

Version: 4.7.1

Published date: July 2025

Copyright © 2025 by ESW. All rights reserved. Information contained herein is subject to change without notice.

Multi Origin (MO)

INTEGRATION GUIDE



**SFCC - ESW Cartridge**

+

Contents

[1. Multi Origin 3](#_Toc189080084)

[2. Multi Origin Workflow 3](#_Toc189080085)

[2.1. Architectural Concept 3](#_Toc189080086)

[2.2. Use Cases 3](#_Toc189080087)

[3. SFCC Implementation 4](#_Toc189080088)

[3.1. Import metadata 4](#_Toc189080089)

[4. Enable MO feature 4](#_Toc189080090)

[5. Attribute mapping 5](#_Toc189080091)

[5.1. Basket LineItem Level 5](#_Toc189080092)

[5.2. Preorder Level 6](#_Toc189080093)

[5.3. Order Confirmation 7](#_Toc189080094)

[5.4. Order LineItem Level 8](#_Toc189080095)

[6. BM order detail page 9](#_Toc189080096)

[7. Storefront implementation 9](#_Toc189080097)

[7.1. Product listing on cart and order detail page 9](#_Toc189080098)

[7.2. Remove line item from basket 9](#_Toc189080099)

[8. Adding logic to fetch products from multiple origins 10](#_Toc189080100)

# **Multi Origin**

The ESW SFCC Cartridge enables retrieval of inventory information from various sources. For instance, if a shopper orders 5 products and the retailer has 3 in IE and 2 in FR, the ESW cartridge can check inventory details in real-time. Retailers simply need to implement logic in an override function to fetch products from different origins.

# **Multi Origin Workflow**

### **Architectural Concept**

**Option 1 (store product origin from product level attribute)**

When adding items to the cart, we will retrieve all origin details from the product attributes and forward them to the pre-order process.

**Option 2 (Get origin from the API upon add to cart)**

When adding an item to the cart, we will invoke an API to retrieve available origins and quantity based those origins using the product ID and then store these origins at the productLineItem level to send them to preorder.

### **Use Cases**

When using multiple origins, there are several scenarios to consider. Based on these scenarios, we will determine whether to create a new line item for the same product or simply increase the quantity in an existing line item. The scenarios are as follows:

* **Line item quantities available on multiple locations**
  + If a shopper orders five units of a product, with two units shipped from France (FR) and three from Ireland (IE), we will create two separate line items for the product to reflect the different shipping origins.
* **Line item has one origin**
  + If product has single origin, then this should be sent to preorder by default
* **Shopper origin available in the line item origins**
  + If a product has multiple origins and the shopper's selected location is among them, then the shopper’s location will be sent in the line item to pre order.
* **Shopper origin is not available in the line item origins**
  + If shopper’s selected country is not available in the multi origin response, then send 1st country as default.
* **Line item has no origin**
  + If there is no origin in the preOrder, then we will send fulfilment country as NULL
* **Customized retailers' origin**
  + As endpoint for getting line items origin could be different for retailers so cartridge will allow retailer to override a function where retailer can add end points to get the inventory information.

# **SFCC Implementation**

### **Import metadata**

Import custom attributes (systemobjects.xml) from **link\_eshopworld\sitesdata\meta.**

The following **ESW** custom attributeswill be available:

**SitePref:**

* ID: eswMultiOriginEnabled
* Title: ESW Multi Origin Enabled
* Type: Boolean
* Group: ESW Multi Origin

**Product level:**

* ID: eswFulfilmentCountryIso
* Title: ESW Fulfilment Country ISO
* Group: ESW Multi Origin
* Type: string

