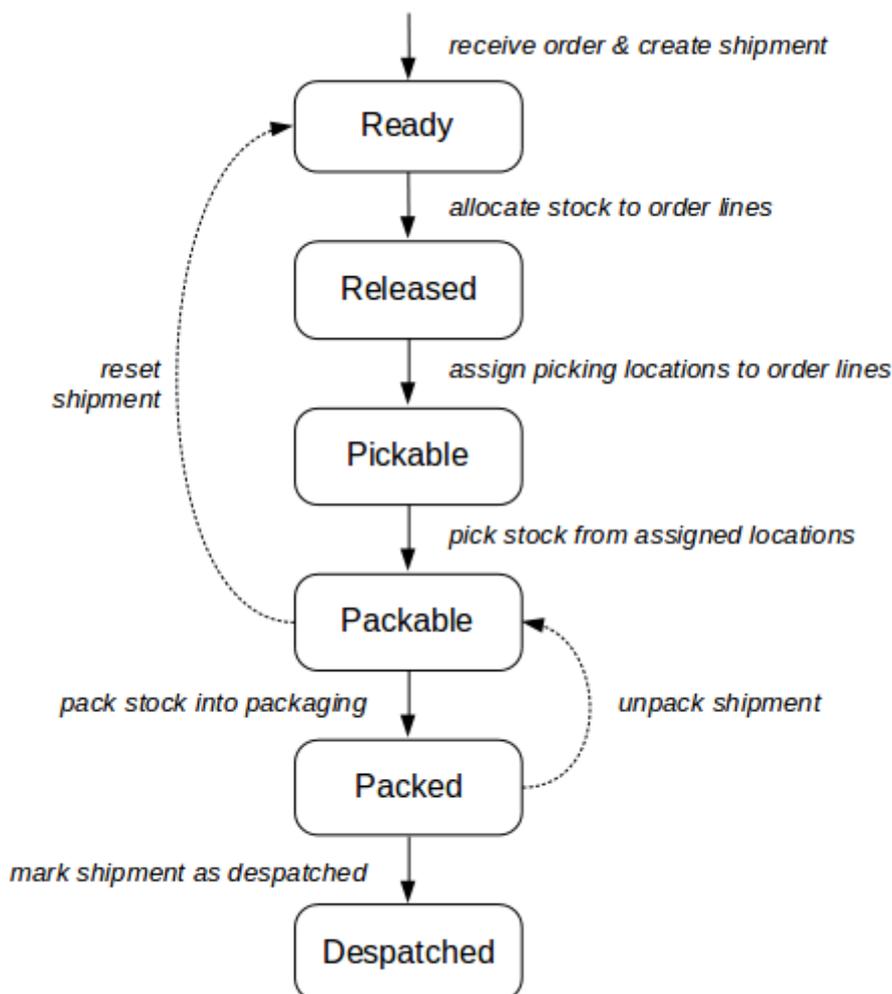
An Order Processing Example

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.

The screenshot displays the 'Orders Dashboard' in the 'RealtimeDespatch' system. The dashboard is titled 'Orders Dashboard' and shows 'Showing all sites' and 'Showing all channels (inc. test)'. The main content area is divided into two sections: 'Outstanding orders by state' and 'Orders added recently'. The 'Outstanding orders by state' section shows a table with the following data:

State	Count
Created	0
Pending	0
Validated	2530
Part despatched	0

The 'Orders added recently' section shows a message: 'No orders have been added onto the system recently.' The dashboard also includes a navigation menu on the left, a top navigation bar with tabs like 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced', and a user profile 'charlie (Workstation unset)' in the top right. A 'feedback' button is visible on the right side of the dashboard.

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.

Stock Allocation and Assignment

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 'Order Lines Dashboard' in the RealtimeDespatch system. The dashboard is divided into several sections:

- Order lines by state:** A table showing the count of order lines in various states: Created (1), Allocated (0), Out of stock (724), Pickable (999), Move pending (837), and Picked (0).
- Move pending products:** A section with a warning icon and text: "The following are the product with the largest number of move pending lines. A stock move is required for these products in order for their order lines to be pickable." Below this is a table with columns Reference, Lines, and Quantity.
- Out of stock products:** A section with a warning icon and text: "The following are the most in demand products, with the largest number of out of stock order lines. New stock must be checked in to fulfil these products." Below this is a table with columns Reference, Lines, and Quantity.

Reference	Lines	Quantity
TRAN805432	776	776
TRAN805426	32	32

Reference	Lines	Quantity
TRAN805423	94	94
TRAN805424	64	64

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.

Shipment Picking

The screenshot displays the 'OrderFlow Demonstration System' interface. The main header shows 'RealtimeDespatch' and user details 'charlie (Workstation unset) | Log Out | About'. A navigation bar includes 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. A left sidebar lists 'Orders', 'Shipments', 'Dashboard', 'Search', 'Lines', 'Consolidate', 'Batches', 'Pick', 'Pack', 'Carriage', 'Manifests', and 'Bulk operations'. The 'Shipments Dashboard' is active, showing 'Showing all sites' and 'Showing all channels (inc. test)'. It contains several panels: 'Outstanding shipments by channel' with a table, 'Outstanding shipments by state' with a list, 'Batched vs individual shipments' with a bar chart, 'Oldest shipments' with a table, and 'Out of stock products' with an information box.

Channel	Total	In Process	Out of Stock	On Hold / Deferred
Amazon NL	0	0	0	0
ElectricTail Amazon UK ⚠️	2476	1751	723	2
Ebay	0	0	0	0
Magento ⚠️	54	54	0	0
Total	2530	1805	723	2

Deferred	1
On hold	1
Out of stock	723
Move pending	832
Pickable	763
Packable	208
Packed	2

Category	Count
Batched	2530
Individual	0

Reference	Age in Days	Current State
A2344000310	18	Move pending

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.

Shipment 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 Demonstration System

RealtimeDespatch

charlie (Workstation unset) | Log Out | About

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

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

Batched shipment picking state

Quantity	Batch Type	Status
100	In paper batch	Ready for printing ^o
200	In paper batch	Printed and possibly picked ^o
642	In handheld batch	Picking not yet started ^o
21	In handheld batch	Currently being picked ^o

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 Task	Associated Batch	Stock Qty	Allocation Qty
cart2	113	13 (Handheld picked)	17	17
cart1	114	14 (Handheld picking)	9	9

Mobile allocations

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

Site	Shipment	State	Location	Product	Quantity	Allocated
Swindon	100000106	Packable	cart2	sw810l	2	2
			cart2	MicroUsbBlackberry	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.

Shipment Packing

- 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.

The screenshot displays the 'Shipment Batch Dashboard' in the OrderFlow RealtimeDespatch system. The interface includes a navigation menu on the left with options like Orders, Shipments, Lines, Consolidate, Batches, Dashboard, Search, New, Pick, Pack, Carriage, Manifests, and Bulk operations. The main dashboard area is divided into several sections:

- Outstanding shipment batches:** A list of counts for different stages: Created (1), Download pending (2), Downloaded (0), Sent to printer (0), Packing (0), and Packed (0).
- Batched vs individual shipments:** A bar chart showing 'Total outstanding shipments: 2530' and 'Total in batch mode: 2530'. The chart compares 'Batched' (approx. 1500) and 'Individual' (approx. 1000) shipments. A legend indicates 'Not in any batch' (blue) and 'Current in specific batch' (grey).
- Pickable shipments:** A message stating 'All 763 pickable shipments in batch mode are currently being batched processed.'
- Partly packed batches:** A table showing one entry for a batch with ID 13, State 'Handheld picked', Picking Location 'cart2', Shipment Count 10, Packed Count 2, and Last Packed on 27/10/2015 at 11:29.

