CAVALLO

SalesPad

Shopify Integration

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.

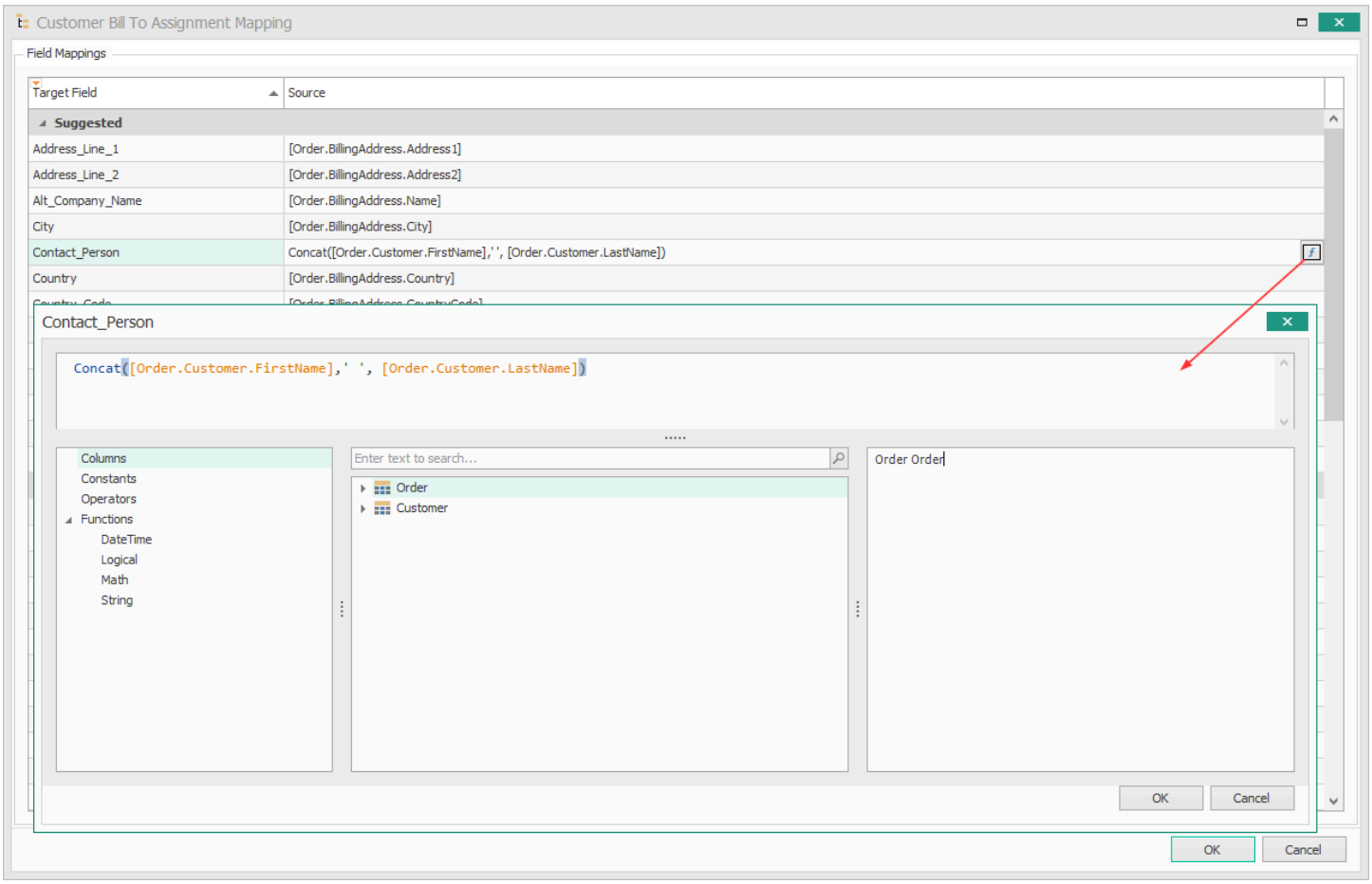
General Information On Expressions

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.

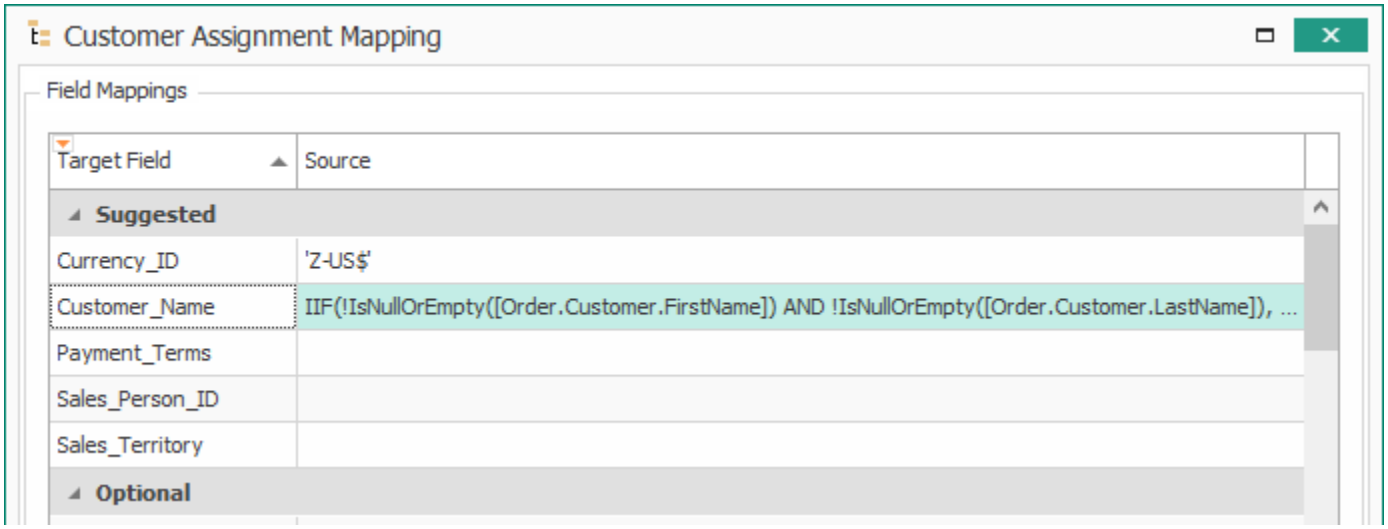
Each expression editor will have a set of source objects to match or map from, and a set of destination objects to be matched or populated. These editors allow additional complexity beyond simply mapping source object fields to destination object fields.

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.

