CAVALLO®

SalesPad

Shopify Integration

Overview **3**
General Information On Expressions **3**
Overview 3
Sibling and Child Relationship 6
Product Export **7**
Overview 7
General Settings 7
Matching Settings 8
Lookup Settings 8
Assignment Settings 8
Scripts 9
Processing 10
Endpoints 10
Get Products 10
Update Products 12
Update Product Variants 13
Create Product 14
Delete Product 15
Create Staged Uploads 15
Upload Media 16
Assign Media to Product 16
Product Image Export **17**
Overview 17
General Settings 17
Matching Settings 17
Lookup Settings 17
Assignment Settings 18
Scripts 18
Processing 18
Endpoints 18
Inventory Level Export **18**
Overview 18
Matching Settings 19
Assignment Settings 20
Processing 20
Endpoints 20
Get Product Variants 20

Activate Inventory 23
Update Inventory Levels 23
Customer and Order Import **24**
Overview 24
General Settings 24
Matching Settings 25
Assignment Settings 31
Scripts 36
Processing 37
Endpoints 38
Get Orders 38
Load Additional Pages 63
Update Processed Order Tag 69
Customer and Customer Address Tracing 69
Order Export **70**
Overview 71
General Settings 71
Matching Settings 71
Assignment Settings 73
Scripts 81
Processing 81
Endpoints 82
Get Order 82
Get Product Variants 91
Create Order 92
Order Update Export **103**
Overview 103
General Settings 103
Matching Settings 104
Assignment Settings 105
Processing 105
Endpoints 105
Get Order Fulfillment Information 105
Create Fulfillment 109
Move Fulfillment Order 110
Get Order 114
Capture Authorization 141
Order Invoiced Export **145**
Overview 145
General Settings 145
Assignment Settings 146
Processing 147
Endpoints 148

Get Order 148
Capture Authorization 157
Send Invoice 161
Close Order 162
Order Voided Export 162
Overview 162
General Settings 163
Assignment Settings 163
Processing 163
Endpoints 163
GraphQL and SalesPad Model Differences 164
Order Import, Order Update Export, and Order Voided Export 164
Order 164
Line Item 167
Transaction 168
Product Export and Product Image Export 169
Product 169
Product Variant 169
Other Export Components 170

Overview

Cavallo's integration with Shopify handles automatic syncing of inventory and sales information between SalesPad/GP and a Shopify website. Product and inventory level information is pushed from SalesPad to the website so that customers have visibility of which products are available. Sales orders created by customers on the website are pulled down to SalesPad so that they can be processed and fulfilled. Payment information can also be imported from the website for visibility in SalesPad. After fulfillment and tracking information is updated for each sales order in SalesPad, this information is pushed to the website for customer visibility. If sales orders are voided in SalesPad or GP, those updates are communicated back to the website.

This document covers configuration for all components of the Shopify integration.

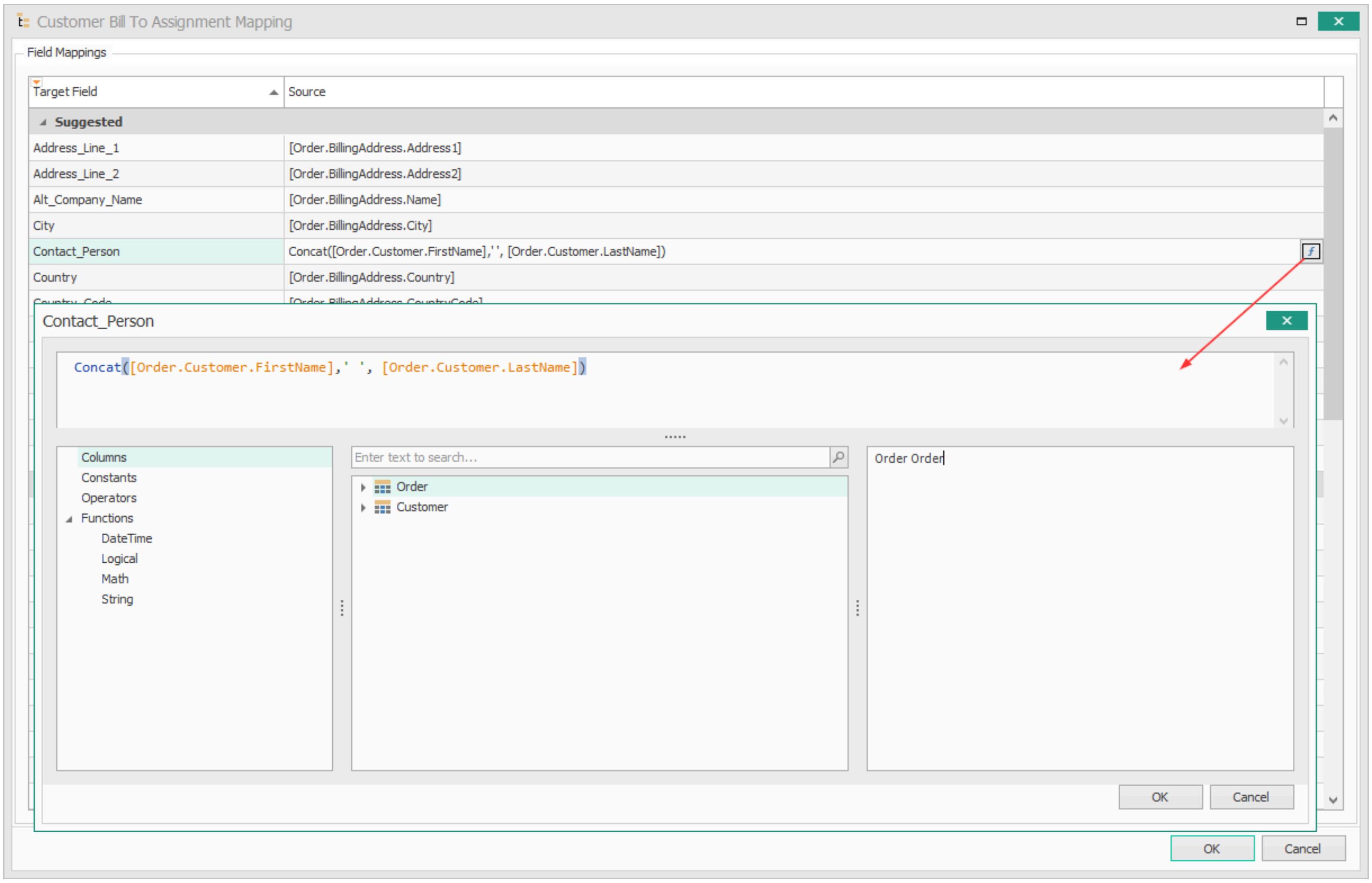
General Information On Expressions

Overview

Many integration settings allow the use of expressions to configure how internal and external entities should be matched, or to designate which values are assigned from external to internal entities.

Each expression editor will have a set of source objects to match or map from, and a set of destination objects to be matched or populated. These editors allow additional complexity beyond simply mapping source object fields to destination object fields.

As an example, below is the default expression for *Customer Bill To Assignment Mapping - Contact_Person.* In this case, the Shopify order is the source object, and a SalesPad CustomerAddr is the destination object. When a new SalesPad CustomerAddr is created in the process of creating a customer for an imported order, this expression is used to populate the Contact_Person field.

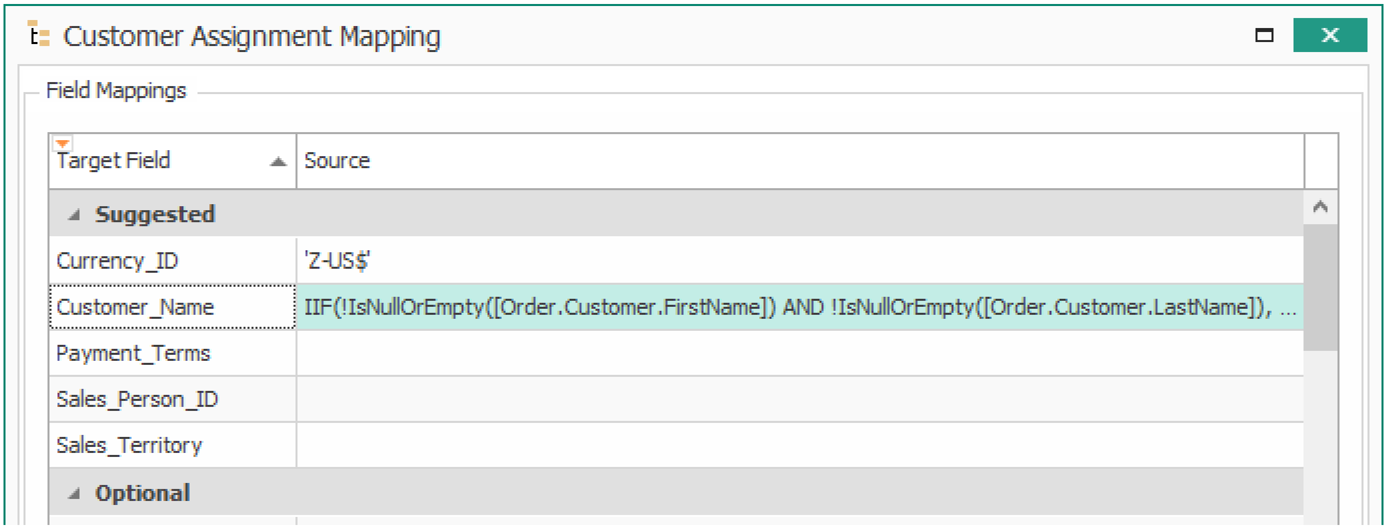


Target Field	Source
Suggested	
Address_Line_1	[Order.BillingAddress.Address1]
Address_Line_2	[Order.BillingAddress.Address2]
Alt_Company_Name	[Order.BillingAddress.Name]
City	[Order.BillingAddress.City]
Contact_Person	Concat([Order.Customer.FirstName],' ', [Order.Customer.LastName])
Country	[Order.BillingAddress.Country]
Country_Code	[Order.BillingAddress.CountryCode]

```
Concat([Order.Customer.FirstName],' ', [Order.Customer.LastName])
```

Rather than a single customer name field with the customer's full name, Shopify has separate first and last name fields, so the Contact_Person field cannot be populated by simply mapping one field to another. The Concat function is used to append the Shopify Customer last name to the first name, with a space in between. If the customer's first name is "Jane" and last name is "Doe", the Contact_Person field will be "Jane Doe".

Expression editors can also be used to assign hard-coded values when populating fields. In the below example configuration of the *Customer Assignment Mapping* setting, all customers will be created with value "Z-US$" in the Currency_ID field.

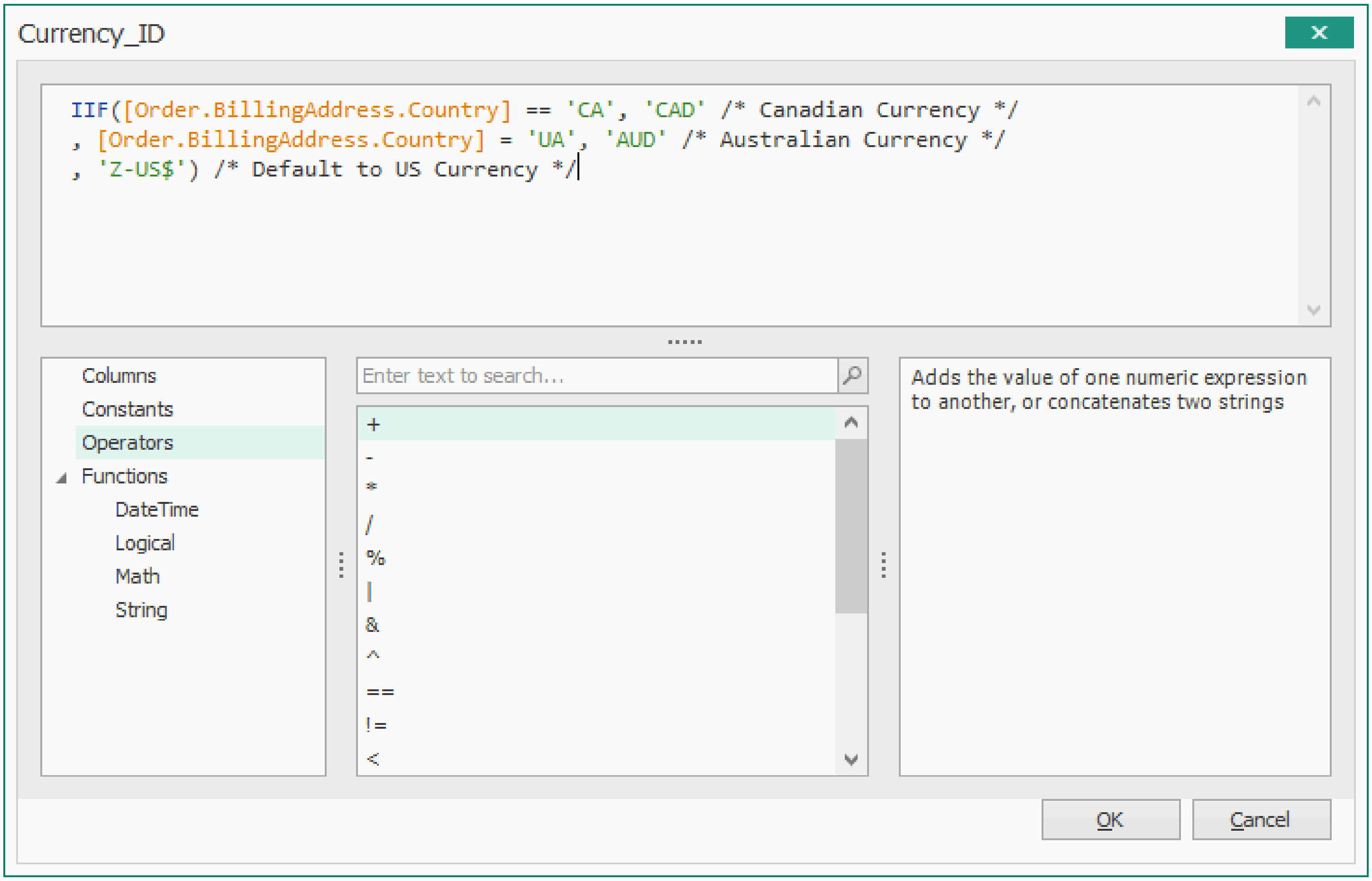


Customer Assignment Mapping

Field Mappings

Target Field	Source
Suggested	
Currency_ID	'Z-US$'
Customer_Name	IIF(!IsNullOrEmpty([Order.Customer.FirstName]) AND !IsNullOrEmpty([Order.Customer.LastName]), ...
Payment_Terms	
Sales_Person_ID	
Sales_Territory	
Optional	

For an example of a more complex use case, the below expression includes an IIF (if statement) conditional. Comments may be included to provide notes for future reference.



```
IIF([Order.BillingAddress.Country] == 'CA', 'CAD' /* Canadian Currency */
, [Order.BillingAddress.Country] = 'UA', 'AUD' /* Australian Currency */
, 'Z-US$') /* Default to US Currency */
```

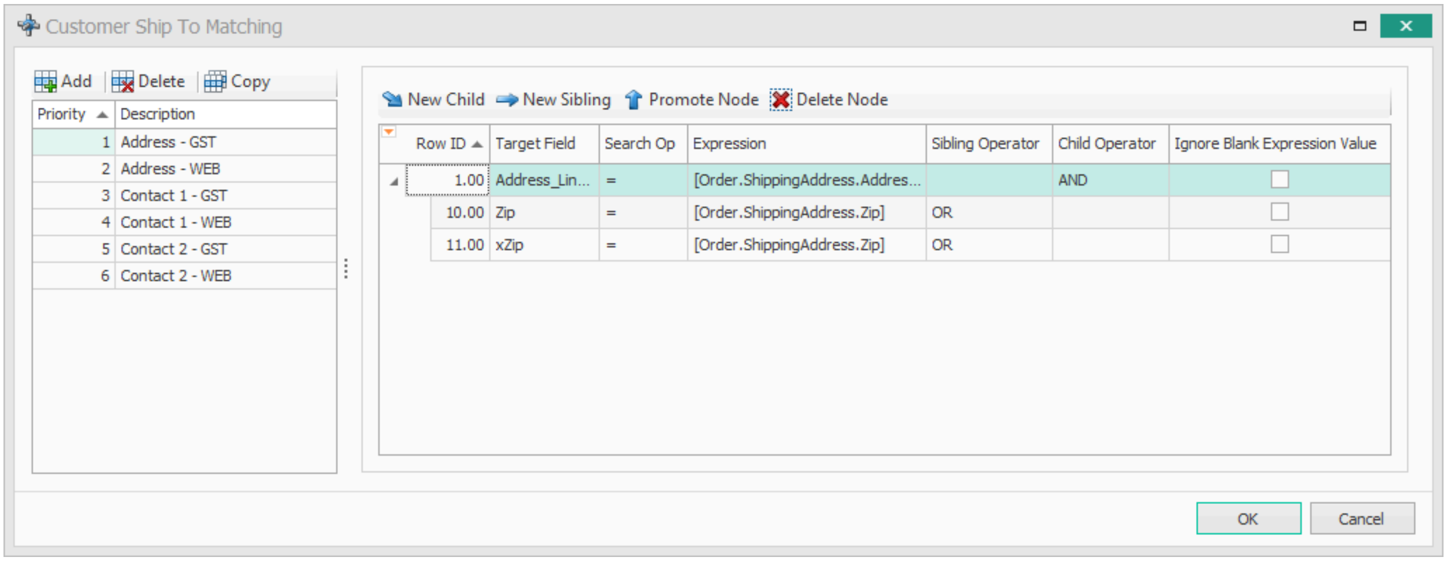
Adds the value of one numeric expression to another, or concatenates two strings

The complete expression language documentation can be found here.

Sibling and Child Relationship

A combination of sibling and child expressions can be used during matching. The expressions can be joined using logical operators such as AND, OR, etc.

For example, to match a customer address on Address Line 1 field and Zip field or xZip user field, set the Address Line 1 child operator to AND and the Zip and xZip sibling operator to OR:



Priority	Description
1	Address - GST
2	Address - WEB
3	Contact 1 - GST
4	Contact 1 - WEB
5	Contact 2 - GST
6	Contact 2 - WEB

Row ID	Target Field	Search Op	Expression	Sibling Operator	Child Operator	Ignore Blank Expression Value
1.00	Address_Lin...	=	[Order.ShippingAddress.Addres...		AND	□
10.00	Zip	=	[Order.ShippingAddress.Zip]	OR		□
11.00	xZip	=	[Order.ShippingAddress.Zip]	OR		□

Product Export

Overview

The Product Export updates products in Shopify based on new and changed item masters in SalesPad. Products which exist in both systems are updated in Shopify, and products which do not yet exist in Shopify are automatically created as part of this process. This component provides flexibility to specify which items and associated values are pushed to Shopify.

General Settings

Export Images During Product Export - *Enable this setting to export images during the Product Export. This could slow down the Product Export considerably. NOTE: Images can still be exported with Product Image Export while this setting is disabled. Defaults to 'False'.*

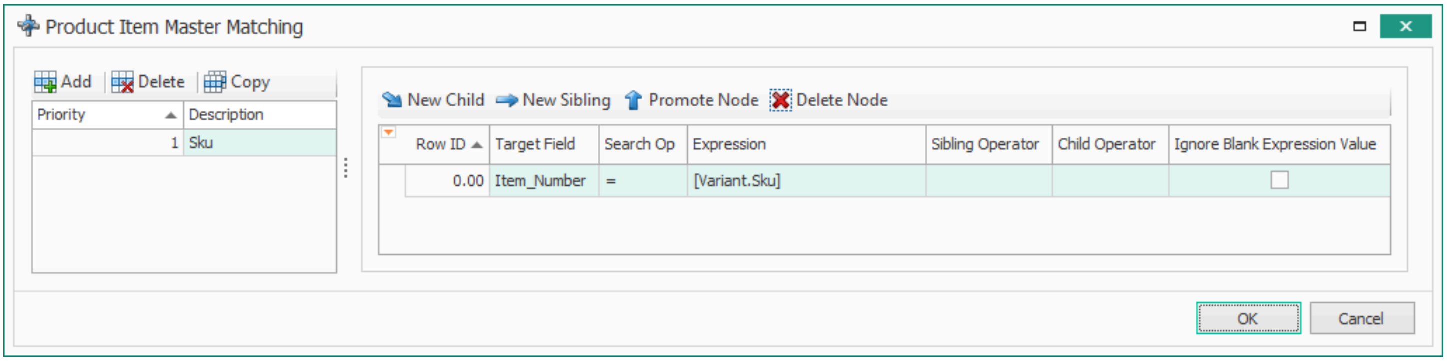
Forward Item Master After Initial Export - *If set to True, the product export will forward item masters when they're first exported to Shopify. Defaults to 'False'.*

Maximum Image Height And Width In Pixels - *Images that have a height or width in pixels greater than this setting's value will be resized to be smaller than this setting's value. If set to 0, images will not be resized but images greater than 4472 x 4472 pixels will not be uploaded. NOTE: This setting applies to both the Product Export and the Product Image Export. Defaults to '0'.*

Product Export Page Size - *Determines the maximum number of Shopify products that SalesPad will export in a single API call. Defaults to '2500'.*

Matching Settings

Product Item Master Matching - *Define the mappings for matching a Shopify Product Variant to a GP Item Master. This matching setting is only used if the items were not already matched by a previous run of the Product Export. NOTE: Item links are saved to the spAAIntegratedProductLink table. NOTE: This setting applies to both the Product Export and the Product Image Export.*



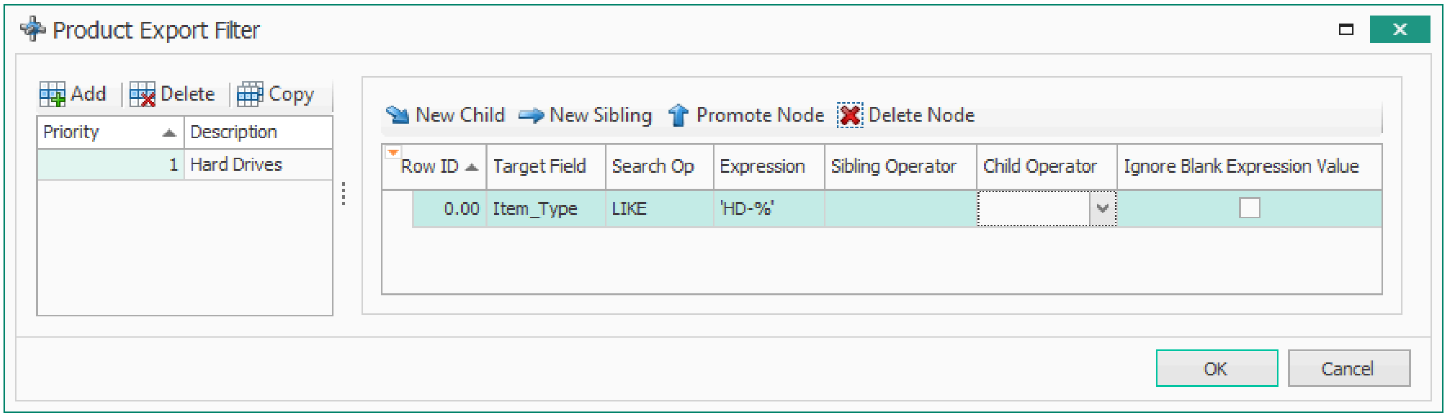
Priority	Description
1	Sku

Row ID	Target Field	Search Op	Expression	Sibling Operator	Child Operator	Ignore Blank Expression Value
0.00	Item_Number	=	[Variant.Sku]			□

Lookup Settings

Product Export Filter - *Define the criteria for determining which items export to Shopify.*

Only one priority row should be added to the left grid. The conditions on the right will be used to determine which Item Masters will be exported to Shopify. In the example below, the system will export all items that start with *HD-*.



Priority	Description
1	Hard Drives

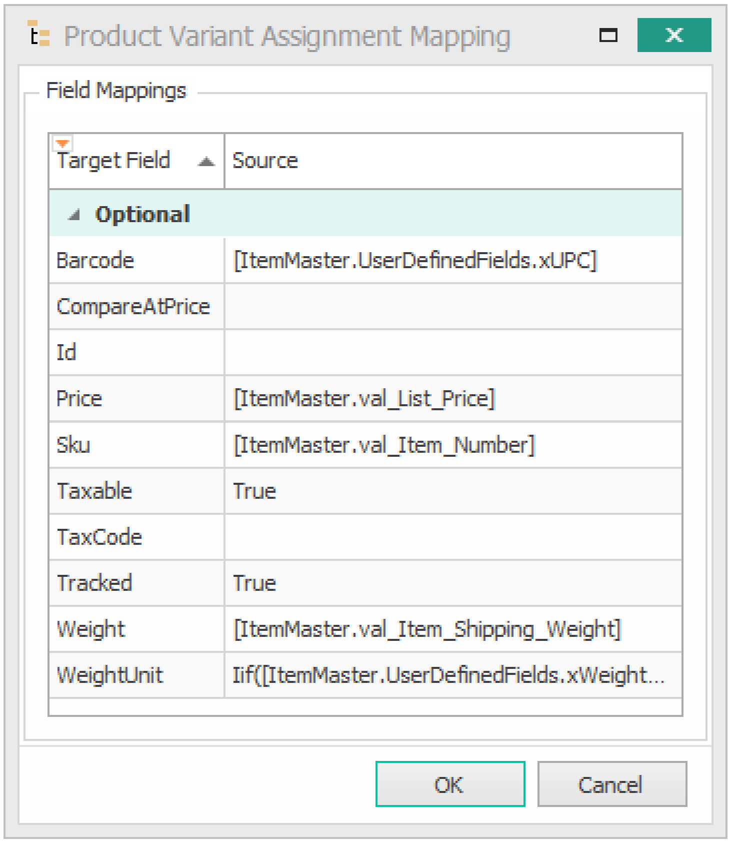
Row ID	Target Field	Search Op	Expression	Sibling Operator	Child Operator	Ignore Blank Expression Value
0.00	Item_Type	LIKE	'HD-%'			□

Assignment Settings

Item Image UDF - *Set the Item Master UDF that contains the product images to send to Shopify. NOTE: This setting applies to Product Export and Product Image Export components.*

Product Assignment Mapping - *Define the mappings to be used when exporting a Product.*

Product Variant Assignment Mapping - *Define the mappings to be used when exporting a Product Variant.*



Target Field	Source
Optional	
Barcode	[ItemMaster.UserDefinedFields.xUPC]
CompareAtPrice	
Id	
Price	[ItemMaster.val_List_Price]
Sku	[ItemMaster.val_Item_Number]
Taxable	True
TaxCode	
Tracked	True
Weight	[ItemMaster.val_Item_Shipping_Weight]
WeightUnit	Iif([ItemMaster.UserDefinedFields.xWeight...

Scripts

Product Pre Export Script - *A C# Script that runs before a Product is exported.*

Parameters: System.ComponentModel.CancelEventArgs ce, SalesPad.Bus.ItemMaster item, SalesPad.Module.Integration.Shopify.Clients.GraphQL.Product product, SalesPad.Module.Integration.Shopify.Clients.GraphQL.Variant variant

This script runs after all mapping and matching settings have run, but before creating or updating a product in Shopify. The CancelEventArgs can be used to cancel the export by setting its Cancel property to True.

Processing

The Product Export bulk loads products from Shopify and matches them to item masters in SalesPad based on the *Product Item Master Matching* setting. The *Product Export Filter* is used to filter out any items which should not be exported to Shopify. Then products that already exist in Shopify are updated if their corresponding SalesPad item has changed, and products that do not yet exist in Shopify are automatically created.

Both creation and updates use the same *Product Assignment Mapping* so that fields are updated the same way. Product variant information for both creation and updates is updated based on the *Product Variant Assignment Mapping*.

As of SalesPad 5.6.2, if *Export Images During Product Export* is set to True, the Product Export will also export item images to Shopify products. The *Item Image UDF* setting specifies which item master image user fields are sent to Shopify.

Endpoints

Get Products

At the beginning of Product Export, the integration bulk loads products and variants from Shopify.

GraphQL query:

None

```
{
  "query": "
                    mutation bulkOperationRunQuery($query: String!) {
                        bulkOperationRunQuery(query: $query) {
                            bulkOperation {
                                id
                                status
                                errorCode
                                url
                            }
                            userErrors {
                                field
                                message
                            }
                        }
                    }",
  "operationName": "bulkOperationRunQuery",
  "variables": {
    "query": "
                {
                  products {
                    edges {
                      node {
                        id
                        descriptionHtml
                        handle
                        productType
                        status
                        tags
                        templateSuffix
                        title
                        vendor
                        variants {
                          edges {
                            node {
                              id
                              barcode
                              compareAtPrice
                              price
                              taxCode
                              taxable
                              inventoryItem {
                                sku
                                tracked
                                measurement {
                                  weight {
                                    value
                                    unit
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }"
  }
}
```

Sample response:

```
None
{"id":"gid:\/\/shopify\/Product\/8061404938543","descriptionHtml":"32 meg SDRAM","handle":"32-sdram","productType":"","status":"ACTIVE","tags":[],"templateSuffix":null,"title":"32 SDRAM","vendor":"Developer Store"}
{"id":"gid:\/\/shopify\/ProductVariant\/44263493435695","barcode":"Test","compareAtPrice":null,"price":"45.00","taxCode":"","taxable":true,"inventoryItem":{"sku":"32 SDRAM","tracked":true,"measurement":{"weight":{"value":0.0,"unit":"POUNDS"}}},"__parentId":"gid:\/\/shopify\/Product\/8061404938543"}
{"id":"gid:\/\/shopify\/Product\/8061404971311","descriptionHtml":"256 meg SDRAM","handle":"256-sdram","productType":"","status":"ACTIVE","tags":[],"templateSuffix":null,"title":"256 SDRAM","vendor":"Developer Store"}
{"id":"gid:\/\/shopify\/ProductVariant\/44263493468463","barcode":"Test","compareAtPrice":null,"price":"275.00","taxCode":"","taxable":true,"inventoryItem":{"sku":"256 SDRAM","tracked":true,"measurement":{"weight":{"value":4.28,"unit":"POUNDS"}}},"__parentId":"gid:\/\/shopify\/Product\/8061404971311"}
```

Update Products

The Product Export then updates products which already exist in both systems.