Shipment Packing

OrderFlow supports many different ways to pack shipments.

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.)

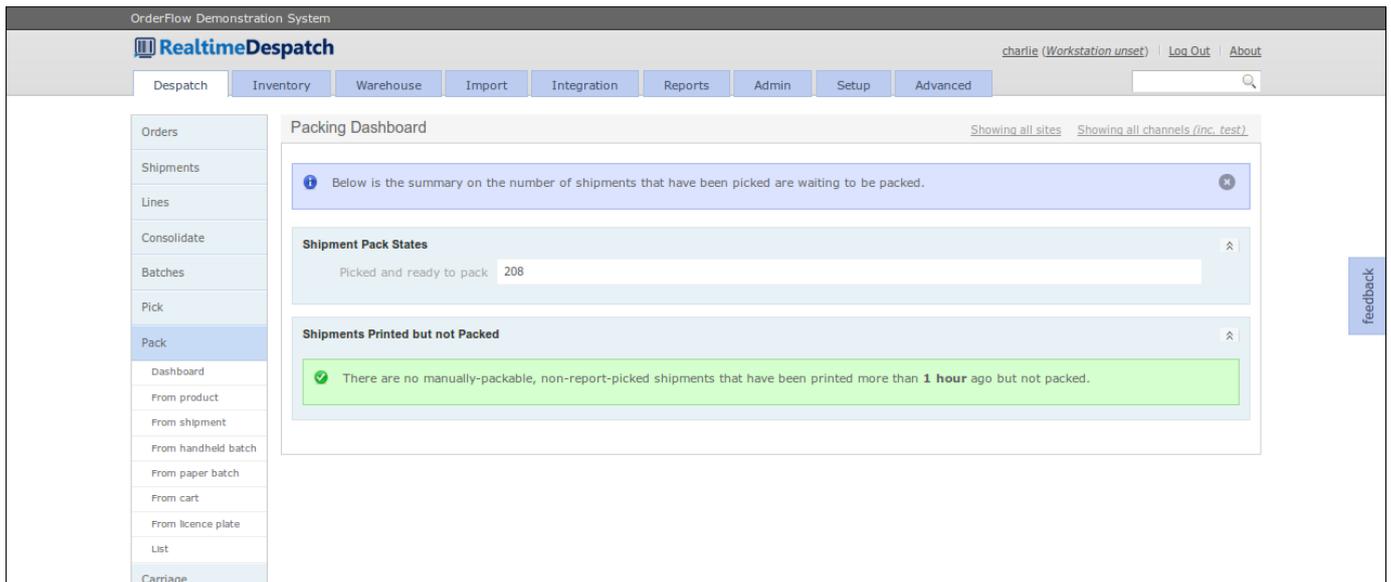
Some organisations want tighter control over packing, i.e. they want to be sure that no mistakes are made during the packing process, which would result in the wrong goods being despatched. This may be the case if they stock many different products, or if products are very similar and can't easily be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

Courier Selection

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



The screenshot shows the 'Packing Dashboard' in the OrderFlow system. The dashboard includes a navigation menu on the left with options like Orders, Shipments, Lines, Consolidate, Batches, Pick, Pack, and Carriage. The main content area displays a summary of shipment pack states, including 'Picked and ready to pack' (208) and 'Shipments Printed but not Packed' (0). A feedback button is visible on the right side.

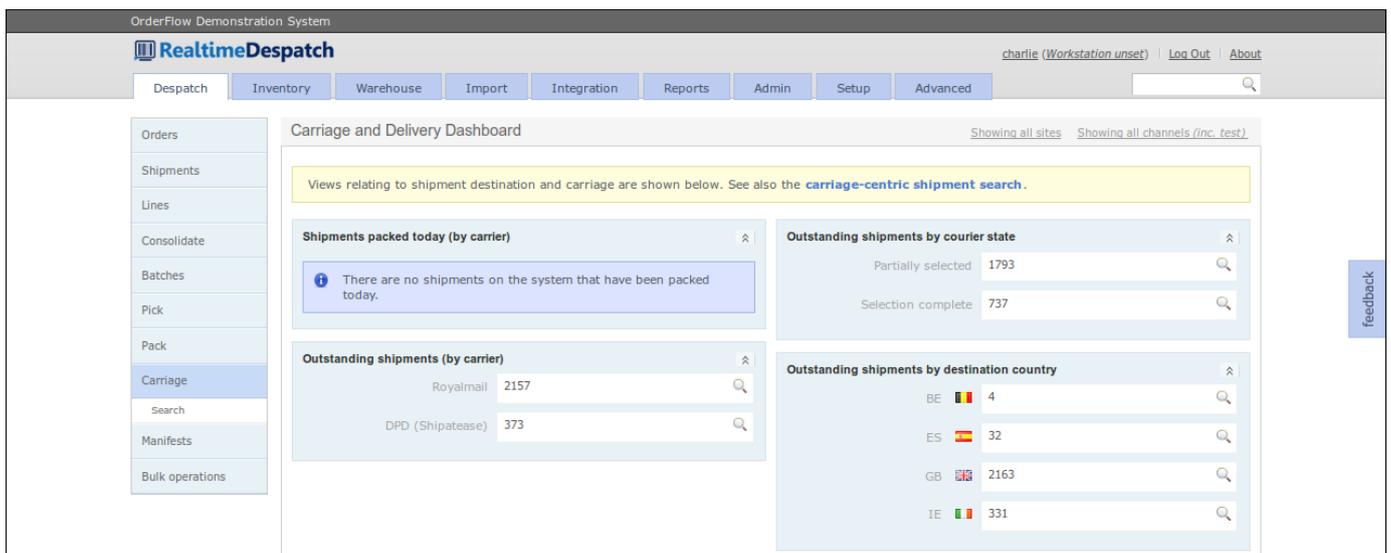
As part of the 'packing consolidation' process, OrderFlow supports other refinements, such as requiring the packer to scan a product's batch code, or to supply a quantity instead of scanning every product.

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:



The screenshot shows the 'Carriage and Delivery Dashboard' in the OrderFlow system. The dashboard includes a navigation menu on the left with options like Orders, Shipments, Lines, Consolidate, Batches, Pick, Pack, Carriage, Search, Manifests, and Bulk operations. The main content area displays views relating to shipment destination and carriage, including 'Shipments packed today (by carrier)', 'Outstanding shipments by courier state', 'Outstanding shipments (by carrier)', and 'Outstanding shipments by destination country'. A feedback button is visible on the right side.

Carrier	Count
Royalmail	2157
DPD (Shipatease)	373

Courier State	Count
Partially selected	1793
Selection complete	737

Destination Country	Count
BE	4
ES	32
GB	2163
IE	331

Courier Selection

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.

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.

Just-in Time Fulfilment

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

Showing all sites | Showing all organisations

Shipment Courier Selection

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

Courier Details

Shipment	A23440003976 to ELIZABETH PERCY	Detail view
Delivery Type	Courier to home	
To Address	5 ORCHARD DRIVE, SITTINGBOURNE, KENT, SN15 5ND, GB	
Courier	Royal Mail (COSS)	
Service	Next Day	
Weight	100 gram	

Modify courier selection

- DPD (Shipatease)
- MetaPack
- Royal Mail (COSS)
- UPS

Please select from one of the available courier services.

- Standard
- Next Day
- High Volume

Please select from each of the available courier service options.

Signature required

- Signature
- No signature

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.

Just-in Time Fulfilment

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.

Further Reading

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

Orders | Shipments | Transfers | Lines | Consolidate | Dashboard | > Locations | Operations | Print | Admin | Batches | Pick | Pack | Carriage | Manifests | Bulk operations

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

Please select a site:

Below are the locations being used for consolidation in site **Second**.