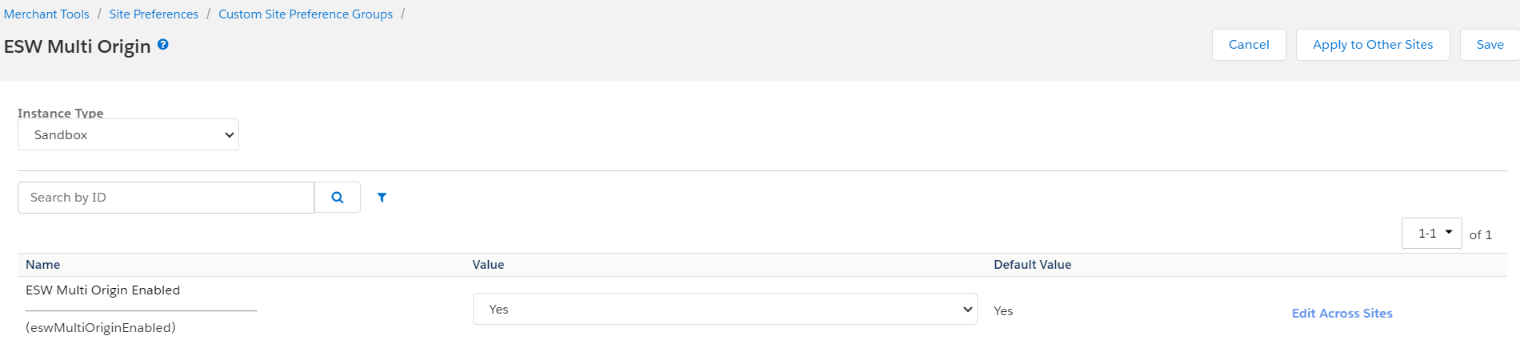
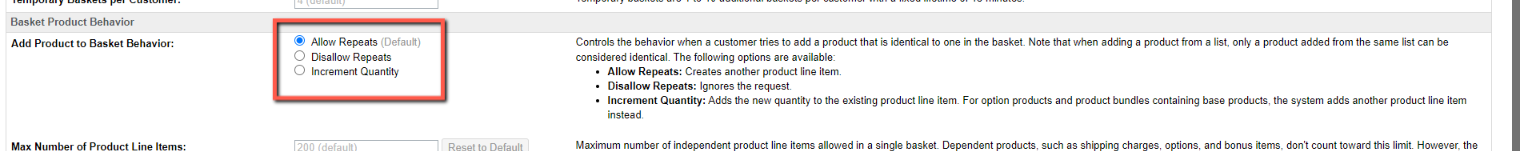
**ProductLineItem level:**

* ID: eswFulfilmentCountryIso
* Title: ESW Fulfilment Country ISO
* Group: ESW Multi Origin
* Type: string
* ID: eswDeliveryOption
* Title: ESW Delivery Option
* Group: ESW Multi Origin
* Type: string

To use the feature, make sure that preference “ESW Multi Origin Enabled” is set to “Yes”.

# **Enable MO feature**

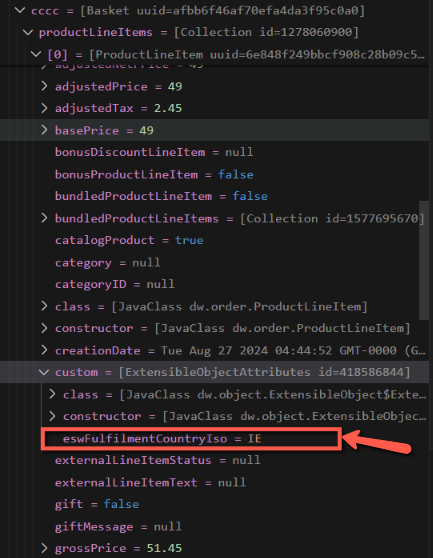
To enable the feature, retailer needs following configurations:

1. Enable feature from site preferences from group “ESW Multi Origin”  
   
2. Retailer needs to make sure that Allow Repeat is enabled from the BM Merchant “Tools > Site Preferences > Baskets”  
   

# **Attribute mapping**

### **Basket LineItem Level**

On basket level the new custom attribute “eswFulfilmentCountryIso” with the product line item is added. Screenshot is attached.



### **Preorder Level**

On the preorder level, the new custom attribute “FulfilmentCountryIso” will be added with the line items. Screenshot is attached.



### **Order Confirmation**

In order confirmation a new attriubute “fulfilmentCountryIso” can be found at line item level.

