

# In8Sync Shopify DirectConnect Setup Guide

Version: 1.0

Date: 09/23/2019

©Copyright 2019 In8Sync, LLC | All Rights Reserved

## Table of Contents

Table of Contents .....	1
<b>1.1 Overview</b> .....	5
<b>Bundle Purpose</b> .....	5
<b>Prerequisites/System Requirements</b> .....	5
<b>2.1 NetSuite Shopify DirectConnect Bundle Version</b> .....	5
<b>3.1 Shopify Version</b> .....	5
<b>4.1 NetSuite Bundle Installation</b> .....	5
<b>4.2 Out of the Box Features</b> .....	5
<b>Components</b> .....	8
<b>Product Flow</b> .....	9
<b>Customer Flow</b> .....	11
<b>Order Flow</b> .....	13
<b>Return Flow</b> .....	14
<b>4.3 Shopify directly from In8Sync (Full Yearly Subscription)</b> .....	15
<b>4.4 Installation Process</b> .....	15
<b>5.2 Setup/Tools</b> .....	18

<b>5.3 Global Settings</b> .....	18
START: REQUIRED SECTIONS FOR SUCCESSFUL STANDARD SETUP.....	19
<b>5.4 Settings</b> .....	19
<b>Connection Setup   Required Settings</b> .....	19
<b>Creating a Private App</b> .....	19
<b>Copy the Shopify URL Domain</b> .....	22
<b>Configure the NetSuite Setting</b> .....	23
<b>Additional Settings</b> .....	27
<b>5.5 Mappings</b> .....	46
<b>cashrefund</b> .....	47
<b>cashsale</b> .....	48
<b>customer</b> .....	48
<b>Customer Mappings from Order Payload Fields</b> .....	49
<b>inventoryitem</b> .....	50
<b>Shopify Item Sync Triggers</b> .....	53
<b>Matrix Option/Attribute Mapping</b> .....	56
<b>Taxable Mapping</b> .....	56
<b>itemfulfillment</b> .....	59
<b>salesorder</b> .....	60
<b>Additional Details/Note Attributes Mapping</b> .....	61
<b>Shopify Order# set in NetSuite Doc# (transid)</b> .....	63
<b>Line-Item Properties Mapping</b> .....	66
<b>5.6 Shipping Mappings</b> .....	68
<b>5.7 Payment Mappings</b> .....	77

<b>5.8 Location Mappings</b> .....	79
<b>END: REQUIRED SECTIONS FOR SUCCESSFUL STANDARD SETUP</b> .....	84
<b>5.9 User Mappings</b> .....	84
<b>5.10 Shopify Requests (Shopify to NS)</b> .....	84
<b>5.11 Requests Logs (Shopify to NS)</b> .....	85
<b>5.12 Shopify Synced Orders</b> .....	85
<b>5.13 Sync Queue (NS to Shopify)</b> .....	86
<b>5.14 Sync Queue Logs (NS to Shopify)</b> .....	86
<b>5.15 Shopify Plus ONLY - Manual Gift Cards Pull</b> .....	86
<b>5.15.1 Manual Gift Card Queue</b> .....	86
<b>5.15.2 Manual Gift Card Logs</b> .....	87
<b>5.15.3 Manual Gift Card Puller</b> .....	88
<b>5.16 Other Transaction Info Results</b> .....	88
<b>5.17 Shopify Payments Payout Reconciliation</b> .....	88
<b>Requirements</b> .....	89
<b>Parameter Mappings</b> .....	89
<b>Settings</b> .....	89
<b>5.17.1 Payout Transactions Results</b> .....	90
<b>5.17.2 Payout Transactions Logs</b> .....	94
<b>5.17.3 Payout Reconciliation Results</b> .....	94
<b>5.17.4 Payout Reconciliation Queue</b> .....	95
<b>5.17.5 Payout Processor logs</b> .....	96
<b>5.17.6 Payout &amp; Transactions Puller</b> .....	96
<b>5.18 Stock Sync Queue</b> .....	97

---

<b>5.19 Stock Sync Logs</b> .....	97
<b>6.1 Testing Installation</b> .....	97
<b>Test An order</b> .....	97
<b>Test Customers</b> .....	98
NetSuite to Shopify .....	98
Shopify to NetSuite .....	98
<b>7.1 Updating the Bundle</b> .....	98
<b>8.1 Uninstalling the Bundle</b> .....	99
<b>9.1 Support</b> .....	99
<b>Troubleshooting</b> .....	99
<b>Support Options</b> .....	99
<b>Contacting Support</b> .....	99
<b>NetSuite Disclaimer</b> .....	100

## 1.1 Overview

### Bundle Purpose

Purpose of this Bundle is to integration and pull Orders, Customer and other data from Shopify into NetSuite. In turn, Updating/Creating Customer, Sales Order and other transactions in NetSuite.