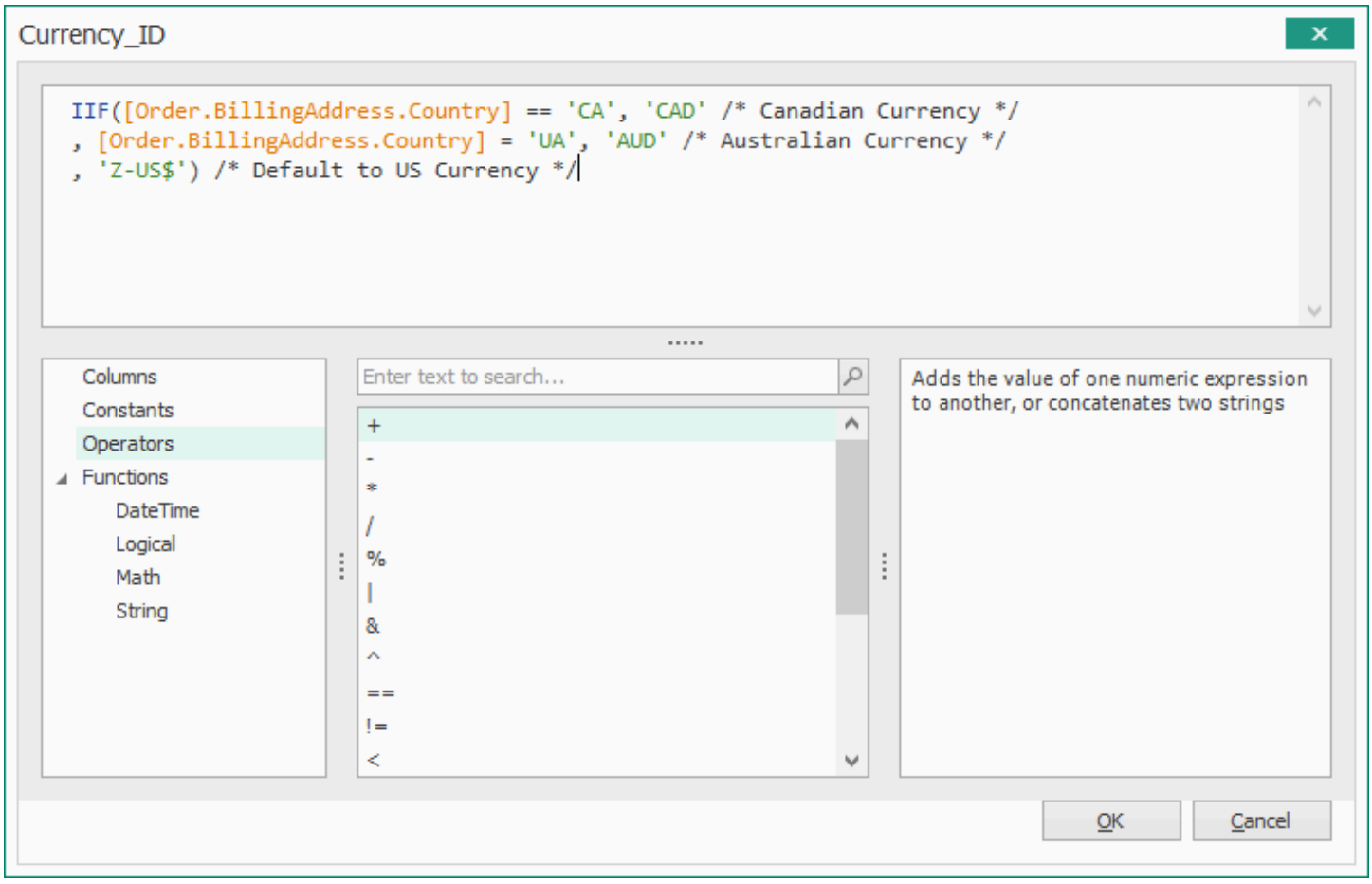


Rather than a single customer name field with the customer's full name, Shopify has separate first and last name fields, so the Contact_Person field cannot be populated by simply mapping one field to another. The Concat function is used to append the Shopify Customer last name to the first name, with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



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.

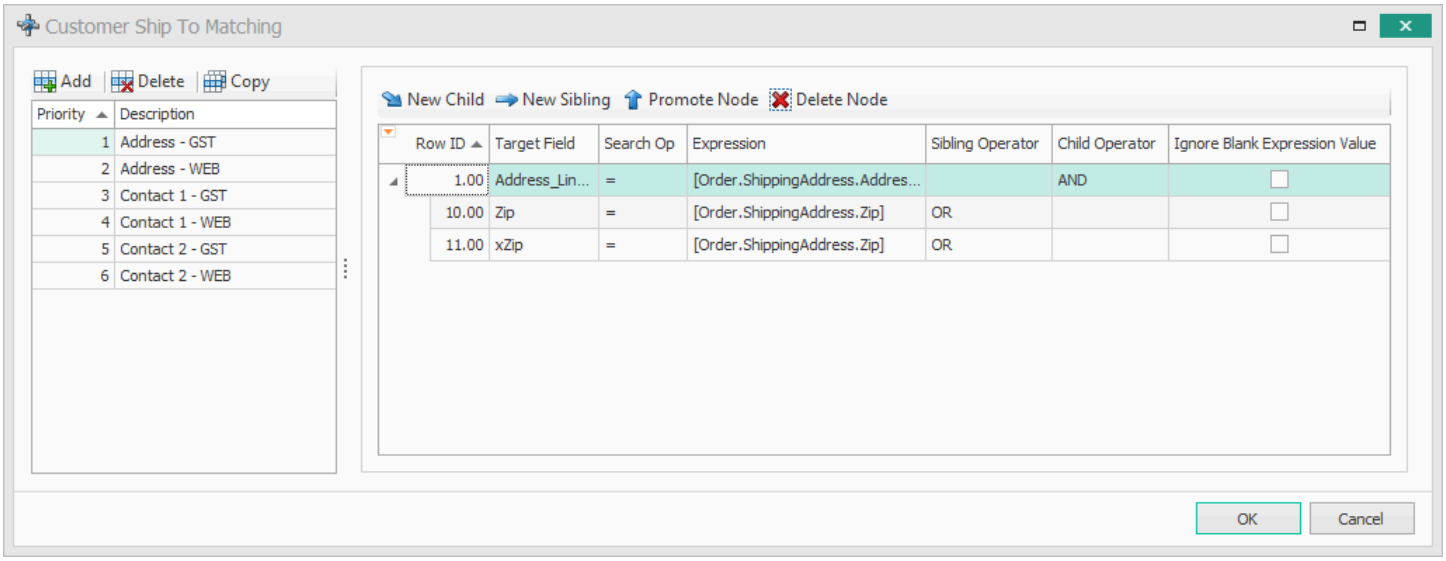


The complete expression language documentation can be found here.

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:



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.

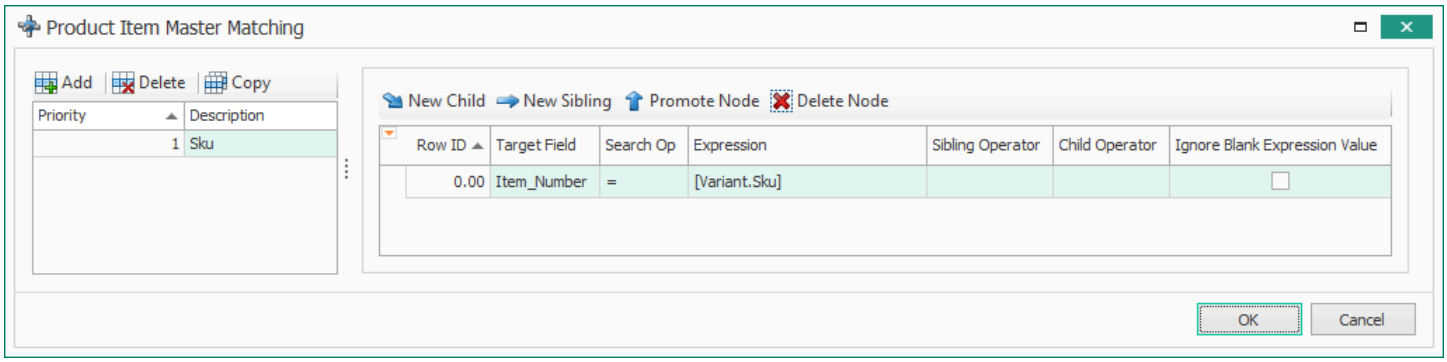
General Settings

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'.*

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

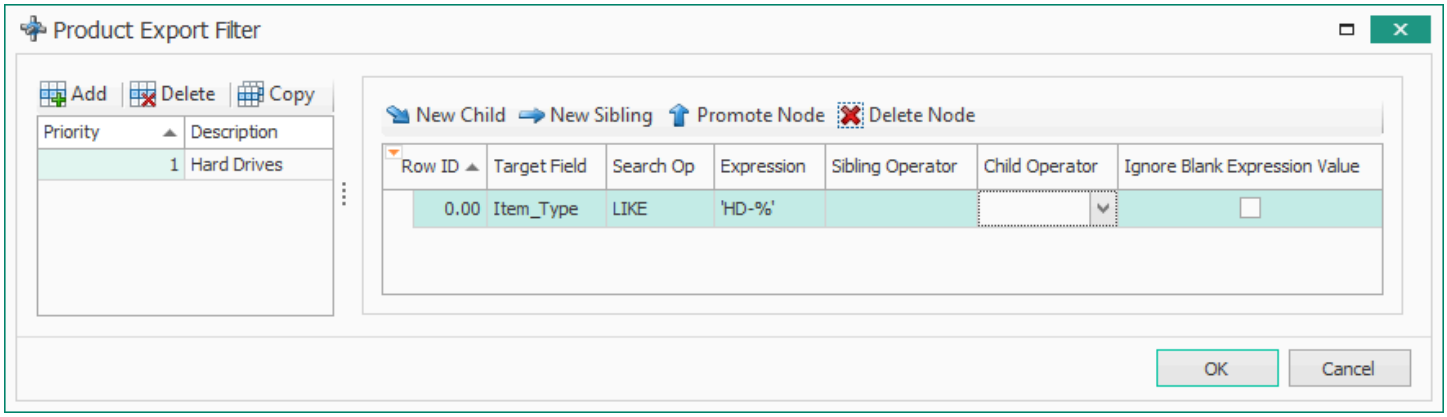
Product Item Master Matching - *Define the mappings for looking up the Item Master for the Product Export.*



Lookup Settings

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

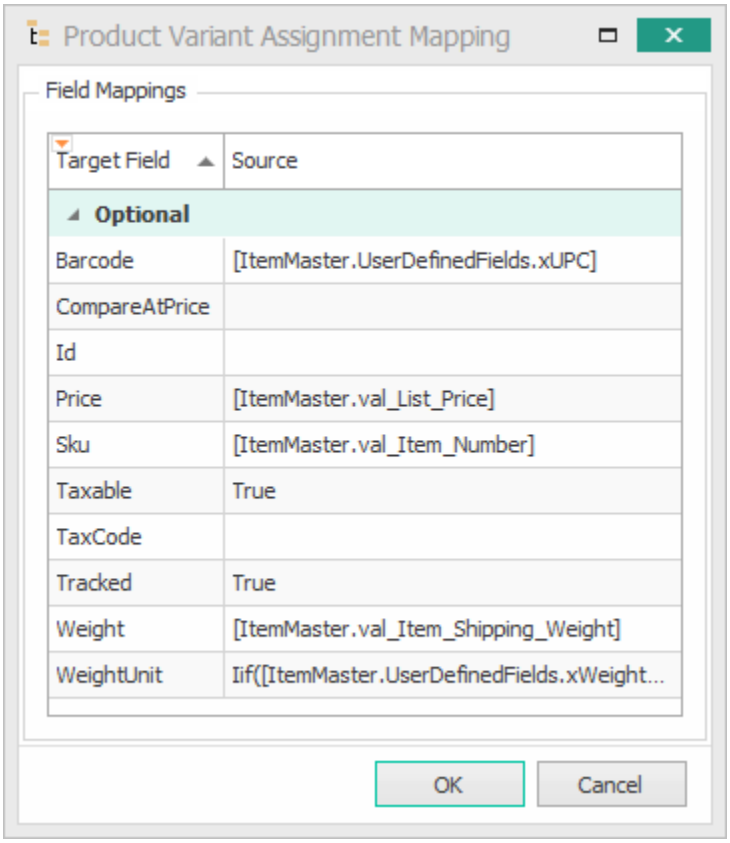
Only one priority row should be added to the left grid. The conditions on the right will be used to determine which Item Masters will be exported to Shopify. In the example below, the system will export all items that start with *HD-*.