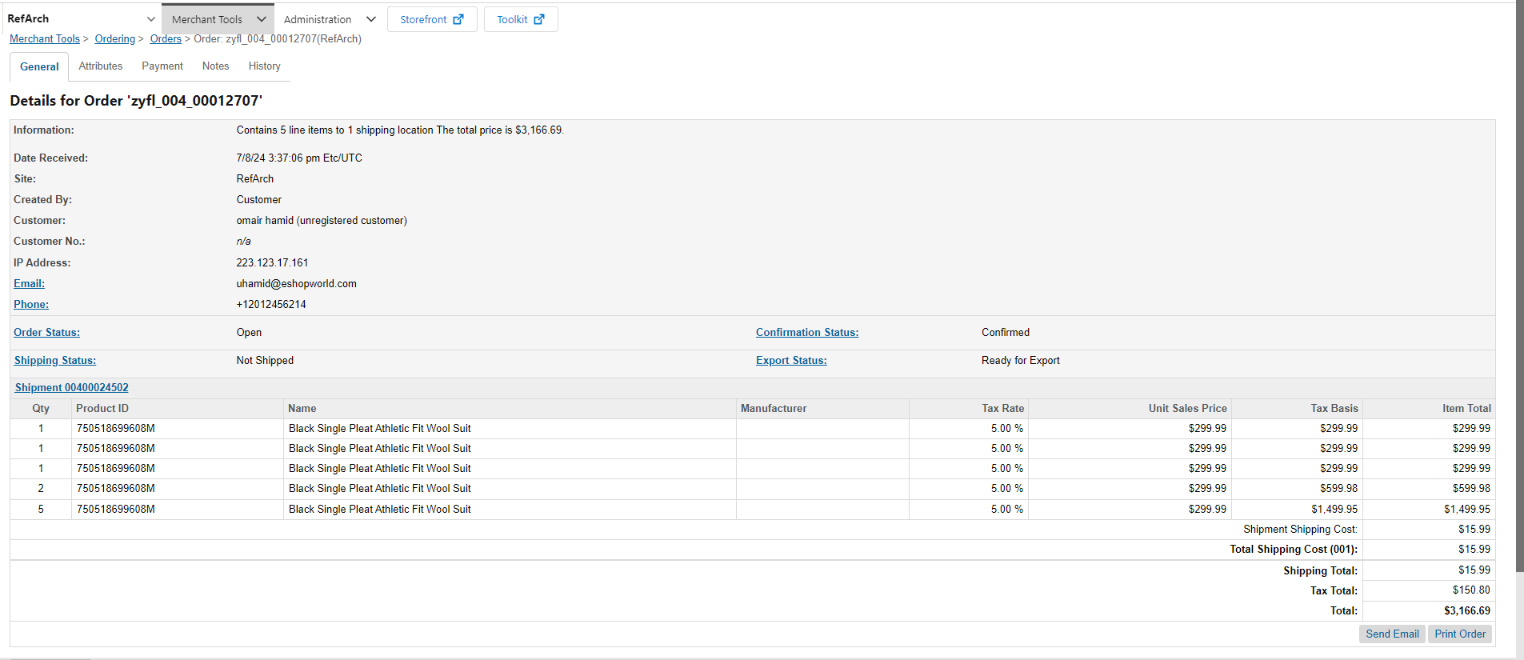


### **Order LineItem Level**

On basket level the new custom attributes “eswFulfilmentCountryIso” and “eswDeliveryOption” with the product line item is added. Screenshot is attached.