### Prerequisites/System Requirements

- Shopify Account
- NetSuite Account
  - Note: Only Administrators can install this bundle

## 2.1 NetSuite Shopify DirectConnect Bundle Version

- Minimum In8Sync Vend Bundle Version 3.2.x and later

## 3.1 Shopify Version

- Must have a valid Shopify Account to connect to.

## 4.1 NetSuite Bundle Installation

### 4.2 Out of the Box Features

Requirements:	Oracle NetSuite	External Application
Platform	Minimum Starter Edition	Shopify

#### 1. **Product Features** (out of the box - setup & configuration required)

##### 1. **Key features**

- Managed 100% within your NetSuite account. No middle-ware
- Managed support and setup
- Shopify risk management supported
- Multi currency support
- Order shipments from NetSuite sync to Shopify immediately
- Other data sync run on scheduled every 15 minutes

## 2. Basic Flows

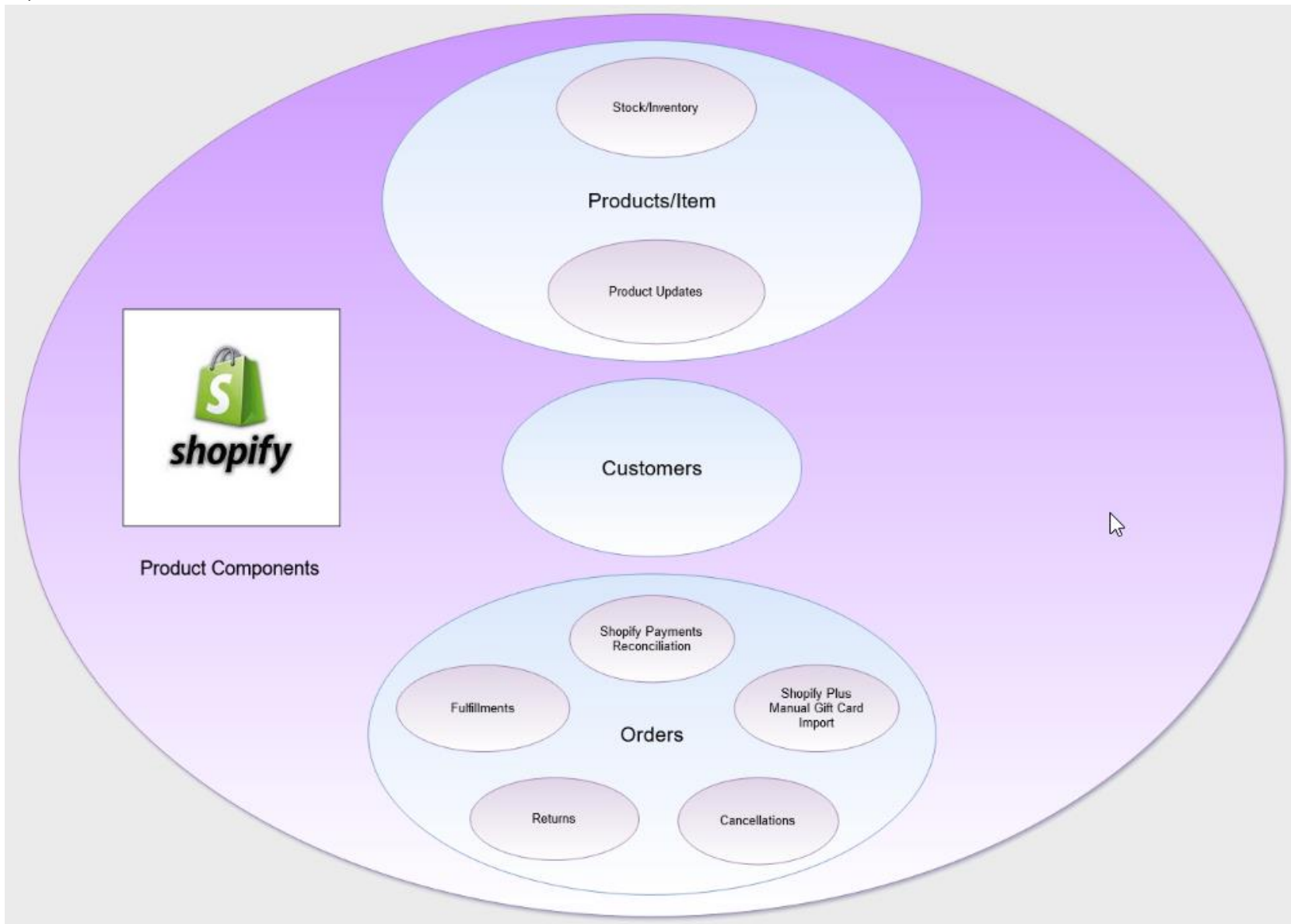
- Customers - Bidirectional
- Orders - Shopify to NetSuite
- Product & Inventory - NetSuite to Shopify
  - Multi-Location Stock Update Support - NetSuite to Shopify
- Basic/Advanced Promotions - NetSuite to Shopify
- Refunds
  - POS Channel - Shopify to NetSuite
  - All Other Channels
    - NetSuite to Shopify
    - Shopify to NetSuite
      - No Loop or Returnly included in Flow, requires other integration or Customization
- Shipment Tracking
  - NetSuite Fulfillments - NetSuite to Shopify
  - Shopify App Fulfillment - Shopify to NetSuite to create Fulfillment
    - No Serial Numbers Support
    - Limited support for Lot Numbers/Bins
- Gift Certificates
  - Shopify → NetSuite
    - Gift Cards on Shopify orders
    - Manual created or app created Shopify gifts cards. **(Shopify Plus Required\*)**
  - Omni-channel: NetSuite → Shopify **(Shopify Plus Required\*)**
- Point of Sale (POS) integration