GraphQL mutation:

```
None
{
  "query": "
                    mutation bulkOperationRunMutation($mutation: String!, $stagedUploadPath: String!) {
                      bulkOperationRunMutation(mutation: $mutation, stagedUploadPath: $stagedUploadPath) {
                        bulkOperation {
                          id
                          status
                          errorCode
                          url
                        }
                        userErrors {
                          field
                          message
                        }
                      }
                    }",
  "operationName": "bulkOperationRunMutation",
  "variables": {
    "mutation": "
                mutation productUpdate($input: ProductInput!) {
                  productUpdate(input: $input) {
                    userErrors {
                      field
                      message
                    }
                  }
                }",
    "stagedUploadPath":
"tmp/70387892527/bulk/3aa9a543-0316-4ab9-8eb6-6be5ed9e7a12/ShopifyGraphProductBulkUpload-20250319_162039.
jsonl"
  }
}
```

Update Product Variants

The Product Export then updates product variants which already exist in both systems.

GraphQL mutation:

None

```
mutation productVariantsBulkUpdate($productId: ID!, $variants: [ProductVariantsBulkInput!]!)
{
  productVariantsBulkUpdate(productId: $productId, variants: $variants) {
    userErrors {
      field
      message
    }
  }
}
```

Create Product

The Product Export then creates products for any SalesPad items that did not have a Shopify product match. It creates the products with the following mutation and then replaces the product's default variant with the productVariantsBulkUpdate mutation.

GraphQL mutation:

None

```
mutation productCreate($product: ProductCreateInput!) {
  productCreate(product: $product) {
    product {
      id
      variants (first: 1) {
        nodes {
          id
        }
      }
    }
    userErrors {
      field
      message
    }
  }
}
```

Sample response:

None

```
{
  "productCreate": {
    "product": {
      "id": "gid://shopify/Product/10530574369071",
      "variants": {
        "nodes": [
          {
            "id": "gid://shopify/ProductVariant/51118270546223"
          }
        ]
      }
    },
    "userErrors": []
  }
}
```

Delete Product

In the unlikely event that the productCreate mutation creates a product but the productVariantsBulkUpdate mutation fails to update the product's variant, the Product Export will delete the newly created product in order to avoid leaving a half-created product in Shopify.

GraphQL mutation:

```
None
mutation productDelete($input: ProductDeleteInput!) {
  productDelete(input: $input) {
    userErrors {
      field
      message
    }
  }
}
```

Create Staged Uploads

After creating or updating a product, the Product Export can export an image to that product. The first step is to create staged uploads for a batch of products.

GraphQL mutation:

```
None
mutation stagedUploadsCreate($input: [StagedUploadInput!]!) {
  stagedUploadsCreate(input: $input) {
    stagedTargets {
      resourceUrl
      url
      parameters {
        name
        value
      }
    }
    userErrors {
      field
      message
    }
  }
}
```

Upload Media

The second step of uploading product images to Shopify is to stream each image to Shopify's staged uploads storage. This does not use Shopify's GraphQL API; it directly uploads the images to the URLs returned by the Create Staged Uploads mutation.

Assign Media to Product

Finally, the Product Export will assign the new media to the appropriate product.

GraphQL mutation:

None

```
mutation productCreateMedia($productId: ID!, $media: [CreateMediaInput!]!) {
  productCreateMedia(productId: $productId, media: $media) {
    media {
      id
    }
    mediaUserErrors {
      code
      message
    }
  }
}
```

Product Image Export

Overview

The Product Image Export exports SalesPad item images to their corresponding Shopify products. This component is an alternative to configuring the Product Export to export product images via the *Export Images During Product Export* setting. Images can take longer to sync than other item data, so it can be helpful to separate that processing so that the Product Export can run faster and more often. This component also offers script hooks that are specific to the image export process.

This component is available as of SalesPad 5.6.2.

General Settings

Maximum Image Height And Width In Pixels - *Images that have a height or width in pixels greater than this setting's value will be resized to be smaller than this setting's value. If set to 0, images will not be resized but images greater than 4472 x 4472 pixels will not be uploaded. NOTE: This setting applies to both the Product Export and the Product Image Export. Defaults to '0'.*

Product Image Export Page Size - *Determines the maximum number of Shopify products that SalesPad will export images for in a single API call. Defaults to '100'.*

Matching Settings

Product Item Master Matching - *Define the mappings for matching a Shopify Product Variant to a GP Item Master. This matching setting is only used if the items were not already matched by a previous run of the Product Export. NOTE: Item links are saved to the spAAIntegratedProductLink table. NOTE: This setting applies to both the Product Export and the Product Image Export.*

Lookup Settings

Product Export Filter - *Define the criteria for determining which items export to Shopify.*

Assignment Settings

Item Image UDF - *Set the Item Master UDF that contains the product images to send to Shopify. NOTE: This setting applies to Product Export and Product Image Export components.*

Scripts

Product Image Pre Export Script - *A C# script that runs before a product image is exported. NOTE: This setting only applies to the Product Image Export.*

Product Image Post Export Script - *A C# script that runs after a product image is exported. NOTE: This setting only applies to the Product Image Export.*

Processing

The Product Image Export bulk loads products from Shopify and matches them to item masters in SalesPad based on the *Product Item Master Matching* setting. The *Product Export Filter* is used to filter out any items which should not be exported to Shopify. Unlike the Product Export, any item masters which aren't filtered out by the *Product Export Filter* but do not match to any Shopify products will be skipped. It is recommended to schedule the Product Export to run before the Product Image Export so that any new products can be created on the website by the Product Export and then have their images set by the Product Image Export.

The *Item Image UDF* setting specifies which item master image user fields are sent to Shopify.

Endpoints

The Product Image Export uses a subset of the endpoints which the Product Export uses. It bulk loads products, creates staged uploads, uploads media, and assigns media to products.

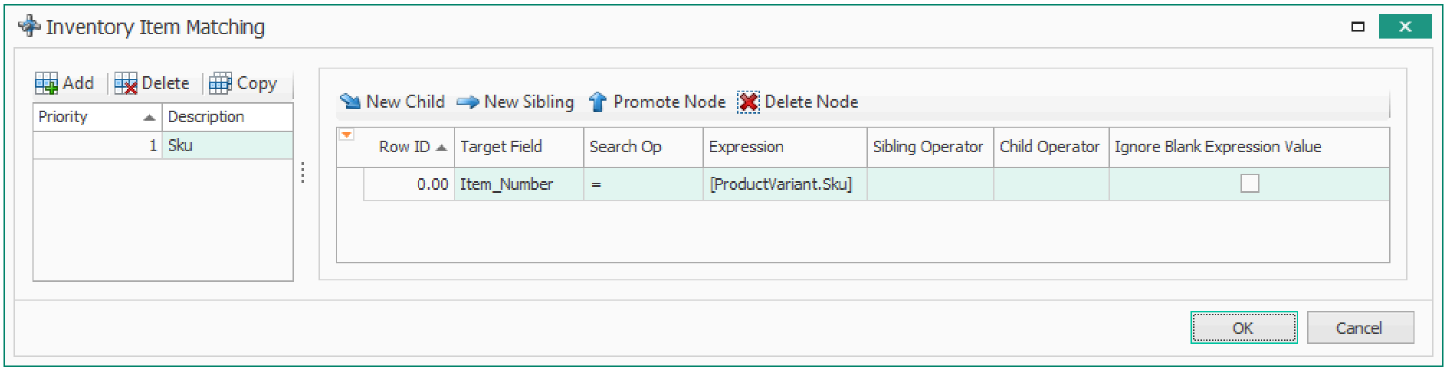
Inventory Level Export

Overview

The Inventory Level Export updates item availability information in Shopify based on current inventory levels in SalesPad warehouses. This ensures that customers placing orders on the website know how much they can order before they will have to wait for backorders.

Matching Settings

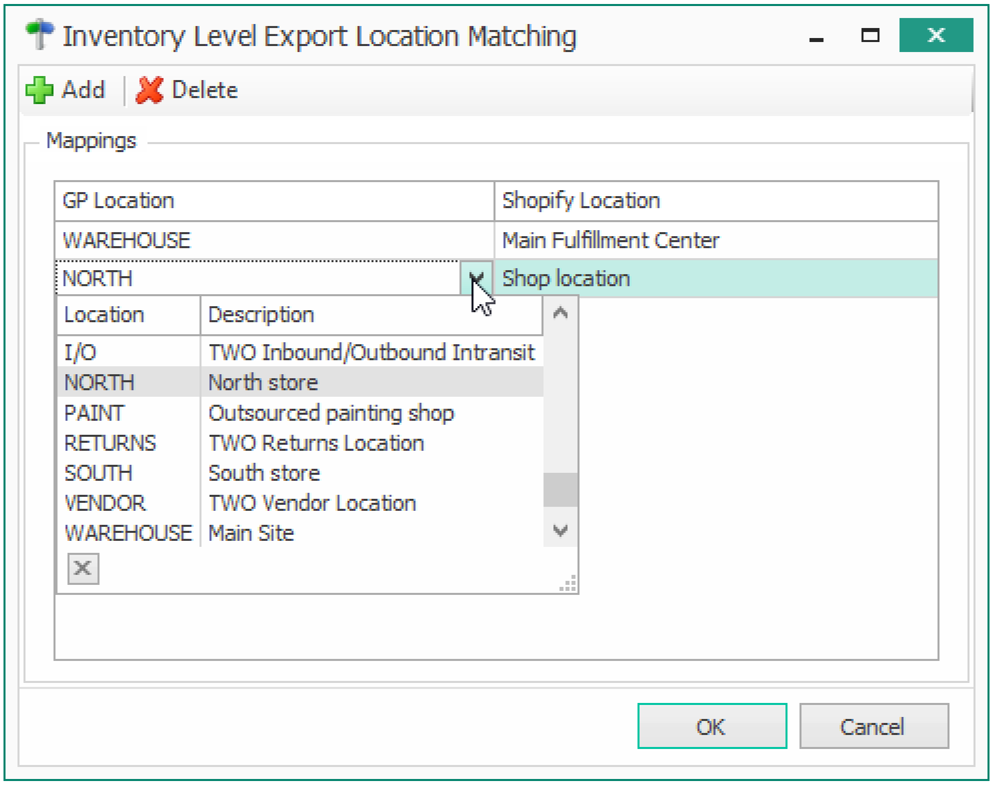
Inventory Item Matching – *Define the mappings for matching a Shopify Product Variant to a GP Item Master. This matching setting is only used if the items were not already matched by the Product Export. NOTE: Item links are saved to the spAAIntegratedProductLink table.*



Priority	Description
1	Sku

Row ID	Target Field	Search Op	Expression	Sibling Operator	Child Operator	Ignore Blank Expression Value
0.00	Item_Number	=	[ProductVariant.Sku]			☐

Inventory Level Export Location Matching – *Define which Shopify location to use for each GP location in the Inventory Level Export. A GP location can be matched to multiple Shopify locations, but multiple GP locations cannot be matched to the same Shopify location. Any unmatched Shopify locations will not be updated.*

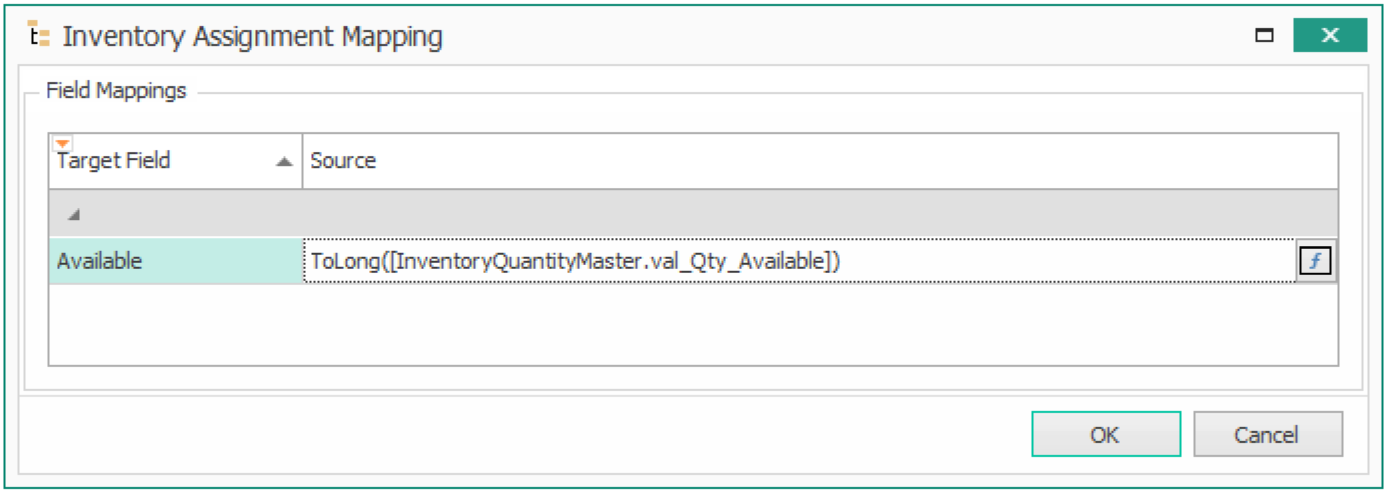


GP Location	Shopify Location
WAREHOUSE	Main Fulfillment Center
NORTH	Shop location

Location	Description
I/O	TWO Inbound/Outbound Intransit
NORTH	North store
PAINT	Outsourced painting shop
RETURNS	TWO Returns Location
SOUTH	South store
VENDOR	TWO Vendor Location
WAREHOUSE	Main Site

Assignment Settings

Inventory Assignment Mapping - *Define the mappings to be used when exporting inventory quantities.*



Target Field	Source
Available	ToLong([InventoryQuantityMaster.val_Qty_Available])

Processing

The Inventory Level Export bulk loads inventory levels and products from Shopify. The *Inventory Item Matching* setting is used to match Shopify products to SalesPad items, and the *Inventory Level Export Location Matching* setting is used to match Shopify locations to SalesPad warehouses. Then the *Inventory Assignment Mapping* setting is used to update the Shopify item availability based on its availability in its corresponding SalesPad item and warehouse.

Endpoints

Get Product Variants

At the beginning of the Inventory Level Export, the integration bulk loads product variants and inventory levels from Shopify.

GraphQL query:

```
None
{
  "query": "
                    mutation bulkOperationRunQuery($query: String!) {
                        bulkOperationRunQuery(query: $query) {
                            bulkOperation {
                                id
                                status
                                errorCode
                                url
                            }
                            userErrors {
                                field
                                message
                            }
                        }
                    }",
  "operationName": "bulkOperationRunQuery",
  "variables": {
    "query": "
                {
                  productVariants {
                    edges {
                      node {
                        id
                        sku
                        inventoryItem {
                          id
                          legacyResourceId
                          inventoryLevels {
                            edges {
                              node {
                                id
                                location {
                                  id
                                  legacyResourceId
                                }
                                quantities (names:[\"available\"]) {
                                  name
                                  quantity
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }"
  }
}
```

Sample response:

```
None
{"id":"gid:\/\/shopify\/ProductVariant\/44263493435695","sku":"32 SDRAM","inventoryItem":{"id":"gid:\/\/shopify\/InventoryItem\/46311720157487","legacyResourceId":"46311720157487"}}
{"id":"gid:\/\/shopify\/InventoryLevel\/112397975855?inventory_item_id=46311720157487","location":{"id":"gid:\/\/shopify\/Location\/76014846255","legacyResourceId":"76014846255"},"quantities":[{"name":"available","quantity":0}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/InventoryLevel\/112499163439?inventory_item_id=46311720157487","location":{"id":"gid:\/\/shopify\/Location\/76116001071","legacyResourceId":"76116001071"},"quantities":[{"name":"available","quantity":413}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/InventoryLevel\/113078632751?inventory_item_id=46311720157487","location":{"id":"gid:\/\/shopify\/Location\/76694225199","legacyResourceId":"76694225199"},"quantities":[{"name":"available","quantity":0}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/InventoryLevel\/113950982447?inventory_item_id=46311720157487","location":{"id":"gid:\/\/shopify\/Location\/77564936495","legacyResourceId":"77564936495"},"quantities":[{"name":"available","quantity":0}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/InventoryLevel\/133742625071?inventory_item_id=46311720157487","location":{"id":"gid:\/\/shopify\/Location\/97242382639","legacyResourceId":"97242382639"},"quantities":[{"name":"available","quantity":413}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/ProductVariant\/44263493468463","sku":"256 SDRAM","inventoryItem":{"id":"gid:\/\/shopify\/InventoryItem\/46311720190255","legacyResourceId":"46311720190255"}}
{"id":"gid:\/\/shopify\/InventoryLevel\/112397975855?inventory_item_id=46311720190255","location":{"id":"gid:\/\/shopify\/Location\/76014846255","legacyResourceId":"76014846255"},"quantities":[{"name":"available","quantity":0}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493468463"}
{"id":"gid:\/\/shopify\/InventoryLevel\/112499163439?inventory_item_id=46311720190255","location":{"id":"gid:\/\/shopify\/Location\/76116001071","legacyResourceId":"76116001071"},"quantities":[{"name":"available","quantity":247}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493468463"}
{"id":"gid:\/\/shopify\/InventoryLevel\/113078632751?inventory_item_id=46311720190255","location":{"id":"gid:\/\/shopify\/Location\/76694225199","legacyResourceId":"76694225199"},"quantities":[{"name":"available","quantity":428}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493468463"}
{"id":"gid:\/\/shopify\/InventoryLevel\/113950982447?inventory_item_id=46311720190255","location":{"id":"gid:\/\/shopify\/Location\/77564936495","legacyResourceId":"77564936495"},"quantities":[{"name":"available","quantity":0}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493468463"}
```

```
{"id":"gid:\/\/shopify\/InventoryLevel\/133742625071?inventory_item_id=46311720190255","location":{"id":"gid:\/\/shopify\/Location\/97242382639","legacyResourceId":"97242382639"},"quantities":[{"name":"available","quantity":247}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493468463"}
```

Activate Inventory

If a product is not assigned to a location, the Inventory Level Export will activate that product at that location.

GraphQL mutation:

None

```
mutation inventoryActivate($inventoryItemId: ID!, $locationId: ID!, $available: Int) {
  inventoryActivate(inventoryItemId: $inventoryItemId, locationId: $locationId, available: $available) {
    userErrors {
      field
      message
    }
  }
}
```

Update Inventory Levels

The Inventory Level Export then updates inventory levels for products which exist in both systems.

GraphQL mutation:

None

```
mutation inventoryAdjustQuantities($input: InventoryAdjustQuantitiesInput!) {
  inventoryAdjustQuantities(input: $input) {
    userErrors {
      field
      message
    }
  }
  }
```

Customer and Order Import

Overview

The Order Import retrieves all orders in Shopify that are ready to be imported. It matches customer and contact information to existing customers and contacts in SalesPad, or it creates customers and addresses as needed, before creating the sales orders. SalesPad orders and lines are created based on configurable mapping settings, then they are linked back to their corresponding Shopify orders, and finally their Shopify orders are flagged as imported.

General Settings

Customer Primary Address - *When the Order Import creates a new customer, it will mark the selected address as the customer's primary address. Defaults to 'Bill To'.*

Enable Order Import Trace - *If enabled, customer and customer address matching information will be logged during order import. This setting should be enabled for troubleshooting purposes only. Defaults to 'False'.*

Financial Status Filter - *Specify one or more financial statuses which a Shopify order must have in order to be imported. Defaults to 'Paid'.*

Forward Document After Import - *If enabled, the imported order will be forwarded in workflow after being saved. Defaults to 'False'.*

Fulfillment Status Filter - *Specify one or more fulfillment statuses which a Shopify order must have in order to be imported. Defaults to 'Unfulfilled'.*

Load Order Payment Terms - *If set to 'True', the Order Import will load each order's payment terms. This requires the 'read_payment_terms' access scope to be selected for your SalesPad connector app in Shopify. NOTE: Only the first 250 payment schedules will be included. Defaults to 'False'.*

Multiple Potential Customers Scenario - Review Queue - *Queue that contains orders where a definitive customer match couldn't be found due to multiple possibilities being present. By default, the* *order will use the customer that has the earliest created date, then the order will be moved to the Workflow Queue designated by this setting to be reviewed.*

Named Notes Tab for Shopify Order Comments - *Specify which tab to import Shopify order comments to on the Sales Document. Defaults to 'Internal Notes'.*

Number Of Days To Look Back - *Specify the number of days to look back from today to import orders. For example, set to 30 to import orders only from the last 30 days. Set to zero to import orders from any time. Defaults to '0'.*

Number Of Orders Per Page - *Specify the number of orders to import for each page. (Maximum of 50). Defaults to '25'.*

Payment Financial Status Filter - *Specify one or more financial statuses for which a payment will be created after a successful import. Defaults to 'Paid'.*

Processed Order Tag - *After an order is imported to SalesPad, this tag will be written to the Shopify order to prevent subsequent imports. Defaults to 'EXPORTED_TO_SALESPAD'.*

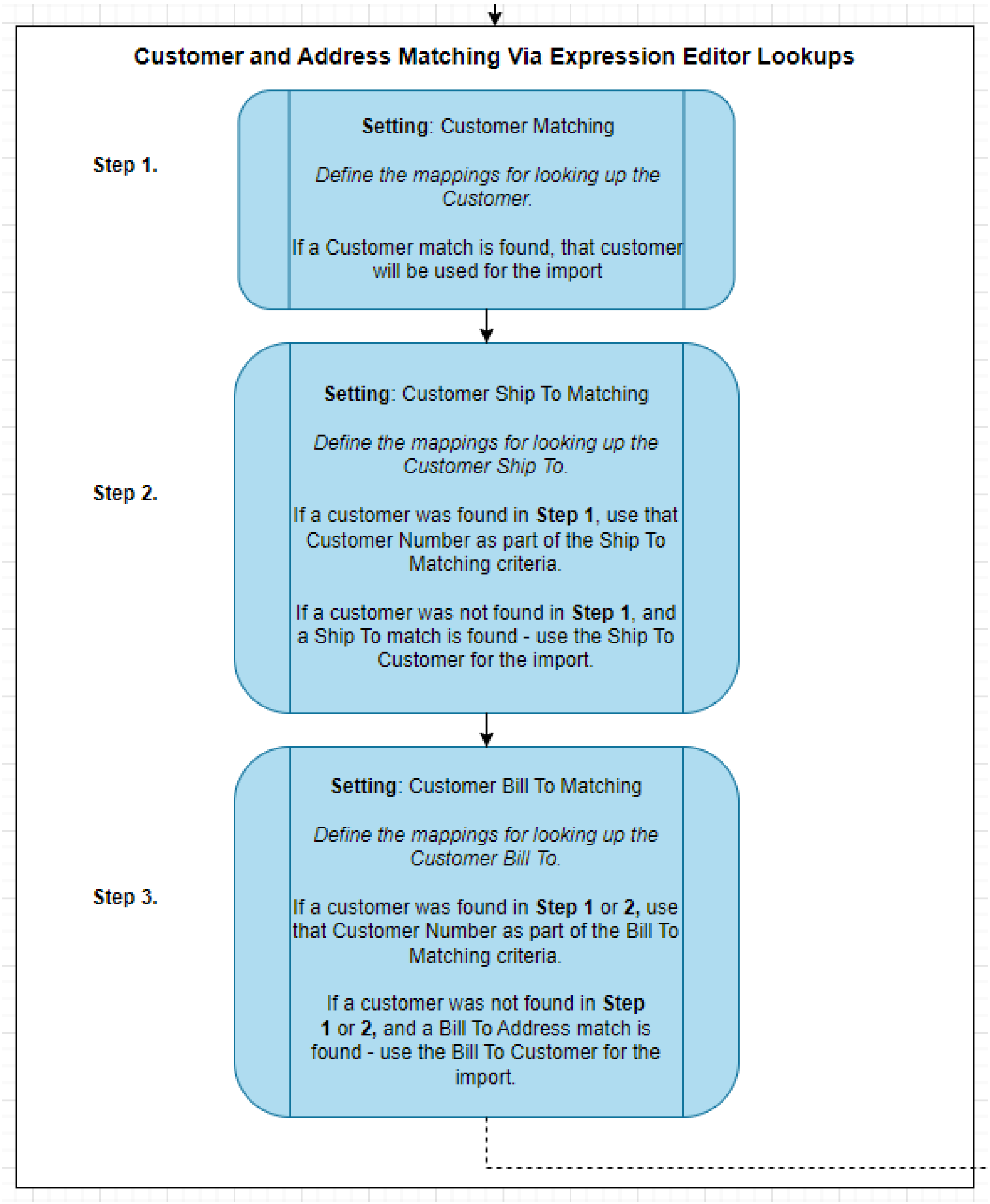
Roll Back Order Import Transaction On Error - *When enabled, the transaction encompassing the order import will be rolled back when an error occurs. This prevents data from a partially completed import from being saved to the database. Defaults to 'True'.*

Matching Settings

These settings use Expression Editors that allow creating custom matching criteria to determine how a Shopify entity should match to a SalesPad entity.

Each Matching setting will build one or more SQL queries to attempt to find a match in the SalesPad database. These queries are executed one at a time in order of Priority (lowest to highest). If an earlier priority query returns a result, that result will be returned to be used in the import, and the subsequent priorities will not execute.

The diagram below illustrates the sequence in which SalesPad attempts to match a SalesPad Customer, Ship To Address, and Bill To Address each time a Shopify Order is imported.



Customer and Address Matching Via Expression Editor Lookups

Step 1.

Setting: Customer Matching

Define the mappings for looking up the Customer.

If a Customer match is found, that customer will be used for the import

Step 2.

Setting: Customer Ship To Matching

Define the mappings for looking up the Customer Ship To.

If a customer was found in **Step 1**, use that Customer Number as part of the Ship To Matching criteria.

If a customer was not found in **Step 1**, and a Ship To match is found - use the Ship To Customer for the import.

Step 3.

Setting: Customer Bill To Matching

Define the mappings for looking up the Customer Bill To.

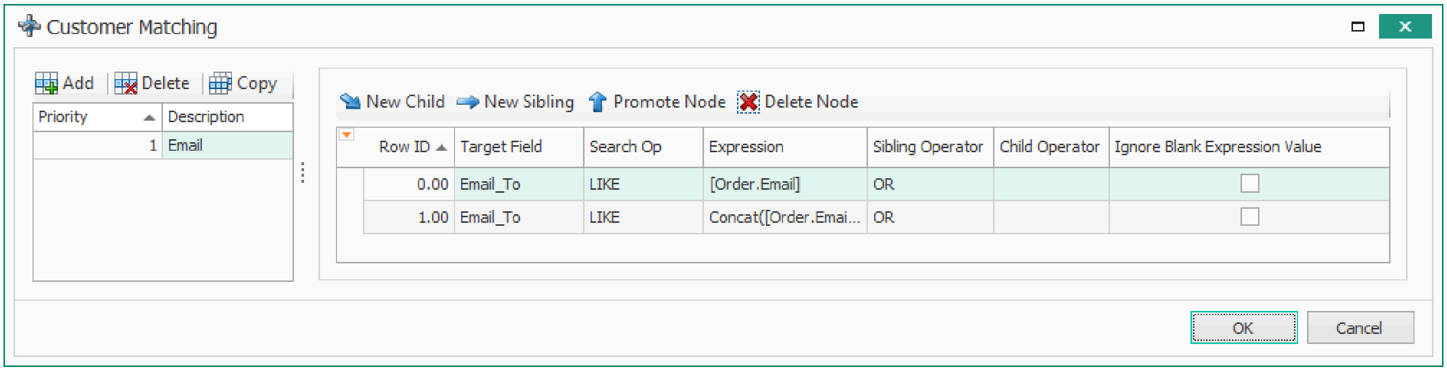
If a customer was found in **Step 1** or **2**, use that Customer Number as part of the Bill To Matching criteria.

If a customer was not found in **Step 1** or **2**, and a Bill To Address match is found - use the Bill To Customer for the import.

Customer Matching – *Define the mappings for looking up the Customer. This matching setting is only used if the customers were not already matched by a previous run of the Order Import. NOTE: Customer links are saved to the spAAIntegratedCustomerLink table.*

This setting is used as the first attempt to match a SalesPad Customer to the Shopify Customer. If a customer is found in this step, that customer's Customer Number will be added to the search criteria when looking up the Ship To and Bill To Addresses.

The default settings attempt to match a SalesPad Customer based on the email of the Shopify Order. The conditions include a LIKE operator combined with an OR sibling operator. The LIKE operator indicates that the Email_To field must contain, but not exactly match, the expression. The OR sibling operator indicates that at least one of the conditions must match in order for a customer record to be considered a match. In the example below, the customer's Email_To field must match either the Order.Email field, or the Order.Email field with a ';' character appended. For example, if the Shopify customer's email is "aaronfitz@gmail.com", a SalesPad customer with "aaronfitz@gmail.com;aaron@fitzelectrical.com" would be a match. The approximate SQL lookup executed for this condition would be:
"WHERE (customer.Email_To LIKE '%aaronfitz@gmail.com%') or customer.Email_To LIKE '%aaronfitz@gmail.com;%'



Priority	Description
1	Email

Row ID	Target Field	Search Op	Expression	Sibling Operator	Child Operator	Ignore Blank Expression Value
0.00	Email_To	LIKE	[Order.Email]	OR		□
1.00	Email_To	LIKE	Concat([Order.Emai...	OR		□

Note that this setting does not need to be populated, since the customer can also be matched indirectly via **Customer Ship To Matching** or **Customer Bill To Matching** settings. If Shopify Orders should be matched by the shipping address instead, then the *Customer Matching* setting can be cleared, and the *Customer Ship To Matchin*g and/or *Customer Bill To Matching* settings will be used to load the Ship To / Bill To and the corresponding customer.

Customer Ship To Matching - *Define the mappings for looking up the Customer Ship To Address.*

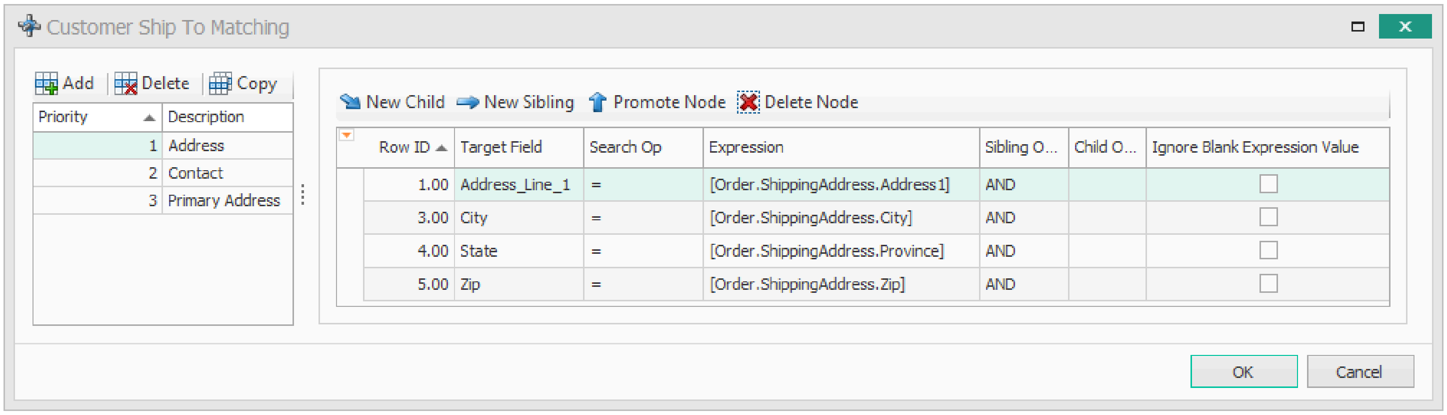
This setting determines how SalesPad will attempt to look up the Ship To Address for an incoming Shopify order. If a SalesPad Customer was matched via the **Customer Matching** setting, the Ship To Address must also belong to that customer. Otherwise, the search will consider address codes across all customers.