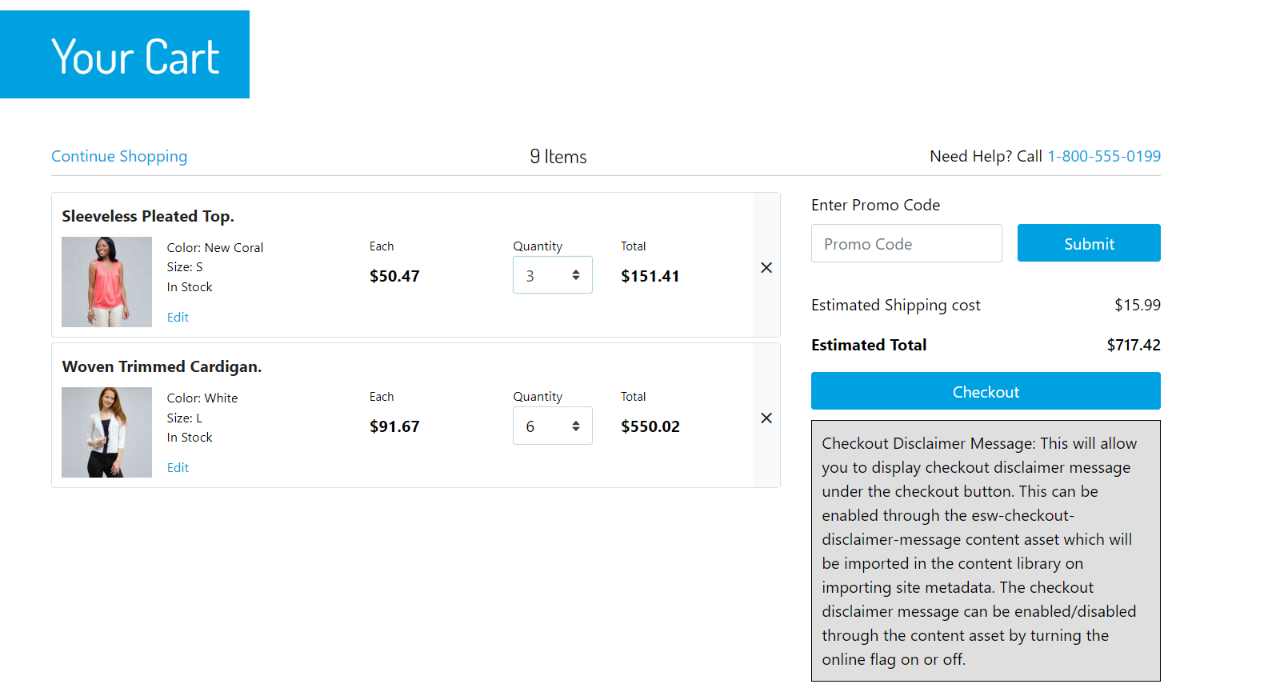
# BM order detail page

On the order detail page in the BM, the line items will shown separately for a single product if origin is different.  


# **Storefront implementation**

### **Product listing on cart and order detail page**

On the cart and order detail pages, shoppers will see the Out of Box (OOB) behaviour of SFCC. This means that for products with the same product IDs, shoppers will see cumulative quantities and prices. This implementation only affects the view and does not alter the basket or order JSON structures. In the order and basket, separate line items will still be maintained if the product origins differ.



### **Remove line item from basket**

If a shopper removes any item from the basket on the storefront, the ESW cartridge will retrieve all line items matching that product ID and remove each one.

# **Adding logic to fetch products from multiple origins**

The ESW SFCC Cartridge will provide a helper function that retailers can extend to send the product origin and quantity information. The purpose of this function is to allow the retailer to create their logic for fulfilment decisions and provide a response giving the origin of each cart item & qty.

Retailers can override a helper function getProductOriginDetails from file \link\_eshopworld\cartridges\int\_eshopworld\_core\cartridge\scripts\helper\eswMultiOriginHelper.js

The basic function definition is as follows:

/\*\*\*\*\*\*\*\*\*Function's Input\*\*\*\*\*\*\*\*\*/

// Cart Contents

var productsWithQuantities = { productId: 'pid1', quantity: 5 };

// Function's content

/\*\*

\* Return the origin and inventory information against products

\* @param {Object} productsWithQuantities - productsWithQuantities

\* @param {string} shopperCountryIso - shopperCountryIso e.g. "IE"

\* @return {Object} - productOriginDetails object

\*/

function getProductOriginDetails(productsWithQuantities, shopperCountryIso) {

// Retailer's Logic to retrieve origin country information for each product

// ...

/\*\* \*\*\*\*\*\*\*If the required quantity isn’t available at one location\*\*\*\*\*\*\*\*\*/

// In cases where the required quantity exceeds availability in one location,

// the remainder is sourced from alternative locations (see productId: pid1)

/\*\* \*\*\*\*\*\*\*If the originIso is not available for a product\*\*\*\*\*\*\*\*\*/

// If a product lacks an origin, retailer will set originIso to NULL

// ESW cartridge will check product level attribute "eswFulfilmentCountryIso"

// and Add originIso from this attribute if availble, (See productId: 'pid3')

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function's Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// The function returns an object with product ids, origins and quantities or empty array ([])

// return [

// { productId: 'pid1', originIso: 'IE', quantity: 2 },

// { productId: 'pid1', originIso: 'FR', quantity: 3 },

// ] | [];

let multiOriginResponse = [];

// If the originIso is not available for products

let nonOriginProductIds = getNonOriginProductIds(multiOriginResponse);

for (let i = 0; i < nonOriginProductIds.length; i++) {

let nonOriginDwProduct = ProductMgr.getProduct(nonOriginProductIds[i]);

// Check, If the product has eswFulfilmentCountryIso attribute

let nonOriginProductOrigin = getOriginFromProductAttribute(nonOriginDwProduct);

if (nonOriginProductOrigin) {

multiOriginResponse = updateMultiOriginIsoInFromProductAttribute(multiOriginResponse, nonOriginDwProduct, nonOriginProductOrigin);

}

}

return multiOriginResponse;

}

The function will take one argument ({ productId: productId, quantity: totalQtyRequested }

) and retailer needs to make sure to return the proposed output, the sample output is as follows:

return [{ productId: 'pid1', originIso: 'IE', quantity: 2 },{ productId: 'pid1', originIso: 'FR', quantity: 3 }] || [];