## 3. Advanced Add-ons (additional configuration required)

- Automated payment reconciliation - Shopify Payments
- Returns Portal (direct NS Connection) - Customer return authorization form in Shopify creates NetSuite RMA record
- Customer specific pricing from NetSuite is displayed and used in Shopify (Requires Being able to Sync Items from NetSuite to Shopify)  
Also Sync Customer/Contacts from NS to Shopify (No Duplicate Emails)
- Customer default price level
- Item specific pricing
- Group pricing
- Quantity based pricing
- Multi-Currency

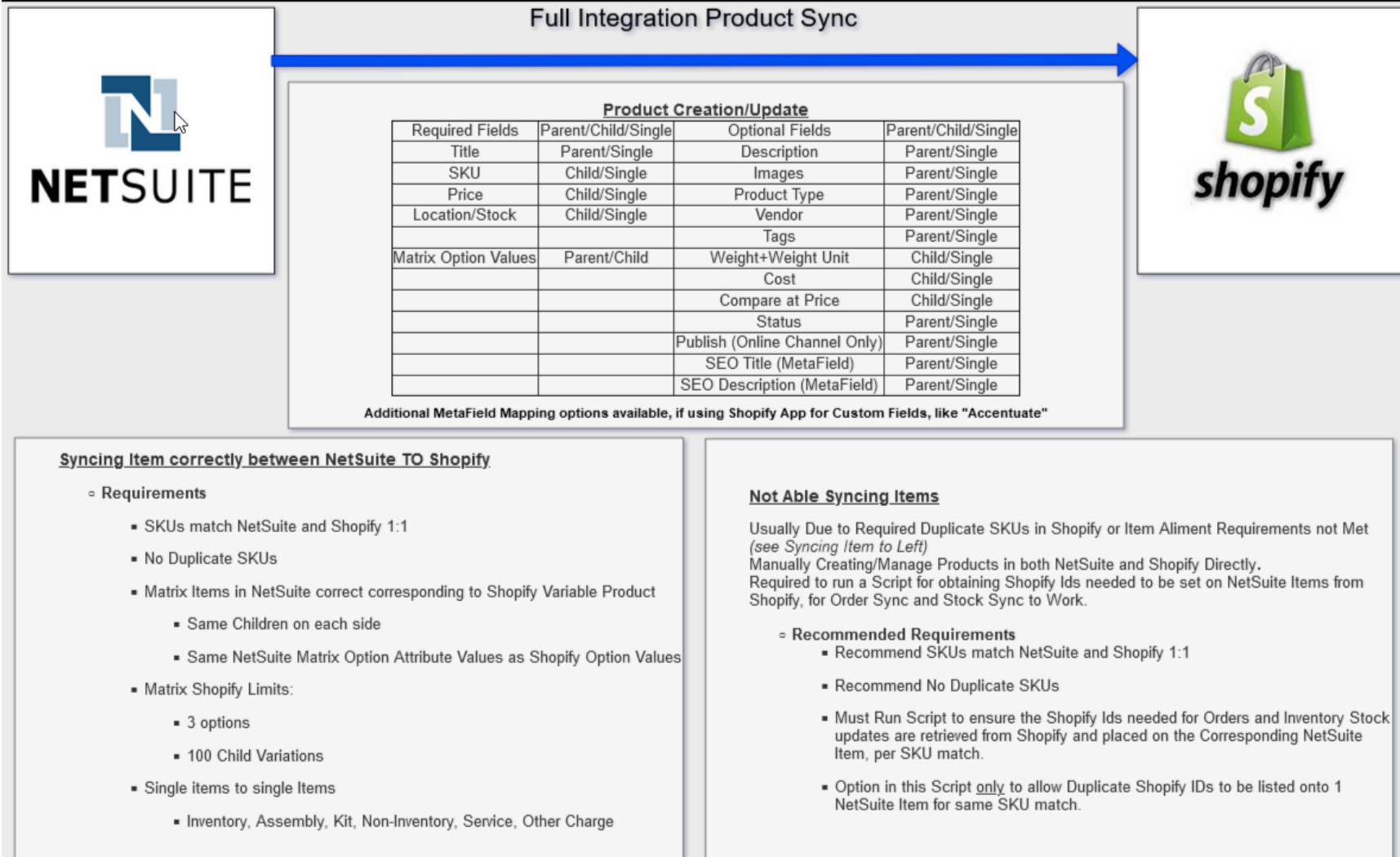
- Variable product sync for non-matrix items (requires item setup training)
- 2. **Shopify Limitations that may affect integrations**
  - Variable Products has a maximum of 100 variants in Shopify
  - Variable Product may have up to 3 options MAX
  - Shopify limits new variant creation to 1000 per day (Shopify Plus excluded from this rule)
  - Wholesale sales channel in Shopify has no API support for customer pricing rules
  - Gift Card API only available with Shopify Plus ONLY!
    - Apps that Create Manually Created Gift Cards that will That Require Shopify PLUS
      - Rise AI
      - Returnly
      - Loop
- 3. **Sync Frequency**
  - Shopify to NetSuite
    - Every 15 Minutes
    - Customers
    - Customer/Order
    - Products
  - Manual Gift Cards (Shopify Plus Only)
    - Every 4 hours
  - Shopify Payments Payout Daily Data
  - NetSuite to Shopify
    - Real-time
      1. Customers
      2. Products
    - Scheduled on and On Demand
      1. Inventory Stock updates