For example, if the customer 'AARONFIT0001' has already been matched, only address codes having a Customer_Num of 'AARONFIT0001' will be considered, even if they otherwise match the criteria in the Ship To Address Matching setting. If a customer has not yet been matched when a Shipping Address is found, the order will be imported under the customer for that shipping address.

The default settings attempt to match a SalesPad Shipping Address in three different steps.

Priority 1

Address: Attempt to find a match based on Address Line 1, City, State, and Zip. All four of these fields must be matched.

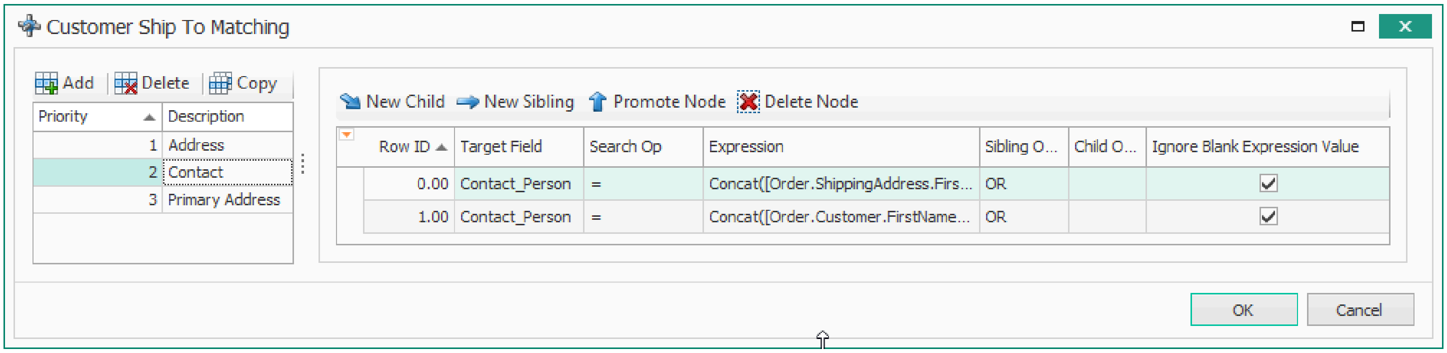


Priority	Description
1	Address
2	Contact
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
1.00	Address_Line_1	=	[Order.ShippingAddress.Address1]	AND		□
3.00	City	=	[Order.ShippingAddress.City]	AND		□
4.00	State	=	[Order.ShippingAddress.Province]	AND		□
5.00	Zip	=	[Order.ShippingAddress.Zip]	AND		□

Priority 2

Contact: Attempt to find a match based on the First and Last Name of the Shopify Customer or Shopify Shipping Address.

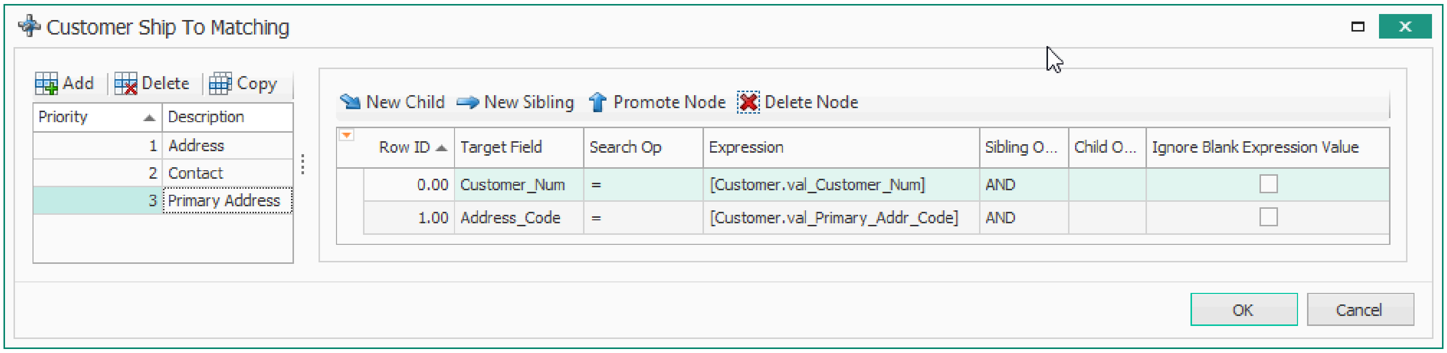


Priority	Description
1	Address
2	Contact
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
0.00	Contact_Person	=	Concat([Order.ShippingAddress.Firs...	OR		☑
1.00	Contact_Person	=	Concat([Order.Customer.FirstName...	OR		☑

Priority 3

Primary Address: Attempt to find a match based on the Primary Address for the Customer. This assumes that the Customer was matched via the *Customer Matching* setting.



Priority	Description
1	Address
2	Contact
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
0.00	Customer_Num	=	[Customer.val_Customer_Num]	AND		□
1.00	Address_Code	=	[Customer.val_Primary_Addr_Code]	AND		□

Customer Bill To Matching – *Define the mappings for looking up the Customer Bill To Address.*

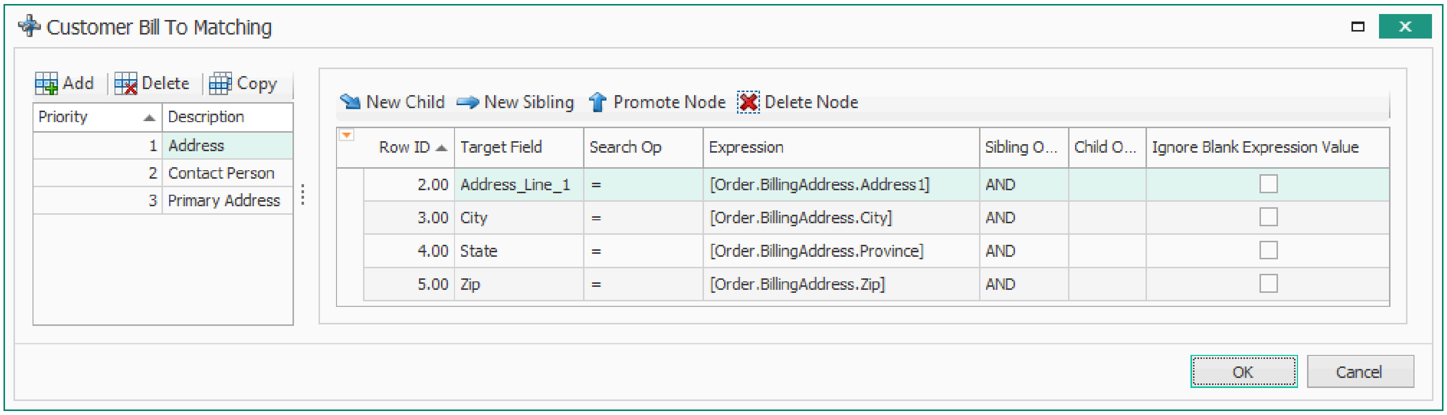
This setting determines how SalesPad will attempt to look up the Bill To Address for an incoming Shopify order. If a SalesPad Customer was matched via the **Customer Matching** setting, the Bill To Address must also belong to that customer. Otherwise, the search will consider address codes across all customers.

For example, if the customer 'AARONFIT0001' has already been matched, only address codes having a Customer_Num of 'AARONFIT0001' will be considered, even if they otherwise match the criteria in the Bill To Address Matching setting. If a customer has not yet been matched when a Bill To Address is found, the order will be imported under the customer for that shipping address.

The default settings attempt to match a SalesPad Billing Address in three different steps.

Priority 1

Address: Attempt to find a match based on Address Line 1, City, State, and Zip. All four of these fields must be matched.

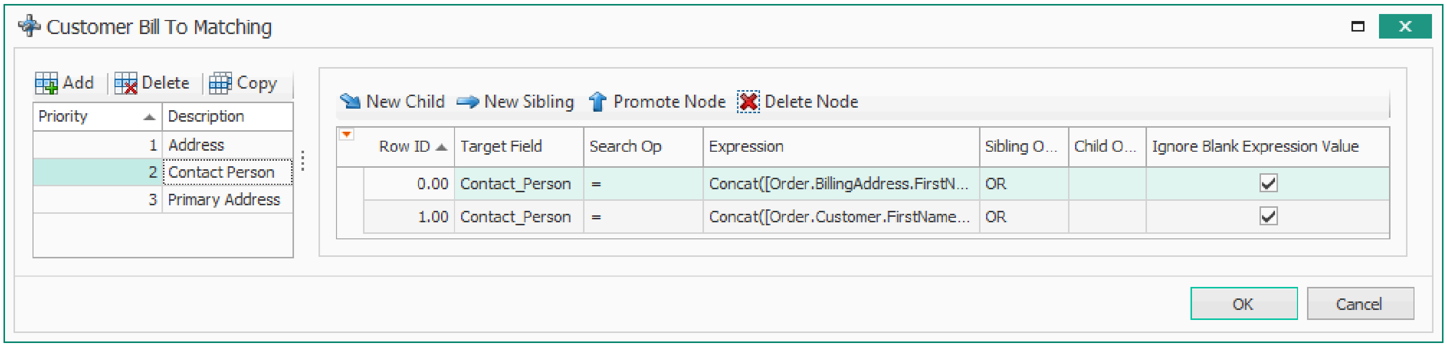


Priority	Description
1	Address
2	Contact Person
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
2.00	Address_Line_1	=	[Order.BillingAddress.Address1]	AND		□
3.00	City	=	[Order.BillingAddress.City]	AND		□
4.00	State	=	[Order.BillingAddress.Province]	AND		□
5.00	Zip	=	[Order.BillingAddress.Zip]	AND		□

Priority 2

Contact Person: Attempt to find a match based on the First and Last Name of the Shopify Customer or Shopify Shipping Address.

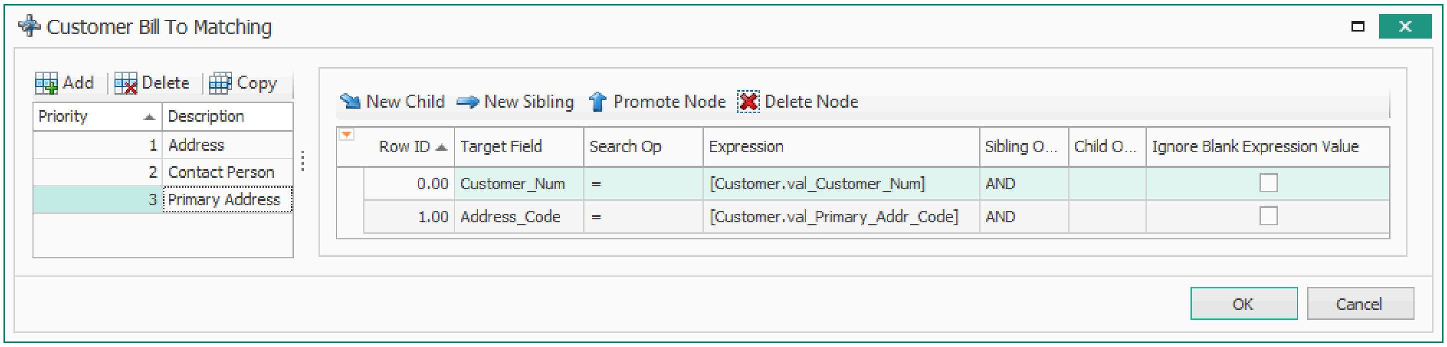


Priority	Description
1	Address
2	Contact Person
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
0.00	Contact_Person	=	Concat([Order.BillingAddress.FirstN...	OR		☑
1.00	Contact_Person	=	Concat([Order.Customer.FirstName...	OR		☑

Priority 3

Primary Address: Attempt to find a match based on the Primary Address for the Customer. This assumes that the Customer was matched via the *Customer Matching* setting.



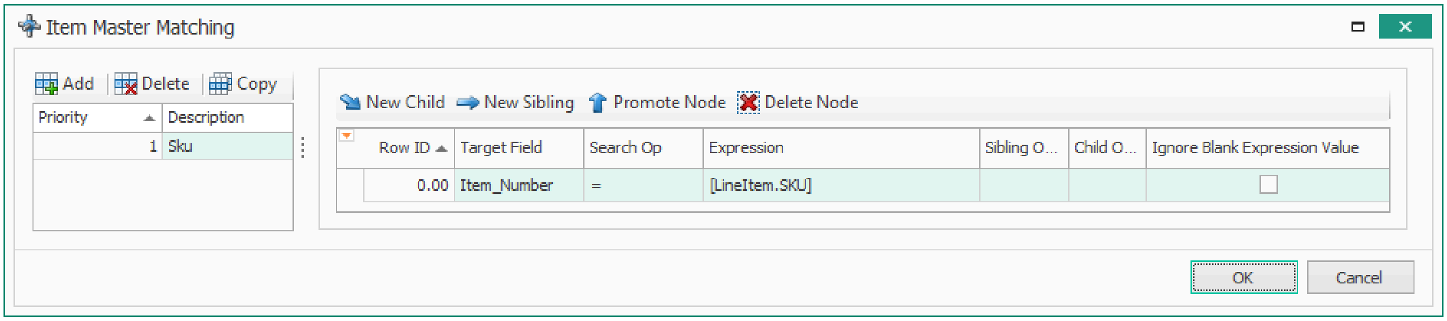
Priority	Description
1	Address
2	Contact Person
3	Primary Address

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
0.00	Customer_Num	=	[Customer.val_Customer_Num]	AND		☐
1.00	Address_Code	=	[Customer.val_Primary_Addr_Code]	AND		☐

Item Master Matching - *Define the mappings for matching a Shopify Line Item to a GP Item Master. This matching setting is only used if the items were not already matched by the Product Export. NOTE: Item links are saved to the spAAIntegratedProductLink table.*

Each Shopify Line Item will use this setting to find the corresponding SalesPad Item.

The default settings attempt to match a Shopify SKU directly to a SalesPad Item Number.



Priority	Description
1	Sku

Row ID	Target Field	Search Op	Expression	Sibling O...	Child O...	Ignore Blank Expression Value
0.00	Item_Number	=	[LineItem.SKU]			□

Assignment Settings

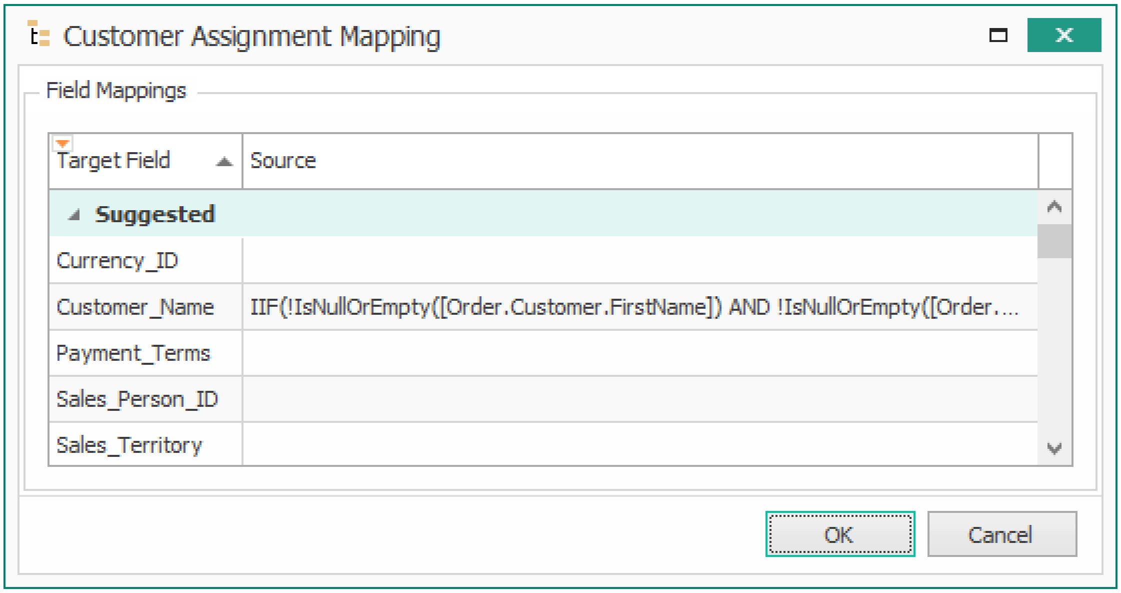
Assignment settings use Expression Editors to define how newly created Customers, Orders and Sales Lines will be populated when created during the order import process.

The target objects will be SalesPad Customers, Contacts, Sales Documents, and Sales Line Items. The source objects will be Shopify Customers, Contacts, Orders, and Line Items.

Each Mapping contains a grid of all the target fields, and an expression that will be used to populate each field. Required fields are grouped under the Suggested category. If a Suggested category field is not populated, an error may occur.

Customer Assignment Mapping – *Define the mappings to be used when creating a new Customer.*

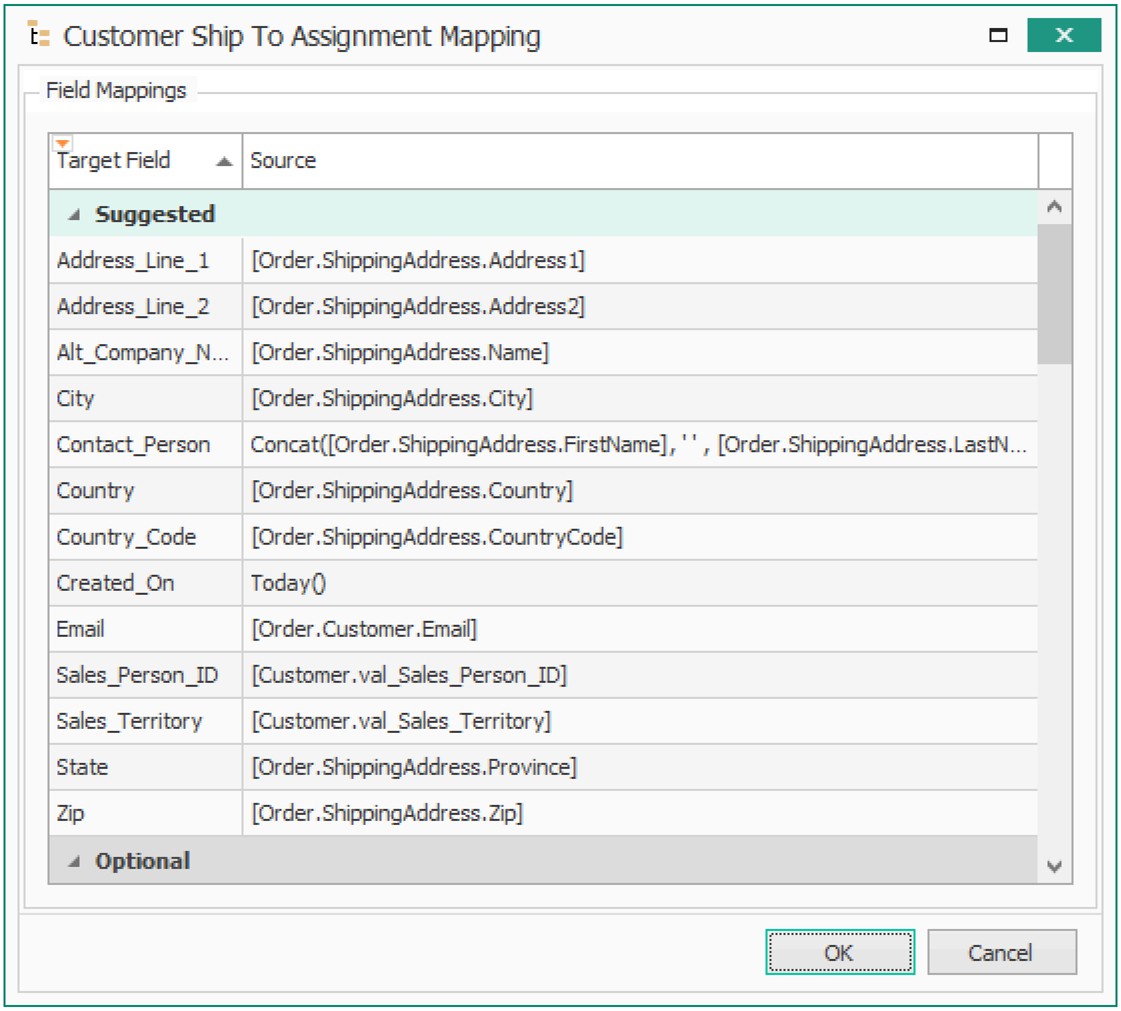
If an existing SalesPad Customer was not found using the matching settings, then a new Customer will be created using the values in the *Customer Assignment Mapping* setting.



Target Field	Source
Suggested	
Currency_ID	
Customer_Name	IIF(!IsNullOrEmpty([Order.Customer.FirstName]) AND !IsNullOrEmpty([Order....
Payment_Terms	
Sales_Person_ID	
Sales_Territory	

Customer Ship To Assignment Mapping - *Define the mappings to be used when creating a new Customer Ship To Address.*

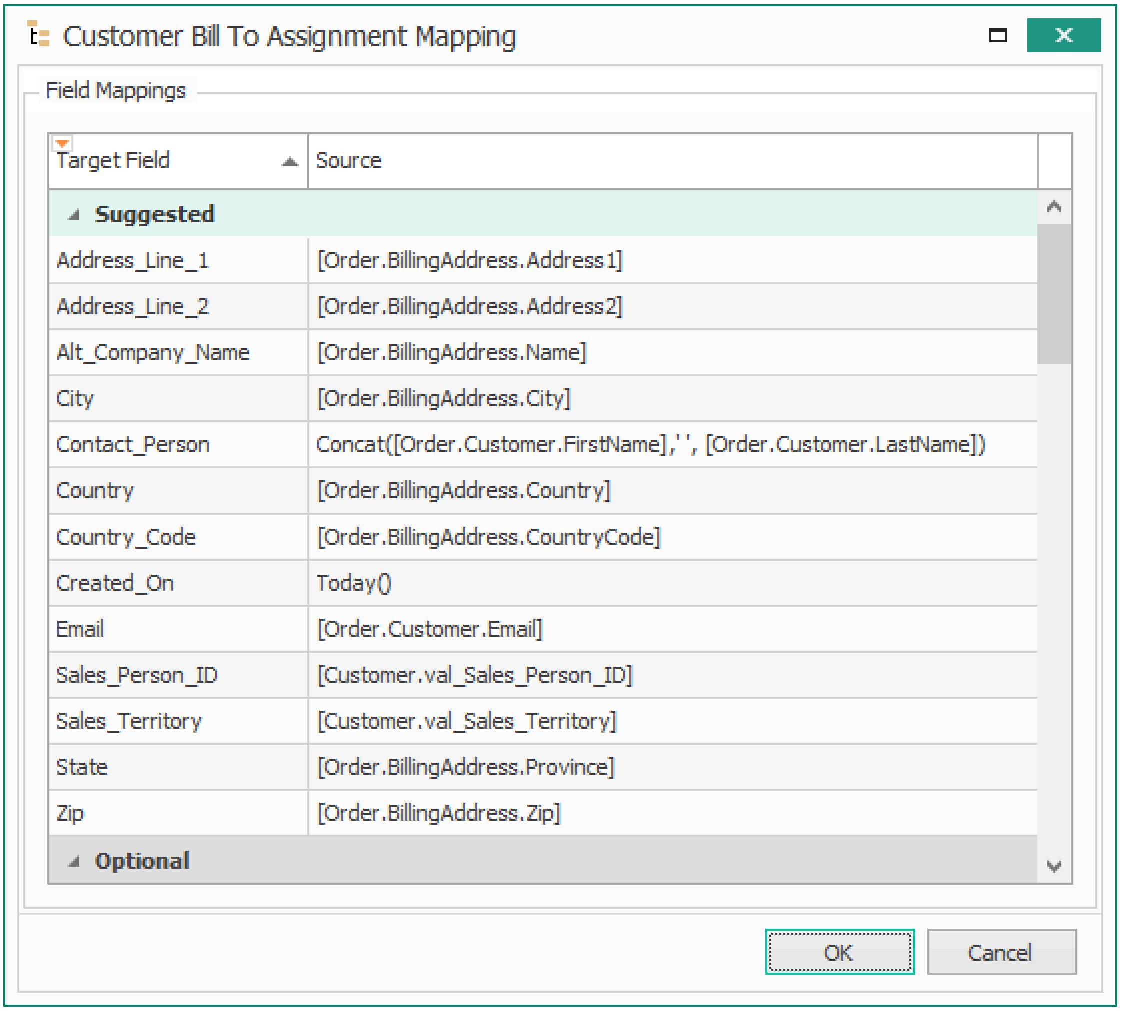
If an existing SalesPad Customer Address was not found using the *Customer Ship To Matching* setting, then a new Customer Address will be created using the values in *Customer Ship To Assignment Mapping* setting.



Target Field	Source
Suggested	
Address_Line_1	[Order.ShippingAddress.Address1]
Address_Line_2	[Order.ShippingAddress.Address2]
Alt_Company_N...	[Order.ShippingAddress.Name]
City	[Order.ShippingAddress.City]
Contact_Person	Concat([Order.ShippingAddress.FirstName], ' ' , [Order.ShippingAddress.LastN...
Country	[Order.ShippingAddress.Country]
Country_Code	[Order.ShippingAddress.CountryCode]
Created_On	Today()
Email	[Order.Customer.Email]
Sales_Person_ID	[Customer.val_Sales_Person_ID]
Sales_Territory	[Customer.val_Sales_Territory]
State	[Order.ShippingAddress.Province]
Zip	[Order.ShippingAddress.Zip]
Optional	

Customer Bill To Assignment Mapping - *Define the mappings to be used when creating a new Customer Bill To Address.*

If an existing SalesPad Customer Address was not found using the *Customer Bill To Matching* setting, then a new Customer Address will be created using the values in the *Customer Bill To Assignment Mapping* setting.



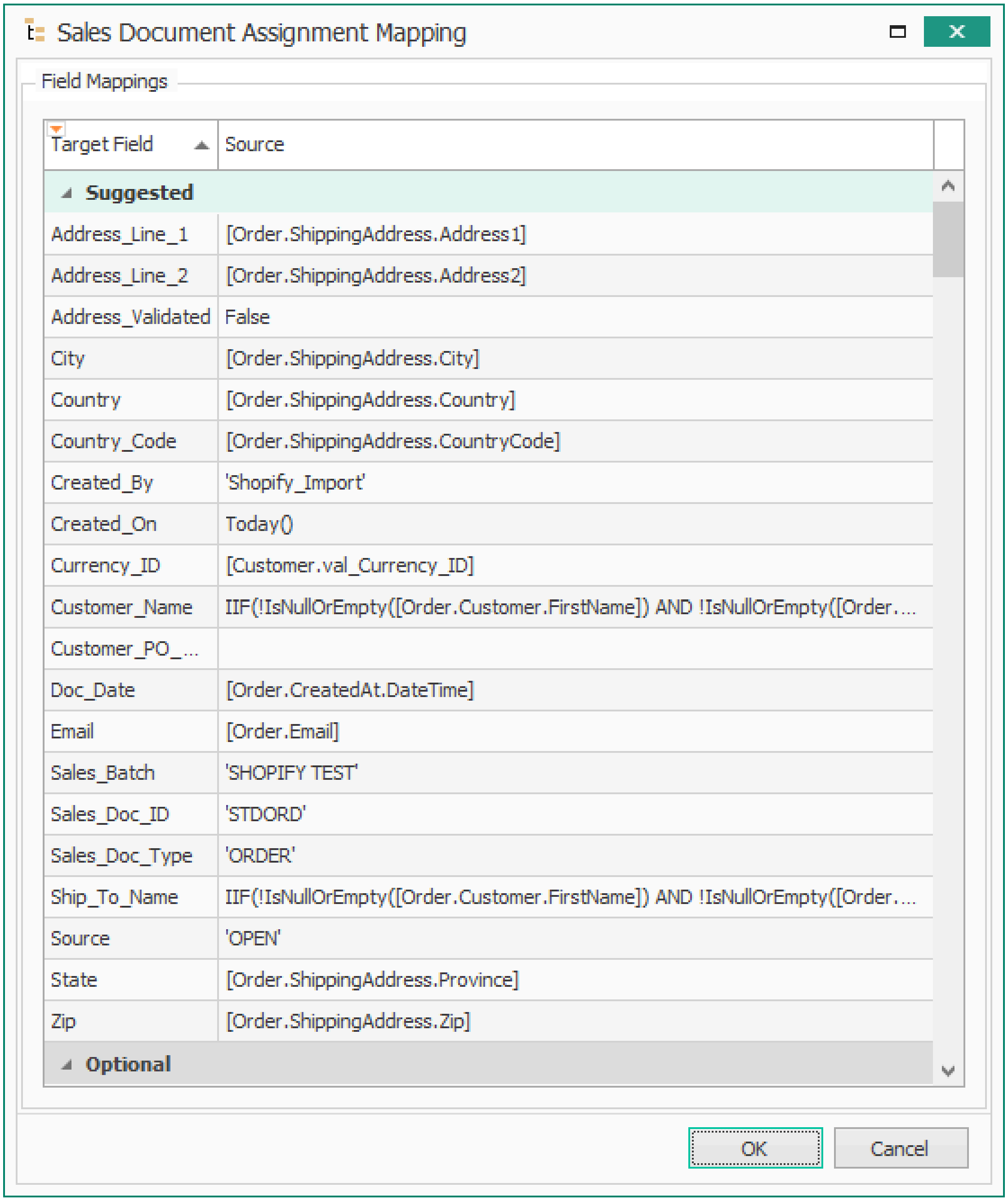
Customer Bill To Assignment Mapping

Field Mappings

Target Field	Source
Suggested	
Address_Line_1	[Order.BillingAddress.Address1]
Address_Line_2	[Order.BillingAddress.Address2]
Alt_Company_Name	[Order.BillingAddress.Name]
City	[Order.BillingAddress.City]
Contact_Person	Concat([Order.Customer.FirstName],' ', [Order.Customer.LastName])
Country	[Order.BillingAddress.Country]
Country_Code	[Order.BillingAddress.CountryCode]
Created_On	Today()
Email	[Order.Customer.Email]
Sales_Person_ID	[Customer.val_Sales_Person_ID]
Sales_Territory	[Customer.val_Sales_Territory]
State	[Order.BillingAddress.Province]
Zip	[Order.BillingAddress.Zip]
Optional	

Sales Document Assignment Mapping - *Define the mappings to be used when creating a new Sales Document.*

When the Shopify Order is imported, this setting will be used to populate the header level fields on the SalesPad Sales Document.