Assignment Settings

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.*



Scripts

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

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*.

Endpoints

Get Products

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

GraphQL query:

```
Unset
{
  "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:

Unset

```
{"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":"4
5.00","taxCode":"","taxable":true,"inventoryItem":{"sku":"32
SDRAM","tracked":true,"measurement":{"weight":{"value":0.0,"unit":"POUNDS"}}},"__parentId":"gid:\/\/shopi
fy\/Product\/8061404938543"}
{"id":"gid:\/\/shopify\/Product\/8061404971311","descriptionHtml":"256 meg
SDRAM","handle":"256-sdram","productType":"","status":"ACTIVE","tags":[],"templateSuffix":null,"title":"2
56 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:

```
Unset
{
  "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:

```
Unset
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:

```
Unset
mutation productCreate($product: ProductCreateInput!) {
  productCreate(product: $product) {
    product {
```

```
      id
      variants (first: 1) {
        nodes {
          id
        }
      }
    }
    userErrors {
      field
      message
    }
  }
}
```

Sample response:

Unset


```
{
  "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:

```
Unset
mutation productDelete($input: ProductDeleteInput!) {
  productDelete(input: $input) {
    userErrors {
      field
      message
    }
  }
}
```

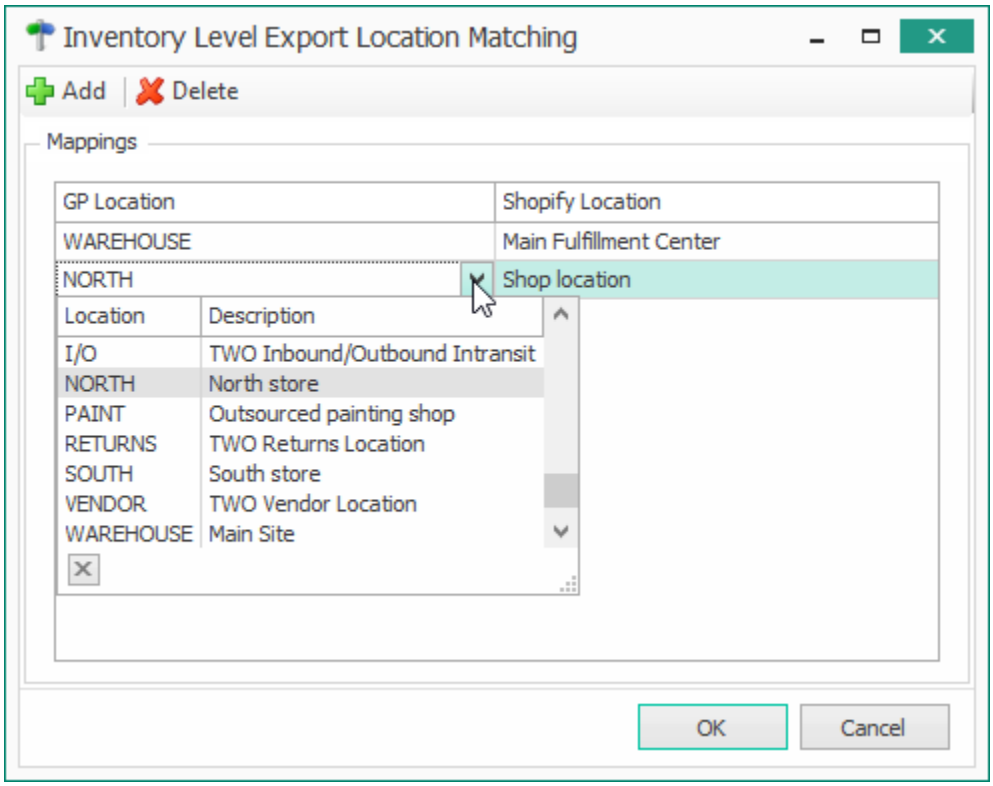
Inventory Level Export

Overview

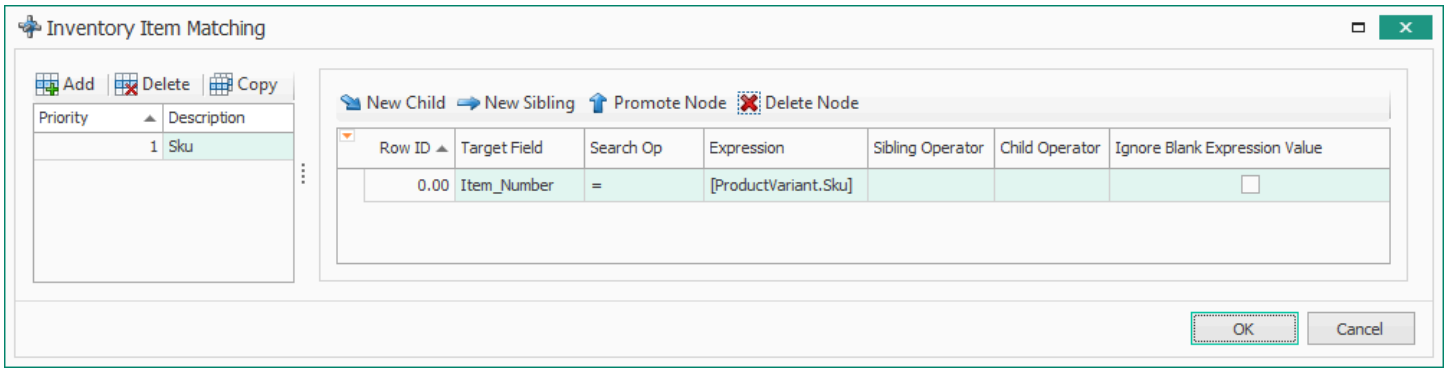
The Inventory Level Export updates item availability information in Shopify based on current inventory levels in SalesPad warehouses. This ensures that customers placing orders on the website know how much they can order before they will have to wait for backorders.

Matching Settings

Inventory 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.*

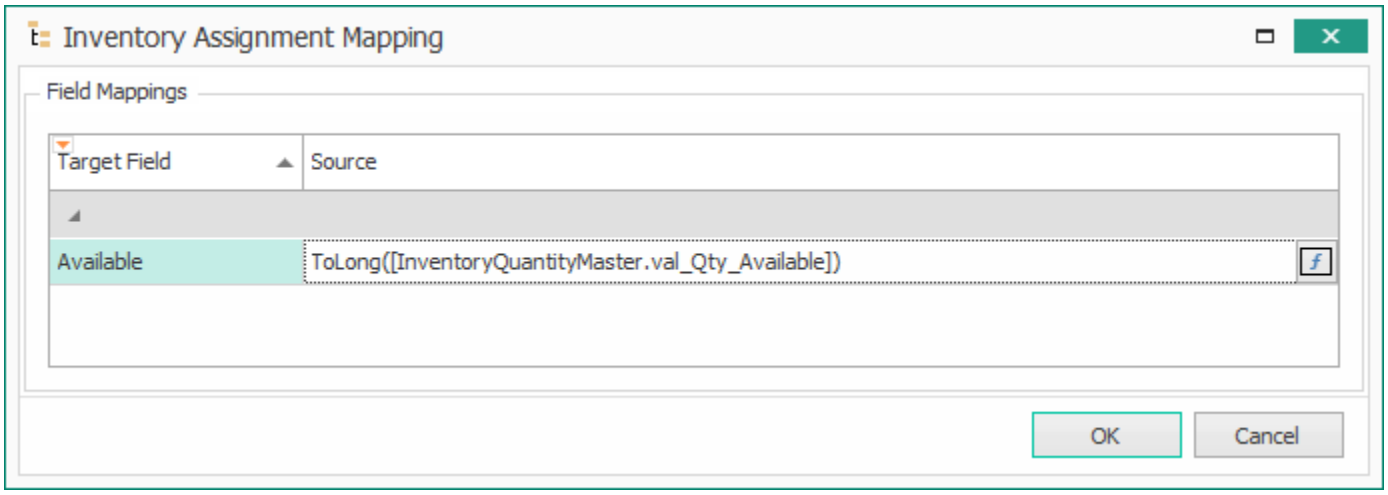


Inventory Item Matching - *Define the criteria for which GP inventory levels to export to Shopify.*



Assignment Settings

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



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:

```
Unset
{
  "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:

Unset


```
{"id":"gid:\/\/shopify\/ProductVariant\/44263493435695","sku":"32
SDRAM","inventoryItem":{"id":"gid:\/\/shopify\/InventoryItem\/46311720157487","legacyResourceId":"4631172
0157487"}}
{"id":"gid:\/\/shopify\/InventoryLevel\/112397975855?inventory_item_id=46311720157487","location":{"id":"
gid:\/\/shopify\/Location\/76014846255","legacyResourceId":"76014846255"},"quantities":[{"name":"availabl
e","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":"availabl
e","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":"availabl
e","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":"availabl
e","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":"availabl
e","quantity":413}],"__parentId":"gid:\/\/shopify\/ProductVariant\/44263493435695"}
{"id":"gid:\/\/shopify\/ProductVariant\/44263493468463","sku":"256
SDRAM","inventoryItem":{"id":"gid:\/\/shopify\/InventoryItem\/46311720190255","legacyResourceId":"4631172
0190255"}}
{"id":"gid:\/\/shopify\/InventoryLevel\/112397975855?inventory_item_id=46311720190255","location":{"id":"
gid:\/\/shopify\/Location\/76014846255","legacyResourceId":"76014846255"},"quantities":[{"name":"availabl
e","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":"availabl
e","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":"availabl
e","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":"availabl
e","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":"availabl
e","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:

```
Unset
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:

```
Unset
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'.*

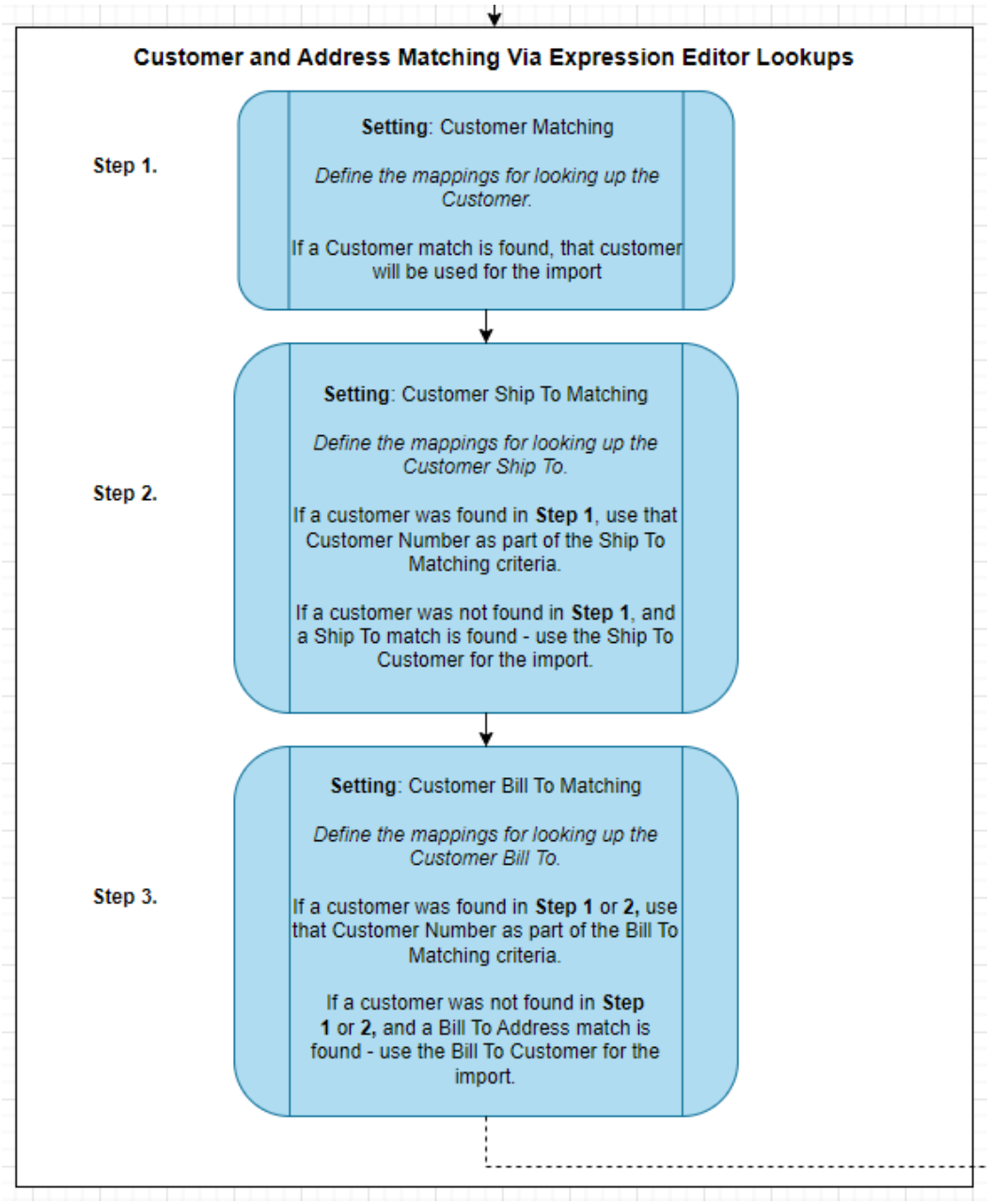
Roll Back Order Import Transaction On Error - *When enabled, the transaction encompassing the order import will be rolled back when an error occurs. This prevents data from a partially completed import from being saved to the database. Defaults to 'True'.*

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.

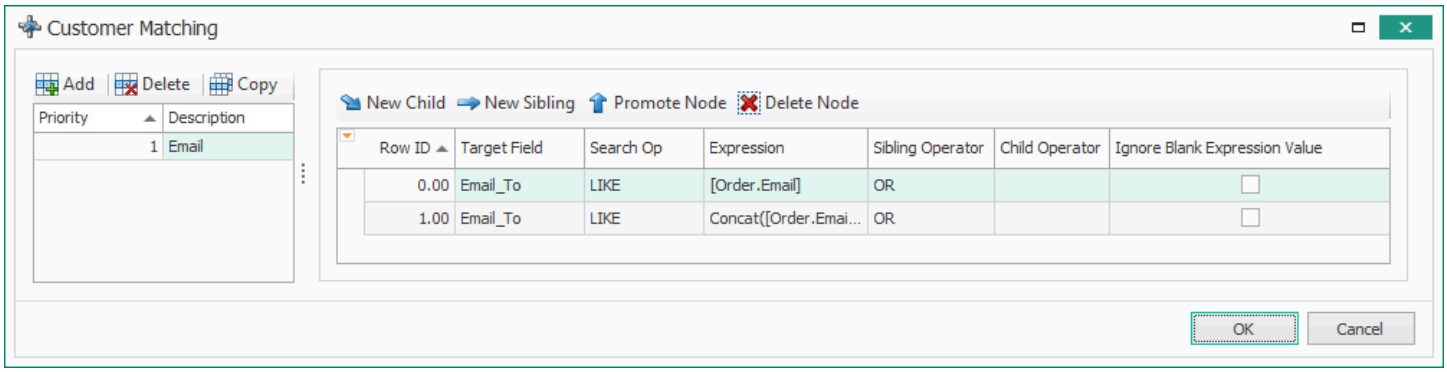


Customer Matching - *Define the mappings for looking up the Customer.*

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;%'



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 Matching* 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.*

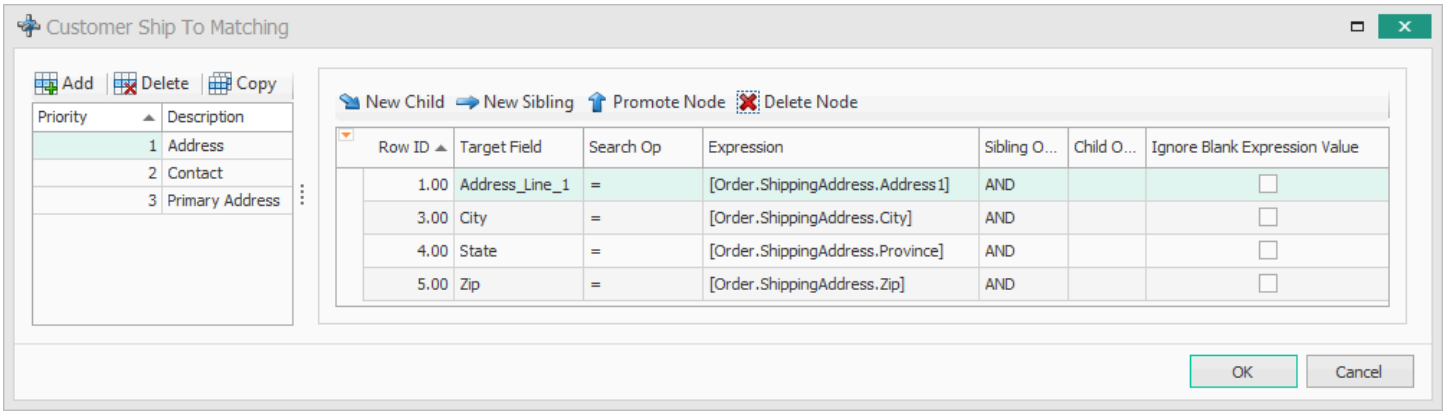
This setting determines how SalesPad will attempt to look up the Ship To Address for an incoming Shopify order. If a SalesPad Customer was matched via the **Customer Matching** setting, the Ship To Address must also belong to that customer. Otherwise, the search will consider address codes across all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

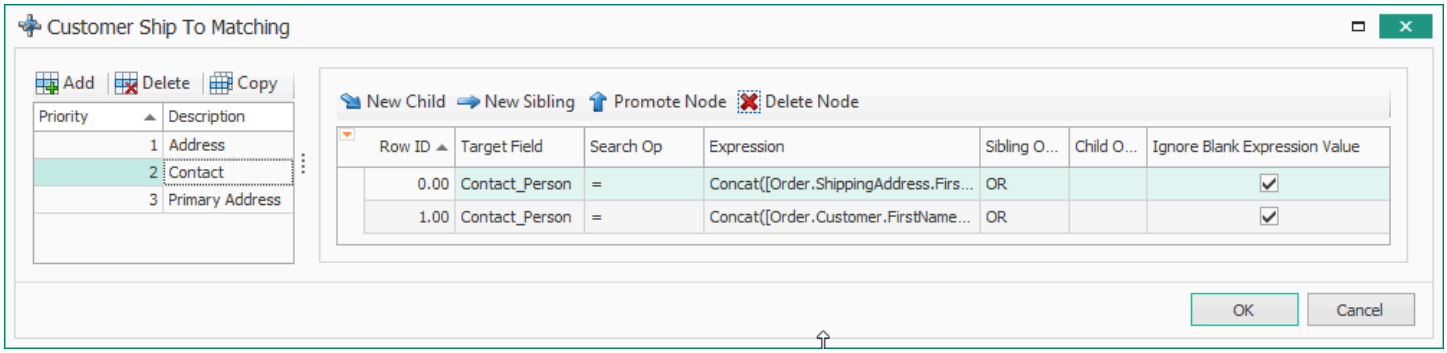
Priority 1

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



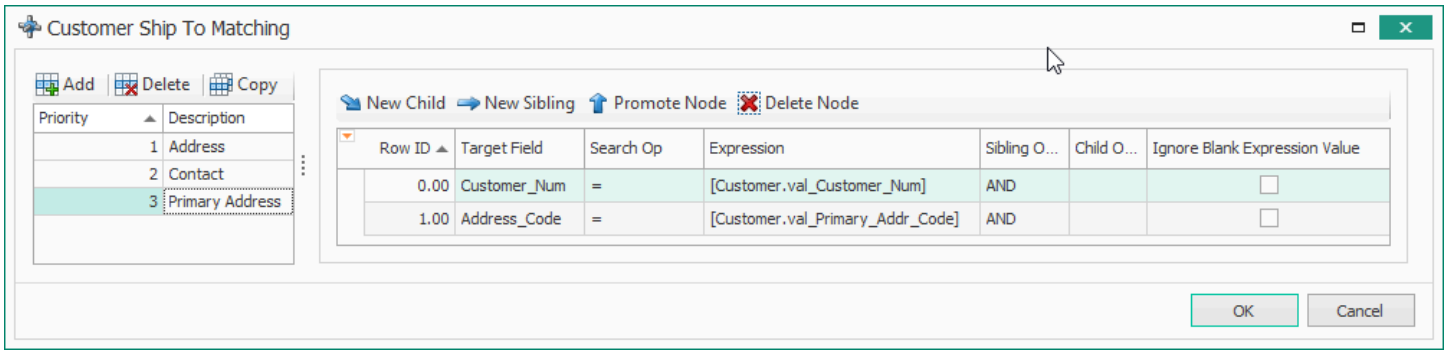
Priority 2

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