## Components



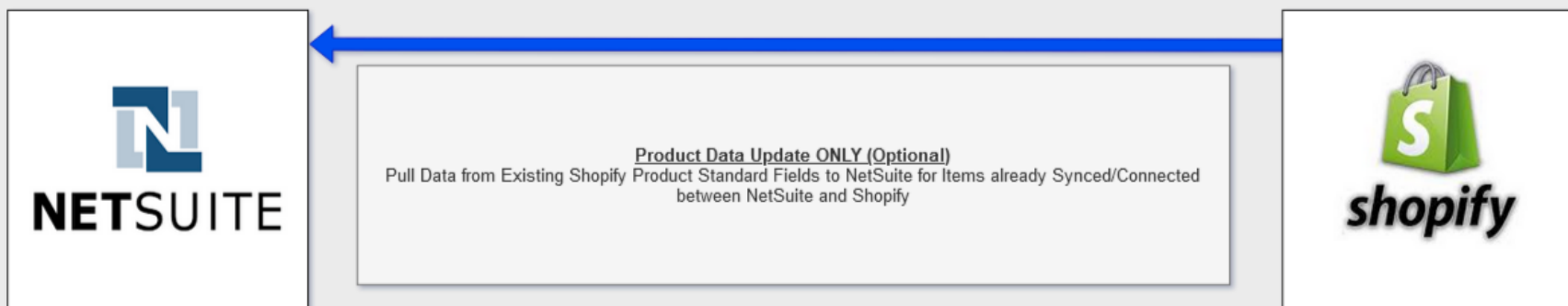


## Product Flow

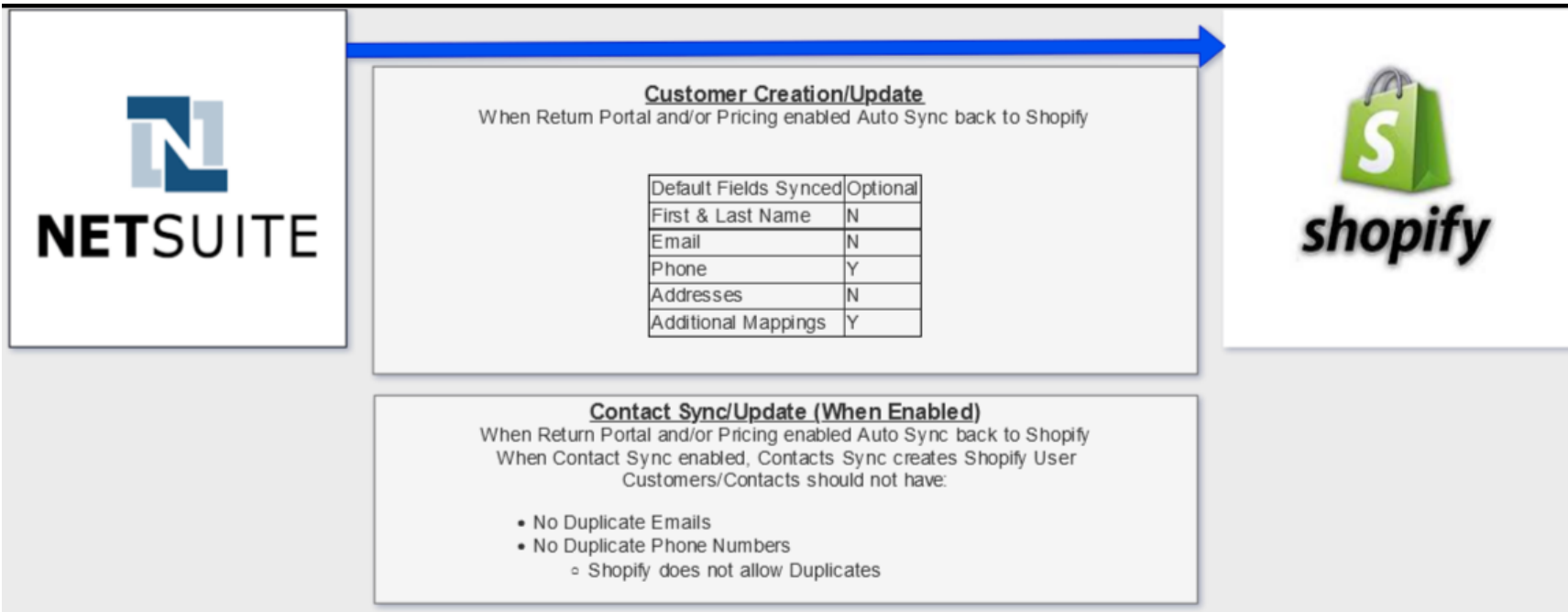


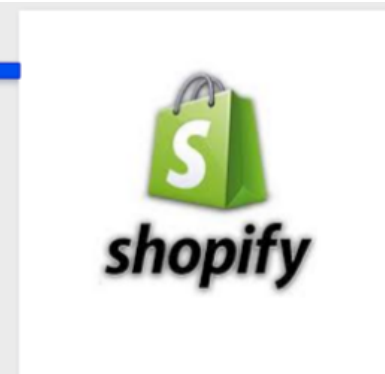
**Stock Sync**

- Shopify to NetSuite
  - Standard Periodic Stock Sync with Options for:
    - Delta Sync (Only Items seen on transactions since Last stock Sync)
    - Items on transaction today
    - All
- Individual Stock Sync Queue
  - Inventory Adjustments
    - Items with Shopify Inventory Item Ids, for Stock update on Completed adjustment, sync Individually through Stock Sync Queue
  - Child Matrix/Single Items, with Shopify Inventory Item Ids, have "Shopify Stock Sync" Button to sync Individually through Stock Sync Queue
  - (Optional, needs deployed per account) Non-Shopify Transactions, Sales Order, Invoice, or Cash sale, with Shopify Items with Inventory Item Ids to be sent directly through Stock Sync Queue