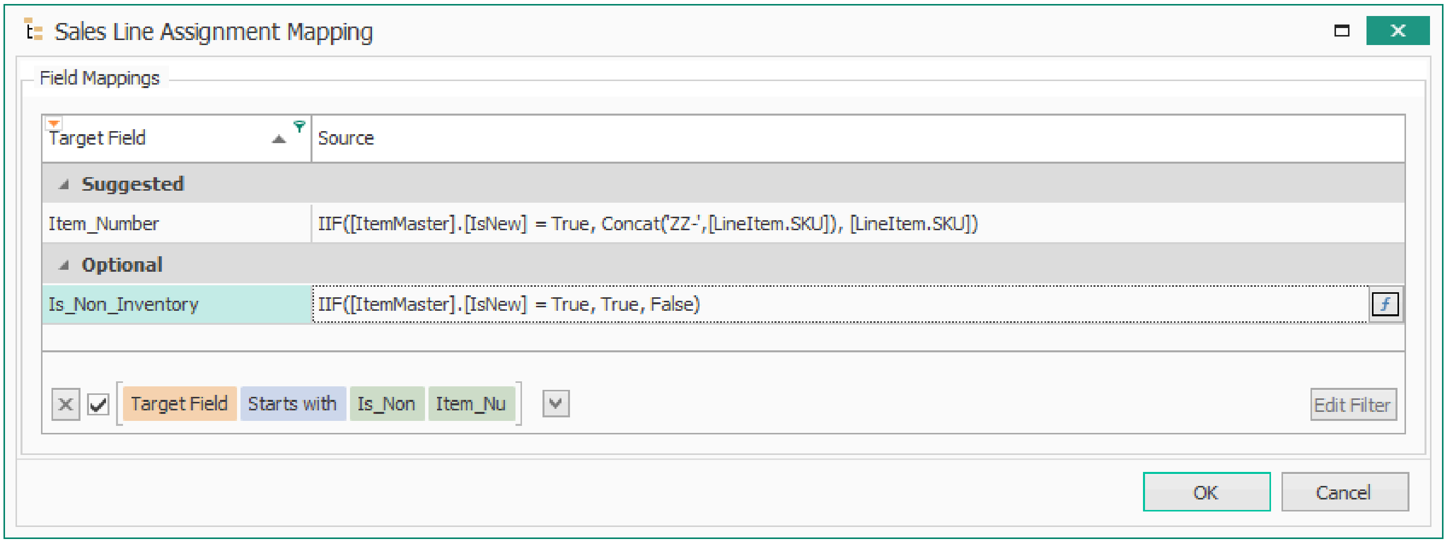


Target Field	Source
Suggested	
Address_Line_1	[Order.ShippingAddress.Address1]
Address_Line_2	[Order.ShippingAddress.Address2]
Address_Validated	False
City	[Order.ShippingAddress.City]
Country	[Order.ShippingAddress.Country]
Country_Code	[Order.ShippingAddress.CountryCode]
Created_By	'Shopify_Import'
Created_On	Today()
Currency_ID	[Customer.val_Currency_ID]
Customer_Name	IIF(!IsNullOrEmpty([Order.Customer.FirstName]) AND !IsNullOrEmpty([Order....
Customer_PO_...	
Doc_Date	[Order.CreatedAt.DateTime]
Email	[Order.Email]
Sales_Batch	'SHOPIFY TEST'
Sales_Doc_ID	'STDORD'
Sales_Doc_Type	'ORDER'
Ship_To_Name	IIF(!IsNullOrEmpty([Order.Customer.FirstName]) AND !IsNullOrEmpty([Order....
Source	'OPEN'
State	[Order.ShippingAddress.Province]
Zip	[Order.ShippingAddress.Zip]
Optional	

Sales Line Assignment Mapping - *Define the mappings to be used when creating a new Sales Line Item.*

When the Shopify Order is imported, each line on that order will be created as a sales line on the SalesPad Sales Document. Each sales line's fields are set based on the *Sales Line Assignment Mapping* setting.

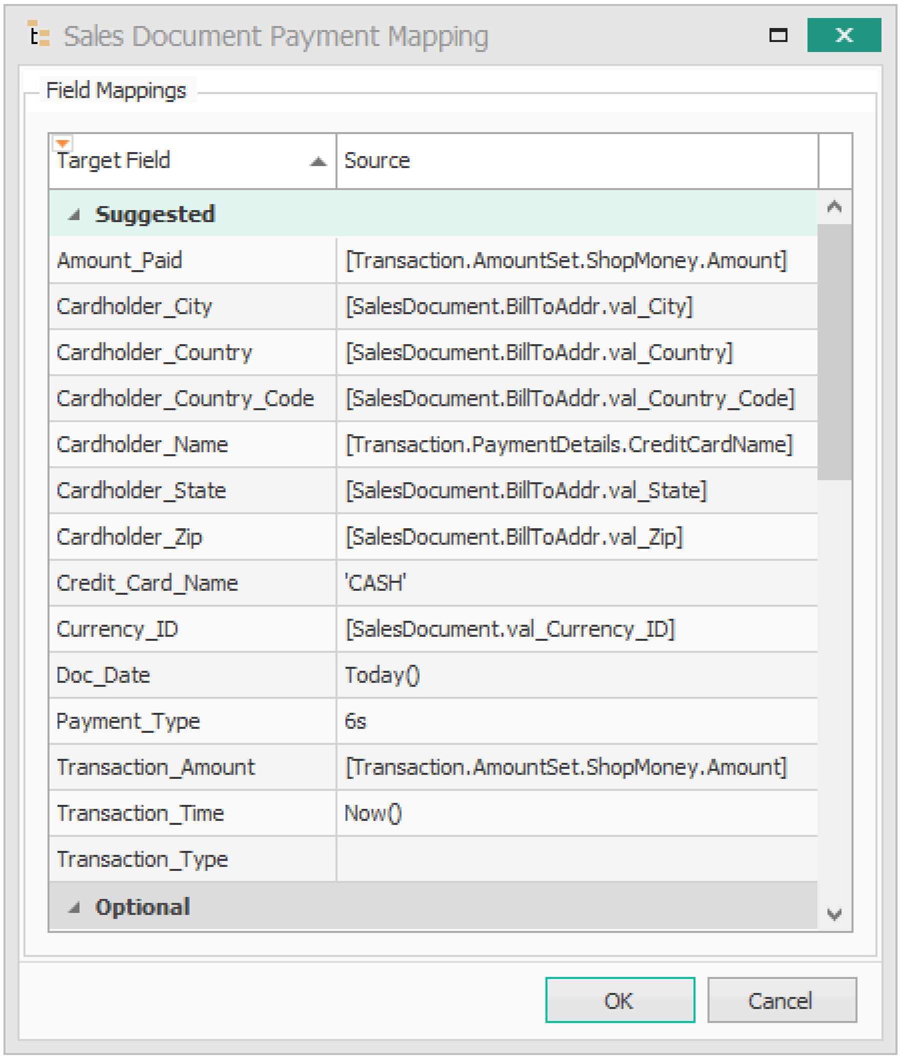
The Item Number is determined by the Item Master that is matched by the *Item Master Matching* setting. If a valid item was not found, then an expression like the one below can be used to import the Shopify Line Item as a Sales Line Item that is a non-inventory item. (Note that *ZZ-* is the default value in the Non-Inventory Prefix setting and should be updated to the local environment's value)


Sales Line Assignment Mapping
Field Mappings
☒ ☑ Target Field Starts with Is_Non Item_Nu
Edit Filter
OK
Cancel

Target Field	Source
Suggested	
Item_Number	IIF([ItemMaster].[IsNew] = True, Concat('ZZ-',[LineItem.SKU]), [LineItem.SKU])
Optional	
Is_Non_Inventory	IIF([ItemMaster].[IsNew] = True, True, False)

A few other line item fields are set by the integration component by default, but they can be overridden by this setting: *Unit_Price*, *Markdown_Amount*, *Tax_Amount*, *Funct_Tax_Amount*.

Sales Document Payment Mapping - *Define the mappings to be used when creating a Sales Document Payment.*



Target Field	Source
Suggested	
Amount_Paid	[Transaction.AmountSet.ShopMoney.Amount]
Cardholder_City	[SalesDocument.BillToAddr.val_City]
Cardholder_Country	[SalesDocument.BillToAddr.val_Country]
Cardholder_Country_Code	[SalesDocument.BillToAddr.val_Country_Code]
Cardholder_Name	[Transaction.PaymentDetails.CreditCardName]
Cardholder_State	[SalesDocument.BillToAddr.val_State]
Cardholder_Zip	[SalesDocument.BillToAddr.val_Zip]
Credit_Card_Name	'CASH'
Currency_ID	[SalesDocument.val_Currency_ID]
Doc_Date	Today()
Payment_Type	6s
Transaction_Amount	[Transaction.AmountSet.ShopMoney.Amount]
Transaction_Time	Now()
Transaction_Type	
Optional	

Scripts

Sales Document Pre Import Script - *A C# Script that runs before a Sales Document is imported.*

Parameters: System.ComponentModel.CancelEventArgs ce, Object sourceDoc, SalesPad.Bus.SalesDocument sd

When importing a Shopify Order, this script runs before any assignments have been executed. This script can be used to cancel the import early by setting ce.Cancel = true. Note that Sales Document fields populated by this script may be overridden by assignment settings.

Customer And Address Matching Script - *A C# Script that runs after the customer and addresses have been matched, and can be used to load a different customer, ship to address, or bill to address.*

Parameters: System.ComponentModel.CancelEventArgs ce, Object sourceDoc, SalesPad.Bus.Customer customer, SalesPad.Bus.CustomerAddr shipToAddr, SalesPad.Bus.CustomerAddr billToAddr, bool multipleCustomersFound

This script will run after the customer and addresses have been matched. It can be used to do more complex assignments or additional validation, and it can cancel the import by setting ce.Cancel = true.

Sales Document Pre Save Script - *A C# Script that runs just before a Sales Document is saved.*

Parameters: System.ComponentModel.CancelEventArgs ce, Object sourceDoc, SalesPad.Bus.SalesDocument sd

After this script runs, the new sales document is saved. This script is the last opportunity to adjust values, or cancel the import by setting ce.Cancel = true.

Sales Document Payment Script - *A C# Script that can be used to override the default payment mappings. (Setting: Sales Document Payment Mapping)*

Parameters: System.ComponentModel.CancelEventArgs ce, Object sourceDoc, SalesPad.Bus.SalesDocument sd

This script can override the default payment mappings, and it can cancel the import by setting ce.Cancel = true.

Item Master Matching Script - *A C# Script that can be used to load SalesPad.Bus.ItemMaster item.*

Parameters: System.ComponentModel.CancelEventArgs ce, Object sourceDoc, Object sourceLine, SalesPad.Bus.SalesDocument sd, SalesPad.Bus.ItemMaster item

This script can override the default matched Item Master, and it can cancel the import by setting ce.Cancel = true.

Processing

For each sales order in the website, the Order Import component attempts to match a SalesPad Customer to the Shopify Customer based on the *Customer Ship To Matching*, *Customer Bill To Matching*, and *Customer Matching settings*. If no match is found, a new customer is created in SalesPad based on the *Customer Assignment Mapping setting*, and address codes for the new customer are determined by the *Customer Ship To Assignment Mapping* and *Customer Bill To Assignment Mapping* settings.

Once the customer and addresses are matched or created, the sales order is created based on the *Sales Document Assignment Mapping*, *Item Master Matching*, and *Sales Line Assignment Mapping* settings. Payment information can be imported for sales orders via the *Sales Document Payment Mapping* setting. A tag is added to each order in Shopify to indicate that importing was completed.

Endpoints

Get Orders

At the beginning of the Order Import, the integration bulk loads orders from Shopify.

GraphQL query:

```
None
query getOrders($first: Int, $after: String, $query: String) {
  orders(first: $first, after: $after, query: $query) {
    nodes {
      legacyResourceId
      id
      billingAddress {
        id
        address1
        address2
        city
        company
        country
        countryCodeV2
        firstName
        lastName
        latitude
        longitude
        name
        phone
        province
        provinceCode
        zip
      }
      clientIp
      customerAcceptsMarketing
```

```
cancelReason
cancelledAt
closedAt
confirmed
createdAt
currencyCode
customer {
  legacyResourceId
  id
  createdAt
  defaultAddress {
    id
    address1
    address2
    city
    company
    country
    countryCodeV2
    firstName
    lastName
    latitude
    longitude
    name
    phone
    province
    provinceCode
    zip
  }
  email
  firstName
  multipassIdentifier
  lastName
  note
  phone
  state
  tags
  taxExempt
```

```
      taxExemptions
      updatedAt
      verifiedEmail
      smsMarketingConsent {
        marketingState
        marketingOptInLevel
        consentUpdatedAt
        consentCollectedFrom
      }
      emailMarketingConsent {
        marketingState
        marketingOptInLevel
        consentUpdatedAt
      }
    }
    customerLocale
    discountApplications(first: 250) {
      nodes {

        allocationMethod
        targetSelection
        targetType
        value {
          ... on MoneyV2 {
            amount
          }
          ... on PricingPercentageValue {
            percentage
          }
        }
        ... on DiscountCodeApplication {
          code
        }
        ... on ManualDiscountApplication {
          title
          description
        }
        ... on AutomaticDiscountApplication {
          title
        }
        ... on ScriptDiscountApplication {
          title
        }
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
    email
    displayFinancialStatus
    displayFulfillmentStatus
    phone
    tags
    lineItems(first: 250) {
      nodes {
        id
        unfulfilledQuantity
        originalUnitPriceSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        product {
          legacyResourceId
        }
        currentQuantity
        quantity
        requiresShipping
```

```
sku
title
variant {
  legacyResourceId
}
variantTitle
name
vendor
isGiftCard
taxable
taxLines {
  priceSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  rate
  title
}
totalDiscountSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
discountAllocations {
  allocatedAmountSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  discountApplication {
    index
  }
}
duties {
  id
  harmonizedSystemCode
  countryCodeOfOrigin
  price {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  taxLines {
    priceSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
```

```
        rate
        title
      }
    }
  }
  pageInfo {
    endCursor
    hasNextPage
  }
}
retailLocation {
  id
  legacyResourceId
  name
}
name
note
customAttributes {
  key
  value
}
statusPageUrl
paymentGatewayNames
processedAt
shippingAddress {
  id
  address1
  address2
  city
  company
  country
  countryCodeV2
  firstName
  lastName
  latitude
  longitude
  name
```

```
      phone
      province
      provinceCode
      zip
    }
    shippingLine {
      id
      carrierIdentifier
      code
      phone
      originalPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      discountedPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      discountAllocations {
        allocatedAmountSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        discountApplication {
          index
        }
      }
      source
      title
      taxLines {
        priceSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        rate
        title
      }
    }
    sourceIdentifier
    sourceName
    subtotalPriceSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
```

```
taxLines {
  priceSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  rate
  title
}
taxesIncluded
test
totalDiscountsSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalTipReceivedSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalTaxSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalWeight
updatedAt
transactions {
  id
  amountSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  authorizationCode
  authorizationExpiresAt
  createdAt
  gateway
```

```
paymentDetails {

  ... on CardPaymentDetails {
    avsResultCode
    bin
    cvvResultCode
    number
    company
    name
    wallet
    expirationMonth
    expirationYear
  }
}
kind
receiptJson
errorCode
status
test
parentTransaction {
  id
}
processedAt
maximumRefundableV2 {
  currencyCode
  amount
}
paymentId
totalUnsettledSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
```

```
        }
      }
      currentTotalDutiesSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      originalTotalDutiesSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      presentmentCurrencyCode
      currentShippingPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      totalShippingPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      totalOutstandingSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      estimatedTaxes
      currentSubtotalPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      currentTotalDiscountsSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
```

```
    currentTotalPriceSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    currentTotalTaxSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    currentTotalAdditionalFeesSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    originalTotalAdditionalFeesSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
```

```
          amount
        }
      }
      poNumber
      taxExempt
    }
    pageInfo {
      endCursor
      hasNextPage
    }
  }
}
```

Sample response:

```
None
{
  "orders": {
    "nodes": [
      {
        "legacyResourceId": "6238605934895",
        "id": "gid://shopify/Order/6238605934895",
        "billingAddress": {
          "id": "gid://shopify/MailingAddress/19882009493807?model_name=Address",
          "address1": "3351 Claystone Street Southeast",
          "address2": null,
          "city": "Grand Rapids",
          "company": "Aaron Fitz Electrical",
          "country": "United States",
          "countryCodeV2": "US",
          "firstName": "Aaron",
          "lastName": "Fitz",
          "latitude": null,
          "longitude": null,
          "name": "Aaron Fitz",
      "phone": null,
      "province": "Michigan",
      "provinceCode": "MI",
      "zip": "49546"
    },
    "clientIp": "184.175.130.74",
    "customerAcceptsMarketing": false,
    "cancelReason": null,
    "cancelledAt": null,
    "closedAt": null,
    "confirmed": true,
    "createdAt": "2025-03-20T12:51:34Z",
    "currencyCode": "USD",
    "customer": {
      "legacyResourceId": "8943179104559",
      "id": "gid://shopify/Customer/8943179104559",
      "createdAt": "2025-03-20T12:50:20Z",
      "defaultAddress": {
        "id": "gid://shopify/MailingAddress/10818200240431?model_name=CustomerAddress",
        "address1": "3351 Claystone Street Southeast",
        "address2": "",
        "city": "Grand Rapids",
        "company": "Aaron Fitz Electrical",
        "country": "United States",
        "countryCodeV2": "US",
        "firstName": "Aaron",
        "lastName": "Fitz",
        "latitude": 42.9237382,
        "longitude": -85.5853301,
        "name": "Aaron Fitz",
        "phone": null,
        "province": "Michigan",
        "provinceCode": "MI",
        "zip": "49546"
      },
      "email": null,
      "firstName": "Aaron",
      "multipassIdentifier": null,
      "lastName": "Fitz",
      "note": "",
      "phone": null,
      "state": "DISABLED",
      "tags": [],
      "taxExempt": false,
      "taxExemptions": [],
      "updatedAt": "2025-03-20T12:51:34Z",
      "verifiedEmail": true,
      "smsMarketingConsent": null,
      "emailMarketingConsent": null
    },
    "customerLocale": "en",
    "discountApplications": {
      "nodes": [],
      "pageInfo": {
        "endCursor": null,
        "hasNextPage": false
      }
    },
    "email": null,
    "displayFinancialStatus": "PAID",
    "displayFulfillmentStatus": "UNFULFILLED",
    "phone": null,
    "tags": [],
    "lineItems": {
      "nodes": [
        {
          "id": "gid://shopify/LineItem/15532308922671",
          "unfulfilledQuantity": 1,
          "originalUnitPriceSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "59.99"
            },
            "presentmentMoney": {
```

```
            "currencyCode": "USD",
            "amount": "59.99"
          }
        },
        "product": {
          "legacyResourceId": "8061405036847"
        },
        "currentQuantity": 1,
        "quantity": 1,
        "requiresShipping": true,
        "sku": "100XLG",
        "title": "100XLG",
        "variant": {
          "legacyResourceId": "44263493533999"
        },
        "variantTitle": null,
        "name": "100XLG",
        "vendor": "Developer Store",
        "isGiftCard": false,
        "taxable": true,
        "taxLines": [
          {
            "priceSet": {
              "shopMoney": {
                "currencyCode": "USD",
                "amount": "3.6"
              },
              "presentmentMoney": {
                "currencyCode": "USD",
                "amount": "3.6"
              }
            },
            "rate": 0.06,
            "title": "Michigan State Tax"
          }
        ],
        "totalDiscountSet": {
                "shopMoney": {
                  "currencyCode": "USD",
                  "amount": "0.0"
                },
                "presentmentMoney": {
                  "currencyCode": "USD",
                  "amount": "0.0"
                }
              },
              "discountAllocations": [],
              "duties": []
            }
          ],
          "pageInfo": {
            "endCursor":
"eyJsYXN0X2lkIjoxNTUzMjMwODkyMjY3MSwibGFzdF92YWx1ZSI6MTU1MzIzMDg5MjI2NzF9",
            "hasNextPage": false
          }
        },
        "retailLocation": null,
        "name": "#1162",
        "note": null,
        "customAttributes": [],
        "statusPageUrl":
"https://developerstore.myshopify.com/70387892527/orders/fddbceb2eba827c0f285cc1cf89f8d7d/aut
henticate?key=1831276ed61b1d65fbd8030175898273",
        "paymentGatewayNames": [
          "manual"
        ],
        "processedAt": "2025-03-20T12:51:33Z",
        "shippingAddress": {
          "id": "gid://shopify/MailingAddress/19882009461039?model_name=Address",
          "address1": "3351 Claystone Street Southeast",
          "address2": null,
          "city": "Grand Rapids",
          "company": "Aaron Fitz Electrical",
          "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Aaron",
      "lastName": "Fitz",
      "latitude": 42.9237382,
      "longitude": -85.5853301,
      "name": "Aaron Fitz",
      "phone": null,
      "province": "Michigan",
      "provinceCode": "MI",
      "zip": "49546"
    },
    "shippingLine": null,
    "sourceIdentifier": null,
    "sourceName": "shopify_draft_order",
    "subtotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      }
    },
    "taxLines": [
      {
        "priceSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          }
        },
        "rate": 0.06,
        "title": "Michigan State Tax"
      }
    ],
    "taxesIncluded": false,
    "test": false,
    "totalDiscountsSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalTipReceivedSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      }
    },
    "totalTaxSet": {
      "shopMoney": {
          "currencyCode": "USD",
          "amount": "3.6"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "3.6"
        }
      },
      "totalWeight": "2268",
      "updatedAt": "2025-03-20T12:51:35Z",
      "transactions": [
        {
          "id": "gid://shopify/OrderTransaction/7589589811503",
          "amountSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "63.59"
            },
            "presentmentMoney": {
              "currencyCode": "USD",
              "amount": "63.59"
            }
          },
          "authorizationCode": null,
          "authorizationExpiresAt": null,
          "createdAt": "2025-03-20T12:51:33Z",
          "gateway": "manual",
          "paymentDetails": null,
          "kind": "SALE",
          "receiptJson": "{}",
          "errorCode": null,
          "status": "SUCCESS",
          "test": false,
          "parentTransaction": null,
          "processedAt": "2025-03-20T12:51:33Z",
          "maximumRefundableV2": null,
          "paymentId": "rKUNK2rWBuhIyxye2kDpEusTW",
          "totalUnsettledSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            },
            "presentmentMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            }
          }
        }
      ],
      "currentTotalDutiesSet": null,
      "originalTotalDutiesSet": null,
      "presentmentCurrencyCode": "USD",
      "currentShippingPriceSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      },
      "totalShippingPriceSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      },
      "totalOutstandingSet": {
        "shopMoney": {
            "currencyCode": "USD",
            "amount": "0.0"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "0.0"
          }
        },
        "estimatedTaxes": false,
        "currentSubtotalPriceSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "59.99"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "59.99"
          }
        },
        "currentTotalDiscountsSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "0.0"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "0.0"
          }
        },
        "currentTotalPriceSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          }
        },
        "currentTotalTaxSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          }
        },
        "currentTotalAdditionalFeesSet": null,
        "originalTotalAdditionalFeesSet": null,
        "poNumber": null,
        "taxExempt": false
      }
    ],
    "pageInfo": {
      "endCursor":
"eyJsYXN0X2lkIjo2MjM4NjA1OTM0ODk1LCJsYXN0X3ZhbHVlIjoxNzQyNDc1MDkzMDAwfQ==",
      "hasNextPage": false
    }
  }
}
```

Load Additional Pages

If hasNextPage is true for the orders, line items, or discount applications, the integration will load the additional pages. The following sample loads a second page of line items.

GraphQL query:

```
None
query getorder($id: ID!, $first: Int, $after: String) {
  order(id: $id) {
    lineItems(first: $first, after: $after) {
      nodes {
        id
        unfulfilledQuantity
        originalUnitPriceSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        product {
          legacyResourceId
        }
        currentQuantity
        quantity
        requiresShipping
        sku
        title
        variant {
          legacyResourceId
        }
        variantTitle
        name
        vendor
        isGiftCard
        taxable
        taxLines {
          priceSet {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          rate
          title
        }
        totalDiscountSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        discountAllocations {
          allocatedAmountSet {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          discountApplication {
            index
          }
        }
        duties {
          id
          harmonizedSystemCode
          countryCodeOfOrigin
```

```
          price {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          taxLines {
            priceSet {
              shopMoney {
                currencyCode
                amount
              }
              presentmentMoney {
                currencyCode
                amount
              }
            }
            rate
            title
          }
        }
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
  }
}
```

Sample response:

None

```
{
  "order": {
    "lineItems": {
      "nodes": [
        {
          "id": "gid://shopify/LineItem/15532353716527",
          "unfulfilledQuantity": 1,
          "originalUnitPriceSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "169.0"
            },
            "presentmentMoney": {
              "currencyCode": "USD",
              "amount": "169.0"
            }
          },
          "product": {
            "legacyResourceId": "8061405004079"
          },
          "currentQuantity": 1,
          "quantity": 1,
          "requiresShipping": true,
          "sku": "128 SDRAM",
          "title": "128 SDRAM",
          "variant": {
            "legacyResourceId": "44263493501231"
          },
          "variantTitle": null,
          "name": "128 SDRAM",
          "vendor": "Developer Store",
          "isGiftCard": false,
          "taxable": true,
          "taxLines": [
            {
              "priceSet": {
                "shopMoney": {
                  "currencyCode": "USD",
                  "amount": "10.14"
                },
                "presentmentMoney": {
                  "currencyCode": "USD",
                  "amount": "10.14"
                }
              },
              "rate": 0.06,
              "title": "Michigan State Tax"
            }
          ],
          "totalDiscountSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            },
            "presentmentMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            }
          },
          "discountAllocations": [],
          "duties": []
        }
      ],
      "pageInfo": {
        "endCursor":
"eyJsYXN0X2lkIjoxNTUzMjM1MzcxNjUyNywibGFzdF92YWx1ZSI6MTU1MzIzNTM3MTY1Mjd9",
        "hasNextPage": false
      }
    }
  }
}
```

Update Processed Order Tag

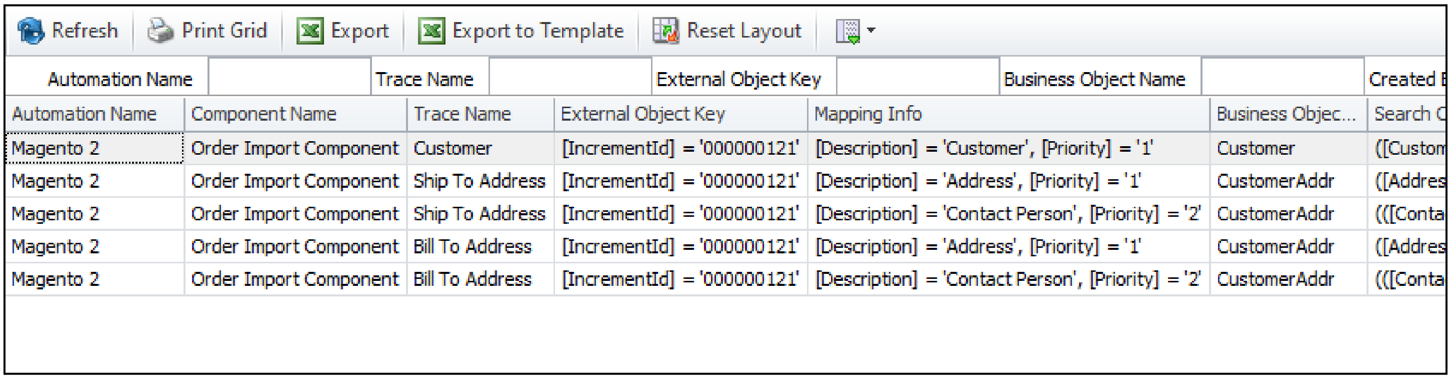
After an order has been successfully imported into SalesPad, the Order Import will tag the Shopify order with the Processed Order Tag setting's value.

GraphQL mutation:

```
None
mutation orderUpdate($input: OrderInput!) {
  orderUpdate(input: $input) {
    userErrors {
      field
      message
    }
  }
}
```

Customer and Customer Address Tracing

Depending on the complexity of the matching configuration, it may be difficult to tell how each customer and customer address is getting matched during order import. Order Import Tracing functionality will log every possible matching result to the spAAIntegrationTrace table in the database, regardless if matching was successful or not. Tracing may be enabled by enabling the *Enable Order Import Trace* setting under Order Import. Note that enabling this setting may cause database bloat, so it is intended only for troubleshooting purposes.



Automation Name	Component Name	Trace Name	External Object Key	Mapping Info	Business Objec...	Search C
Magento 2	Order Import Component	Customer	[IncrementId] = '000000121'	[Description] = 'Customer', [Priority] = '1'	Customer	([Custom
Magento 2	Order Import Component	Ship To Address	[IncrementId] = '000000121'	[Description] = 'Address', [Priority] = '1'	CustomerAddr	([Addres
Magento 2	Order Import Component	Ship To Address	[IncrementId] = '000000121'	[Description] = 'Contact Person', [Priority] = '2'	CustomerAddr	(([Conta
Magento 2	Order Import Component	Bill To Address	[IncrementId] = '000000121'	[Description] = 'Address', [Priority] = '1'	CustomerAddr	([Addres
Magento 2	Order Import Component	Bill To Address	[IncrementId] = '000000121'	[Description] = 'Contact Person', [Priority] = '2'	CustomerAddr	(([Conta

The following quick report can be used to query the spAAIntegrationTrace table in SalesPad:

```
<report name="AA Integration Trace" AutoLinks="true" GroupFooterShowMode="Expanded"
bestFitAll="true" AutoFit="false">
 <description />
 <query addWhere="true">SELECT *
FROM (
        SELECT Automation_Name = ai.Instance_Name
                ,Automation_Description = ai.Instance_Description
                ,Component_Name = aic.Component_Name
                ,Trace_Name = ait.Trace_Name
                ,Group_ID = ait.Group_ID
                ,External_Object_Key = ait.External_Object_Key
                ,External_Object = ait.External_Object
                ,Mapping_Info = ait.Mapping_Info
                ,Business_Object_Name = ait.Business_Object_Name
                ,Search_Clause = ait.Search_Clause
                ,Results = ait.Results
                ,Created_On = ait.Created_On
                ,Created_By = ait.Created_By
        FROM spAAIntegrationTrace AS ait WITH (NOLOCK)
        LEFT JOIN spAAInstance AS ai WITH (NOLOCK) ON ai.AA_Instance_ID =
ait.AA_Instance_ID
        LEFT JOIN spAAInstanceComponent AS aic WITH (NOLOCK) ON aic.AA_Instance_ID =
ait.AA_Instance_ID
                AND aic.Component_ID = ait.AA_Component_ID
        ) AS a</query>
 <search name="Automation Name" column="Automation_Name" searchOp="LIKE" Type="Text"
/>
 <search name="Trace Name" column="Trace_Name" searchOp="LIKE" Type="Text" />
 <search name="External Object Key" column="External_Object_Key" searchOp="LIKE"
Type="Text" />
 <search name="Business Object Name" column="Business_Object_Name" searchOp="LIKE"
Type="Text" />
 <search name="Created On" column="Created_On" searchOp="=" Type="DateTime" />
 <OnRunScript />
</report>
```

The embedded SQL query can be used directly from SSMS.