Used Consolidation Locations

The following view shows consolidation locations currently being used.

Location	Shipment	Priority	State	Product	Required ^⓪	Present ^⓪	Shortfall ^⓪
consolidation_13	AC_multiline_3	Normal	Consolidating	sennheiser_mm500	1	0	1
				ipod5	3	3	0
				woodworm_zoom	2	0	2
consolidation_14	AC_multiline_5	Normal	Consolidating	sennheiser_mm500	3	0	3
				ipod5	3	1	2
				woodworm_zoom	1	0	1

[View free consolidation locations in this site](#)
[View orphaned consolidation locations in this site](#)

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

Further Reading

Warehouse Processes Guide (to be published)

Warehouse Management

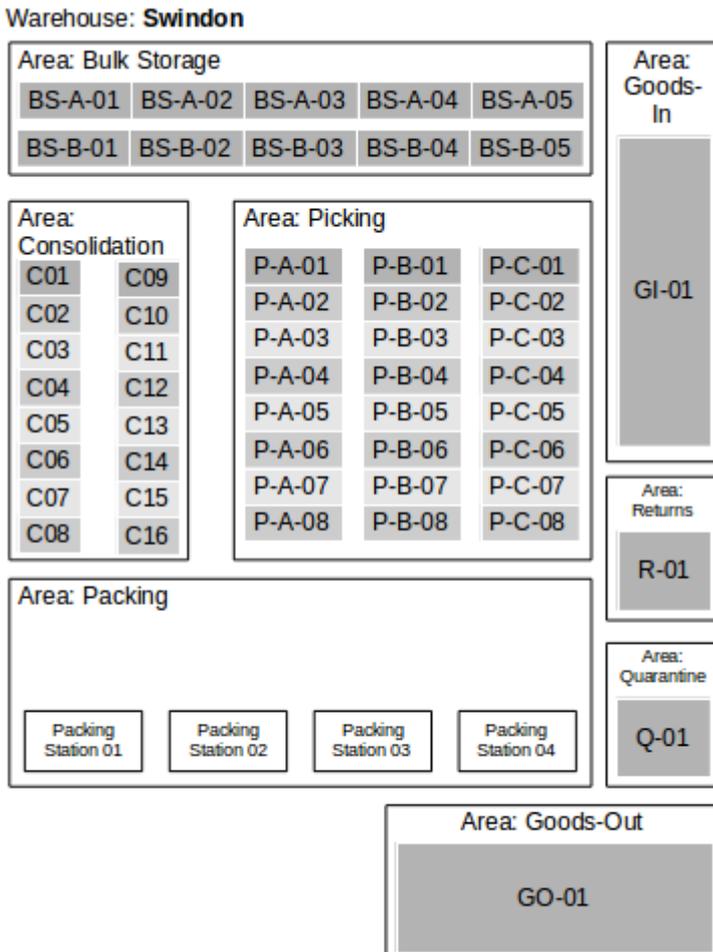
OrderFlow has user-friendly Inventory and Warehouse functions which make it easy to manage stock in an efficient and cost-effective way.

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.



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).

Warehouse Management Features

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.

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.

RealtimeDespatch charlie (Workstation unset) | Log Out | About

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

Delivery Detail Showing all sites | Showing all channels (inc. test) | Back to list | Next >>

Delivery Details

ID	83	Created	7 May, 2015 14:26:03
Site	Swindon	Type	Standard
Organisation	ElectricEtail	State	Created
Purchase Order	DMS676563009		
Purchase Order State	Partially applied		
Supplier	DMS Imports		
Supplier Reference	DMS676563009-150507		
Default Location	Not set		
Comments	No comments have been left for this Delivery. + Add a comment to this Delivery		

[Delete](#)

[Apply](#)

Use the 'Apply' button above to complete entry of the delivery into the warehouse. This will update stock levels in the warehouse.

Delivery Lines

So far, this delivery contains the following lines.

Product	Description	State	Location	Date	Quantity
TRAN805426	Coffee maker	Created	location_003	07/05/2015 14:26:42	30

[Add ...](#)

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.

Warehouse Management Features

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.

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 shows the RealtimeDespatch interface with a 'Warehouse' tab selected. The 'Tasks Dashboard' is displayed, showing several key sections:

- Open stock move tasks:** A table listing recent tasks with columns for Organisation, Type, State, Source lines, and Target lines.
- Products to put away:** A list of product references and their quantities.
- Background replenishment (from storage):** A list of product references and their quantities.
- Stock move tasks applied today:** A message indicating no tasks were applied today.

Organisation	Type	State	Source lines	Target lines
ElectricEtail	Putaway	Ready	5	5
Myco Incorporated	Handheld picking	In progress	6	6
Myco Incorporated	Putaway	Accepted	6	4
ElectricEtail	Handheld picking	In progress	11	11
ElectricEtail	Handheld picking	In progress	11	11
ElectricEtail	Handheld picking	In progress	15	15

Reference	Quantity
UniCharge35	541
TRAN805426	200
TRAN805432	50
TRAN805422	20
TRAN805431	10
KitchenGiftSet	5

Reference	Quantity
TRAN805432	1000

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.

Warehouse Management Features

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 product details

The following details have been configured for product **UniCharge35** for warehouse **Swindon**.

Strategy and Locations		Replenishment	
Picking strategy	Primary only	Pick Location Max	50
Primary picking location	location_008	Pick Replenish Threshold	20
Primary storage location	storage_1	Primary Storage Max	500
Replenishment area	Not set	Storage Replenish Threshold	200
Storage area	Not set		
Physical Storage Type	Pallet		

Reorder	
Reorder Threshold	50
Target Quantity	100

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:

Further Reading

RealtimeDespatch charlie (Packing Station 1) | [Log Out](#) | [About](#)

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

Purchase orders
ASNs
Deliveries
Work orders
Licence plates
Picking
Tasks
Returns
Search
Authorisation
New
Suppliers

Return Line Detail

Showing all sites Showing all channels (inc. test)

Details for the return line are shown below.

Return Details	
Return ID	1
Type	stock
Product	iPhone3GS
Location	quarantine_3
Quantity	1
State	Created
Reason	E - Quality/Manufacturing fault
Condition	As new
Original Line	iPhone3GS
Note	Won't turn on
Image	

[Back to return](#)

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

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.

The screenshot displays the 'Import Batches' section of the RealtimeDespatch application. The top navigation bar includes 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. The 'Import' tab is active. On the left, a sidebar contains options like 'Fetch', 'Post', 'Setup', 'Upload', 'History', and '> Batches'. The main content area shows search criteria for 'Entity' and 'Reference', with a dropdown menu open for 'Entity' listing various data types. Below the search criteria is a table of 'Import batch search results' with the following data:

ID	Type	Entity	Succeeded	Failed	Duplicates	Time Taken	Date
4526	xml	Myco I	1	0	0	303 ms	06/01/2016 17:05:37
4525	csv	Electric	2476	0	0	72995 ms	21/12/2015 10:37:30
4524	xml	Magent	54	0	0	2350 ms	21/12/2015 10:35:15
4518	xml	Myco I	1	0	0	247 ms	02/12/2015 14:26:44
4515	xml	Myco I	8	0	0	327 ms	11/11/2015 11:57:59
4511	xml	Myco I	8	0	0	319 ms	12/10/2015 16:11:27
4506	xml	Myco I	1	0	0	45 ms	11/09/2015 15:00:46
4505	xml	Myco Incorporated	1	0	0	1038 ms	11/09/2015 14:54:00
4488	xml	Myco Incorporated	8	0	0	338 ms	07/05/2015 14:47:34
4476	xml	Myco Incorporated	8	0	0	167 ms	11/03/2015 15:54:52
4475	xml	Myco Incorporated	1	0	0	277 ms	11/03/2015 15:54:05
4470	xml	Myco Incorporated	8	0	0	705 ms	20/02/2015 08:06:09
4469	xls	ElectricEtail	1	0	2	60 ms	18/02/2015 11:51:40
4468	xls	ElectricEtail	2	1	0	126 ms	18/02/2015 11:50:58
4458	xml	Myco Incorporated	1	0	0	461 ms	04/02/2015 14:33:04

At the bottom of the table, it says 'Page 1 of 2' and 'Viewing 1 - 15 of 18'. A note at the bottom right states: 'Note: the current search is limited to data up to 400 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 ".