Customer Flow





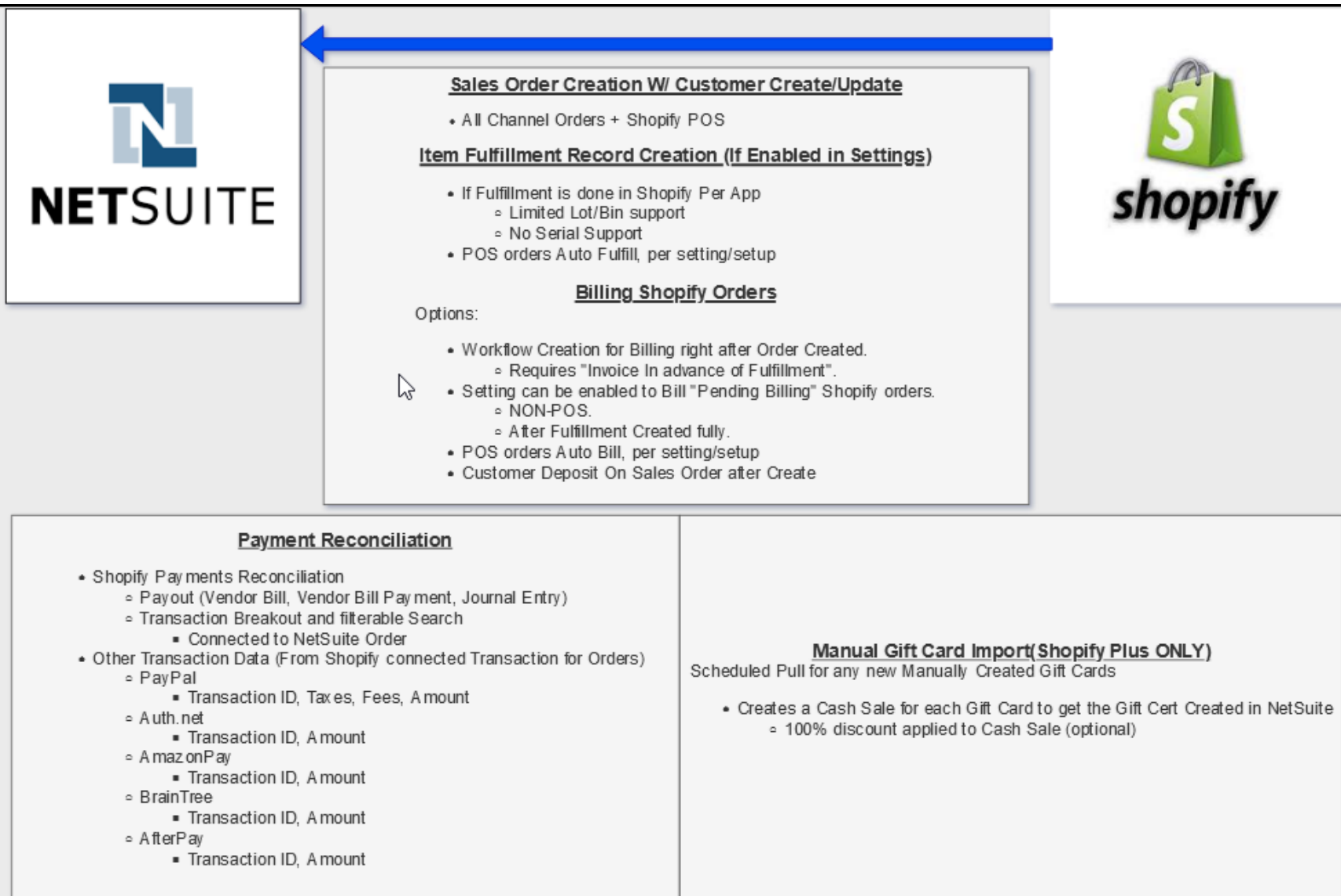
**Customer Creation/Updates**  
 With Orders  
 Customer Update Only (when Enabled)  
 New Shopify Customers Search Connection Process:

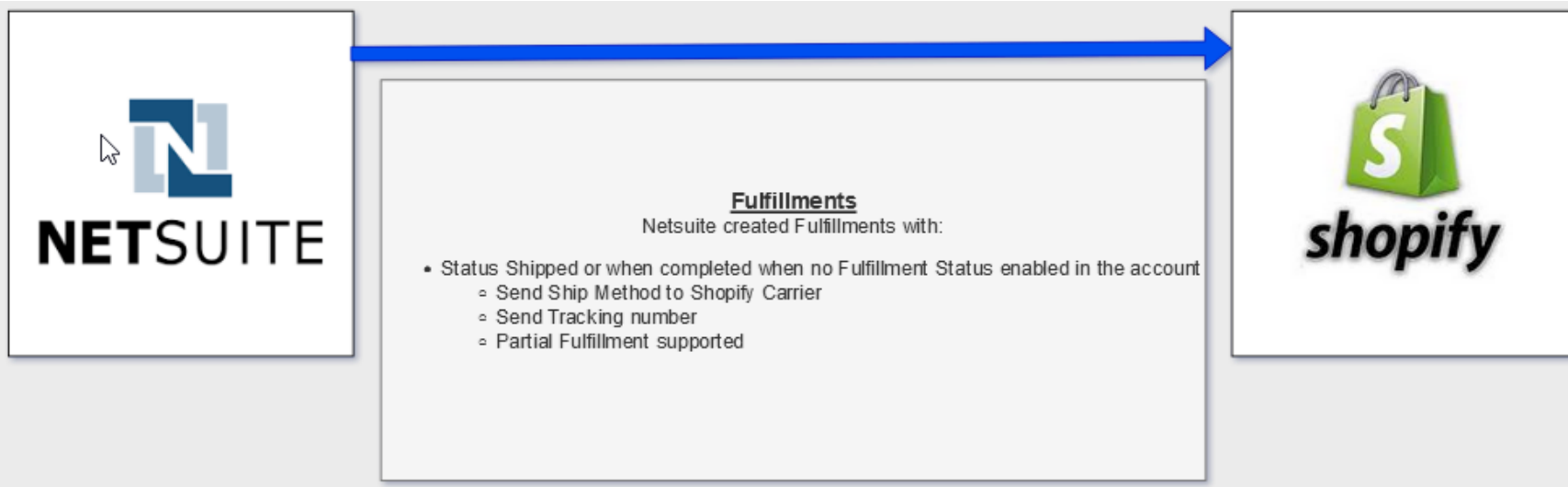
- Shopify to NetSuite Search for Customer or Contact (when Enabled) with same Shopify ID.
- If Shopify ID not found, search by Email & Subsidiary
- If Email and Subsidiary not found, create new Customer
  - Shopify Id then attached to Customer/Contact for Future Connection between both systems