Order Export

Overview

The Order Export component creates orders in Shopify based on SalesPad sales orders. It exports open SalesPad orders which are in specific workflow queues, creates each corresponding website order based on configurable mapping settings, and then it forwards the SalesPad orders in workflow once processing is complete.

This component is available as of SalesPad 5.6.4.

General Settings

Export Orders In Workflow Queue - *Specify the workflow queues that will be processed to export documents to the website.*

Exported Order Tag - *The Order Export will write this tag to exported Shopify orders. The Order Import will then filter out orders with this tag to help ensure that exported orders aren't imported. Defaults to 'IMPORTED_FROM_SALESPAD'.*

Inventory Behavior - *The strategy for handling updates to Shopify inventory when exporting an order to Shopify. Defaults to 'Decrement Obeying Policy'.*

Line Num Custom Attribute Name - *The name of the line item custom attribute which contains the linked sales line item's Line Num. This attribute is required for the Order Export to be able to write External IDs to each sales line item after creating the order. Beginning the attribute's name with an underscore (_) signals to Shopify to hide the attribute from places such as the order status page. Defaults to '_SalesPad Line Num'.*

Order Export Failure Queue - *In the event of an unsuccessful export, the document will be placed into this queue.*

Send Receipt - *Whether to send an order confirmation to the customer. Defaults to 'False'.*

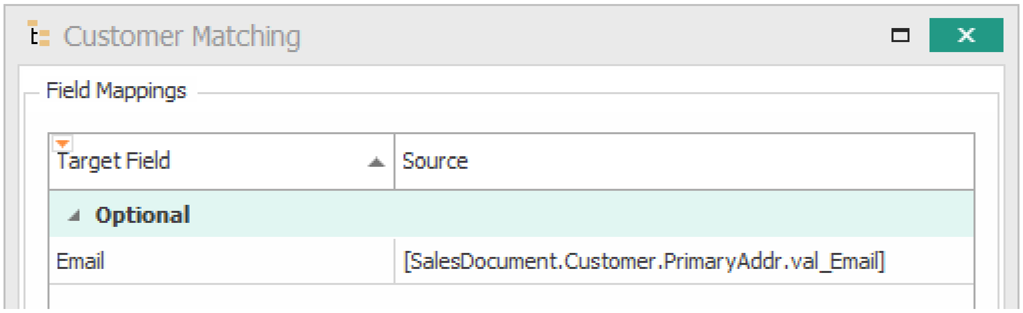
Matching Settings

Customer Matching - *Define the mappings for finding the website customer based on the SalesPad customer. This matching setting is only used if the customers were not already matched by the Order Import. NOTE: Customer links are saved to the spAAIntegratedCustomerLink table.*

This matching setting is simpler than the Order Import's Customer Matching setting. The setting allows users to select which SalesPad field to match to a Shopify customer's primary email address. The email addresses must exactly match.

By default, Customer Matching will try to match the sales document's customer's primary address's email address to a Shopify customer's primary email address. Users can change which SalesPad field to use, but cannot change which Shopify field to use or add more fields to match on.

If the Order Import had previously matched a SalesPad customer to a Shopify customer, the Order Export will use that match instead of using this setting.


Customer Matching

Field Mappings

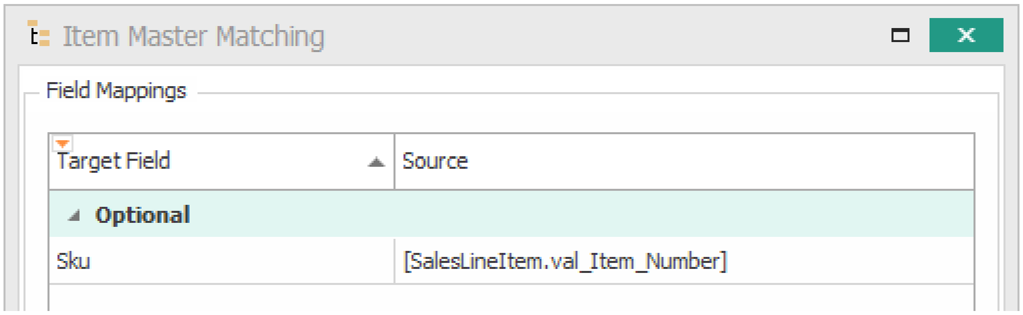
Target Field	Source
Optional	
Email	[SalesDocument.Customer.PrimaryAddr.val_Email]

Item Master Matching – *Define the mappings for finding the website product variant based on the SalesPad item. This matching setting is only used if the items were not already matched by the Product Export. NOTE: Item links are saved to the spAAIntegratedProductLink table.*

This matching setting is simpler than the Order Import's Item Master Matching setting. The setting allows users to select which SalesPad field to match to a Shopify product variant's SKU. The fields must exactly match.

By default, Item Master Matching will try to match the sales line item's item number to a Shopify product variant's SKU. Users can change which SalesPad field to use, but cannot change which Shopify field to use or add more fields to match on.

If the Product Export had previously matched a SalesPad item master to a Shopify product variant, the Order Export will use that match instead of using this setting.



Item Master Matching

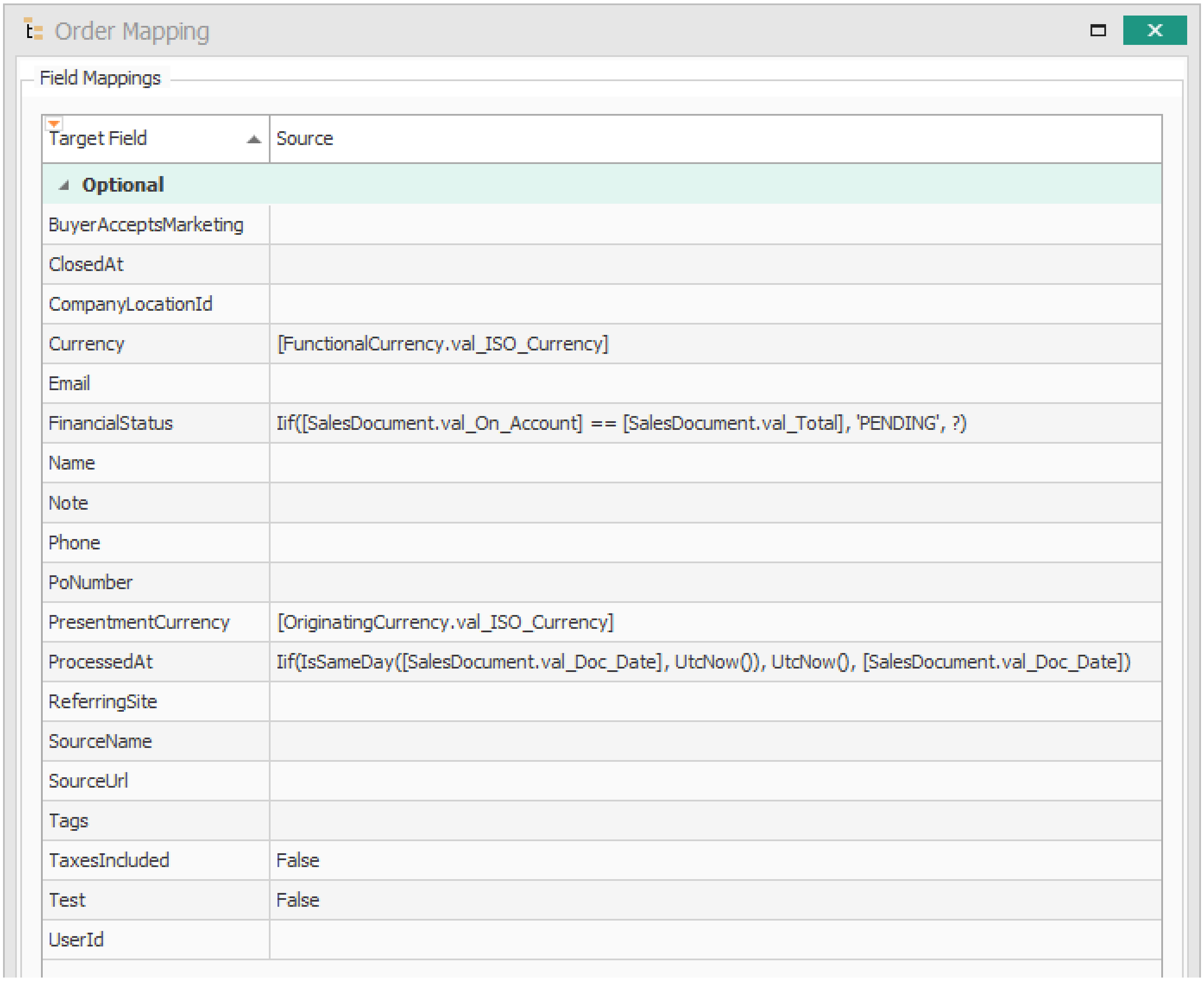
Field Mappings

Target Field	Source
Optional	
Sku	[SalesLineItem.val_Item_Number]

Assignment Settings

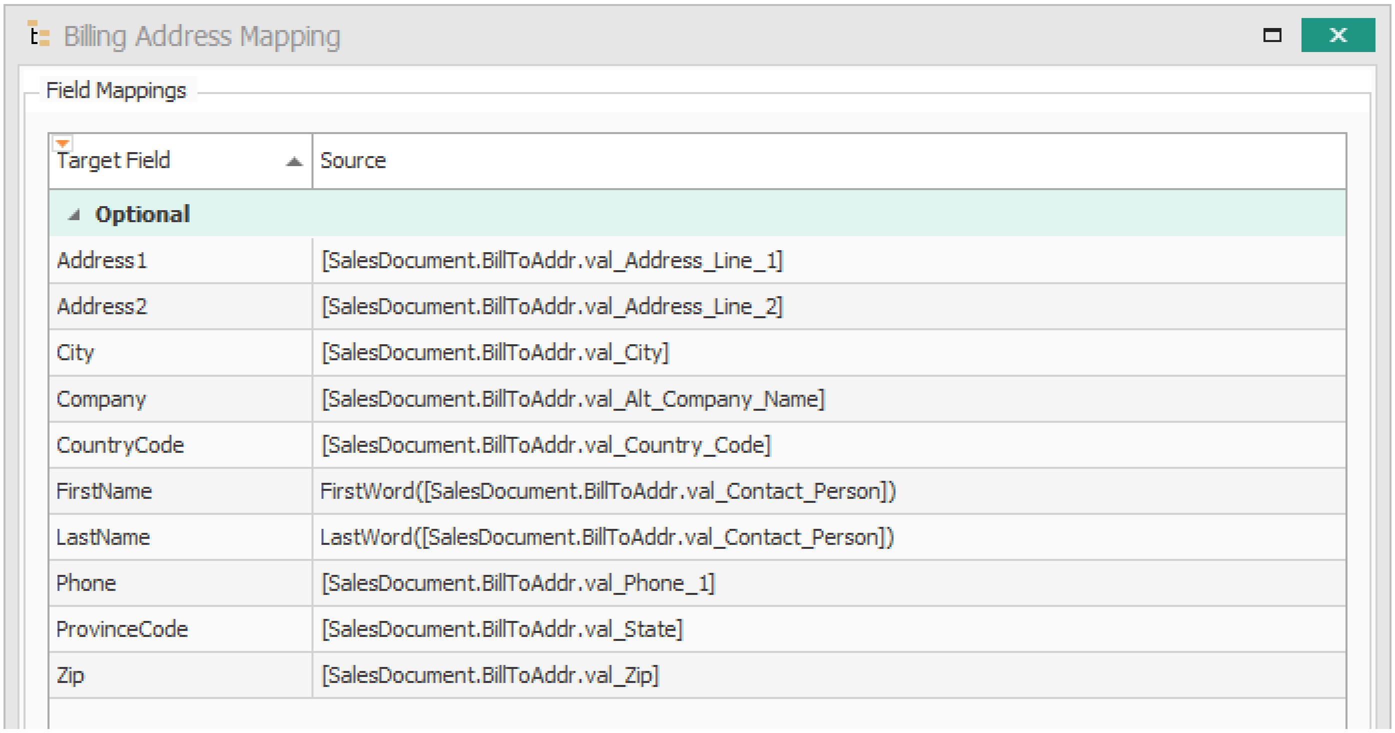
Order Mappings – *Define the mappings to be used when creating a new order on the website. NOTE: The Currency and Presentment Currency fields will roll down to various monetary fields, and are meant to always correspond to the GP functional currency and the sales document's currency respectively.*

This setting controls some of the basic header information for the order. Various other assignment settings control building out the rest of the order.



Target Field	Source
Optional	
BuyerAcceptsMarketing	
ClosedAt	
CompanyLocationId	
Currency	[FunctionalCurrency.val_ISO_Currency]
Email	
FinancialStatus	Iif([SalesDocument.val_On_Account] == [SalesDocument.val_Total], 'PENDING', ?)
Name	
Note	
Phone	
PoNumber	
PresentmentCurrency	[OriginatingCurrency.val_ISO_Currency]
ProcessedAt	Iif(IsSameDay([SalesDocument.val_Doc_Date], UtcNow()), UtcNow(), [SalesDocument.val_Doc_Date])
ReferringSite	
SourceName	
SourceUrl	
Tags	
TaxesIncluded	False
Test	False
UserId	

Billing Address Mapping – *Define the mappings to be used when creating a billing address for a new order on the website.*



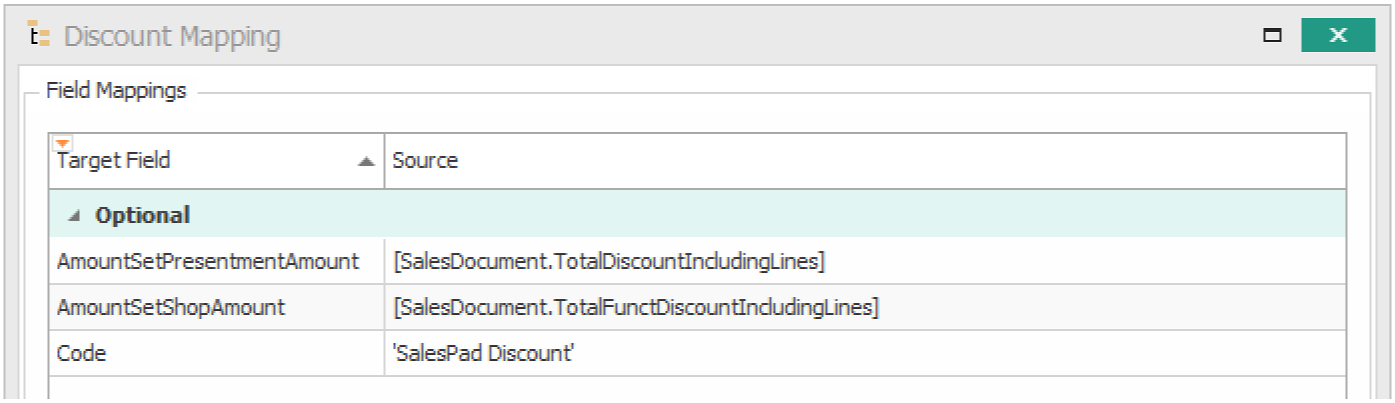
Billing Address Mapping

Field Mappings

Target Field	Source
Optional	
Address1	[SalesDocument.BillToAddr.val_Address_Line_1]
Address2	[SalesDocument.BillToAddr.val_Address_Line_2]
City	[SalesDocument.BillToAddr.val_City]
Company	[SalesDocument.BillToAddr.val_Alt_Company_Name]
CountryCode	[SalesDocument.BillToAddr.val_Country_Code]
FirstName	FirstWord([SalesDocument.BillToAddr.val_Contact_Person])
LastName	LastWord([SalesDocument.BillToAddr.val_Contact_Person])
Phone	[SalesDocument.BillToAddr.val_Phone_1]
ProvinceCode	[SalesDocument.BillToAddr.val_State]
Zip	[SalesDocument.BillToAddr.val_Zip]

Discount Mapping – *Define the mappings to be used when creating a discount for a new order on the website.*

While Shopify normally has line-level discounts, they are not available when creating an order via Shopify's API, so all line discounts are placed on the header.



Discount Mapping

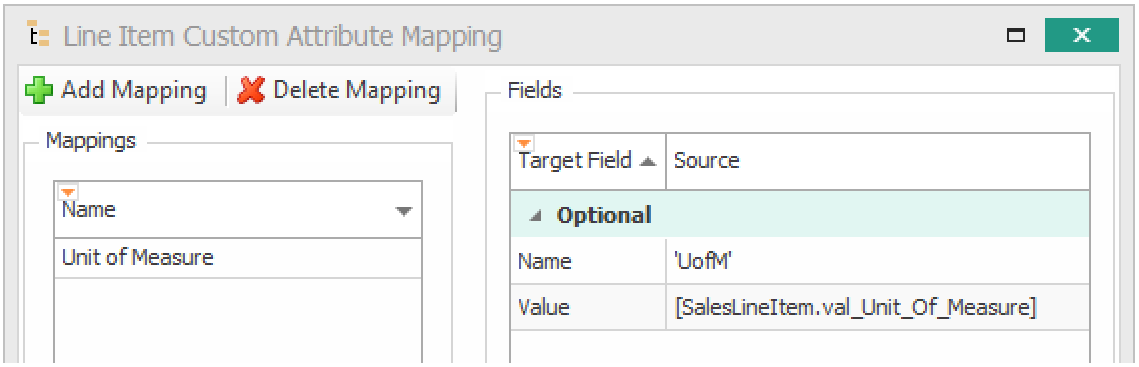
Field Mappings

Target Field	Source
Optional	
AmountSetPresentmentAmount	[SalesDocument.TotalDiscountIncludingLines]
AmountSetShopAmount	[SalesDocument.TotalFunctDiscountIncludingLines]
Code	'SalesPad Discount'

Line Item Custom Attribute Mapping – *Assign values to a line item's custom attributes.*

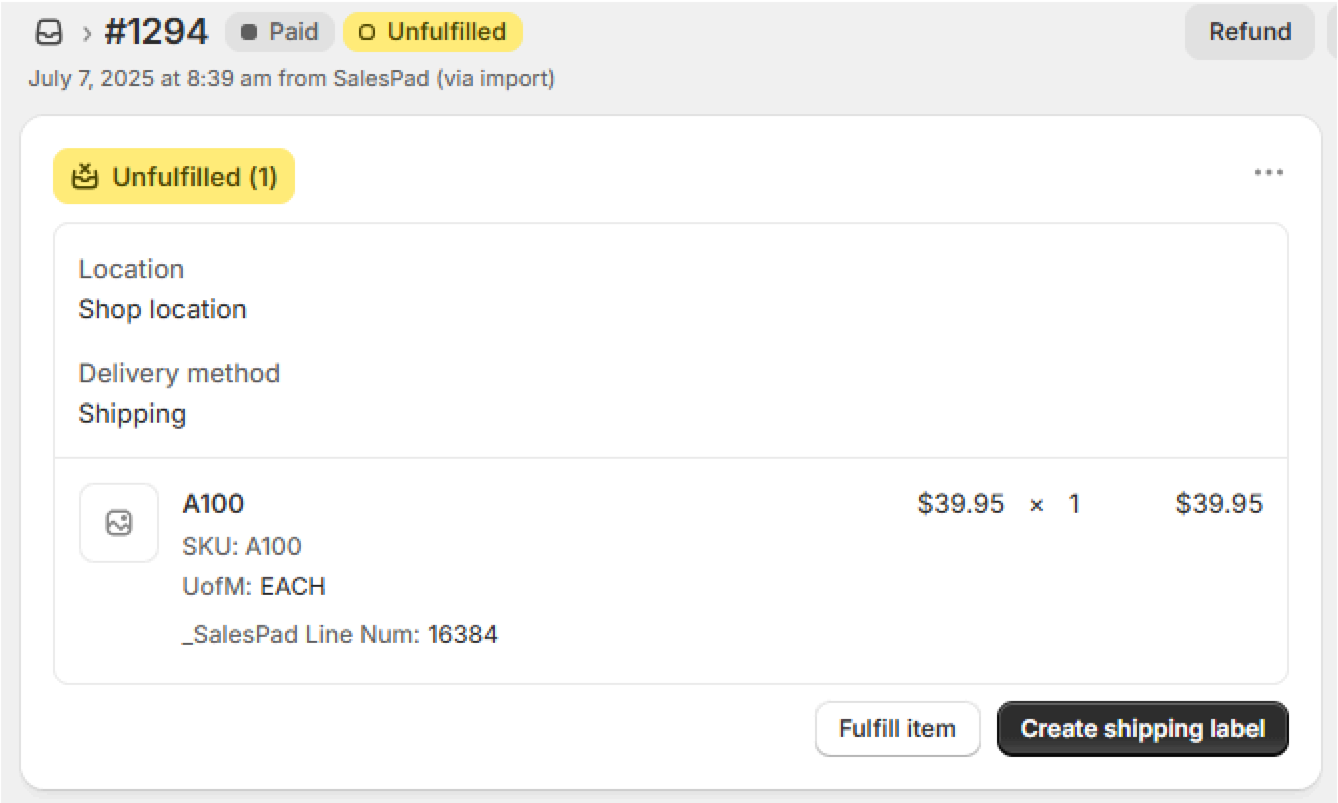
Shopify line items have custom attributes which can store arbitrary information. To create a custom attribute, click the Add Mapping button. In the right grid, give the attribute a name and map the value to a field or expression. Note that the name in the right grid is what is sent to Shopify, while the name in the left grid is only for reference.

In the following example, the Order Export will send each sales line item's unit of measure to Shopify. This attribute is visible on Shopify's admin hub as well as the order status page. Note that attributes which have a name starting with a '_' will only be visible on the admin hub by default.



Name
Unit of Measure

Target Field	Source
Optional	
Name	'UofM'
Value	[SalesLineItem.val_Unit_Of_Measure]

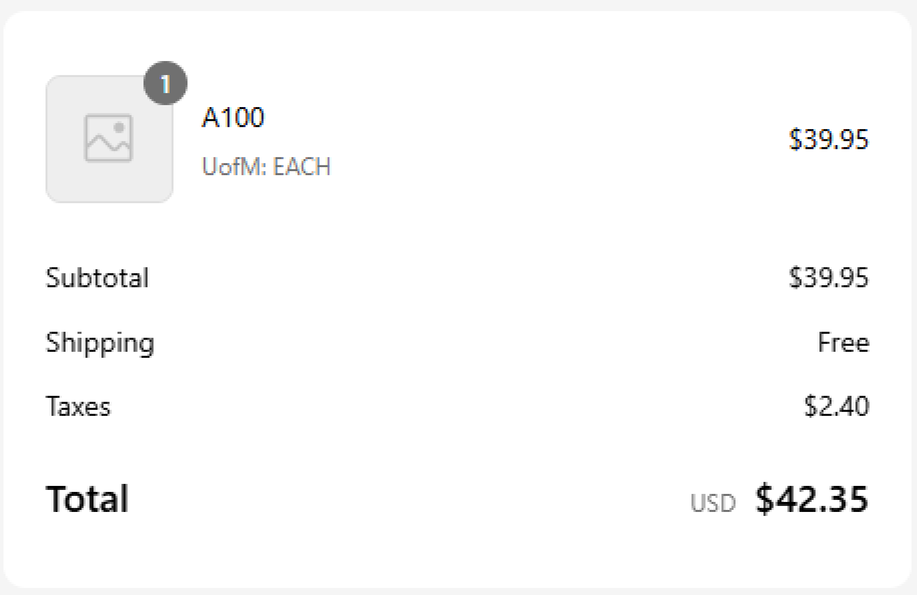


July 7, 2025 at 8:39 am from SalesPad (via import)

Location
Shop location

Delivery method
Shipping

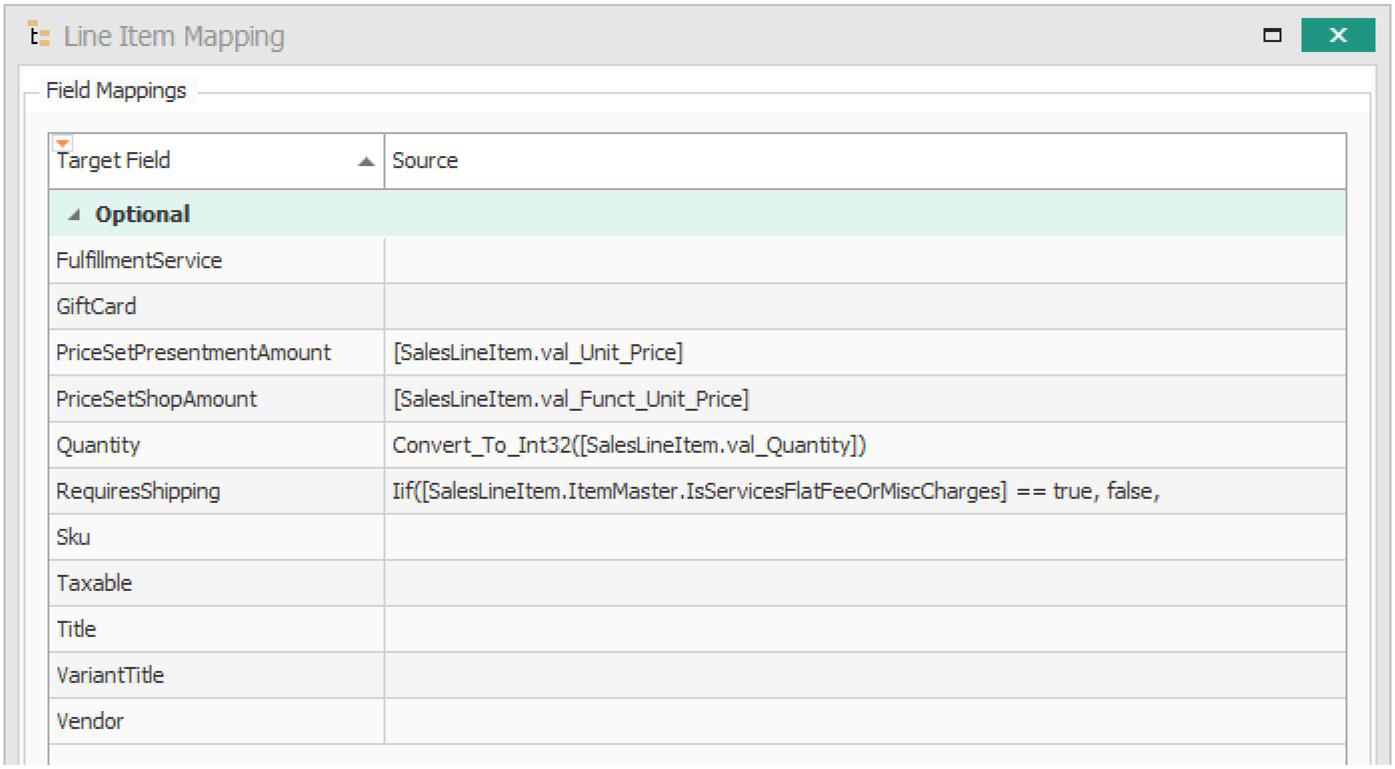
A100 $39.95 × 1 $39.95
SKU: A100
UofM: EACH
_SalesPad Line Num: 16384



1 A100 $39.95
UofM: EACH

Subtotal	$39.95
Shipping	Free
Taxes	$2.40
Total	USD **$42.35**

Line Item Mapping - *Define the mappings to be used when creating a new line item on the website.*



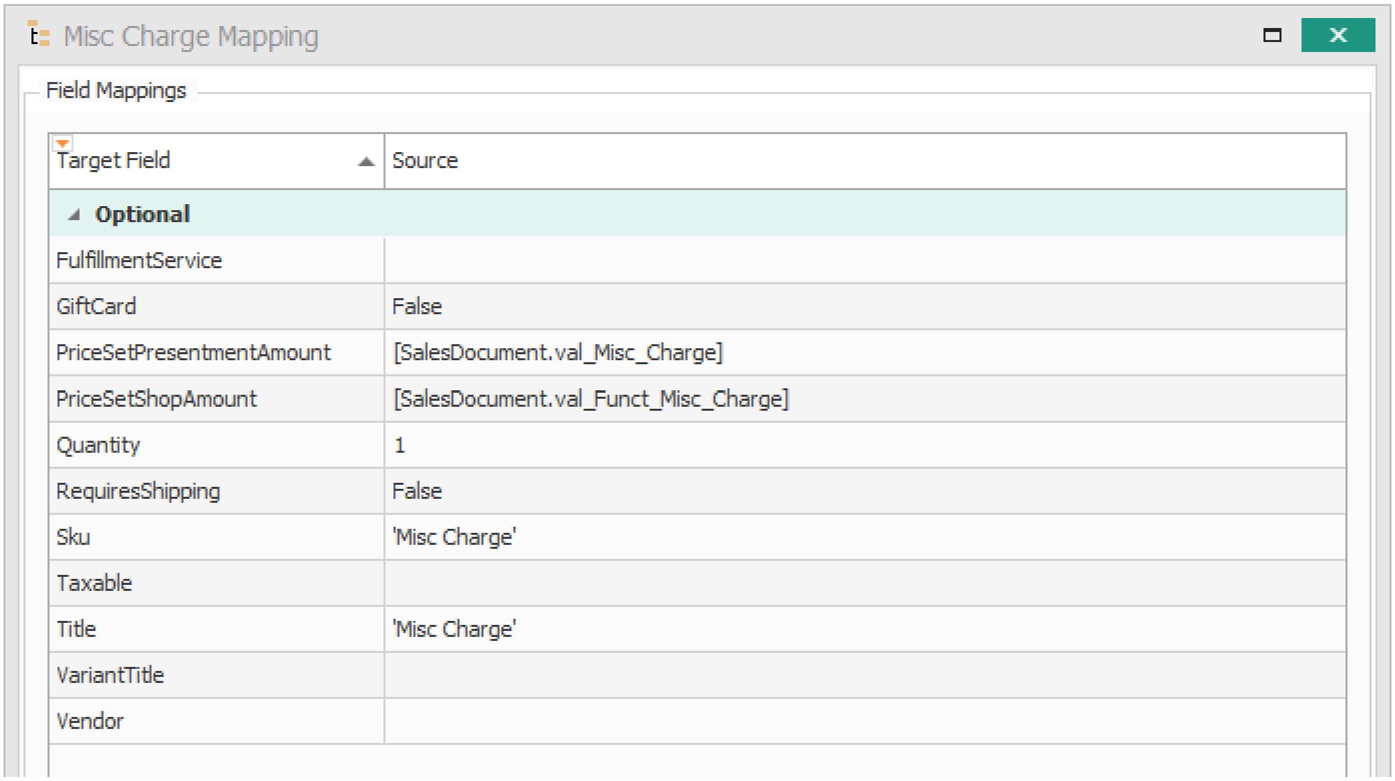
Line Item Mapping

Field Mappings

Target Field	Source
Optional	
FulfillmentService	
GiftCard	
PriceSetPresentmentAmount	[SalesLineItem.val_Unit_Price]
PriceSetShopAmount	[SalesLineItem.val_Funct_Unit_Price]
Quantity	Convert_To_Int32([SalesLineItem.val_Quantity])
RequiresShipping	Iif([SalesLineItem.ItemMaster.IsServicesFlatFeeOrMiscCharges] == true, false,
Sku	
Taxable	
Title	
VariantTitle	
Vendor	

Misc Charge Mapping – *Define the mappings to be used when creating a new line item representing the sales document's Misc Charge to the website.*

The Order Export exports all sales document data which contributes to the document's total. Line item prices, discounts, freight, and taxes all have Shopify equivalents, but misc charge does not. Therefore, misc charges are exported to Shopify as an extra line item.