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

Remote messages

Dashboard

> Search

Message types

API Definitions

API Operations

Process Definitions

Process Operations

Test

Search Outgoing Messages Showing all sites | Showing all channels (inc. test)

Search criteria - click to view criteria applied for this search

Remote message search results

ID	Channel	Purpose	Type	State	Reference	Retries	Created	Sent	Time Taken
1959	Magento	Inventory	Api magento soap	Sent		0	11/01/2016 15:30:02	11/01/2016	979 ms
1958		Stock change	Stock change notification	Error	1452526097080	5	11/01/2016 15:28:17		0 ms
1957	Magento	Inventory	Api magento soap	Sent		0	11/01/2016 15:28:03	11/01/2016	986 ms
1956		Stock change	Stock change notification	Error	1452526052316	5	11/01/2016 15:27:32		1 ms
1955		Stock change	Stock change notification	Error	1452526044336	5	11/01/2016 15:27:24		0 ms
1954		Stock change	Stock change notification	Error	1452525978202	5	11/01/2016 15:26:18		0 ms
1953		Stock change	Stock change notification	Error	1452525977993	5	11/01/2016 15:26:18		0 ms
1952		Stock change	Stock change notification	Error	1452525977439	5	11/01/2016 15:26:17		0 ms
1951		Stock change	Stock change notification	Error	1452525976808	5	11/01/2016 15:26:16		0 ms
1950		Stock change	Stock change notification	Error	1452525976730	5	11/01/2016 15:26:16		0 ms
1949		Stock change	Stock change notification	Error	1452525975574	5	11/01/2016 15:26:16		0 ms
1948	Magento	Inventory	Api magento soap	Sent		0	11/01/2016 15:24:02	11/01/2016	959 ms
1947	Magento	Shipment despatched	Api magento soap	Sent	100000139	0	11/01/2016 15:22:20	11/01/2016	3693 ms
1946	Magento	Order event	Order notification	Error	100000139	5	11/01/2016 15:22:17		80 ms
1945		Stock change	Stock change notification	Error	1452524313806	5	11/01/2016 14:58:33		0 ms

Page 1 of 2 | Viewing 1 - 15 of 17

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

© Realtime Despatch Software Ltd 2011-2016 Return to the top

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

API Definition Showing all sites | Showing all channels (inc. test)

Summary Info

External Reference: magento
 Name: Magento
 Type: Shopping cart
 Module: rtd2-integration-magento
 Default Message Type: api_magento_soap

[Edit API definition](#)

Channels

Channels supported for this API are shown below.

Name	Base URLs	
Magento	http://localhost/magento/api/rest http://demo.realtimedespach.co.uk/magento	View properties

Process Configurations

Process configurations linked to this API definition are shown below.

Definition	Detail	Activated	
Fetch catalogue	(Built-in)	No	<input checked="" type="checkbox"/>
Recent inventory notification	(Built-in)	Yes	<input checked="" type="checkbox"/>
Embedded inventory notification	inventory_notification_invoke	Yes	<input checked="" type="checkbox"/>
Shipment event notification	(Built-in)	Yes	<input checked="" type="checkbox"/>

Operation Definitions

Operations on the system for this API definition are shown below.

Reference	Name	Entity	Synchronous	Paginated	Timestamped	
fetch_catalogue_soap	Fetch Catalogue (SOAP)	Product	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
fetch_catalogue_rest	Fetch Catalogue (REST)	Product	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
inventory_notification_soap	Inventory Notification (SOAP)	ProductInventory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
shipment_despatch_notification_soap	Shipment despatch notification (SOAP)	Shipment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

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.

RealtimeDespatch
charlie (Packing Station 1) | [Log Out](#) | [About](#)

Despatch
Inventory
Warehouse
Import
Integration
Reports
Admin
Setup
Advanced

- Remote messages
- Message types
- API Definitions
- API Operations
- Process Definitions
- Process Operations
- Test

Process definitions

Showing all sites Showing all channels (inc. test)

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

Type [Ⓔ]	Reference and Name	Item order	Site Global [Ⓔ]	Activated
Api	action_api_remote_entry Action API remote entry			No ✘
Api	delivery_applied_notification Delivery Applied Notification	13		Yes ✘
Api	delivery_line_applied_notification Delivery Line Applied Notification	9		Yes ✘
Api	embedded_inventory_notification Embedded inventory notification	1	✓	Yes ✘
Api	fetch_catalogue Fetch catalogue	301	✓	Yes ✘
Api	fetch_new_orders Fetch new orders	311	✓	Yes ✘
Api	fetch_open_orders Fetch open orders	315	✓	Yes ✘
Api	fetch_outstanding_products Fetch outstanding products		✓	No ✘
Api	order_event_notification Order event notification	105		Yes ✘
Api	order_imported_notification Order imported notification		✓	Yes ✘
Api	product_imported Product imported			Yes ✘
Api	pull_catalogue Fetch catalogue			No ✘
Api	purchase_order_completed_notification Purchase Order Completed Notification	17		Yes ✘
Api	recent_inventory_notification Recent inventory notification		✓	Yes ✘
Api	refetch_failed_orders Refetch failed orders	319	✓	Yes ✘
Api	refetch_failed_products Refetch failed products		✓	Yes ✘
Api	remote_order_update Remote order update			No ✘
Api	remote_product_update Remote product update		✓	No ✘
Api	shipment_event_notification Shipment event notification	101		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 *warehouse* reports configured.

The screenshot shows the 'RealtimeDespatch' interface with the 'Reports' tab selected. The 'Warehouse Reports' section is active, displaying a list of reports categorized into Warehouse, Periodic, and Location reports. A search bar is visible at the top of the report list.

Warehouse Reports

Showing all sites Showing all channels (inc. test)

Below is a list of the warehouse reports that are available to run.

Type Text

Report Name	Description	Type	Formats
Barcodes	Basecodes report, giving list of barcode for products and category matches	Warehouse	jrxml
Locations (multiple) A4	Barcode and location references on A4	Warehouse	jrxml
Product Inventory by product	Product Inventory as spreadsheet	Warehouse	csv xls
Product Reorder	Product reorder report, giving list of products whose available quantity has fallen below the target reorder threshold	Warehouse	jrxml
Product Reorder (Ungrouped)	Product reorder report, giving list of products whose available quantity has fallen below the target reorder threshold. No grouping of results.	Warehouse	jrxml
Retruns 2	Returns recorded between two dates, with a status of 'applied'	Warehouse	jrxml

Report Name	Description	Type	Formats
Deliveries - end of day	Deliveries over date range, suitable as end of day report	Periodic	csv
Stock - end of day	Stock report, suitable for use as end of day report	Periodic	csv

Report Name	Description	Type	Formats
Location (multi) 107x57	Barcodes for matching location references on 107 x 57 mm label, matched by location or area	Location	jrxml
Location (single) 107x57	Single barcode and location reference on 107 x 57 mm label	Location	jrxml
Location (single) A4	Single barcode and location reference on A4	Location	jrxml

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

Print Item 1400 Showing all sites | Showing all channels (inc. test)

<< Previous | Back to list | Next >>

Details for the printed item **1400** are shown below.