Default Fields Synced	Optional
First & Last Name	N
Email	N
Phone	Y
Addresses	N
Additional Mappings	Y

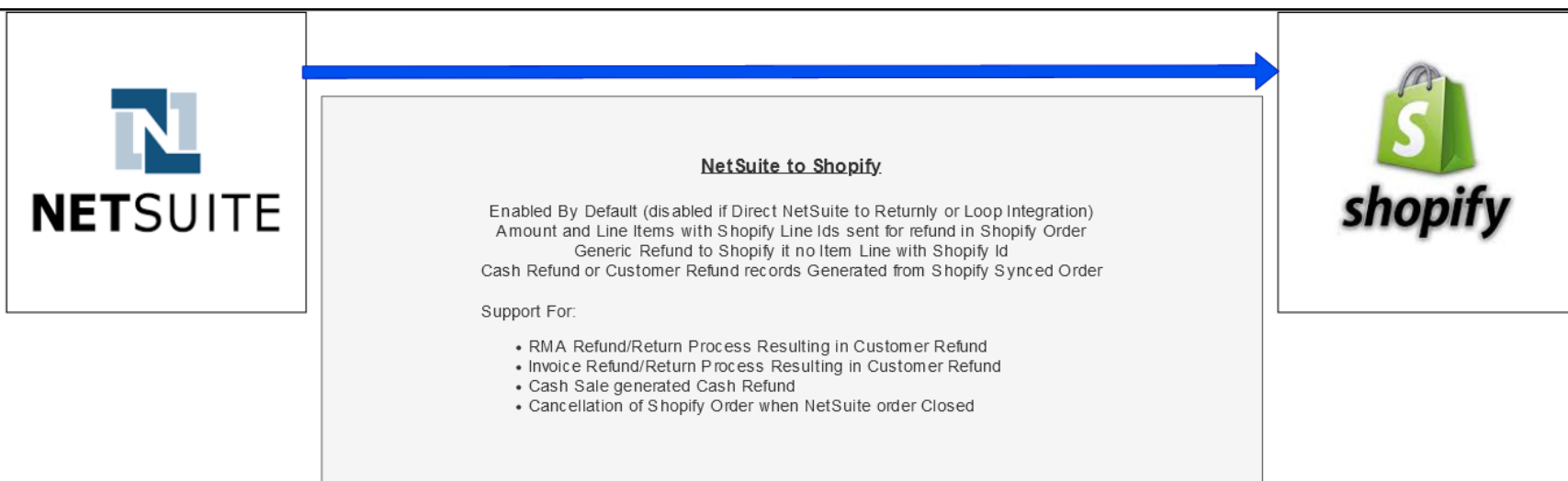
**Contact Sync/Update**  
 (When Enabled)  
 Sync back to NetSuite search Checks for Contacts 1st then Customers 2nd

Order Flow





Return Flow





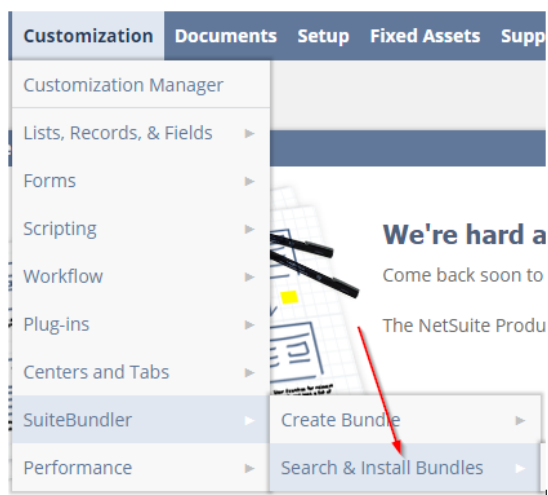
### 4.3 Shopify directly from In8Sync (Full Yearly Subscription)

If Shopify is purchased Directly from In8Sync, the original License Key will continue to be used for Annual License and Renewals, once purchased.