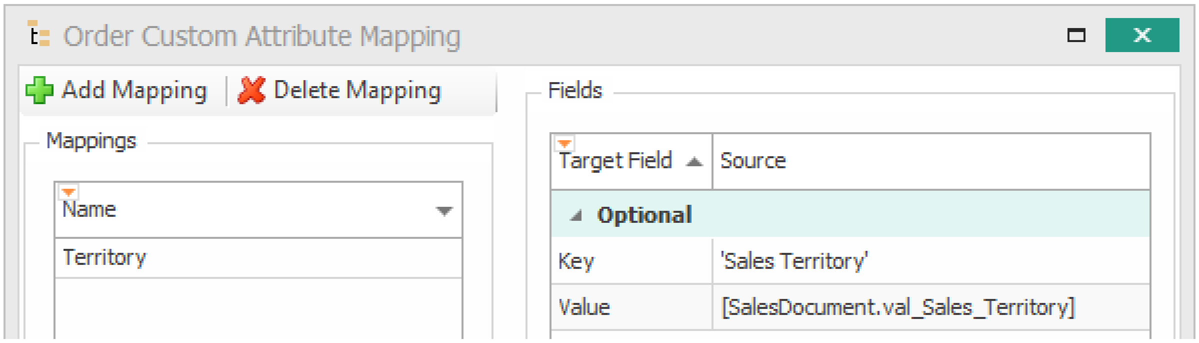


Target Field	Source
Optional	
FulfillmentService	
GiftCard	False
PriceSetPresentmentAmount	[SalesDocument.val_Misc_Charge]
PriceSetShopAmount	[SalesDocument.val_Funct_Misc_Charge]
Quantity	1
RequiresShipping	False
Sku	'Misc Charge'
Taxable	
Title	'Misc Charge'
VariantTitle	
Vendor	

Order Custom Attribute Mapping – *Assign values to an order's custom attributes.*

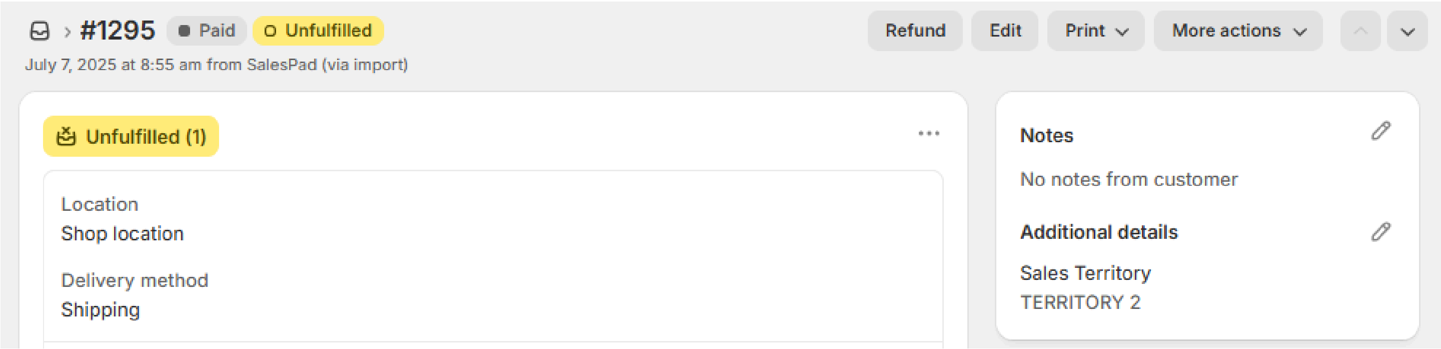
Shopify orders have custom attributes which can store arbitrary information. To create a custom attribute, click the Add Mapping button. In the right grid, give the attribute a name and map the value to a field or expression. Note that the name in the right grid is what is sent to Shopify, while the name in the left grid is only for reference.

In the following example, the Order Export will send each sales document's sales territory to Shopify. This attribute is visible on Shopify's admin hub in the additional details section.



Name
Territory

Target Field	Source
Optional	
Key	'Sales Territory'
Value	[SalesDocument.val_Sales_Territory]



July 7, 2025 at 8:55 am from SalesPad (via import)

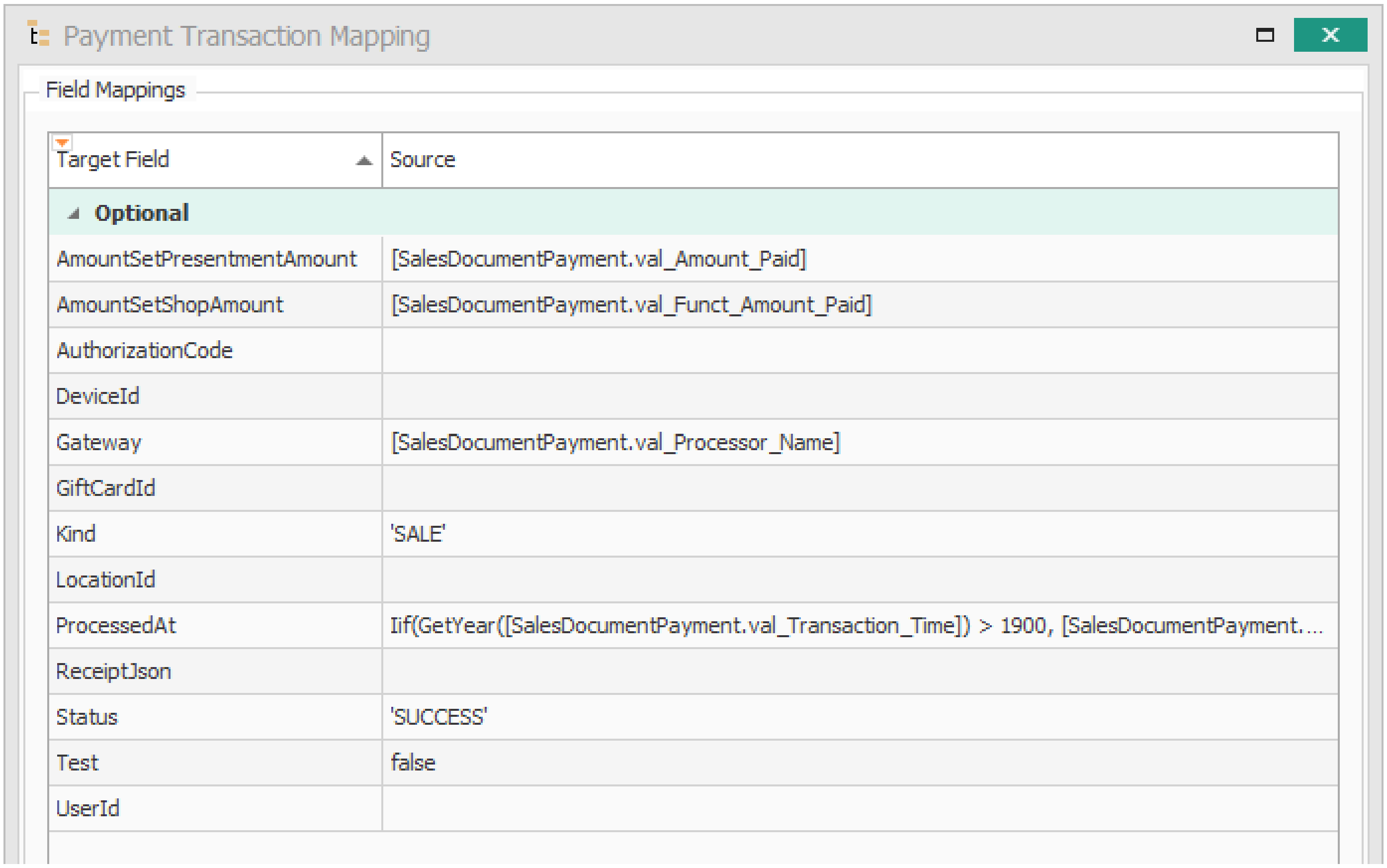
Location
Shop location

Delivery method
Shipping

Notes
No notes from customer

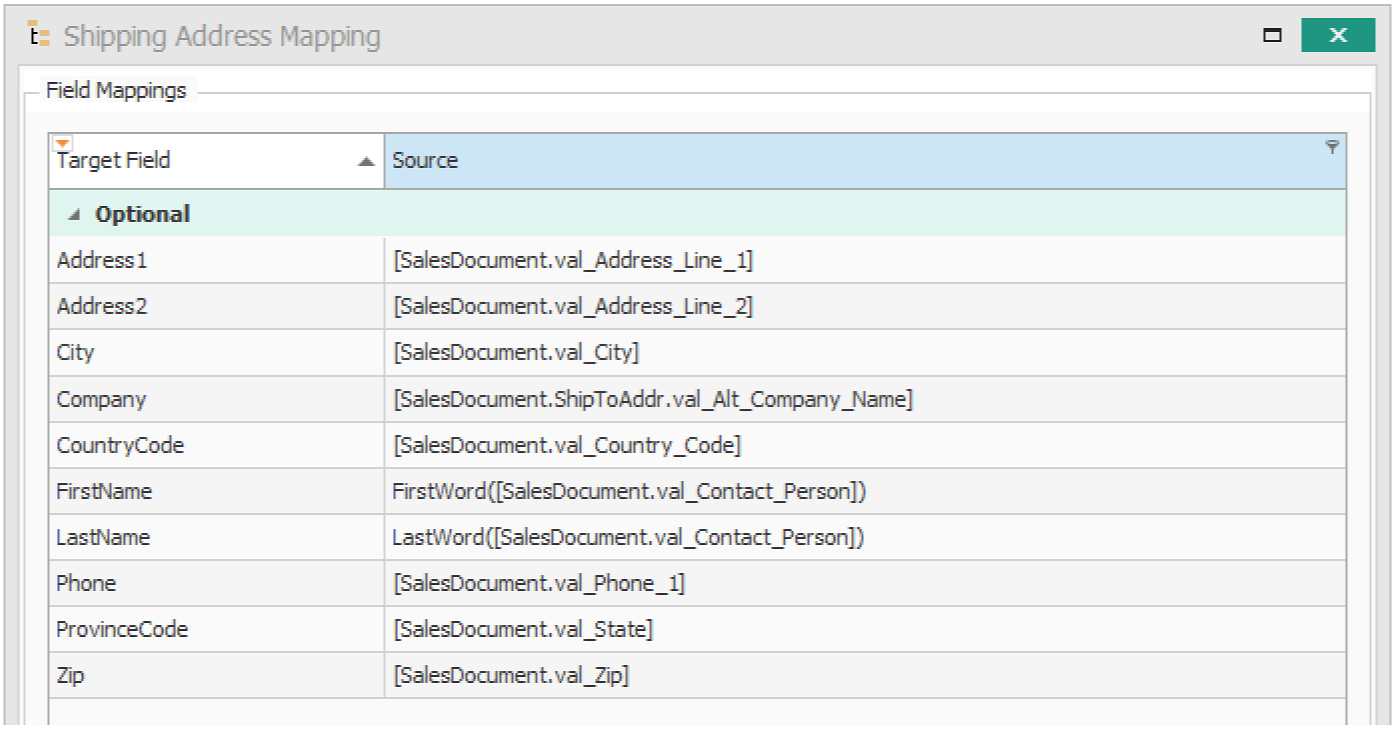
Additional details
Sales Territory
TERRITORY 2

Payment Transaction Mapping – *Define the mappings to be used when creating a new payment transaction for a new order on the website based on SalesPad payment information.*



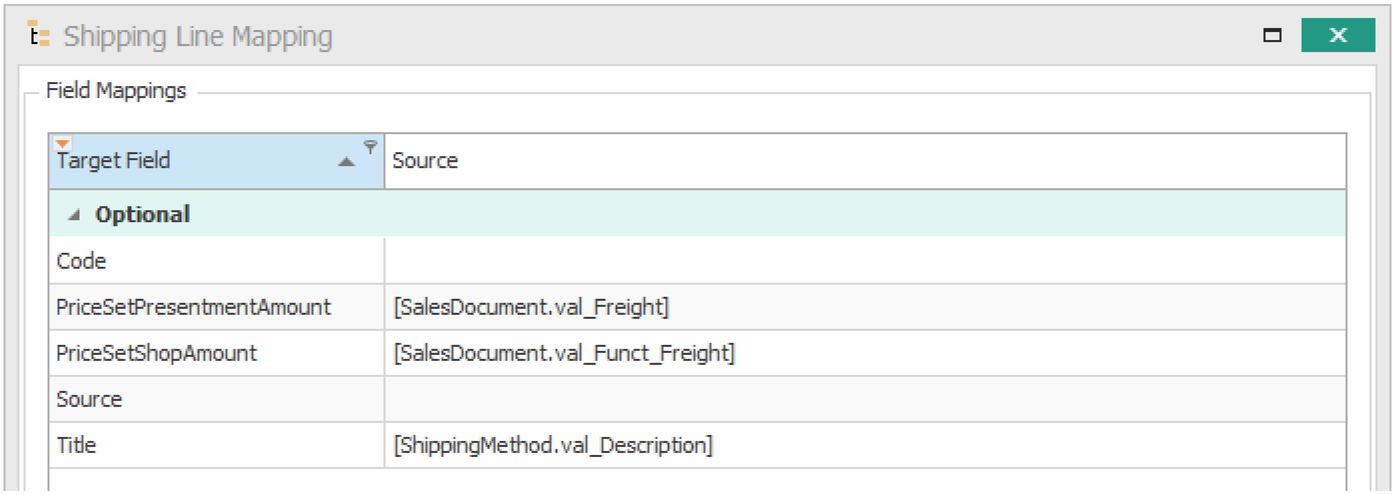
Target Field	Source
Optional	
AmountSetPresentmentAmount	[SalesDocumentPayment.val_Amount_Paid]
AmountSetShopAmount	[SalesDocumentPayment.val_Funct_Amount_Paid]
AuthorizationCode	
DeviceId	
Gateway	[SalesDocumentPayment.val_Processor_Name]
GiftCardId	
Kind	'SALE'
LocationId	
ProcessedAt	Iif(GetYear([SalesDocumentPayment.val_Transaction_Time]) > 1900, [SalesDocumentPayment....
ReceiptJson	
Status	'SUCCESS'
Test	false
UserId	

Shipping Address Mapping - *Define the mappings to be used when creating a shipping address for a new order on the website.*



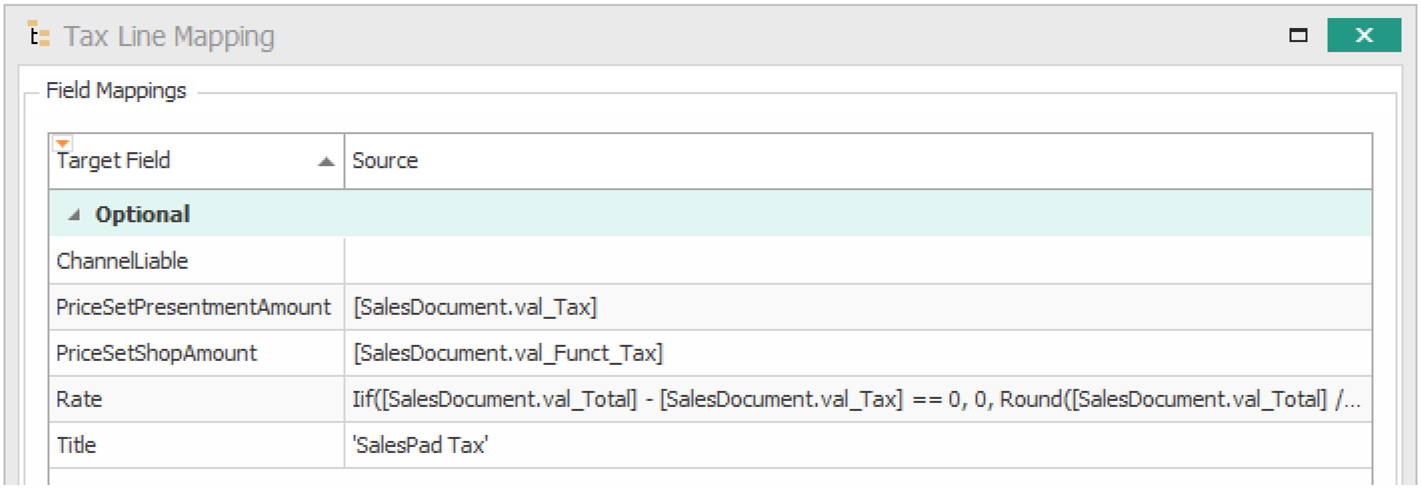
Target Field	Source
Optional	
Address1	[SalesDocument.val_Address_Line_1]
Address2	[SalesDocument.val_Address_Line_2]
City	[SalesDocument.val_City]
Company	[SalesDocument.ShipToAddr.val_Alt_Company_Name]
CountryCode	[SalesDocument.val_Country_Code]
FirstName	FirstWord([SalesDocument.val_Contact_Person])
LastName	LastWord([SalesDocument.val_Contact_Person])
Phone	[SalesDocument.val_Phone_1]
ProvinceCode	[SalesDocument.val_State]
Zip	[SalesDocument.val_Zip]

Shipping Line Mapping – *Define the mappings to be used when creating a new shipping line on the website. The Order Export will only create a single shipping line, and only if the shipping line has a price.*



Target Field	Source
Optional	
Code	
PriceSetPresentmentAmount	[SalesDocument.val_Freight]
PriceSetShopAmount	[SalesDocument.val_Funct_Freight]
Source	
Title	[ShippingMethod.val_Description]

Tax Line Mapping – *Define the mappings to be used when creating a new tax line on the website. The Order Export will only create a single tax line on the order header.*



Target Field	Source
Optional	
ChannelLiable	
PriceSetPresentmentAmount	[SalesDocument.val_Tax]
PriceSetShopAmount	[SalesDocument.val_Funct_Tax]
Rate	Iif([SalesDocument.val_Total] - [SalesDocument.val_Tax] == 0, 0, Round([SalesDocument.val_Total] /...
Title	'SalesPad Tax'

Scripts

Order Pre Export Script - *A C# Script that runs before an Order is exported.*

Parameters: System.ComponentModel.CancelEventArgs ce, SalesPad.Bus.SalesDocument sd, SalesPad.Shopify.OrderCreateOrderInput order, SalesPad.Shopify.OrderCreateOptionsInput options

This script runs after all mapping and matching settings have run, but before creating the order in Shopify. The CancelEventArgs can be used to cancel the export by setting its Cancel property to True.

Order Post Export Script - *A C# Script that runs after an Order is exported.*

Parameters: SalesPad.Bus.SalesDocument sd, SalesPad.Shopify.OrderCreateOrderOutput order

This script runs after the Order Export creates the Shopify order, but before saving the order and line IDs to the sales document.

Processing

The Order Export processes all sales documents which are in any of the queues specified by the *Export Orders In Workflow Queue* setting.

SalesPad only allows a sales document to be linked to a single eCommerce website. If an Order Import component created the sales document, then any Order Export components will not export the sales document to another website. Likewise, once an Order Export component exports a sales document to one website, other Order Export components will not export the sales document. A sales document's External ID field having a value indicates that the document is linked to a website order.

The Order Export runs all of the mapping settings to build most of the Shopify order. If the Order Import had previously matched the sales document's customer to a Shopify customer, the Order Export will assign the order to that Shopify customer. Otherwise it will use the *Customer Matching* setting to determine which customer to use. Likewise, if the Product Export had previously matched a sales line item's item master to a product variant, the Order Export will use that product variant. Otherwise it will use the *Item Master Matching* setting.

The Order Export then creates an order in Shopify. It writes the website order's ID to the sales document's External ID in SalesPad, and each website line item's ID to each SalesPad sales line item's External ID. Finally, it saves the sales document and forwards the document to its next workflow queue.

The Order Export checks for the possibility that a previous run of the component created an order in Shopify but SalesPad did not receive a response from Shopify or failed to save the sales document. It checks for open orders in Shopify which have a source identifier corresponding to the sales document. If it finds a matching order, the export will skip ahead to writing the External IDs to the sales document and lines, so that it does not create a duplicate website order.

Endpoints

Get Order

The Order Export first checks if the sales document was already exported to Shopify. If it was and the order has over 250 line items, the component will load the additional line item pages.

GraphQL query:

```
None
{
  "query": "
                    query getOrders($first: Int, $query: String) {
                      orders(first: $first, query: $query) {
                        nodes {
                          id
                          legacyResourceId
                          billingAddress {
                            address1
                            address2
  city
  company
  countryCodeV2
  firstName
  lastName
  phone
  provinceCode
  zip
}
customerAcceptsMarketing
closedAt
currencyCode
customAttributes {
  key
  value
}
customer {
  id
  displayName
}
discountCodes
totalDiscountsSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
email
displayFinancialStatus
lineItems(first: 250) {
  nodes {
    id
    isGiftCard
```

```
        originalUnitPriceSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        customAttributes {
          key
          value
        }
        quantity
        requiresShipping
        sku
        taxable
        title
        variant {
          id
          sku
        }
        variantTitle
        vendor
      }
    pageInfo {
      endCursor
      hasNextPage
    }
  }
name
note
phone
poNumber
presentmentCurrencyCode
processedAt
```

```
shippingAddress {
  address1
  address2
  city
  company
  countryCodeV2
  firstName
  lastName
  phone
  provinceCode
  zip
}
shippingLine {
  code
  originalPriceSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  source
  title
}
sourceIdentifier
sourceName
tags
taxesIncluded
taxLines {
  channelLiable
  priceSet {
    shopMoney {
      currencyCode
      amount
```

```
                            }
                            presentmentMoney {
                              currencyCode
                              amount
                            }
                          }
                          rate
                          title
                        }
                        test
                        transactions {
                          amountSet {
                            shopMoney {
                              currencyCode
                              amount
                            }
                            presentmentMoney {
                              currencyCode
                              amount
                            }
                          }
                          authorizationCode
                          gateway
                          kind
                          processedAt
                          receiptJson
                          status
                          test
                        }
                      }
                      pageInfo {
                        endCursor
                        hasNextPage
                      }
                    }
                  }",
"variables": {
    "first": 1,
    "query": "source_identifier:SalesPad Order SHOP0357 from database TWO AND status:OPEN"
  }
}
```

Sample response:

None

```
{
  "orders": {
    "nodes": [
      {
        "id": "gid://shopify/Order/6378718396719",
        "legacyResourceId": "6378718396719",
        "billingAddress": {
          "address1": "One Microsoft Way",
          "address2": "",
          "city": "Redmond",
          "company": "Aaron Fitz Electrical",
          "countryCodeV2": "US",
          "firstName": "Bob",
          "lastName": "Fitz",
          "phone": "",
          "provinceCode": "WA",
          "zip": "98052-6399"
        },
        "customerAcceptsMarketing": false,
        "closedAt": null,
        "currencyCode": "USD",
        "customAttributes": [
          {
            "key": "Sales Territory",
            "value": "TERRITORY 2"
          }
        ],
      "customer": {
        "id": "gid://shopify/Customer/8943179104559",
        "displayName": "Aaron Fitz"
      },
      "discountCodes": [],
      "totalDiscountsSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      },
      "email": "aaron.fitz@aaronfitzelectrical.net",
      "displayFinancialStatus": "PENDING",
      "lineItems": {
        "nodes": [
          {
            "id": "gid://shopify/LineItem/15780272537903",
            "isGiftCard": false,
            "originalUnitPriceSet": {
              "shopMoney": {
                "currencyCode": "USD",
                "amount": "39.95"
              },
              "presentmentMoney": {
                "currencyCode": "USD",
                "amount": "39.95"
              }
            },
            "customAttributes": [
              {
                "key": "UofM",
                "value": "EACH"
              },
                {
                  "key": "_SalesPad Line Num",
                  "value": "16384"
                }
              ],
              "quantity": 1,
              "requiresShipping": true,
              "sku": "A100",
              "taxable": true,
              "title": "A100",
              "variant": {
                "id": "gid://shopify/ProductVariant/51118270546223",
                "sku": "A100"
              },
              "variantTitle": null,
              "vendor": "Developer Store"
            }
          ],
          "pageInfo": {
            "endCursor":
"eyJsYXN0X2lkIjoxNTc4MDI3MjUzNzkwMywibGFzdF92YWx1ZSI6MTU3ODAyNzI1Mzc5MDN9",
            "hasNextPage": false
          }
        },
        "name": "#1297",
        "note": null,
        "phone": null,
        "poNumber": null,
        "presentmentCurrencyCode": "USD",
        "processedAt": "2025-07-07T16:16:47Z",
        "shippingAddress": {
          "address1": "11403 45 St. South",
          "address2": "",
          "city": "Chicago",
          "company": "Aaron Fitz Electrical",
          "countryCodeV2": "US",
          "firstName": "Bob",
          "lastName": "Fitz",
          "phone": "",
          "provinceCode": "IL",
          "zip": "60603-0776"
        },
        "shippingLine": null,
        "sourceIdentifier": "SalesPad Order SHOP0358 from database TWO",
        "sourceName": "22751084545",
        "tags": [
          "IMPORTED_FROM_SALESPAD"
        ],
        "taxesIncluded": false,
        "taxLines": [
          {
            "channelLiable": null,
            "priceSet": {
              "shopMoney": {
                "currencyCode": "USD",
                "amount": "2.4"
              },
              "presentmentMoney": {
                "currencyCode": "USD",
                "amount": "2.4"
              }
            },
            "rate": 0.0601,
            "title": "SalesPad Tax"
          }
        ],
        "test": false,
        "transactions": []
      }
    ],
    "pageInfo": {
      "endCursor":
"eyJsYXN0X2lkIjo2Mzc4NzE4Mzk2NzE5LCJsYXN0X3ZhbHVlIjoxNzUxOTA1MDA3MDAwfQ==",
      "hasNextPage": false
    }
  }
}
```

Get Product Variants

If any sales line items are not already linked to product variants via the Product Export, the Order Export will try to load them from Shopify based on their SKU. It loads product variants in bulk.

There is not a customer version of this API call because the component handles on-the-fly customer matching via OrderCreateAssociateCustomerAttributesInput's email field instead.

GraphQL query:

```
None
{
  "query": "
                  query getProductVariants($first: Int, $after: String, $query: String) {
                    productVariants(first: $first, after: $after, query: $query) {
                      nodes {
                        id
                        sku
                      }
                      pageInfo {
                        endCursor
                        hasNextPage
                      }
                    }
                  }",
  "variables": {
    "first": 250,
    "after": null,
    "query": "sku:'A100'"
  }
}
```

Sample response:

```
None
{
  "productVariants": {
    "nodes": [
      {
        "id": "gid://shopify/ProductVariant/51118270546223",
        "sku": "A100"
      }
    ],
    "pageInfo": {
      "endCursor":
"eyJsYXN0X2lkIjo1MTExODI3MDU0NjIyMywibGFzdF92YWx1ZSI6IjUxMTE4MjcwNTQ2MjIzIn0=",
      "hasNextPage": false
    }
  }
}
```

Create Order

Finally, the Order Export creates an order. If the order has over 250 line items, the component will load the additional line item pages.