Printed item	
ID	1400
Entity	Shipment
Site	Swindon
Entity Reference	100000139
State	Sent
Entity ID	15210
Created	11/01/2016 15:21:50
File Name	despatch_note_15210.xml
Printed	11/01/2016 15:22:02
Format	jasper_report_xml
Type	Despatch note
File Mime Type	application/xml
Batched	Yes (in batch 73)
Queue	APPLET
Report key	rtd2-paperwork:despatchnote_simple

[View configuration](#)

Download

Order Reference: 100000139
Order Date: 11/01/2016
Address for Returns: Returns Business
Returns Address
Returns Address
Returns Address
Returns Address
AB1 1BA

1	sw 810i	Sony Ericsson W810i
1	iPhone3GS	iPhone 3GS

Page 1 of 1

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).

The screenshot displays the 'Memory Dashboard' in the RealtimeDespatch application. The dashboard includes a navigation menu on the left with options like Users, Roles, Logging, Errors, Events, Audit entries, Jobs, Housekeeping, Health check, Performance, Dashboard, Search, > Memory, and Couriers. The main content area shows 'Views relating to memory use on the system.' and a 'Memory utilisation' table. A warning note states: 'Note that this view only shows the memory utilisation of the JVM to which the application is connected.' The table lists memory components with their used, max, and percentage values.

Name	Type	Used	Max	Percentage
Code Cache	Non-heap memory	16,653.7	50,331.6	33.1
PS Eden Space	Heap memory	83,905.1	686,030.8	12.2
PS Survivor Space	Heap memory	3,735.6	7,733.2	48.3
PS Old Gen	Heap memory	181,754.8	1,402,339.3	13.0
PS Perm Gen	Non-heap memory	111,446.6	174,063.6	64.0

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

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.

RealtimeDespatch Setup Dashboard

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

Currently on the system there is/are:

- 5 active **couriers**
- 4 **batch types**
- 5 **print queues**
- 3 **workstations**

Organisation	Organisation State	Channel(s)	Channel State	Products
ElectricEtail	Live	ElectricEtail Amazon UK ⚠️ Ebay	Test Live	29
Gee Whiz Widgets	Live	Amazon NL	Live	0
Myco Incorporated	Live	Magento ⚠️	Test	9
Total				38

Quick property group search links:

- Address
- Archive
- ASN
- Batch Processing
- Consolidation
- Correos
- Courier
- Courier (Generic)
- Courier (Shared)
- customs
- Delivery
- Despatch
- Handheld
- Import
- Inventory
- Licence Plate
- Magento API
- Messaging
- MetaPack
- Notification
- Order Entry
- Packages
- Paperwork
- Postal Sort
- Pricing
- Printing
- Product
- Purchase Order
- Remote Import
- Reports
- Returns
- Royal Mail - DMO
- Royal Mail - Tracked
- Ship@Ease
- State
- System
- Unset
- UPS
- Warehouse
- Worldship

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

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.

Setup

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

The screenshot displays the 'Setup Dashboard' in the RealtimeDespatch application. The current system state is 'operational'. A summary of system entities is provided: 5 active couriers, 4 batch types, 5 print queues, and 3 workstations. Below this, a table lists the active organisations with their states, channels, and product counts.

Organisation	Organisation State	Channel(s)	Channel State	Products
ElectricEtail	Live	ElectricEtail Amazon UK ⚠ Ebay	Test Live	29
Gee Whiz Widgets	Live	Amazon NL	Live	0
Myco Incorporated	Live	Magento ⚠	Test	9
Total				38

Below the table, there is a 'Property Groups' section with a search link and a list of property categories: Address, Archive, ASN, Batch Processing, Consolidation, Correos, Courier, Courier (Generic), Courier (Shared), customs, Delivery, Despatch, Handheld, Import, Inventory, Licence Plate, Magento API, Messaging, MetaPack, Notification, Order Entry, Packages, Paperwork, Postal Sort, Pricing, Printing, Product, Purchase Order, Remote Import, Reports, Returns, Royal Mail - DMO, Royal Mail - Tracked, Ship@Ease, State, System, Unset, UPS, Warehouse, and Worldship. An 'Override' button is visible at the bottom right of the dashboard.

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.

Setup

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.

The screenshot displays the RealtimeDespatch software interface. The top navigation bar includes tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. The user is logged in as 'charlie (Packing Station 1)'. The main content area shows the 'Property Detail' for 'Courier selection script'. The property details include: 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 this property across different scopes:

Scope	Site	Active	Value	Last updated
Shared		Yes	return;	25/03/2015 10:25:18
Gee Whiz Widgets		Yes	def delivery=values.courierDeliveryItem def countrycode ...	05/03/2013 20:18:44
Magento		Yes	def delivery=values.courierDeliveryItem def countrycode ...	25/03/2015 10:26:16
ElectricEtail Amazon UK	Swindon	Yes	def delivery=values.courierDeliveryItem def countrycode ...	16/02/2015 22:10:16

An 'Edit Property Value' dialog box is open, showing the 'Edit Value' form for the 'Courier selection script' property. The dialog includes a title, description, and applicability (Channel (Magento)). The value field contains the following script:

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

Setup

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.

The screenshot shows the 'Courier Detail' configuration page in the RealtimeDespatch application. The page is titled 'Courier Detail' and shows details for the courier 'royalmail_dmo'. The interface includes a navigation menu on the left with options like System, Channels, Organisations, Sites, Properties, Couriers, List, New, Batch types, Packaging, Print queues, Workstations, File Resources, and Countries. The main content area is divided into several sections:

- Summary:** A form with fields for Reference (royalmail_dmo), Name (Royal Mail (DMO)), Activated (No), Module (rtd2-courier-royalmail (royalMailDmo)), and Property Group (Royalmail dmo). Below these are various options with their status: Separate label printing is enabled, Label reprinting support is enabled, Shipment-level package label printing is disabled, Customs document printing support is disabled, Courier validation is not required, Courier cancellation is required, Pack scanning does not need to be courier specific, Despatch notes are not dependent on this courier, Service code selection is required, Display of the service reference is disabled, Selection of a service option(s) is required, Cancellation of the courier will not clear the selected service, and Generation of a courier manifest is not required.
- Shipment services:** A table listing services. One service is shown: Reference PK101, Name Packetsort 8 1st Class Daily Rate, Description Packetsort 8 1st Class Daily Rate, International X, Active Yes (with a red X icon). A '+ New Service' button is at the bottom right.
- Shipment Courier States:** A form with fields for Import State (Not set), Before Pack State (Selection complete), Picking State (Not set), and After Pack State (Selection complete).
- Despatch Options:** A form with fields for Despatch Ref Required (Not set) and Despatch Ref Source (Remote).

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

Setup

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.

The screenshot displays the 'Edit Batch Type' configuration page for a batch type named 'specialoffers'. The page is organized into several sections:

- Batch Type details:** Includes fields for Reference (specialoffers), Description (Single line orders for special offer (high-volume) items. Individual shipments within a batch will all be for the same high-volume SKU.), Sort Indicator (1), Print Queue (Default), Packing Type (Scan to Pack), Completion State (Sent to printer), Active (checked), Is Default (unchecked), Courier Dependent (unchecked), and Defer Despatch Note Printing (checked).
- Batch size:** Includes Min Batch Size (8), Max Batch Size (15), and an Allow Max Batch Size Override checkbox (unchecked). It also features links for 'Set channel-specific thresholds' and 'Set site-specific thresholds'.
- Picking Detail:** Includes Picking Type (Report), Picking Target (Packing), Picking Report Type (List), Picking Report Keys (rtd2-batch:picking_list_a4_barcode), Handheld Task Report Keys, Report Picking Sort (shipment.externalReference, product.externalReference), and Handheld Picking Sort (location.externalReference, product.externalReference).
- Batch creation:** Includes Batch Creation Strategy (uniqueSKUCreationStrategy).