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.



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

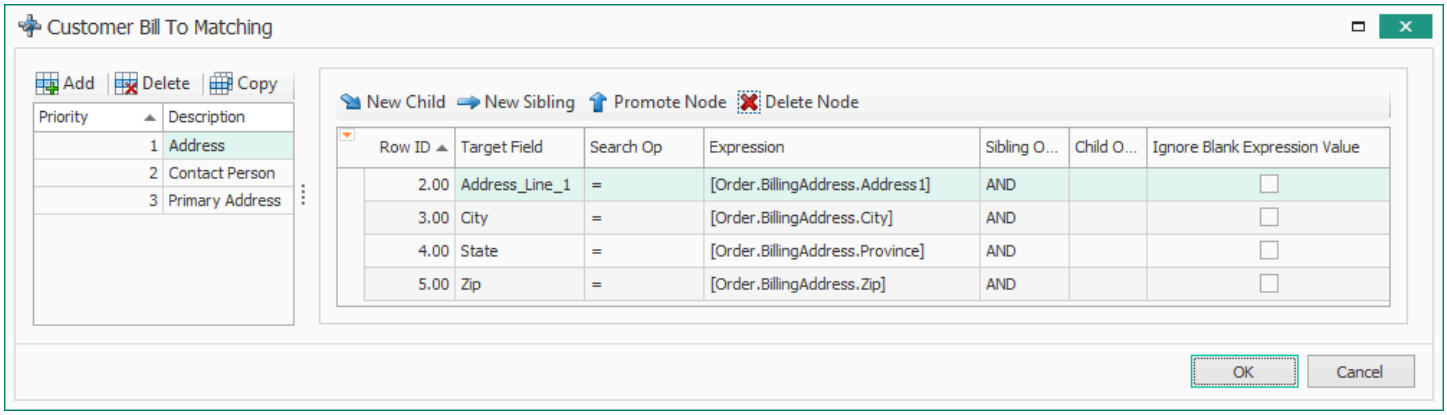
This setting determines how SalesPad will attempt to look up the Bill To Address for an incoming Shopify order. If a SalesPad Customer was matched via the **Customer Matching** setting, the Bill To Address must also belong to that customer. Otherwise, the search will consider address codes across all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

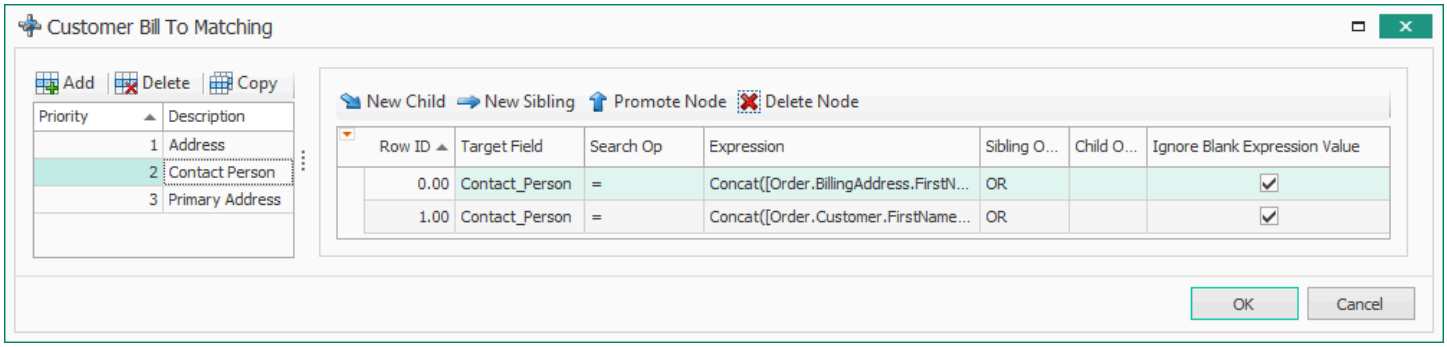
Priority 1

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



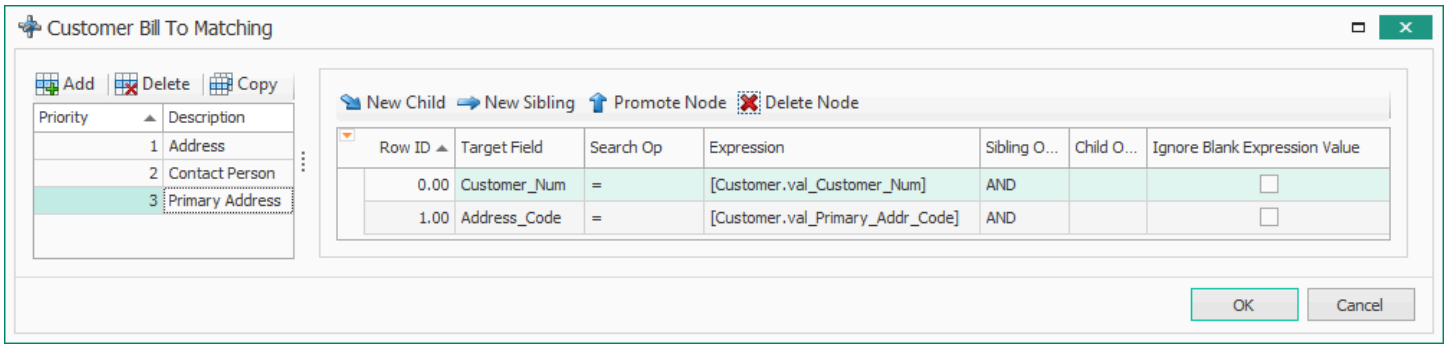
Priority 2

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



Priority 3

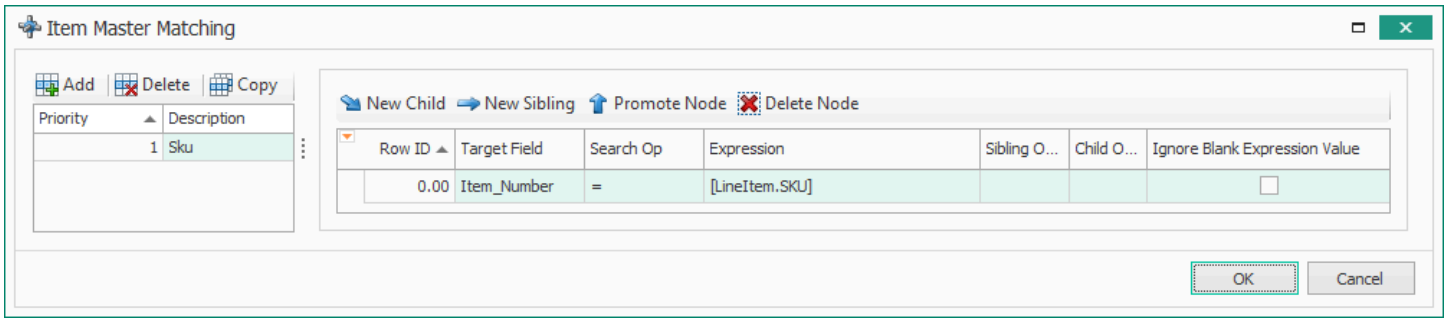
Primary Address: Attempt to find a match based on the Primary Address for the Customer. This assumes that the Customer was matched via the *Customer Matching* setting.



Item Master Matching - *Define the mappings for matching Shopify items to a GP Item Master.*

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.



Assignment Settings

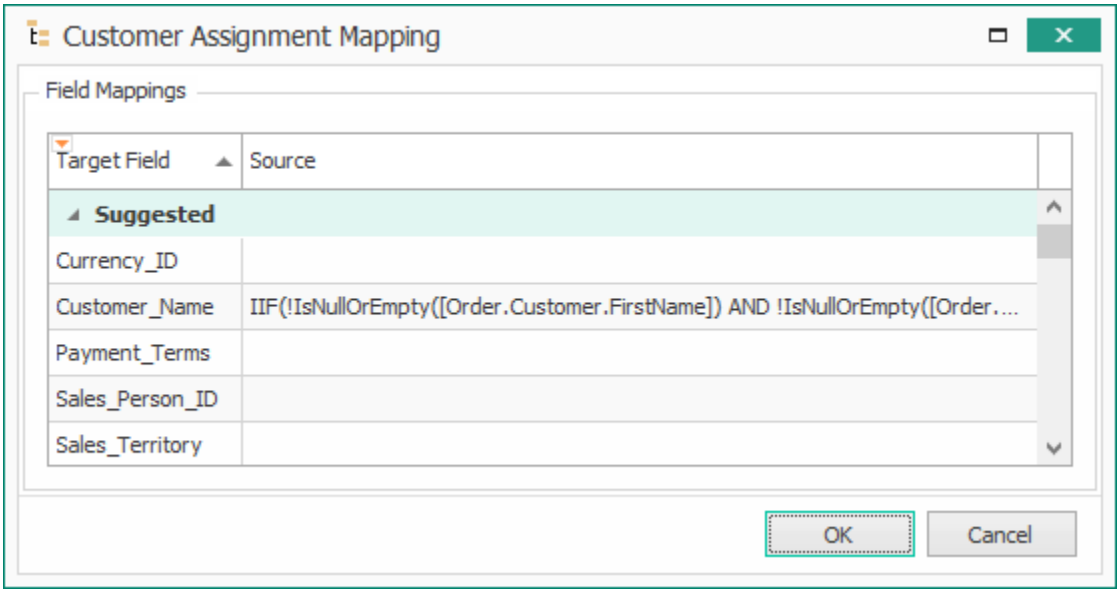
Assignment settings use Expression Editors to define how newly created Customers, Orders and Sales Lines will be populated when created during the order import process.