GraphQL mutation:

```
None
{
  "query": "
                    mutation orderCreate($order: OrderCreateOrderInput!, $options:
OrderCreateOptionsInput) {
                      orderCreate(order: $order, options: $options) {
                        order {
                          id
                          legacyResourceId
                          billingAddress {
```

```
  address1
  address2
  city
  company
  countryCodeV2
  firstName
  lastName
  phone
  provinceCode
  zip
}
customerAcceptsMarketing
closedAt
currencyCode
customAttributes {
  key
  value
}
customer {
  id
  displayName
}
discountCodes
totalDiscountsSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
email
displayFinancialStatus
lineItems(first: 250) {
  nodes {
    id
    isGiftCard
    originalUnitPriceSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    customAttributes {
      key
      value
    }
    quantity
    requiresShipping
    sku
    taxable
    title
    variant {
      id
      sku
    }
    variantTitle
    vendor
  }
  pageInfo {
    endCursor
    hasNextPage
  }
}
name
note
phone
poNumber
```

```
presentmentCurrencyCode
processedAt
shippingAddress {
  address1
  address2
  city
  company
  countryCodeV2
  firstName
  lastName
  phone
  provinceCode
  zip
}
shippingLine {
  code
  originalPriceSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  source
  title
}
sourceIdentifier
sourceName
tags
taxesIncluded
taxLines {
  channelLiable
  priceSet {
    shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      rate
      title
    }
    test
    transactions {
      amountSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      authorizationCode
      gateway
      kind
      processedAt
      receiptJson
      status
      test
    }
  }
  userErrors {
    field
    message
  }
}
```

```
                    }",
  "variables": {
    "order": {
      "billingAddress": {
        "address1": "One Microsoft Way",
        "address2": "",
        "city": "Redmond",
        "company": "Aaron Fitz Electrical",
        "countryCode": "US",
        "firstName": "Bob",
        "lastName": "Fitz",
        "phone": "",
        "provinceCode": "WA",
        "zip": "98052-6399"
      },
      "currency": "USD",
      "customAttributes": [
        {
          "key": "Sales Territory",
          "value": "TERRITORY 2"
        }
      ],
      "customer": {
        "toAssociate": {
          "email": "aaron.fitz@aaronfitzelectrical.net"
        }
      },
      "financialStatus": "PENDING",
      "lineItems": [
        {
          "priceSet": {
            "shopMoney": {
              "amount": 39.95000,
              "currencyCode": "USD"
            },
            "presentmentMoney": {
              "amount": 39.95000,
              "currencyCode": "USD"
            }
          },
          "properties": [
            {
              "name": "UofM",
              "value": "EACH"
            },
            {
              "name": "_SalesPad Line Num",
              "value": "16384"
            }
          ],
          "quantity": 1,
          "requiresShipping": true,
          "variantId": "gid://shopify/ProductVariant/51118270546223"
        }
      ],
      "presentmentCurrency": "USD",
      "processedAt": "2025-07-07T16:16:47.0949984+00:00",
      "shippingAddress": {
        "address1": "11403 45 St. South",
        "address2": "",
        "city": "Chicago",
        "company": "Aaron Fitz Electrical",
        "countryCode": "US",
        "firstName": "Bob",
        "lastName": "Fitz",
        "phone": "",
        "provinceCode": "IL",
        "zip": "60603-0776"
      },
      "sourceIdentifier": "SalesPad Order SHOP0358 from database TWO",
      "tags": "IMPORTED_FROM_SALESPAD",
      "taxesIncluded": false,
      "taxLines": [
        {
          "priceSet": {
            "shopMoney": {
              "amount": 2.40000,
              "currencyCode": "USD"
            },
            "presentmentMoney": {
              "amount": 2.40000,
              "currencyCode": "USD"
            }
          },
          "rate": 0.0601,
          "title": "SalesPad Tax"
        }
      ],
      "test": false,
      "transactions": []
    },
    "options": {
      "inventoryBehaviour": "DECREMENT_OBEYING_POLICY",
      "sendFulfillmentReceipt": false,
      "sendReceipt": false
    }
  }
}
```

Sample response:

None

```
{
  "orderCreate": {
    "order": {
      "id": "gid://shopify/Order/6378718396719",
      "legacyResourceId": "6378718396719",
      "billingAddress": {
        "address1": "One Microsoft Way",
        "address2": "",
        "city": "Redmond",
        "company": "Aaron Fitz Electrical",
        "countryCodeV2": "US",
        "firstName": "Bob",
        "lastName": "Fitz",
        "phone": "",
        "provinceCode": "WA",
        "zip": "98052-6399"
      },
      "customerAcceptsMarketing": false,
      "closedAt": null,
      "currencyCode": "USD",
      "customAttributes": [
        {
          "key": "Sales Territory",
          "value": "TERRITORY 2"
        }
      ],
      "customer": {
        "id": "gid://shopify/Customer/8943179104559",
        "displayName": "Aaron Fitz"
      },
      "discountCodes": [],
      "totalDiscountsSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      },
      "email": "aaron.fitz@aaronfitzelectrical.net",
      "displayFinancialStatus": "PENDING",
      "lineItems": {
        "nodes": [
          {
            "id": "gid://shopify/LineItem/15780272537903",
            "isGiftCard": false,
            "originalUnitPriceSet": {
              "shopMoney": {
                "currencyCode": "USD",
                "amount": "39.95"
              },
              "presentmentMoney": {
                "currencyCode": "USD",
                "amount": "39.95"
              }
            },
            "customAttributes": [
              {
                "key": "UofM",
                "value": "EACH"
              },
              {
                "key": "_SalesPad Line Num",
                "value": "16384"
              }
            ],
            "quantity": 1,
            "requiresShipping": true,
            "sku": "A100",
            "taxable": true,
            "title": "A100",
            "variant": {
              "id": "gid://shopify/ProductVariant/51118270546223",
              "sku": "A100"
            },
            "variantTitle": null,
            "vendor": "Developer Store"
          }
        ],
      "pageInfo": {
        "endCursor":
"eyJsYXN0X2lkIjoxNTc4MDI3MjUzNzkwMywibGFzdF92YWx1ZSI6MTU3ODAyNzI1Mzc5MDN9",
        "hasNextPage": false
      }
    },
    "name": "#1297",
    "note": null,
    "phone": null,
    "poNumber": null,
    "presentmentCurrencyCode": "USD",
    "processedAt": "2025-07-07T16:16:47Z",
    "shippingAddress": {
      "address1": "11403 45 St. South",
      "address2": "",
      "city": "Chicago",
      "company": "Aaron Fitz Electrical",
      "countryCodeV2": "US",
      "firstName": "Bob",
      "lastName": "Fitz",
      "phone": "",
      "provinceCode": "IL",
      "zip": "60603-0776"
    },
    "shippingLine": null,
    "sourceIdentifier": "SalesPad Order SHOP0358 from database TWO",
    "sourceName": "22751084545",
    "tags": [
      "IMPORTED_FROM_SALESPAD"
    ],
    "taxesIncluded": false,
    "taxLines": [
      {
        "channelLiable": null,
        "priceSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "2.4"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "2.4"
          }
        },
        "rate": 0.0601,
        "title": "SalesPad Tax"
      }
    ],
    "test": false,
    "transactions": []
  },
  "userErrors": []
}
}
```

Order Update Export

Overview

The Order Update Export component sends tracking numbers and fulfillments back to Shopify for orders that were originally created by the Order Import component. It targets specific workflow queues, and it forwards orders out of those queues once processing is complete. This component can also trigger Shopify to capture the order's authorization and/or to notify the customer that order updates have occurred.

General Settings

Capture Payment Upon Completing Fulfillment - *If set to True, pending authorizations on Shopify orders will be captured by Shopify once the order is completely fulfilled. Defaults to 'False'.*

Export Failure Queue - *In the event of an unsuccessful export, the document will be placed into this queue.*

If there is an exception during the order export process, the document will be moved to this queue and the error will be logged to the Automation Agent Action Center.

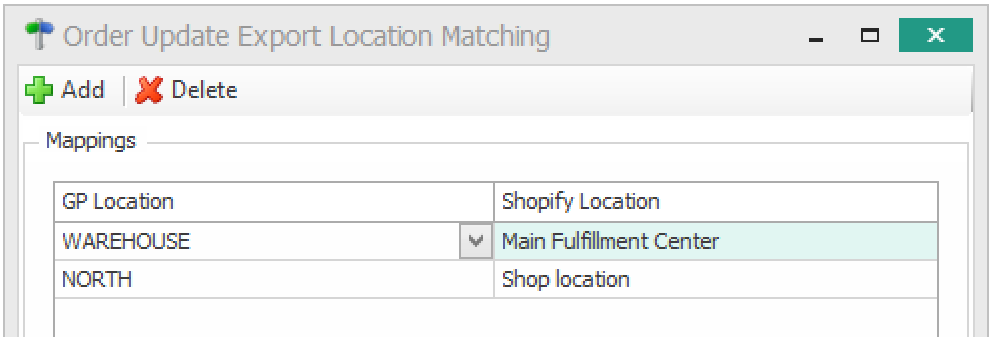
Notify Customers - *If set to True, Shopify will email the order's customer when the Order Update Export creates or updates fulfillments. Defaults to 'True'.*

Update Orders In Workflow Queue - *Queues that contain orders ready to be exported to Shopify. This component will ignore orders which do not have an External ID, but will assume that any orders which have an External ID belong to the linked Shopify store even if the order originated from a different automation instance.*

Orders that were imported from Shopify will need to be directed into one of these queues so that they can export fulfillment and tracking information back to Shopify. They will wait to be processed, and afterwards they will be forwarded in workflow.

Matching Settings

Order Update Export Location Matching - *Define which Shopify location to use for each GP location in the Order Update Export. A GP location can only be matched to one Shopify location, but multiple GP locations can be matched to the same Shopify location.*



GP Location	Shopify Location
WAREHOUSE	Main Fulfillment Center
NORTH	Shop location

Sales Line Matching - *Define the optional mappings for matching a Sales Line to a Shopify line. Can be used to override the default line matching.*

By default, the Order Update Export will use the External ID column on sales line items to determine which Shopify line corresponds to it. The Sales Line Matching setting can override this behavior if needed, but it is recommended to leave the setting blank.

Assignment Settings

Captured Payment Mapping - *Define the mappings to be used when creating a Sales Document Payment after capturing an authorization.*

Processing

The Order Update Export processes all sales orders which are in any of the queues specified by the *Update Orders In Workflow Queue* setting. It uses the link that was created when each order was imported to call the Shopify API and update the corresponding Shopify order. For sales documents, the External ID designates the linked Shopify Order ID. For sales lines, the *Sales Line Matching* setting is used to find the corresponding Shopify line. The sales line also has an External ID that stores the ID of the corresponding Shopify line item; if the *Sales Line Matching* setting has been left blank, or did not find a match for a given line, then the External ID is used to find the Shopify line.

Fulfillments and tracking information are added for each Shopify sales line. If *Notify Customers* is enabled, then a flag is sent to trigger Shopify to notify the customer that this information has been updated. If *Capture Payment Upon Completing Fulfillment* is enabled, then a flag is sent to trigger Shopify to capture the pending payment.

The SalesPad order is forwarded in workflow if all processing was successful, and otherwise it is moved to the workflow queue in the *Export Failure Queue* setting.

Endpoints

Get Order Fulfillment Information

The Order Update Export first loads the order from Shopify. If hasNextPage is true for any of the connections, the integration will load the additional pages for importing.

GraphQL query:

None

```
query getOrder($id: ID!) {
  order(id: $id) {
    id
    fulfillmentOrders(first: 250) {
      nodes {
        id
        assignedLocation {
          location {
            id
            legacyResourceId
            name
          }
        }
        lineItems(first: 250) {
          nodes {
            id
            lineItem {
              id
              unfulfilledQuantity
            }
            remainingQuantity
          }
          pageInfo {
            endCursor
            hasNextPage
          }
        }
        supportedActions {
          action
        }
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
    fulfillments {
      id
      fulfillmentLineItems(first: 250) {
        nodes {
          lineItem {
            id
            unfulfilledQuantity
          }
          quantity
        }
        pageInfo {
          endCursor
          hasNextPage
        }
      }
      status
      trackingInfo {
        number
      }
    }
    lineItems(first: 250) {
      nodes {
        id
        unfulfilledQuantity
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
  }
}
```

Sample response:

```
None
{
  "order": {
    "id": "gid://shopify/Order/6238605934895",
    "fulfillmentOrders": {
      "nodes": [
        {
          "id": "gid://shopify/FulfillmentOrder/7211687149871",
          "assignedLocation": {
            "location": {
              "id": "gid://shopify/Location/76116001071",
              "legacyResourceId": "76116001071",
              "name": "WAREHOUSE"
            }
          },
          "lineItems": {
            "nodes": [
              {
                "id": "gid://shopify/FulfillmentOrderLineItem/15699563872559",
                "lineItem": {
                  "id": "gid://shopify/LineItem/15532308922671",
                  "unfulfilledQuantity": 1
                },
                "remainingQuantity": 1
              }
            ],
            "pageInfo": {
              "endCursor":
"eyJsYXN0X2lkIjoxNTY5OTU2Mzg3MjU1OSwibGFzdF92YWx1ZSI6MTU2OTk1NjM4NzI1NTl9",
              "hasNextPage": false
            }
          },
          "supportedActions": [
            {
              "action": "CREATE_FULFILLMENT"
            },
            {
              "action": "MOVE"
            },
            {
              "action": "HOLD"
            }
          ]
        }
```

```
      ],
      "pageInfo": {
        "endCursor":
"eyJsYXN0X2lkIjo3MjExNjg3MTQ5ODcxLCJsYXN0X3ZhbHVlIjoiNzIxMTY4NzE0OTg3MSJ9",
        "hasNextPage": false
      }
    },
    "fulfillments": [],
    "lineItems": {
      "nodes": [
        {
          "id": "gid://shopify/LineItem/15532308922671",
          "unfulfilledQuantity": 1
        }
      ],
      "pageInfo": {
        "endCursor":
"eyJsYXN0X2lkIjoxNTUzMjMwODkyMjY3MSwibGFzdF92YWx1ZSI6MTU1MzIzMDg5MjI2NzF9",
        "hasNextPage": false
      }
    }
  }
}
```

Create Fulfillment

Next, the Order Update Export will create a fulfillment.

GraphQL mutation:

None

```
mutation fulfillmentCreate($fulfillment: FulfillmentInput!) {
  fulfillmentCreate(fulfillment: $fulfillment) {
    userErrors {
      field
      message
    }
  }
}
```

Move Fulfillment Order

Shopify makes assumptions about which location an order will be fulfilled from. If these assumptions do not match how the order was fulfilled in SalesPad, or if the SalesPad order has fulfillments from multiple locations, the Order Update Export must move or split a fulfillment order.

GraphQL mutation:

```
None
mutation fulfillmentOrderMove($id: ID!, $newLocationId: ID!) {
  fulfillmentOrderMove(id: $id, newLocationId: $newLocationId) {
    movedFulfillmentOrder {
      id
      assignedLocation {
        location {
          id
          legacyResourceId
          name
        }
      }
      lineItems(first: 250) {
        nodes {
          id
          lineItem {
            id
            unfulfilledQuantity
          }
          remainingQuantity
        }
        pageInfo {
          endCursor
          hasNextPage
        }
      }
      supportedActions {
        action
      }
    }
    originalFulfillmentOrder {
      id
      assignedLocation {
        location {
          id
          legacyResourceId
          name
        }
      }
      lineItems(first: 250) {
        nodes {
          id
          lineItem {
            id
            unfulfilledQuantity
          }
          remainingQuantity
        }
        pageInfo {
          endCursor
          hasNextPage
        }
      }
      supportedActions {
        action
      }
    }
    userErrors {
      field
      message
    }
  }
}
```

Sample response:

None

```
{
  "fulfillmentOrderMove": {
    "movedFulfillmentOrder": {
      "id": "gid://shopify/FulfillmentOrder/7211780538671",
      "assignedLocation": {
        "location": {
          "id": "gid://shopify/Location/76694225199",
          "legacyResourceId": "76694225199",
          "name": "NORTH"
        }
      },
      "lineItems": {
        "nodes": [
          {
            "id": "gid://shopify/FulfillmentOrderLineItem/15699716145455",
            "lineItem": {
              "id": "gid://shopify/LineItem/15532446286127",
              "unfulfilledQuantity": 1
            },
            "remainingQuantity": 1
          }
        ],
        "pageInfo": {
          "endCursor":
"eyJsYXN0X2lkIjoxNTY5OTcxNjE0NTQ1NSwibGFzdF92YWx1ZSI6MTU2OTk3MTYxNDU0NTV9",
          "hasNextPage": false
        }
      },
      "supportedActions": [
        {
          "action": "CREATE_FULFILLMENT"
        },
        {
          "action": "MOVE"
        },
        {
          "action": "HOLD"
        }
      ]
    },
    "originalFulfillmentOrder": {
      "id": "gid://shopify/FulfillmentOrder/7211777229103",
      "assignedLocation": {
        "location": {
          "id": "gid://shopify/Location/76116001071",
          "legacyResourceId": "76116001071",
          "name": "WAREHOUSE"
        }
      },
      "lineItems": {
        "nodes": [
          {
            "id": "gid://shopify/FulfillmentOrderLineItem/15699710214447",
            "lineItem": {
              "id": "gid://shopify/LineItem/15532446253359",
              "unfulfilledQuantity": 0
            },
            "remainingQuantity": 0
          }
        ],
        "pageInfo": {
          "endCursor":
"eyJsYXN0X2lkIjoxNTY5OTcxMDIxNDQ0NywibGFzdF92YWx1ZSI6MTU2OTk3MTAyMTQ0NDd9",
          "hasNextPage": false
        }
      },
      "supportedActions": []
    },
    "userErrors": []
  }
}
```

Get Order

If the ‘Capture Payment Upon Completing Fulfillment’ setting is set to True, or if the ‘Sales Line Matching’ setting has a value, the Order Update Export will also load the Shopify order information which the Order Import loads. This is done so that the Order Update Export has access to the order’s transactions, and so that the ‘Captured Payment Mapping’ and ‘Sales Line Matching’ settings have access to the normal number of order and line item fields.

GraphQL query:

```
None
query getOrder($id: ID!) {
  order(id: $id) {
    legacyResourceId
    id
    billingAddress {
      id
      address1
      address2
      city
      company
      country
      countryCodeV2
      firstName
      lastName
      latitude
      longitude
      name
      phone
      province
      provinceCode
      zip
    }
    clientIp
    customerAcceptsMarketing
    cancelReason
    cancelledAt
    closedAt
    confirmed
    createdAt
    currencyCode
    customer {
      legacyResourceId
      id
      createdAt
      defaultAddress {
        id
        address1
        address2
        city
        company
        country
        countryCodeV2
        firstName
        lastName
        latitude
        longitude
        name
        phone
        province
        provinceCode
        zip
      }
      email
```

```
      firstName
      multipassIdentifier
      lastName
      note
      phone
      state
      tags
      taxExempt
      taxExemptions
      updatedAt
      verifiedEmail
      smsMarketingConsent {
        marketingState
        marketingOptInLevel
        consentUpdatedAt
        consentCollectedFrom
      }
      emailMarketingConsent {
        marketingState
        marketingOptInLevel
        consentUpdatedAt
      }
    }
    customerLocale
    discountApplications(first: 250) {
      nodes {

        allocationMethod
        targetSelection
        targetType
        value {
          ... on MoneyV2 {
            amount
          }
          ... on PricingPercentageValue {
            percentage
          }
        }
        ... on DiscountCodeApplication {
          code
        }
        ... on ManualDiscountApplication {
          title
          description
        }
        ... on AutomaticDiscountApplication {
          title
        }
        ... on ScriptDiscountApplication {
          title
        }
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
    email
    displayFinancialStatus
    displayFulfillmentStatus
    phone
    tags
    lineItems(first: 250) {
      nodes {
        id
        unfulfilledQuantity
        originalUnitPriceSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        product {
          legacyResourceId
        }
        currentQuantity
        quantity
        requiresShipping
        sku
        title
        variant {
          legacyResourceId
        }
        variantTitle
        name
        vendor
        isGiftCard
        taxable
        taxLines {
          priceSet {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          rate
          title
        }
        totalDiscountSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
        discountAllocations {
          allocatedAmountSet {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          discountApplication {
            index
          }
        }
        duties {
          id
          harmonizedSystemCode
          countryCodeOfOrigin
          price {
            shopMoney {
              currencyCode
              amount
            }
            presentmentMoney {
              currencyCode
              amount
            }
          }
          taxLines {
            priceSet {
              shopMoney {
```

```
                currencyCode
                amount
              }
              presentmentMoney {
                currencyCode
                amount
              }
            }
            rate
            title
          }
        }
      }
      pageInfo {
        endCursor
        hasNextPage
      }
    }
    retailLocation {
      id
      legacyResourceId
      name
    }
    name
    note
    customAttributes {
      key
      value
    }
    statusPageUrl
    paymentGatewayNames
    processedAt
    shippingAddress {
      id
      address1
      address2
      city
      company
      country
      countryCodeV2
      firstName
      lastName
      latitude
      longitude
      name
      phone
      province
      provinceCode
      zip
    }
    shippingLine {
      id
      carrierIdentifier
      code
      phone
      originalPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      discountedPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
```

```
  }
  discountAllocations {
    allocatedAmountSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    discountApplication {
      index
    }
  }
  source
  title
  taxLines {
    priceSet {
      shopMoney {
        currencyCode
        amount
      }
      presentmentMoney {
        currencyCode
        amount
      }
    }
    rate
    title
  }
}
sourceIdentifier
sourceName
subtotalPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
taxLines {
  priceSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
  rate
  title
}
taxesIncluded
test
totalDiscountsSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalTipReceivedSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalTaxSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalWeight
updatedAt
transactions {
  id
  amountSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
```

```
    currencyCode
    amount
  }
}
authorizationCode
authorizationExpiresAt
createdAt
gateway
paymentDetails {

  ... on CardPaymentDetails {
    avsResultCode
    bin
    cvvResultCode
    number
    company
    name
    wallet
    expirationMonth
    expirationYear
  }
}
kind
receiptJson
errorCode
status
test
parentTransaction {
  id
}
processedAt
maximumRefundableV2 {
  currencyCode
  amount
}
paymentId
totalUnsettledSet {
    shopMoney {
      currencyCode
      amount
    }
    presentmentMoney {
      currencyCode
      amount
    }
  }
}
currentTotalDutiesSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
originalTotalDutiesSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
presentmentCurrencyCode
currentShippingPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalShippingPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
totalOutstandingSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
estimatedTaxes
currentSubtotalPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
currentTotalDiscountsSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
currentTotalPriceSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
currentTotalTaxSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
currentTotalAdditionalFeesSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
originalTotalAdditionalFeesSet {
  shopMoney {
    currencyCode
    amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
poNumber
taxExempt
}
}
```

Sample response:

None

```
{
  "order": {
    "legacyResourceId": "6238876303663",
    "id": "gid://shopify/Order/6238876303663",
    "billingAddress": {
      "id": "gid://shopify/MailingAddress/19882377740591?model_name=Address",
      "address1": "3351 Claystone Street Southeast",
      "address2": null,
      "city": "Grand Rapids",
      "company": "Aaron Fitz Electrical",
      "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Aaron",
      "lastName": "Fitz",
      "latitude": null,
      "longitude": null,
    "name": "Aaron Fitz",
    "phone": null,
    "province": "Michigan",
    "provinceCode": "MI",
    "zip": "49546"
  },
  "clientIp": "184.175.130.74",
  "customerAcceptsMarketing": false,
  "cancelReason": null,
  "cancelledAt": null,
  "closedAt": null,
  "confirmed": true,
  "createdAt": "2025-03-20T16:22:13Z",
  "currencyCode": "USD",
  "customer": {
    "legacyResourceId": "8943179104559",
    "id": "gid://shopify/Customer/8943179104559",
    "createdAt": "2025-03-20T12:50:20Z",
    "defaultAddress": {
      "id": "gid://shopify/MailingAddress/10818200240431?model_name=CustomerAddress",
      "address1": "3351 Claystone Street Southeast",
      "address2": "",
      "city": "Grand Rapids",
      "company": "Aaron Fitz Electrical",
      "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Aaron",
      "lastName": "Fitz",
      "latitude": 42.9237382,
      "longitude": -85.5853301,
      "name": "Aaron Fitz",
      "phone": null,
      "province": "Michigan",
      "provinceCode": "MI",
      "zip": "49546"
    },
    "email": null,
      "firstName": "Aaron",
      "multipassIdentifier": null,
      "lastName": "Fitz",
      "note": "",
      "phone": null,
      "state": "DISABLED",
      "tags": [],
      "taxExempt": false,
      "taxExemptions": [],
      "updatedAt": "2025-03-20T16:22:13Z",
      "verifiedEmail": true,
      "smsMarketingConsent": null,
      "emailMarketingConsent": null
    },
    "customerLocale": "en",
    "discountApplications": {
      "nodes": [],
      "pageInfo": {
        "endCursor": null,
        "hasNextPage": false
      }
    },
    "email": null,
    "displayFinancialStatus": "AUTHORIZED",
    "displayFulfillmentStatus": "FULFILLED",
    "phone": null,
    "tags": [
      "EXPORTED_TO_SALESPAD"
    ],
    "lineItems": {
      "nodes": [
        {
          "id": "gid://shopify/LineItem/15532750078255",
          "unfulfilledQuantity": 0,
          "originalUnitPriceSet": {
            "shopMoney": {
              "currencyCode": "USD",
    "amount": "59.99"
  },
  "presentmentMoney": {
    "currencyCode": "USD",
    "amount": "59.99"
  }
},
"product": {
  "legacyResourceId": "8061405036847"
},
"currentQuantity": 1,
"quantity": 1,
"requiresShipping": true,
"sku": "100XLG",
"title": "100XLG",
"variant": {
  "legacyResourceId": "44263493533999"
},
"variantTitle": null,
"name": "100XLG",
"vendor": "Developer Store",
"isGiftCard": false,
"taxable": true,
"taxLines": [
  {
    "priceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      }
    },
    "rate": 0.06,
    "title": "Michigan State Tax"
```

```
            }
          ],
          "totalDiscountSet": {
            "shopMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            },
            "presentmentMoney": {
              "currencyCode": "USD",
              "amount": "0.0"
            }
          },
          "discountAllocations": [],
          "duties": []
        }
      ],
      "pageInfo": {
        "endCursor":
"eyJsYXN0X2lkIjoxNTUzMjc1MDA3ODI1NSwibGFzdF92YWx1ZSI6MTU1MzI3NTAwNzgyNTV9",
        "hasNextPage": false
      }
    },
    "retailLocation": null,
    "name": "#1166",
    "note": null,
    "customAttributes": [],
    "statusPageUrl":
"https://developerstore.myshopify.com/70387892527/orders/a82937ba45103f0c7579ad2e64f11799/aut
henticate?key=2f3f296968f7fc052e45242a0a77df0b",
    "paymentGatewayNames": [
      "shopify_payments"
    ],
    "processedAt": "2025-03-20T16:22:11Z",
    "shippingAddress": {
      "id": "gid://shopify/MailingAddress/19882377707823?model_name=Address",
      "address1": "3351 Claystone Street Southeast",
      "address2": null,
      "city": "Grand Rapids",
      "company": "Aaron Fitz Electrical",
      "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Aaron",
      "lastName": "Fitz",
      "latitude": 42.9237382,
      "longitude": -85.5853301,
      "name": "Aaron Fitz",
      "phone": null,
      "province": "Michigan",
      "provinceCode": "MI",
      "zip": "49546"
    },
    "shippingLine": null,
    "sourceIdentifier": null,
    "sourceName": "shopify_draft_order",
    "subtotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      }
    },
    "taxLines": [
      {
        "priceSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "3.6"
          }
        },
        "rate": 0.06,
        "title": "Michigan State Tax"
      }
    ],
    "taxesIncluded": false,
    "test": true,
    "totalDiscountsSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalTipReceivedSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      }
    },
    "totalTaxSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      }
    },
    "totalWeight": "2268",
    "updatedAt": "2025-03-20T16:23:13Z",
    "transactions": [
      {
        "id": "gid://shopify/OrderTransaction/7589956682031",
        "amountSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          }
        },
        "authorizationCode": "pi_3R4lrEQxz3jysvRG0TuNgwjY",
        "authorizationExpiresAt": "2025-03-27T16:22:12Z",
        "createdAt": "2025-03-20T16:22:11Z",
        "gateway": "shopify_payments",
        "paymentDetails": {
          "avsResultCode": "Y",
          "bin": "555555",
          "cvvResultCode": "M",
          "number": "•••• •••• •••• 4444",
          "company": "Mastercard",
          "name": "Aaron Fitz",
     "wallet": null,
     "expirationMonth": 6,
     "expirationYear": 2025
    },
    "kind": "AUTHORIZATION",
    "receiptJson":
"{\"id\":\"pi_3R4lrEQxz3jysvRG0TuNgwjY\",\"object\":\"payment_intent\",\"amount\":6359,\"amou
nt_capturable\":6359,\"amount_received\":0,\"canceled_at\":null,\"cancellation_reason\":null,
\"capture_method\":\"manual\",\"charges\":{\"object\":\"list\",\"data\":[{\"id\":\"ch_3R4lrEQ
xz3jysvRG0o3dFonF\",\"object\":\"charge\",\"amount\":6359,\"application_fee\":null,\"balance_
transaction\":null,\"captured\":false,\"created\":1742487732,\"currency\":\"usd\",\"failure_c
ode\":null,\"failure_message\":null,\"fraud_details\":{},\"livemode\":false,\"metadata\":{\"m
anual_entry\":\"false\",\"order_id\":\"rNFXtnHLztgeHyXiCTBSIAIjc\",\"order_transaction_id\":\
"7589956682031\",\"payments_charge_id\":\"2942828773679\",\"shop_id\":\"70387892527\",\"shop_
name\":\"Developer
Store\",\"shopify_payments_next\":\"true\"},\"outcome\":{\"advice_code\":null,\"network_advic
e_code\":null,\"network_decline_code\":null,\"network_status\":\"approved_by_network\",\"reas
on\":null,\"risk_level\":\"normal\",\"risk_score\":11,\"seller_message\":\"Payment
complete.\",\"type\":\"authorized\"},\"paid\":true,\"payment_intent\":\"pi_3R4lrEQxz3jysvRG0T
uNgwjY\",\"payment_method\":\"pm_1R4lrEQxz3jysvRGQJmVzg9t\",\"payment_method_details\":{\"car
d\":{\"amount_authorized\":6359,\"authorization_code\":null,\"brand\":\"mastercard\",\"captur
e_before\":1745079732,\"checks\":{\"address_line1_check\":\"pass\",\"address_postal_code_chec
k\":\"pass\",\"cvc_check\":\"pass\"},\"country\":\"US\",\"description\":\"MasterCard Prepaid
Card
(Non-US)\",\"ds_transaction_id\":null,\"exp_month\":6,\"exp_year\":2025,\"extended_authorizat
ion\":{\"status\":\"enabled\"},\"fingerprint\":\"W42R39dc2U266g0i\",\"funding\":\"credit\",\"
iin\":\"555555\",\"incremental_authorization\":{\"status\":\"unavailable\"},\"installments\":
null,\"issuer\":\"CIAGROUP\",\"last4\":\"4444\",\"mandate\":null,\"moto\":true,\"multicapture
\":{\"status\":\"unavailable\"},\"network\":\"mastercard\",\"network_token\":{\"used\":false}
,\"network_transaction_id\":\"MCCJTPOW60320\",\"overcapture\":{\"maximum_amount_capturable\":
6359,\"status\":\"unavailable\"},\"overcapture_supported\":false,\"payment_account_reference\
":\"0zGFFYNQCX24uQFJVlH3Xbf1yJ6Le\",\"regulated_status\":\"unregulated\",\"three_d_secure\":n
ull,\"wallet\":null},\"type\":\"card\"},\"refunded\":false,\"source\":null,\"status\":\"succe
eded\",\"mit_params\":{\"network_transaction_id\":\"MCCJTPOW60320\"}}],\"has_more\":false,\"t
otal_count\":1,\"url\":\"\\/v1\\/charges?payment_intent=pi_3R4lrEQxz3jysvRG0TuNgwjY\"},\"conf
irmation_method\":\"manual\",\"created\":1742487732,\"currency\":\"usd\",\"last_payment_error
\":null,\"livemode\":false,\"metadata\":{\"manual_entry\":\"false\",\"order_id\":\"rNFXtnHLzt
geHyXiCTBSIAIjc\",\"order_transaction_id\":\"7589956682031\",\"payments_charge_id\":\"2942828
773679\",\"shop_id\":\"70387892527\",\"shop_name\":\"Developer
Store\",\"shopify_payments_next\":\"true\"},\"next_action\":null,\"payment_method\":\"pm_1R4l
rEQxz3jysvRGQJmVzg9t\",\"payment_method_types\":[\"card\"],\"source\":null,\"status\":\"requi
res_capture\"}",
        "errorCode": null,
        "status": "SUCCESS",
        "test": true,
        "parentTransaction": null,
        "processedAt": "2025-03-20T16:22:11Z",
        "maximumRefundableV2": null,
        "paymentId": "rNFXtnHLztgeHyXiCTBSIAIjc",
        "totalUnsettledSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "63.59"
          }
        }
      }
    ],
    "currentTotalDutiesSet": null,
    "originalTotalDutiesSet": null,
    "presentmentCurrencyCode": "USD",
    "currentShippingPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalShippingPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalOutstandingSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "estimatedTaxes": false,
    "currentSubtotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "59.99"
      }
    },
    "currentTotalDiscountsSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "currentTotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "63.59"
      }
    },
    "currentTotalTaxSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "3.6"
      }
    },
    "currentTotalAdditionalFeesSet": null,
    "originalTotalAdditionalFeesSet": null,
    "poNumber": null,
    "taxExempt": false
  }
}
```

Capture Authorization

If the ‘Capture Payment Upon Completing Fulfillment’ setting is set to True and the order has an authorization, the Order Update Export will capture that authorization in Shopify and import the new transaction into SalesPad.