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.

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

System

Channels

Organisations

Sites

Properties

Couriers

Batch types

Packaging

Print queues

Workstations

List

New

Java

PDF

File Resources

Countries

Edit Workstation Showing all sites Showing all channels (inc. test)

Edit workstation entry **packing1**.

Workstation details

Reference: packing1

Name: Packing Station 1

Description: Left hand warehouse packing station

Site: Swindon

Area: Workstations

Cancel Clone Update

Workstation properties

Name	Value		
despatch.note.print.queue	APPLET		
shipatease.print.queue	DPD		
shipatease.workstation.id	1		
crossdock.print.queue			
shipment.batch.print.queue			

New Property

© Realtime Despatch Software Ltd 2011-2016 Return to the top

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.

Advanced

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.

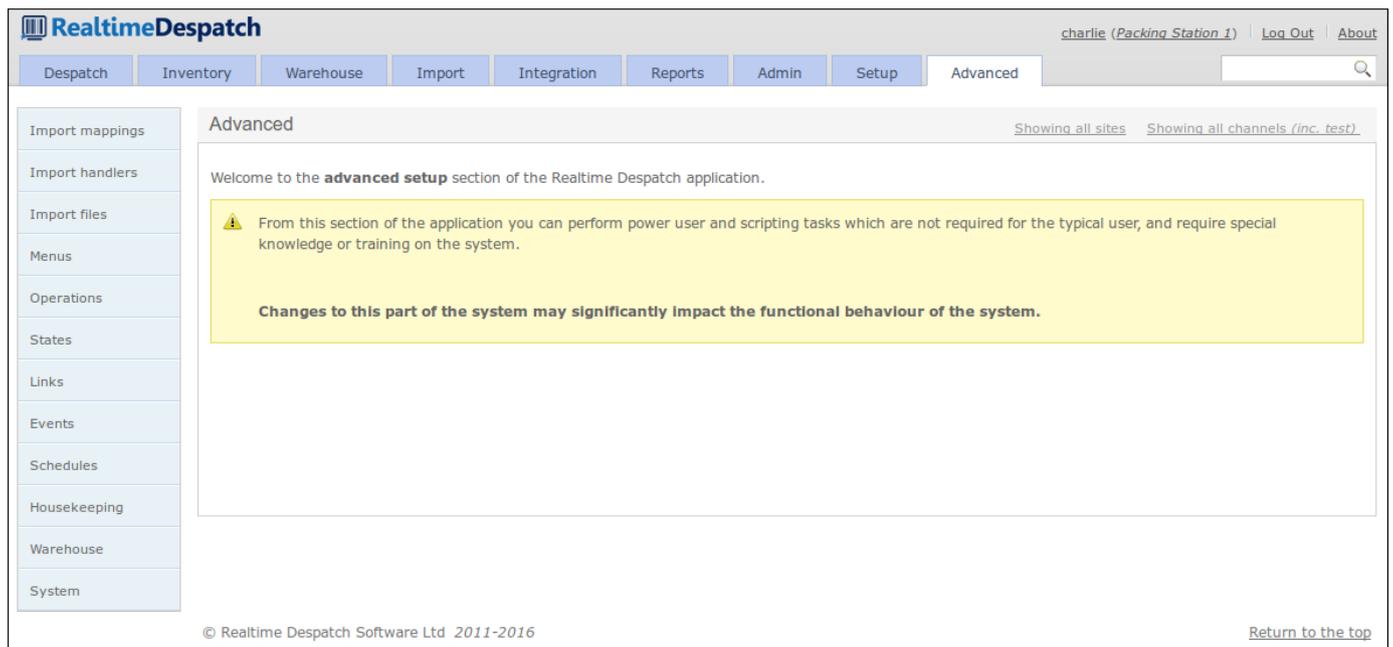
Advanced

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.



The screenshot shows the 'Advanced' configuration page in the RealtimeDespatch application. The page has a navigation bar with tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. The 'Advanced' tab is selected. On the left, there is a sidebar menu with options: Import mappings, Import handlers, Import files, Menus, Operations, States, Links, Events, Schedules, Housekeeping, Warehouse, and System. The main content area is titled 'Advanced' and contains a welcome message: 'Welcome to the **advanced setup** section of the Realtime Despatch application.' Below this is a yellow warning box with a triangle icon and the text: 'From this section of the application you can perform power user and scripting tasks which are not required for the typical user, and require special knowledge or training on the system. Changes to this part of the system may significantly impact the functional behaviour of the system.' At the bottom of the page, there is a copyright notice: '© Realtime Despatch Software Ltd 2011-2016' and a link: 'Return to the top'.

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.

Realtimedespach
charlie (Packing Station 1) | [Log Out](#) | [About](#)

Despatch
Inventory
Warehouse
Import
Integration
Reports
Admin
Setup
Advanced

- Import mappings
- Import handlers
- Import files
- List
- New
- Menus
- Operations
- States
- Links
- Events
- Schedules
- Housekeeping
- Warehouse
- System

Input File Definition Detail

[Showing all sites](#) | [Showing all channels \(inc. test\)](#)

Details for current input 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	import/xml
Activated	Yes

Folder locations

Below are the folder locations in use for this input 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	Path used for imports	Exists	Readable	Writable
Gee Whiz Widgets	/organisation-geewhiz/asn/import/xml/file	✓	✓	✓
ElectricEtail	/organisation-electricetail/asn/import/xml/file	✓	✓	✓
Mycos Incorporated	/organisation-mycos/asn/import/xml/file	✓	✓	✓

[Edit definition](#)

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

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.

RealtimeDespatch

charlie (Packing Station 1) | Log Out | About

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

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

Edit Input Handler Definition

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

Definition details

Reference

Name

Description

Entity

Handler

Format

Encoding

Transformation

Content Script

Activated

Cancel Clone Update

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

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.

RealtimeDespatch charlie (Packing Station 1) | [Log Out](#) | [About](#)

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

Edit Import Mapping [Showing all sites](#) [Showing all channels \(inc. test\)](#)

Enter details for current import mapping.

Applicability

Entity:

Channel:

Unmodifiable Fields:

Activated:

Pre-translations

Script detail [ⓘ]

```
<fieldmapper>
<mappings useinput = "true">
  <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>
  <mapping qualifier = "shipment" to = "inBatchMode">'true'</mapping>
  <mapping qualifier = "shipment" to = "userDefined4">context.storeId</mapping>
  <mapping qualifier = "order" remove = "storeId"/>
</mappings>
</fieldmapper>
```

Post-translations

Script detail [ⓘ]

```
<transformations>
  <transformation qualifier="shipment" handler="weightCalculator">
  </transformation>
  <transformation qualifier="shipment" handler="courierSetterTransformer">
  </transformation>
  <transformation qualifier="shipment" handler="batchSetterTransformer">
  </transformation>
</transformations>
```

Notes

[Cancel](#) [Clone](#) [Update](#)

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

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.

RealtimeDespatch

charlie (Packing Station 1) | Log Out | About

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

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

Search criteria

Reference:

Title:

Parent Reference: none (top-level)

Reset Search

Menu search results

Reference	Title	Parent	Mode	Order	URI
despatch	Despatch		Web	2	/despatch/view/shipment.htm
inventory	Inventory		Web	4	/warehouse/view/dashboard.htm
warehouse	Warehouse		Web	5	/warehouse/view/operations.htm
import	Import		Web	7	/import/view/dashboard.htm
integration	Integration		Web	8	/integration/home.htm
reports	Reports		Web	9	/reports/view/dashboard.htm
admin	Admin		Web	11	/admin/view/dashboard.htm
setup	Setup		Web	13	/setup/view/dashboard.htm
advanced	Advanced		Web	15	/advanced/home.htm
mobile	Mobile		Mobile	21	/mobile/dashboard.htm?screen=mobile

Page 1 of 1 | Viewing 1 - 10 of 10

© Realtime Despatch Software Ltd 2011-2016 | Return to the top

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.