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.

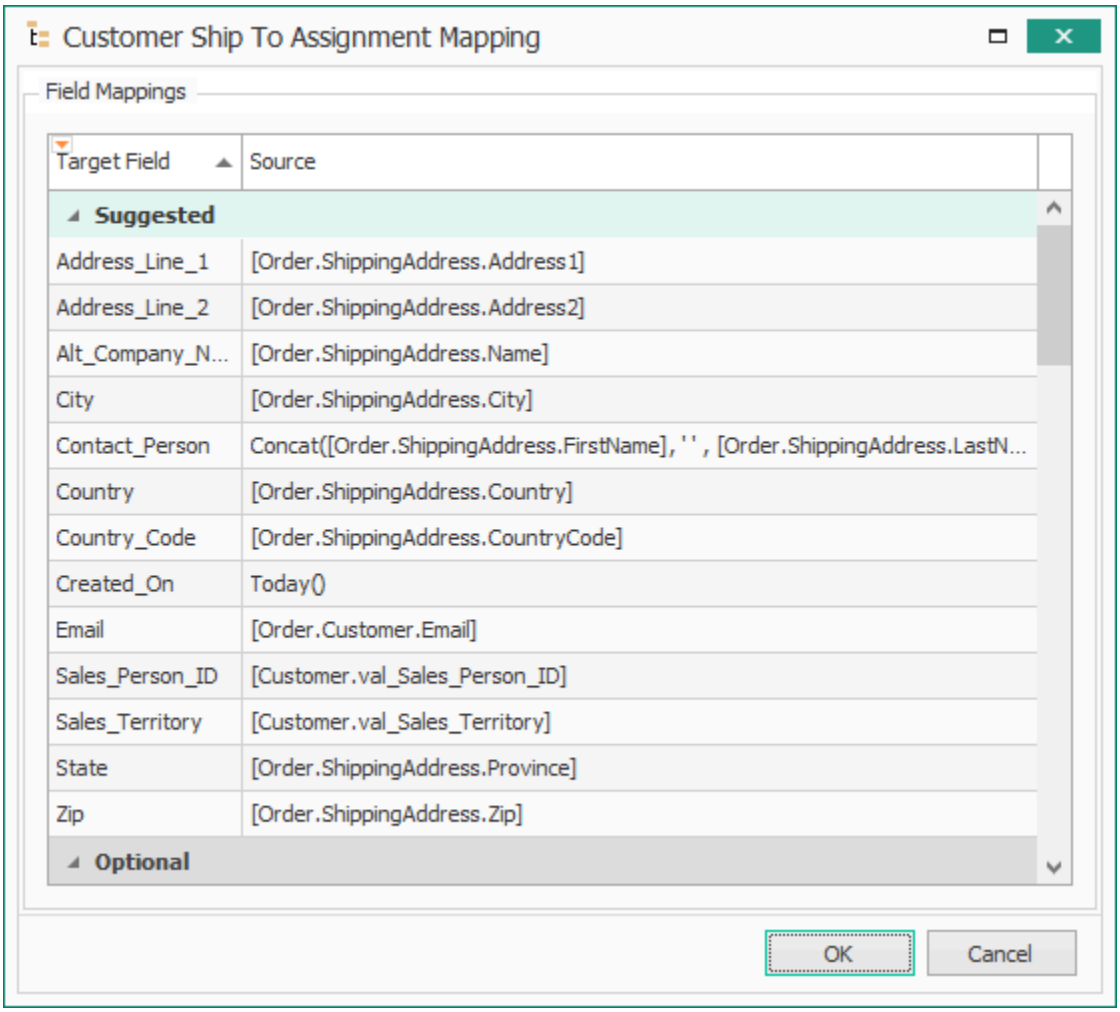
Customer Assignment Mapping - *Define the mappings to be used when creating a new Customer.*

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.



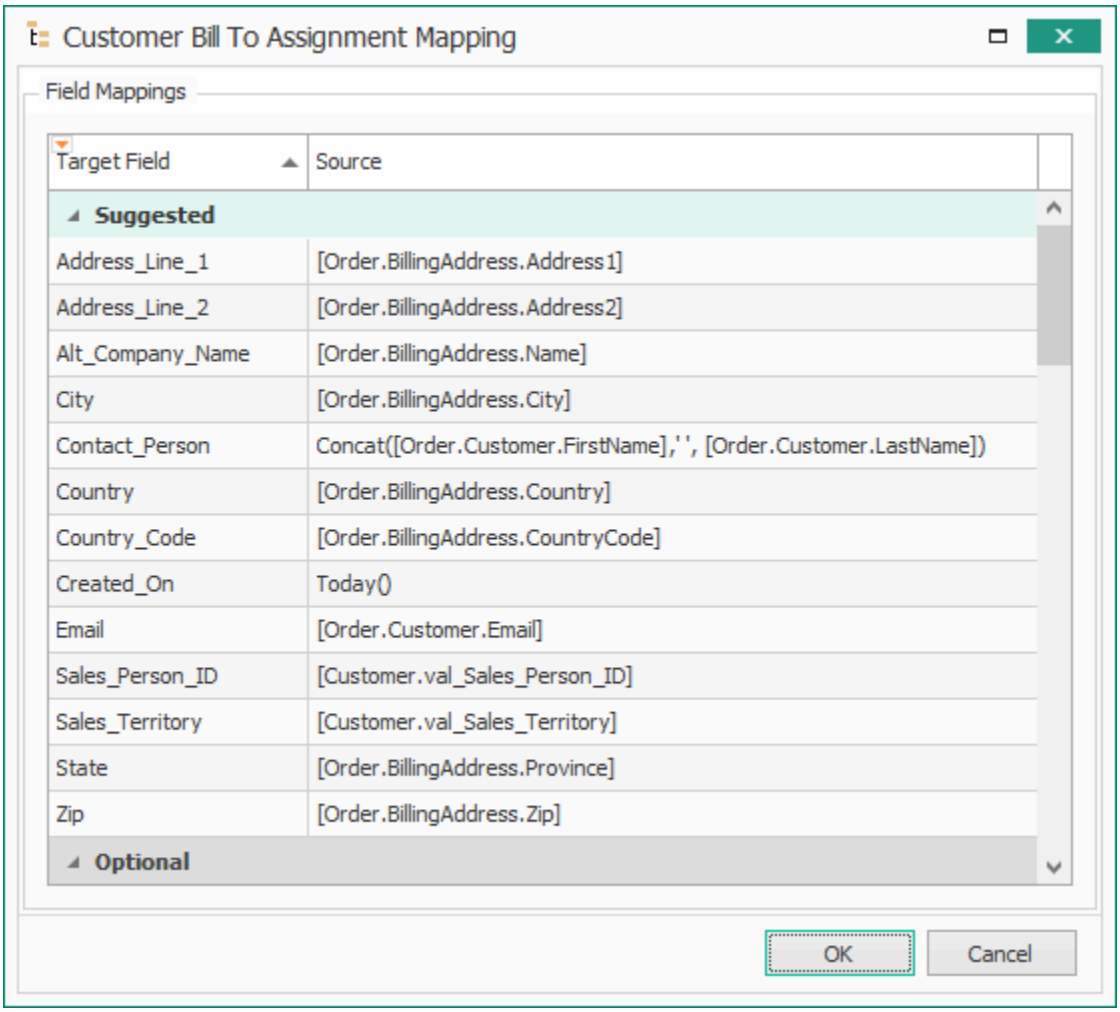
Customer Ship To Assignment Mapping - *Define the mappings to be used when creating a new Customer Ship To Address.*

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.



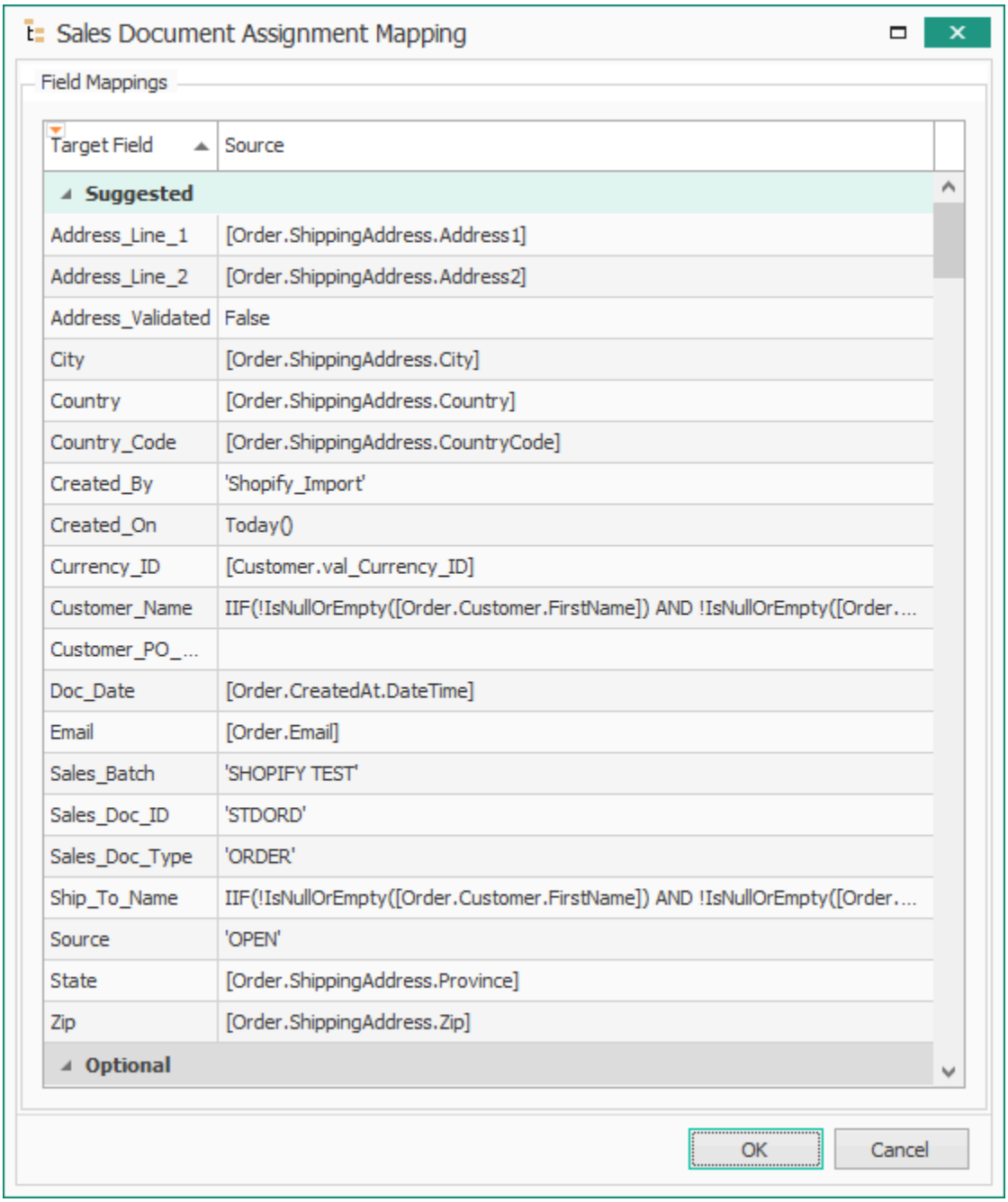
Customer Bill To Assignment Mapping - *Define the mappings to be used when creating a new Customer Bill To Address.*

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.



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

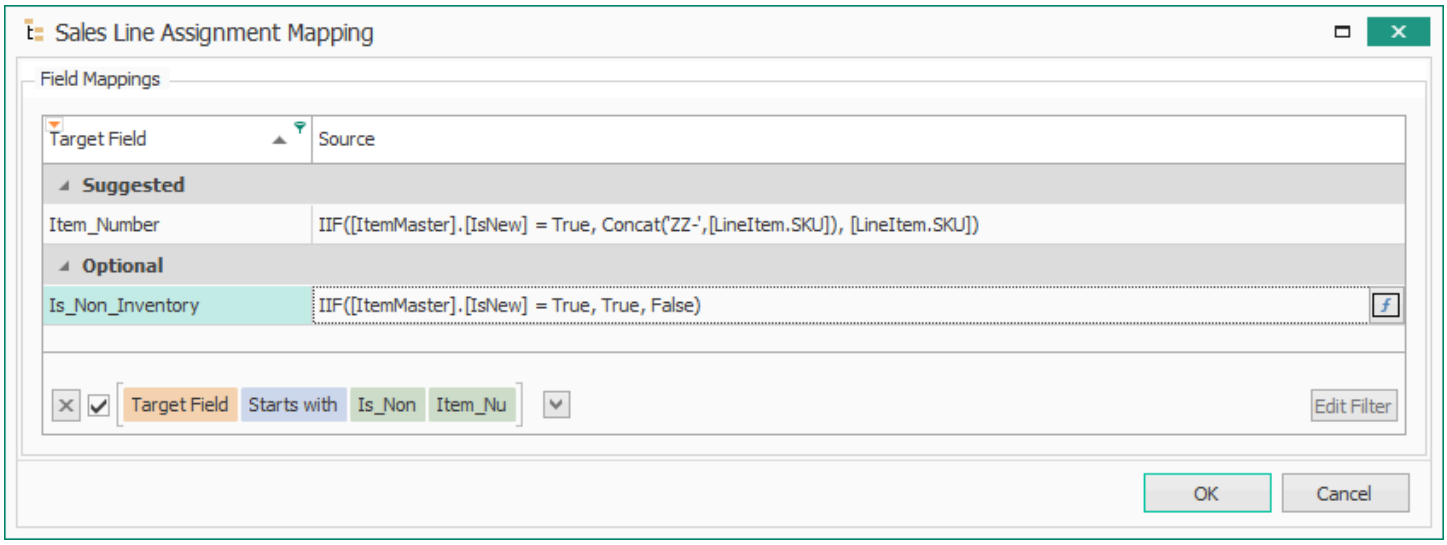
When the Shopify Order is imported, this setting will be used to populate the header level fields on the SalesPad Sales Document.



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

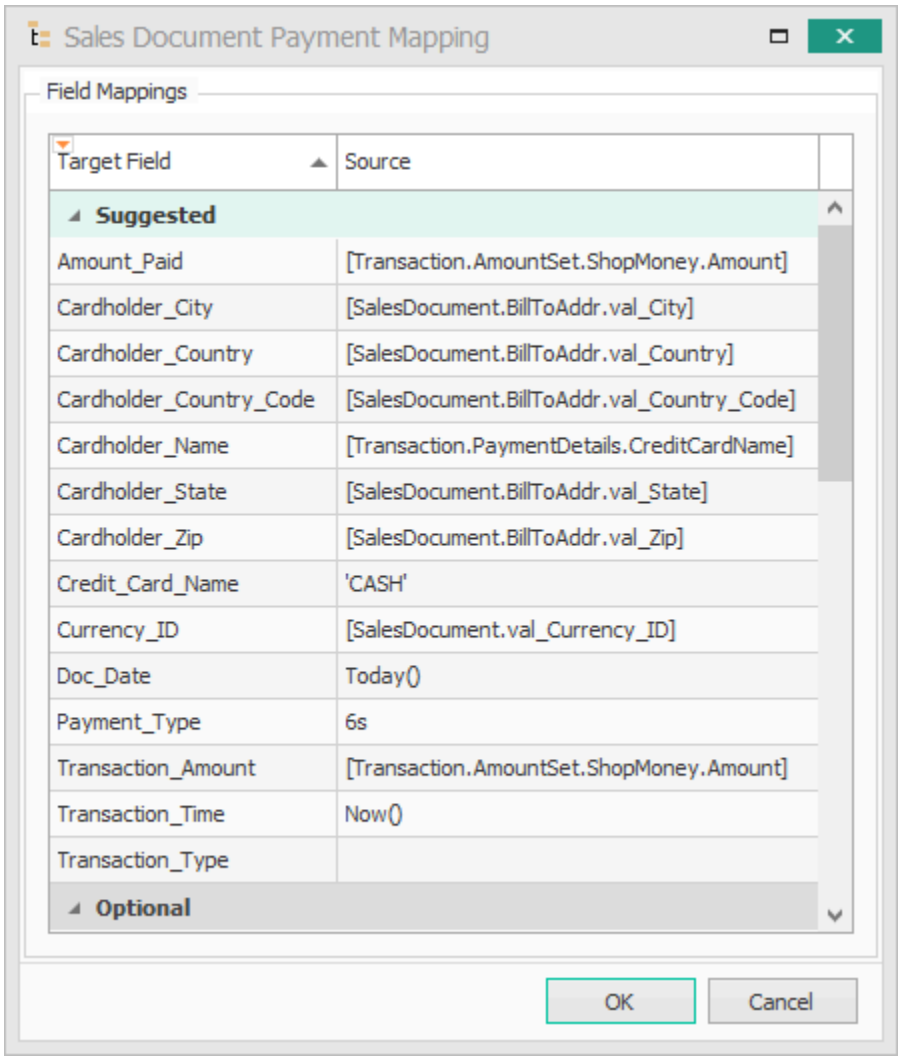
When the Shopify Order is imported, each line on that order will be created as a sales line on the SalesPad Sales Document. Each sales line's fields are set based on the *Sales Line Assignment Mapping* setting.

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)



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.*



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:

```
Unset
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:

Unset


```
{
  "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:


```
Unset
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:

Unset


```
{
  "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

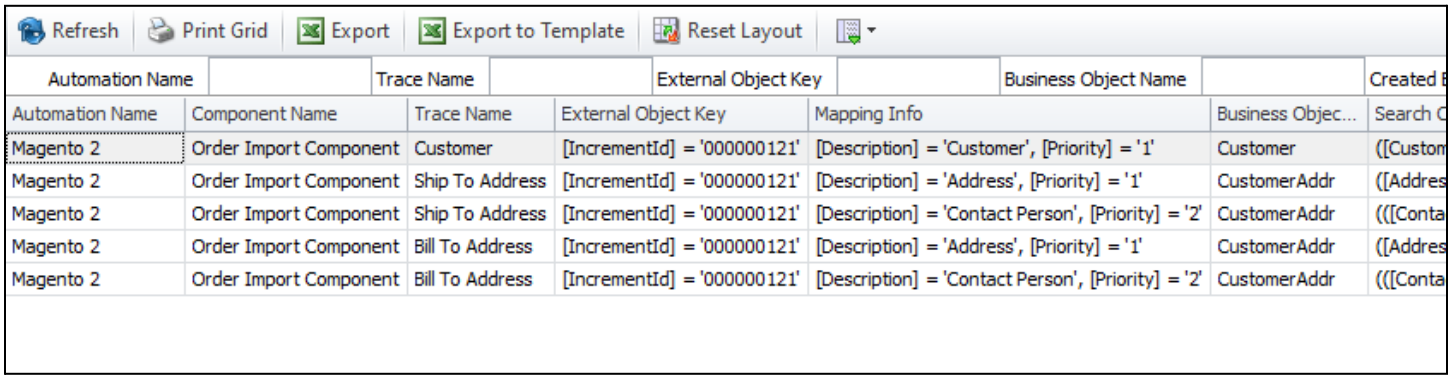
After an order has been successfully imported into SalesPad, the Order Import will tag the Shopify order with the Processed Order Tag setting's value.

GraphQL mutation:

```
Unset
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.



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 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 a specific workflow queue, and it forwards orders out of that queue 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 Queue - *Queue that contains 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 this queue so that they can export fulfillment and tracking information back to Shopify. They will wait to be processed, and afterward they will be forwarded in workflow.

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.

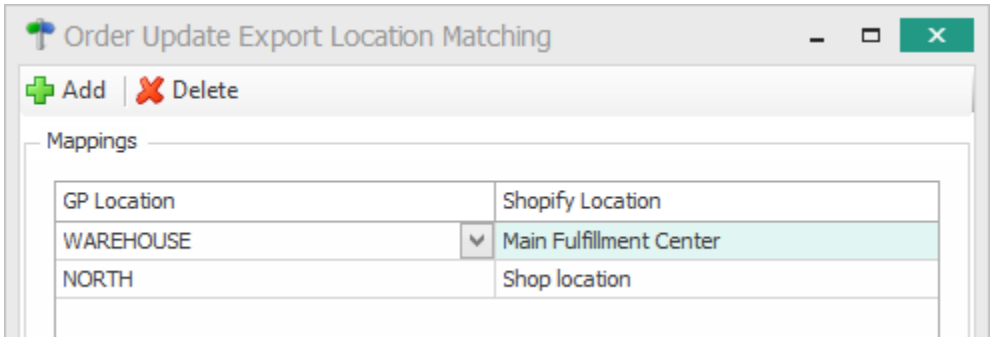
Notify Customers - *If set to True, Shopify will email the order's customer when the Order Update Export creates or updates fulfillments. Defaults to 'True'.*

Number Of Orders Per Export Page - *Specify the number of orders in each page of the order export. Defaults to '50'.*

Use this setting to minimize the number of orders that SalesPad attempts to load and process at the same time.

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.*



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 in the *Export 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:

```
Unset
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:

Unset


```
{
  "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:

```
Unset
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:

```
Unset
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:

Unset

```
{
  "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:

```
Unset
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:

Unset


```
{
  "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:

```
Unset
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:

Unset


```
{
  "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 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'.*

Number Of Voided Orders Per Export Page - *Specify the number of voided orders in each page of the Voided Order Export. Defaults to '50'.*

Matching Settings

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

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:

```
Unset
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

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
VariantTitle	variantTitle
Name	name
Vendor	vendor
GiftCard	isGiftCard
Taxable	taxable
TaxLines	taxLines
TotalDiscount	totalDiscountSet.shopMoney.amount
TotalDiscountSet	totalDiscountSet

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

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