GraphQL mutation:

None

```
mutation orderCapture($input: OrderCaptureInput!) {
  orderCapture(input: $input) {
    transaction {
      id
      amountSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      authorizationCode
      authorizationExpiresAt
      createdAt
      gateway
      paymentDetails {

        ... on CardPaymentDetails {
          avsResultCode
          bin
          cvvResultCode
          number
          company
          name
          wallet
          expirationMonth
          expirationYear
        }
      }
      kind
      receiptJson
      errorCode
      status
      test
      parentTransaction {
        id
      }
      processedAt
      maximumRefundableV2 {
        currencyCode
        amount
      }
      paymentId
      totalUnsettledSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
    }
    userErrors {
      field
      message
    }
  }
}
```

Sample response:

```
None
{
  "orderCapture": {
    "transaction": {
      "id": "gid://shopify/OrderTransaction/7589979095343",
      "amountSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "63.59"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "63.59"
        }
      },
      "authorizationCode": "pi_3R4lrEQxz3jysvRG0TuNgwjY",
      "authorizationExpiresAt": null,
      "createdAt": "2025-03-20T16:28:56Z",
      "gateway": "shopify_payments",
      "paymentDetails": {
        "avsResultCode": "Y",
        "bin": "555555",
        "cvvResultCode": "M",
        "number": "•••• •••• •••• 4444",
        "company": "Mastercard",
        "name": "Aaron Fitz",
        "wallet": null,
        "expirationMonth": 6,
        "expirationYear": 2025
      },
      "kind": "CAPTURE",
      "receiptJson":
"{\"id\":\"pi_3R4lrEQxz3jysvRG0TuNgwjY\",\"object\":\"payment_intent\",\"amount\":6359,\"amou
nt_capturable\":0,\"amount_received\":6359,\"canceled_at\":null,\"cancellation_reason\":null,
\"capture_method\":\"manual\",\"charges\":{\"object\":\"list\",\"data\":[{\"id\":\"ch_3R4lrEQ
xz3jysvRG0o3dFonF\",\"object\":\"charge\",\"amount\":6359,\"application_fee\":\"fee_1R4lxkQxz
3jysvRGsUfI24Ab\",\"balance_transaction\":{\"id\":\"txn_3R4lrEQxz3jysvRG0ofa3Lml\",\"object\"
:\"balance_transaction\",\"exchange_rate\":null},\"captured\":true,\"created\":1742487732,\"c
urrency\":\"usd\",\"failure_code\":null,\"failure_message\":null,\"fraud_details\":{},\"livem
ode\":false,\"metadata\":{\"manual_entry\":\"false\",\"order_id\":\"rNFXtnHLztgeHyXiCTBSIAIjc
\",\"order_transaction_id\":\"7589956682031\",\"payments_charge_id\":\"2942828773679\",\"shop
_id\":\"70387892527\",\"shop_name\":\"Developer
Store\",\"shopify_payments_next\":\"true\"},\"outcome\":{\"advice_code\":null,\"network_advic
e_code\":null,\"network_decline_code\":null,\"network_status\":\"approved_by_network\",\"reas
on\":null,\"risk_level\":\"normal\",\"risk_score\":11,\"seller_message\":\"Payment
complete.\",\"type\":\"authorized\"},\"paid\":true,\"payment_intent\":\"pi_3R4lrEQxz3jysvRG0T
uNgwjY\",\"payment_method\":\"pm_1R4lrEQxz3jysvRGQJmVzg9t\",\"payment_method_details\":{\"car
d\":{\"amount_authorized\":6359,\"authorization_code\":null,\"brand\":\"mastercard\",\"captur
e_before\":1745079732,\"checks\":{\"address_line1_check\":\"pass\",\"address_postal_code_chec
k\":\"pass\",\"cvc_check\":\"pass\"},\"country\":\"US\",\"description\":\"MasterCard Prepaid
Card
(Non-US)\",\"ds_transaction_id\":null,\"exp_month\":6,\"exp_year\":2025,\"extended_authorizat
ion\":{\"status\":\"enabled\"},\"fingerprint\":\"W42R39dc2U266g0i\",\"funding\":\"credit\",\"
iin\":\"555555\",\"incremental_authorization\":{\"status\":\"unavailable\"},\"installments\":
null,\"issuer\":\"CIAGROUP\",\"last4\":\"4444\",\"mandate\":null,\"moto\":true,\"multicapture
\":{\"status\":\"unavailable\"},\"network\":\"mastercard\",\"network_token\":{\"used\":false}
,\"network_transaction_id\":\"MCCJTPOW60320\",\"overcapture\":{\"maximum_amount_capturable\":
6359,\"status\":\"unavailable\"},\"overcapture_supported\":false,\"payment_account_reference\
":\"0zGFFYNQCX24uQFJVlH3Xbf1yJ6Le\",\"regulated_status\":\"unregulated\",\"three_d_secure\":n
ull,\"wallet\":null},\"type\":\"card\"},\"refunded\":false,\"source\":null,\"status\":\"succe
eded\",\"mit_params\":{\"network_transaction_id\":\"MCCJTPOW60320\"}}],\"has_more\":false,\"t
otal_count\":1,\"url\":\"\\/v1\\/charges?payment_intent=pi_3R4lrEQxz3jysvRG0TuNgwjY\"},\"conf
irmation_method\":\"manual\",\"created\":1742487732,\"currency\":\"usd\",\"last_payment_error
\":null,\"livemode\":false,\"metadata\":{\"manual_entry\":\"false\",\"order_id\":\"rNFXtnHLzt
geHyXiCTBSIAIjc\",\"order_transaction_id\":\"7589956682031\",\"payments_charge_id\":\"2942828
773679\",\"shop_id\":\"70387892527\",\"shop_name\":\"Developer
Store\",\"shopify_payments_next\":\"true\",\"transaction_fee_total_amount\":\"214\"},\"next_a
ction\":null,\"payment_method\":\"pm_1R4lrEQxz3jysvRGQJmVzg9t\",\"payment_method_types\":[\"c
ard\"],\"source\":null,\"status\":\"succeeded\"}",
      "errorCode": null,
      "status": "SUCCESS",
      "test": true,
      "parentTransaction": {
        "id": "gid://shopify/OrderTransaction/7589956682031"
      },
      "processedAt": "2025-03-20T16:28:57Z",
      "maximumRefundableV2": null,
      "paymentId": "#1166.2",
      "totalUnsettledSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      }
    },
    "userErrors": []
  }
}
```

Order Invoiced Export

Overview

The Order Invoiced Export can run after a Shopify-linked sales document is transferred to an invoice. This component can perform various updates such as capturing an authorization or closing the order. It targets specific workflow queues, and it forwards invoices out of those queues once processing is complete.

This component is available as of SalesPad 5.6.2.

General Settings

Capture Payment – *Trigger Shopify to capture the pending Shopify authorization once the SalesPad order is fully invoiced. Defaults to 'False'.*

Close Website Order – *Close the Shopify order once the SalesPad order is fully invoiced. Defaults to 'False'.*

Close Website Order Financial Status Filter – *Specify one or more financial statuses which a Shopify order must have in order to close the order. This setting will be ignored if it's blank or if Close Website Order is set to 'False'. Defaults to 'Paid'.*

Invoice Export Failure Queue – *In the event of an unsuccessful export, the document will be placed into this queue.*

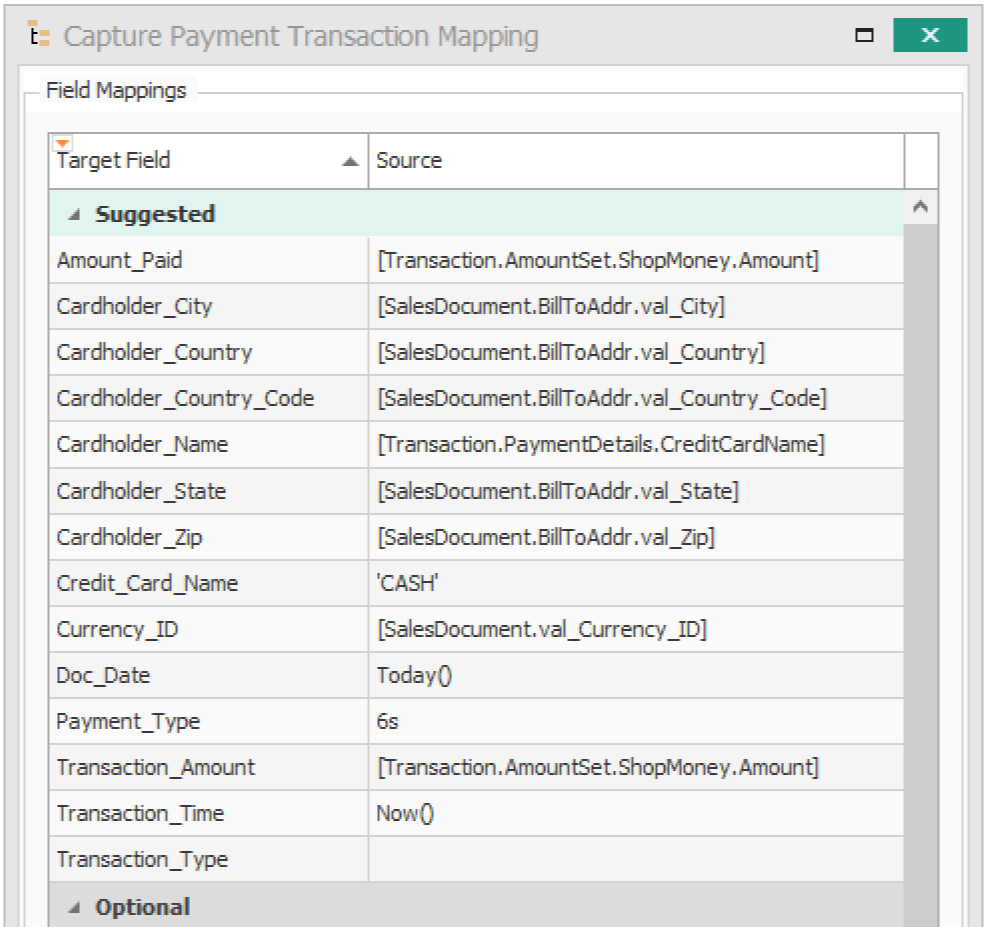
Invoice Export Queue – *Queues that contain invoices ready to be exported to Shopify. This component will ignore invoices which do not have an External ID, but will assume that any invoices which have an External ID belong to the linked Shopify store even if the corresponding order originated from a different automation instance.*

Invoice Notify – Send an invoice to the customer. Defaults to 'False'.

Invoice Notify Financial Status Filter – *Specify one or more financial statuses which a Shopify order must have in order to send an invoice. This setting will be ignored if it's blank or if Invoice Notify is set to 'False'. Defaults to 'Pending'.*

Assignment Settings

Capture Payment Transaction Mapping – *Define the mappings to be used when creating a Sales Document Payment after the Order Invoiced Export captures an authorization.*



Target Field	Source
Suggested	
Amount_Paid	[Transaction.AmountSet.ShopMoney.Amount]
Cardholder_City	[SalesDocument.BillToAddr.val_City]
Cardholder_Country	[SalesDocument.BillToAddr.val_Country]
Cardholder_Country_Code	[SalesDocument.BillToAddr.val_Country_Code]
Cardholder_Name	[Transaction.PaymentDetails.CreditCardName]
Cardholder_State	[SalesDocument.BillToAddr.val_State]
Cardholder_Zip	[SalesDocument.BillToAddr.val_Zip]
Credit_Card_Name	'CASH'
Currency_ID	[SalesDocument.val_Currency_ID]
Doc_Date	Today()
Payment_Type	6s
Transaction_Amount	[Transaction.AmountSet.ShopMoney.Amount]
Transaction_Time	Now()
Transaction_Type	
Optional	

Invoice Notify Email Mapping - *Define the mappings to be used when sending an invoice.*

This setting can be left blank, in which case Shopify will compose the email based on order and customer data.

Processing

The Order Invoiced Export is capable of doing 3 different things depending on settings and the state of the Shopify order.

If the *Capture Payment* setting is enabled, the SalesPad order is fully invoiced, and the Shopify order has an authorization, the component will capture that authorization. If successful, the component will write back the captured transaction to the SalesPad invoice.

If the *Invoice Notify* setting is enabled and the *Invoice Notify Financial Status Filter* setting is either blank or contains the order's current financial status, Shopify will send an invoice reminder to the customer.

If the *Close Website Order* setting is enabled and the *Close Website Order Financial Status Filter* setting is either blank or contains the order's current financial status, the component will close the Shopify order. Note that this setting is redundant with Shopify's *Automatically archive the order* setting, and it may make more sense to use the Shopify setting.

Finally, the Order Invoiced Export will forward the invoice in workflow if there were no processing errors; otherwise, it will move the invoice to the workflow queue in the *Invoice Export Failure Queue* setting. If the export did not update the Shopify order in any way due to settings or the order's status, the invoice will still forward and the export will still be reported as successful.

Endpoints

Get Order

The Order Invoiced Export first loads the order from Shopify.

GraphQL query:

None

```
query getOrder($id: ID!) {
  order(id: $id) {
    legacyResourceId
    id
    billingAddress {
      id
      address1
      address2
      city
      company
      country
      countryCodeV2
      firstName
      lastName
      latitude
      longitude
      name
      phone
      province
      provinceCode
      zip
    }
    customer {
      legacyResourceId
      id
      email
      firstName
      lastName
    }
    email
    displayFinancialStatus
    displayFulfillmentStatus
    name
    shippingAddress {
      id
      address1
      address2
      city
      company
      country
      countryCodeV2
      firstName
      lastName
      latitude
      longitude
      name
      phone
      province
      provinceCode
      zip
    }
    transactions {
      id
      amountSet {
        shopMoney {
          currencyCode
          amount
```

```
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
authorizationCode
authorizationExpiresAt
createdAt
gateway
paymentDetails {

  ... on CardPaymentDetails {
    avsResultCode
    bin
    cvvResultCode
    number
    company
    name
    wallet
    expirationMonth
    expirationYear
  }
}
kind
receiptJson
errorCode
status
test
parentTransaction {
  id
}
processedAt
maximumRefundableV2 {
  currencyCode
  amount
}
```

```
        paymentId
        totalUnsettledSet {
          shopMoney {
            currencyCode
            amount
          }
          presentmentMoney {
            currencyCode
            amount
          }
        }
      }
      currentShippingPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      totalOutstandingSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      currentSubtotalPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      currentTotalDiscountsSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      currentTotalPriceSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
      currentTotalTaxSet {
        shopMoney {
          currencyCode
          amount
        }
        presentmentMoney {
          currencyCode
          amount
        }
      }
    }
  }
```

Sample response:

None

```
{
  "order": {
    "legacyResourceId": "6378807951663",
    "id": "gid://shopify/Order/6378807951663",
    "billingAddress": {
      "id": "gid://shopify/MailingAddress/20125831168303?model_name=Address",
      "address1": "11403 45 St. South",
      "address2": null,
      "city": "Chicago",
      "company": "Aaron Fitz Electrical",
      "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Bob",
      "lastName": "Fitz",
      "latitude": 41.813099,
      "longitude": -87.6115503,
      "name": "Bob Fitz",
      "phone": null,
      "province": "Illinois",
      "provinceCode": "IL",
      "zip": "60603-0776"
    },
    "customer": {
      "legacyResourceId": "8943179104559",
      "id": "gid://shopify/Customer/8943179104559",
      "email": "aaron.fitz@aaronfitzelectrical.net",
      "firstName": "Aaron",
      "lastName": "Fitz"
    },
    "email": "aaron.fitz@aaronfitzelectrical.net",
    "displayFinancialStatus": "AUTHORIZED",
    "displayFulfillmentStatus": "FULFILLED",
    "name": "#1298",
    "shippingAddress": {
      "id": "gid://shopify/MailingAddress/20125831135535?model_name=Address",
      "address1": "11403 45 St. South",
      "address2": null,
      "city": "Chicago",
      "company": "Aaron Fitz Electrical",
      "country": "United States",
      "countryCodeV2": "US",
      "firstName": "Bob",
      "lastName": "Fitz",
      "latitude": 41.813099,
      "longitude": -87.6115503,
      "name": "Bob Fitz",
      "phone": null,
      "province": "Illinois",
      "provinceCode": "IL",
      "zip": "60603-0776"
    },
    "transactions": [
      {
        "id": "gid://shopify/OrderTransaction/7798302966063",
        "amountSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "40.0"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "40.0"
          }
        },
        "authorizationCode": "pi_3RiIfwQxz3jysvRG0LabmmRi",
        "authorizationExpiresAt": "2025-07-14T17:17:57Z",
        "createdAt": "2025-07-07T17:17:56Z",
        "gateway": "shopify_payments",
        "paymentDetails": {
          "avsResultCode": "Y",
          "bin": "555555",
          "cvvResultCode": "M",
          "number": "•••• •••• •••• 4444",
          "company": "Mastercard",
          "name": "Aaron Fitz",
          "wallet": null,
          "expirationMonth": 5,
          "expirationYear": 2026
        },
        "kind": "AUTHORIZATION",
        "receiptJson": [redacted],
        "errorCode": null,
        "status": "SUCCESS",
        "test": true,
        "parentTransaction": null,
        "processedAt": "2025-07-07T17:17:56Z",
        "maximumRefundableV2": null,
        "paymentId": "rVT7zdEpehodozez6aGnbURTl",
        "totalUnsettledSet": {
          "shopMoney": {
            "currencyCode": "USD",
            "amount": "40.0"
          },
          "presentmentMoney": {
            "currencyCode": "USD",
            "amount": "40.0"
          }
        }
      }
    ],
    "currentShippingPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "totalOutstandingSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "currentSubtotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "40.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "40.0"
      }
    },
    "currentTotalDiscountsSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    },
    "currentTotalPriceSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "40.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "40.0"
      }
    },
    "currentTotalTaxSet": {
      "shopMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      },
      "presentmentMoney": {
        "currencyCode": "USD",
        "amount": "0.0"
      }
    }
  }
}
```

Capture Authorization

The Order Invoiced Export can capture authorizations. If it does so, it will also reload the Shopify order in order to have the order's updated financial status.

GraphQL mutation:

```
None
{
  "query": "
                    mutation orderCapture($input: OrderCaptureInput!) {
                      orderCapture(input: $input) {
                        transaction {
                          id
                          amountSet {
                            shopMoney {
                              currencyCode
                              amount
  }
  presentmentMoney {
    currencyCode
    amount
  }
}
authorizationCode
authorizationExpiresAt
createdAt
gateway
paymentDetails {

  ... on CardPaymentDetails {
    avsResultCode
    bin
    cvvResultCode
    number
    company
    name
    wallet
    expirationMonth
    expirationYear
  }
}
kind
receiptJson
errorCode
status
test
parentTransaction {
  id
}
processedAt
maximumRefundableV2 {
  currencyCode
  amount
}
```

```
                          paymentId
                          totalUnsettledSet {
                            shopMoney {
                              currencyCode
                              amount
                            }
                            presentmentMoney {
                              currencyCode
                              amount
                            }
                          }
                        }
                        userErrors {
                          field
                          message
                        }
                      }
                    }",
  "variables": {
    "input": {
      "amount": 40.0,
      "currency": "USD",
      "finalCapture": true,
      "id": "gid://shopify/Order/6378807951663",
      "parentTransactionId": "gid://shopify/OrderTransaction/7798302966063"
    }
  }
}
```

Sample response:

```
None
{
  "orderCapture": {
    "transaction": {
```

```
"id": "gid://shopify/OrderTransaction/7798339141935",
"amountSet": {
  "shopMoney": {
    "currencyCode": "USD",
    "amount": "40.0"
  },
  "presentmentMoney": {
    "currencyCode": "USD",
    "amount": "40.0"
  }
},
"authorizationCode": "pi_3RiIfwQxz3jysvRG0LabmmRi",
"authorizationExpiresAt": null,
"createdAt": "2025-07-07T17:35:19Z",
"gateway": "shopify_payments",
"paymentDetails": {
  "avsResultCode": "Y",
  "bin": "555555",
  "cvvResultCode": "M",
  "number": "•••• •••• •••• 4444",
  "company": "Mastercard",
  "name": "Aaron Fitz",
  "wallet": null,
  "expirationMonth": 5,
  "expirationYear": 2026
},
"kind": "CAPTURE",
"receiptJson": [redacted],
"errorCode": null,
"status": "SUCCESS",
"test": true,
"parentTransaction": {
  "id": "gid://shopify/OrderTransaction/7798302966063"
},
"processedAt": "2025-07-07T17:35:20Z",
"maximumRefundableV2": null,
"paymentId": "#1298.2",
      "totalUnsettledSet": {
        "shopMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        },
        "presentmentMoney": {
          "currencyCode": "USD",
          "amount": "0.0"
        }
      }
    },
    "userErrors": []
  }
}
```

Send Invoice

The Order Invoiced Export can send an invoice to the customer.

GraphQL mutation:

```
None
{
  "query": "
                    mutation orderInvoiceSend($orderId: ID!, $email: EmailInput) {
                      orderInvoiceSend(id: $orderId, email: $email) {
                        userErrors {
                          field
                          message
                        }
                      }
                    }",
  "variables": {
    "orderId": "gid://shopify/Order/6378807951663",
    "email": {
      "to": null,
      "from": null,
      "bcc": null,
      "subject": null,
      "customMessage": null
    }
  }
}
```

Close Order

The Order Invoiced Export can close orders.

GraphQL mutation:

```
None
mutation orderClose($input: OrderCloseInput!) {
  orderClose(input: $input) {
    userErrors {
      field
      message
    }
  }
}
```

Order Voided Export

Overview

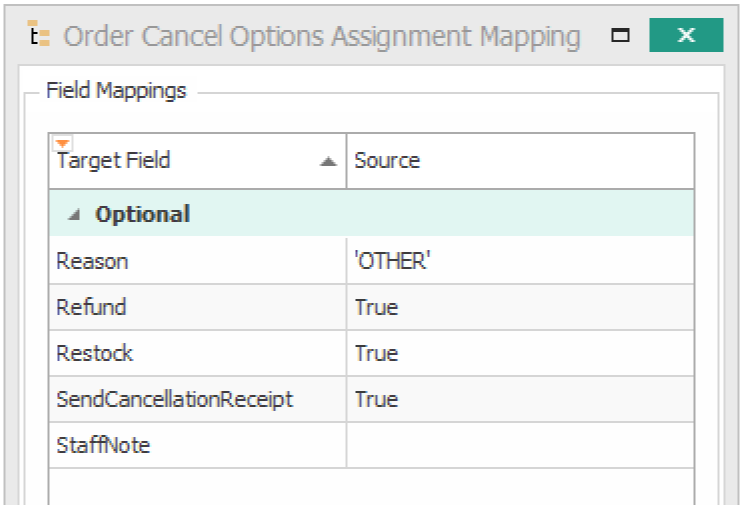
The Order Voided Export component is used to automatically cancel orders in Shopify after they have been voided in SalesPad.

General Settings

Number Of Days To Look Back - *Specify the number of days to look back from today to export voided orders. For example, set to 30 to export voided orders only from the last 30 days. Set to zero to export voided orders from any time. Defaults to '30'.*

Assignment Settings

Order Cancel Options Assignment Mapping - *Define the mappings to be used when canceling an order.*



Target Field	Source
Optional	
Reason	'OTHER'
Refund	True
Restock	True
SendCancellationReceipt	True
StaffNote	

Processing

The Order Voided Export loads SalesPad orders that have been voided in the timeframe defined in the *Number Of Days To Look Back* setting and are linked to a Shopify order. For each order, if it has not already been voided in Shopify, then the *Order Cancel Options Assignment Mapping* setting is used to cancel it in Shopify.

Endpoints

GraphQL mutation:

```
None
mutation orderCancel($notifyCustomer: Boolean, $orderId: ID!, $reason: OrderCancelReason!,
$refund: Boolean!, $restock: Boolean!, $staffNote: String) {
  orderCancel(notifyCustomer: $notifyCustomer, orderId: $orderId, reason: $reason, refund:
$refund, restock: $restock, staffNote: $staffNote) {
    userErrors {
      field
      message
    }
  }
}
```

GraphQL and SalesPad Model Differences

Matching, mapping, and script settings often show a list of Shopify fields. These fields do not always use the same names as Shopify's GraphQL API, and they are sometimes located in different places within the model. This is because SalesPad uses a more static version of Shopify's model in order to reduce the number of breaking changes between SalesPad versions.

This section lists the field mappings for the entities which have the most notable differences between SalesPad's Shopify model and Shopify's GraphQL model.

Order Import, Order Update Export, and Order Voided Export

Order

SalesPad Field	**Shopify GraphQL Field**
Id	legacyResourceId
AdminGraphQLAPIId	id
BillingAddress	billingAddress
BrowserIp	clientIp
BuyerAcceptsMarketing	customerAcceptsMarketing
CancelReason	cancelReason
CancelledAt	cancelledAt
ClosedAt	closedAt

Confirmed	confirmed
CreatedAt	createdAt
Currency	currencyCode
Customer	customer
CustomerLocale	customerLocale
DiscountApplications	discountApplications
Email	email
FinancialStatus	displayFinancialStatus
FulfillmentStatus	displayFulfillmentStatus
Phone	phone
Tags	tags
LineItems	lineItems
LocationId	retailLocation.legacyResourceId
Name	name
Note	note
NoteAttributes	customAttributes
OrderStatusUrl	statusPageUrl
PaymentGatewayNames	paymentGatewayNames
ProcessedAt	processedAt
ShippingAddress	shippingAddress
ShippingLine	shippingLine
SourceIdentifier	sourceIdentifier
SourceName	sourceName
SubtotalPrice	subtotalPriceSet.shopMoney.amount
TaxLines	taxLines
TaxesIncluded	taxesIncluded

Test	test
TotalDiscounts	totalDiscountsSet.shopMoney.amount
TotalLineItemsPrice	subtotalPriceSet.shopMoney.amount
TotalTipReceived	totalTipReceivedSet.shopMoney.amount
TotalPrice	totalPriceSet.shopMoney.amount
TotalTax	totalTaxSet
TotalWeight	totalWeight
UpdatedAt	updatedAt
Transactions	transactions
CurrentTotalDutiesSet	currentTotalDutiesSet
OriginalTotalDutiesSet	originalTotalDutiesSet
PresentmentCurrency	presentmentCurrencyCode
TotalLineItemsPriceSet	subtotalPriceSet
TotalDiscountsSet	totalDiscountsSet
CurrentShippingPriceSet	currentShippingPriceSet
TotalShippingPriceSet	totalShippingPriceSet
SubtotalPriceSet	subtotalPriceSet
TotalPriceSet	totalPriceSet
TotalOutstanding	totalOutstandingSet.shopMoney.amount
TotalTaxSet	totalTaxSet
EstimatedTaxes	estimatedTaxes
CurrentSubtotalPriceSet	currentSubtotalPriceSet
CurrentTotalDiscounts	currentTotalDiscountsSet.shopMoney.amount
CurrentTotalDiscountsSet	currentTotalDiscountsSet
CurrentTotalPrice	currentTotalPriceSet.shopMoney.amount

CurrentTotalPriceSet	currentTotalPriceSet
CurrentTotalTax	currentTotalTaxSet.shopMoney.amount
CurrentTotalTaxSet	currentTotalTaxSet
PaymentTerms	paymentTerms
CurrentTotalAdditionalFeesSet	currentTotalAdditionalFeesSet
OriginalTotalAdditionalFeesSet	originalTotalAdditionalFeesSet
PoNumber	poNumber
TaxExempt	taxExempt
TotalTipReceivedSet	totalTipReceivedSet
TotalOutstandingSet	totalOutstandingSet

Line Item

SalesPad Field	**Shopify GraphQL Field**
Id	id (number portion only)
AdminGraphQLAPIId	id
FulfillableQuantity	unfulfilledQuantity
Price	originalUnitPriceSet.shopMoney.amount
ProductId	product.legacyResourceId
CurrentQuantity	currentQuantity
Quantity	quantity
RequiresShipping	requiresShipping
SKU	sku
Title	title
VariantId	variant.legacyResourceId
VariantGlobalId	variant.id
VariantTitle	variantTitle
Name	name

Vendor	vendor
GiftCard	isGiftCard
Taxable	taxable
TaxLines	taxLines
TotalDiscount	totalDiscountSet.shopMoney.amount
TotalDiscountSet	totalDiscountSet
Properties	customAttributes
DiscountAllocations	discountAllocations
ProductExists	‘True’ if product is not null
PriceSet	originalUnitPriceSet
Duties	duties

Transaction

SalesPad Field	**Shopify GraphQL Field**
Id	id (number portion only)
AdminGraphQLAPIId	id
Amount	amountSet.shopMoney.amount
Authorization	authorizationCode
AuthorizationExpiresAt	authorizationExpiresAt
CreatedAt	createdAt
Gateway	gateway
Kind	kind
Receipt	receiptJson
ErrorCode	errorCode
Status	status
Test	test
Currency	amount.presentmentMoney.currencyCode

ParentId	parentTransaction.id (number portion only)
ProcessedAt	processedAt
MaximumRefundable	maximumRefundableV2.amount
PaymentId	paymentId
TotalUnsettledSet	totalUnsettledSet
AmountSet	amountSet

Product Export and Product Image Export

Product

SalesPad's Shopify model and Shopify's GraphQL model are currently identical for products. This includes SalesPad's Id field including the "gid://shopify/Product/" portion of the Id.

Product Variant

SalesPad Field	**Shopify GraphQL Field**
Id	id
Barcode	barcode
CompareAtPrice	compareAtPrice
Price	price
TaxCode	taxCode
Taxable	taxable
Sku	inventoryItem.sku
Tracked	inventoryItem.tracked
Weight	inventoryItem.measurement.weight.value
WeightUnit	inventoryItem.measurement.weight.unit

Other Export Components

The other export components do not have notable differences between the SalesPad and Shopify models.