RealtimeDespatch

charlie (Packing Station 1) | Log Out | About

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

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

Edit Menu

Back to list | Next >>

Current menu item **inventory/area/list**.

Menu details

Reference: inventory/area/list

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: Web

Roles:

- all
- packer
- picker
- customer_services_operator
- batch_handler
- remote_order_viewer
- remote_order_modifier
- remote_warehouse_viewer
- general
- user_administrator

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 'Courier invalid' state definition for the shipment object.

The screenshot shows the 'RealtimeDespatch' application interface. The top navigation bar includes 'Despatch', 'Inventory', 'Warehouse', 'Import', 'Integration', 'Reports', 'Admin', 'Setup', and 'Advanced'. The 'Advanced' tab is selected. The main content area is titled 'Edit State Definition' and shows the configuration for the 'courier_invalid' state for the 'rtd.domain.Shipment' entity. The state definition details include:

- Name:** courier_invalid
- Entity:** Shipment
- Title:** Courier invalid
- Description:** Shipments have failed courier validation. These shipments must be modified, if necessary, and resubmitted in order to be accepted by the courier.
- Dashboard Base URI:** (empty)
- Dashboard Base URI Description:** (empty)
- Operations:** A list of operations with checkboxes:
 - accept address
 - add order line
 - add to manifest
 - advance earliest ship
 - approve
 - approve for picking
 - bypass datasheet printing
 - bypass manifest
 - cancel
 - cancel courier
 - cancel packages
 - capture product batches
 - change courier
 - change state
 - change validate courier
 - complete consolidation
 - complete packages
 - cross dock
 - payment response
 - pick consolidated
 - prepare for package division
 - print
 - print collection label
 - print courier label
 - print customs document
 - print datasheets
 - print packages
 - rebatch
 - regenerate
 - reintroduce
 - reject address
 - release in stock lines
 - remove assignments
 - reprint
 - reprint collection label
 - reprint courier label

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.

Advanced

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.

The screenshot shows the 'Edit Operation Definition' interface in the RealtimeDespatch application. The page title is 'Edit Operation Definition' and it shows the current operation definition for 'cancel_packages' for the entity 'rtd.domain.Shipment'. The form includes the following fields and options:

- Name:** cancel_packages
- Title:** (empty)
- Entity:** Shipment
- Description:** Cancel all packages in a shipment (and remove courier validation).
- Module:** rtd2-order (with a green checkmark)
- Handler:** shipment:cancel_packages
- Check handler if allowed:**
- Target State (Scriptable):** (empty)
- Error State (Scriptable):** (empty)
- Update parent state:** Results in parent set to target state once change has been applied to all children.
- Target Parent State (Scriptable):** (empty)
- Partial Parent State (Scriptable):** (empty)
- Activated:**
- Initially Hidden:**
- Requires Confirmation:**
- Confirmation Prompt:** Are you sure you want to cancel all packages in this shipment?
- Input URI:** (empty)
- Modes:**
 - admin
 - batch
 - detail
 - list
 - pack
 - expedite
 - remote

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

Import mappings | Import handlers | Import files | Menus | Operations | States | **Links** | List | New | Events | Schedules | Housekeeping | Warehouse | System

Edit Link Definition

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

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

Link definition details

Name:

Title:

Entity:

Description:

Base URI:

Modes

<input type="checkbox"/> admin	<input type="checkbox"/> list
<input type="checkbox"/> batch	<input type="checkbox"/> pack
<input checked="" type="checkbox"/> detail	<input type="checkbox"/> expedite
	<input type="checkbox"/> remote

States

<input checked="" type="checkbox"/> allocated	<input checked="" type="checkbox"/> out of stock
<input type="checkbox"/> cancelled	<input checked="" type="checkbox"/> packable
<input type="checkbox"/> consolidated	<input type="checkbox"/> packed
<input type="checkbox"/> consolidating	<input type="checkbox"/> paused
<input checked="" type="checkbox"/> courier invalid	<input type="checkbox"/> pending bundle extraction
<input type="checkbox"/> courier not selected	<input type="checkbox"/> pending orderline merge
<input checked="" type="checkbox"/> courier not validated	<input type="checkbox"/> pending product activation
<input checked="" type="checkbox"/> created	<input type="checkbox"/> pending site selection
<input checked="" type="checkbox"/> cross dockable	<input type="checkbox"/> pending split
<input checked="" type="checkbox"/> cross docking	<input checked="" type="checkbox"/> pickable
<input type="checkbox"/> deferred	<input checked="" type="checkbox"/> picked
<input type="checkbox"/> despatched	<input type="checkbox"/> picked for consolidation
<input type="checkbox"/> held after reset	<input checked="" type="checkbox"/> ready
<input type="checkbox"/> manifested	<input checked="" type="checkbox"/> released
<input type="checkbox"/> manifesting	<input type="checkbox"/> reset
<input checked="" type="checkbox"/> move pending	
<input type="checkbox"/> on hold	

Inclusion Rule:

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

Import mappings | Import handlers | Import files | Menus | Operations | States | Links | **Events** | List | New | Schedules | Housekeeping | Warehouse | System

Edit Event Definition Showing all sites | Showing all channels (inc. test)

Current event definition **shipment_despatched**.

Event Definition Details

Name:

Description:

Source Type:

Source Entity:

Source Data:

Filter Condition:

Activated:

Persistent:

Target Delay:

[Cancel](#) [Update](#)

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

Name	Module	Handler	Activated	
shipment_despatch_listener	rtd2-notification	order_notification	No	✘
shipment_despatch_process_listener	rtd2-process	process_event	Yes	✘
shipment_despatch_courier_notifier	rtd2-courier	shipment_courier_update	No	✘

[New listener definition](#)

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

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.

RealtimeDespatch charlie (Packing Station 1) | Log Out | About

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

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

Edit Schedule Definition

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.

Run	Name	Scope	Sites	Handler	Activated
	shipment_reintroduce	Amazon NL, Amazon UK, Ebay, Magento	swindon	shipmentReintroduction (rd2-order)	Yes
	shipment_ready	Amazon NL, Amazon UK, Ebay, Magento	swindon	genericStateOperation (rd2-state)	Yes
	shipment_extract_bundle			invokeStateOperation (rd2-state)	No
	shipment_merge_order_lines			invokeStateOperation (rd2-state)	No
	shipment_send_payment_request		swindon	sendPaymentRequest (rd2-payment)	No
	consolidate_to_pickable			consolidateToPickable (rd2-order-consolidation)	No
	set_consolidation_picking_queues			invokeProcess (rd2-process)	No
	shipment_split_by_site		-	siteNotSetStateOperation (rd2-state)	No
	shipment_split_from_script		-	genericStateOperation (rd2-state)	No

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.

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

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

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

« Previous | Back to list | Next »

Enter details for location type **Incoming_preconsolidation**.

Summary

Reference: Incoming_preconsolidation

Name: Incoming - preconsolidation

Description: Incoming locations to which stock is initially credited prior to identifying a target workflow for further post goods-in operations.

Attributes

- Allocatable
- Pickable
- Picking
- Storage
- Multi-product
- Overlapping
- Tracked
- Incoming
- Outgoing
- Mobile
- Temporary
- Premium
- Quarantined
- Damaged
- System
- Unconfirmed
- Licence Plate
- Nests Licence Plate
- Consolidation
- Workstation
- Pre-receipt
- Enclosing
- Subdivided
- Subsidiary

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.

Edit Selection Definition Showing all sites | Showing all organisations (inc. test)

« Previous | Back to list | Next »

Enter details for selection definition **Initial_putaway_target**. ▶ Test

Summary

Reference:

Description:

Entity:

Type:

Activated:

Filtering

Include Attributes:

Exclude Attributes:

Selection Filter Script:

Special Entries Script:

Result Weighting Script:

Sort Expression:

Min Results:

Max Results:

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 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.

- Import mappings
- Import handlers
- Import files
- Menus
- Operations
- States
- Links
- Events
- Schedules
- Housekeeping
- Warehouse**
- Location types
- Product types
- Grouped products
- Selection definitions
- > Task definitions
- Picking definitions
- Reason codes
- System
- Test

Edit Task Definition [Showing all sites](#) [Showing all organisations \(inc. test\)](#)

[Back to list](#) | [Next >>](#)

Enter details for task definition **background_replenishment**.

Summary

Reference

Name

Description

Printed Report Key

Item Order

Action Delegate

Activated

Logging

Logging Enabled Max Log Size

Inputs

Line Report Key [🔍 View](#)

Default Product Count

Query Boost Factor

Input Parameters

Max Weight

Weight Units

Task Strategies

Strategy class

Source Line Strategy

Target Line Strategy

Line Selection Strategies

1st Source Selection [🔍 View](#)

2nd Source Selection

1st Target Selection [🔍 View](#)

2nd Target Selection [🔍 View](#)

Source allocator class

Target allocator class

Line Sorting Strategies

Source line sort expression

Target line sort expression

Applicability

Creation Modes Desktop
 Handheld
 Embedded
 Imported

Processing Modes Desktop
 Handheld
 Embedded

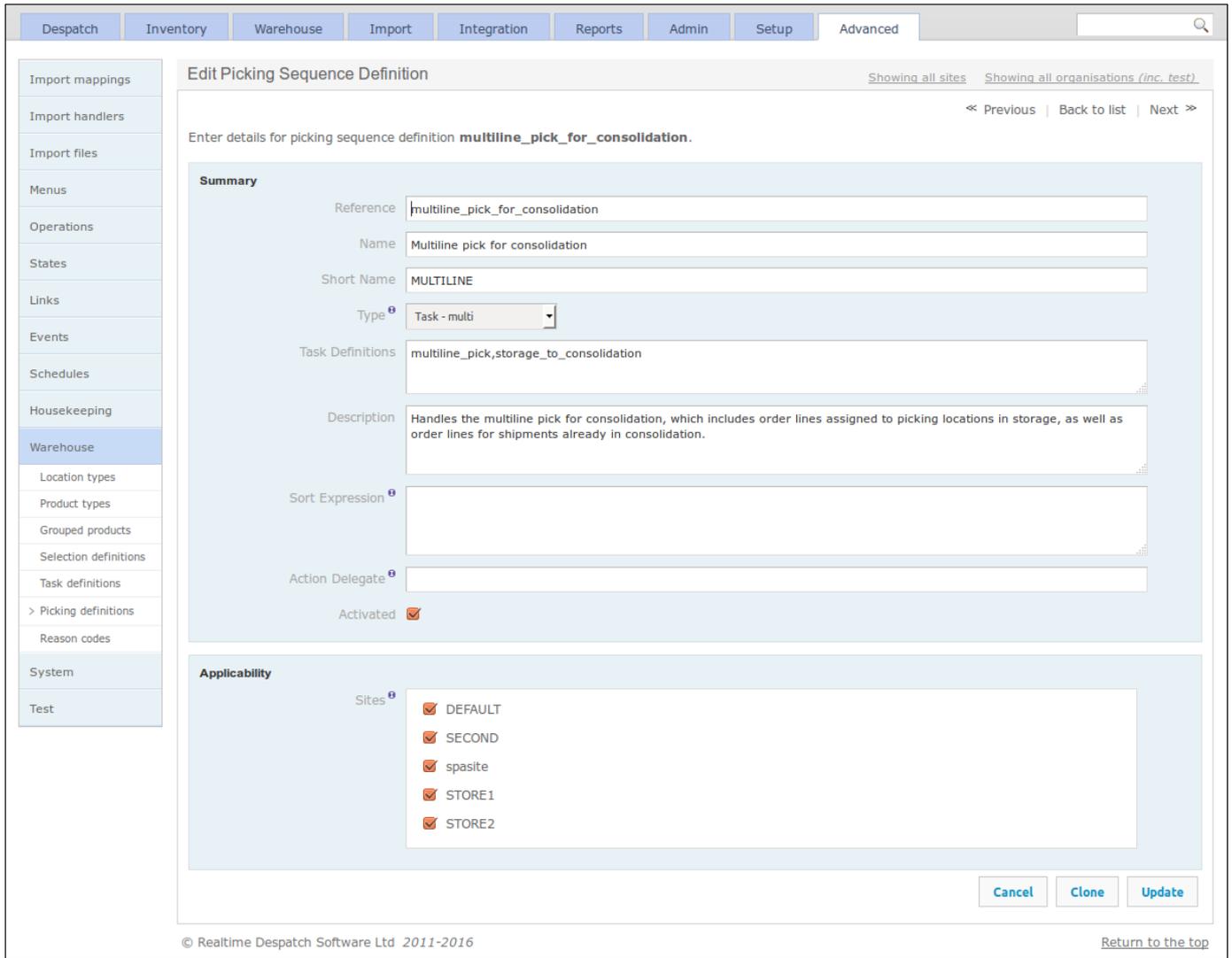
Organisations acme

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'.



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.

Despatch
Inventory
Warehouse
Import
Integration
Reports
Admin
Setup
Advanced
🔍

- Import mappings
- Import handlers
- Import files
- Menus
- Operations
- States
- Links
- Events
- Schedules
- Housekeeping
- Warehouse
- Location types
- Product types
- Grouped products
- Selection definitions
- Task definitions
- Picking definitions
- > Reason codes
- System
- Test

Edit Warehouse Reason Code

[Showing all sites](#) [Showing all organisations \(inc. test\)](#)

[Back to list](#) | [Next >>](#)

Enter details for warehouse reason code **stock_adjustment_damaged_in_process**.

Summary

Reference:

Title:

Short Title:

Description:

Type:

Type Applicability:

Order:

Activated:

Applicability

Desktop:

Handheld:

Handheld Operations (Scriptable):

Site-specific:

Sites:

- Default
- Second
- Spa Site
- Store 1
- Store 2

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

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.

Further Reading

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.

The screenshot shows the 'System' section of the OrderFlow administration interface. The top navigation bar includes tabs for Despatch, Inventory, Warehouse, Import, Integration, Reports, Admin, Setup, and Advanced. A search bar is located on the right. The left sidebar contains a list of menu items: Import mappings, Import handlers, Import files, Menus, Operations, States, Links, Events, Schedules, Housekeeping, Warehouse, System (highlighted), Loaded modules, Module profiles, Devices, and Test. The main content area is titled 'System' and includes the following elements:

- Buttons: [Showing all sites](#) and [Showing all organisations \(inc. test\)](#)
- Information box: From this section you can access advanced system-oriented information, as well as perform tasks related to the modules used for this application.
- Section: **Modules** (with expand/collapse icon)
 - Text: View currently loaded modules, or view/edit the currently loaded module profile.
 - Links: [Currently loaded modules](#), [Module profiles](#), [Handheld application](#), [Mobile application](#)
- Section: **Environment** (with expand/collapse icon)
 - Text: View information on the current application environment, including *system properties* and *loaded fonts*.
 - Links: [Caching](#), [Data Source](#), [Memory usage](#), [System properties](#), [Fonts](#)

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

Further Reading