### 4.4 Installation Process

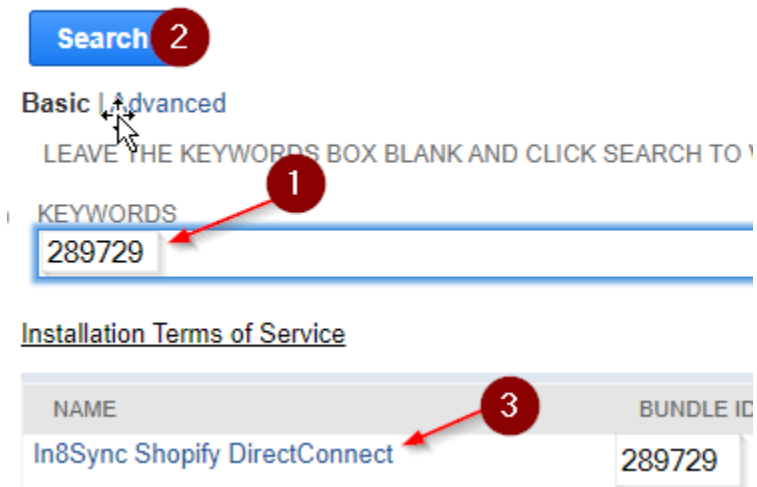
Installation of the In8Sync Shopify Bundle only requires that you follow the instructions below from your NetSuite account. In8Sync must make the Bundle available to your NetSuite Account ID. Please contact In8Sync if you need to have it made available

Go to *Customization Tab > SuiteBundler > Search & Install Bundles*



Search for Bundle 289729 then click on the name "In8Sync Shopify DirectConnect"

## Search & Install Bundles



Under the Bundle Details page click Install



## Bundle Details

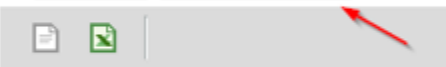
### In8Sync Shopify DirectConnect



NAME  
In8Sync Shopify DirectConnect

On the Preview Bundle Install page, click Install Bundle

## Preview Bundle Install

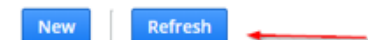


Name: [redacted] Version: [redacted]

BUNDLE CONTENTS

On the Installed Bundles page, you can see the bundle installing with progress shown under Status. Keep hitting refresh until completes

## Installed Bundles



ACTION	NAME	BUNDLE ID	VERSION	MANAGED	ABSTRACT	ADMIN DOCUMENTATION	STATUS
	[redacted]	[redacted]		No			Pending...

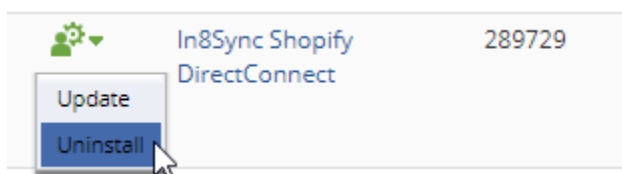
When installation is complete there will be a green check mark under status

## Installed Bundles

New Refresh

ACTION	NAME ▲	BUNDLE ID	VERSION	MANAGED	ABSTRACT	ADMIN DOCUMENTATION	STATUS
				No	NetSuite integration with Vend created by In8 (www.ineight.net).		

Once the Bundle is installed successfully, there is a Role called “In8Sync Integration Role” that does not require 2FA, that can be used and assigned to the NetSuite User Seat allowing In8Sync Integration Team to log in and setup the Integration with their Customer’s Project.



The following Sections are listed under **Setup Tab > In8 Shopify**

## 5.2 Setup/Tools

This Menu Section is where Current and Future features will reside regarding:

Existing:

- Twice Daily Error reports for Transaction & Customer Payloads pulled from Shopify that end up in error.
- Enabling scheduled and Manual Stock Sync from NetSuite to Shopify
- Clearing Old logs from the “Shopify Requests (Shopify to NS)” Payload Queue

Future:

- Dashboard for Error logs
- Adjustments to Main configuration Pulls

## 5.3 Global Settings

Global Setting is mainly used for the current option setting they would apply to ALL Shopify Site integration in NetSuite.

Currently there are the following Two options:

- “Sync Items Through Queue”

- All Site would have Items sync through the “Sync Queue (NS to Shopify)”
- See Section [6.13 Sync Queue \(NS to Shopify\)](#)
- Flatten Matrix Items
  - *This should never be turned on without advanced help from In8Sync. It changed the way Matrix Items in NetSuite would sync to Shopify.*
  - In Short, Matrix Parent Items no longer sync and the Sync considers all Items in NetSuite as Single Items

## START: REQUIRED SECTIONS FOR SUCCESSFUL STANDARD SETUP

### 5.4 Settings

Each Shopify Site Settings are the “Site Global” setting for Connection and additional options available to control the processes for Integration per site!

The Connection Setup, required settings, and a few other settings are needed for the Main Integration to work properly.

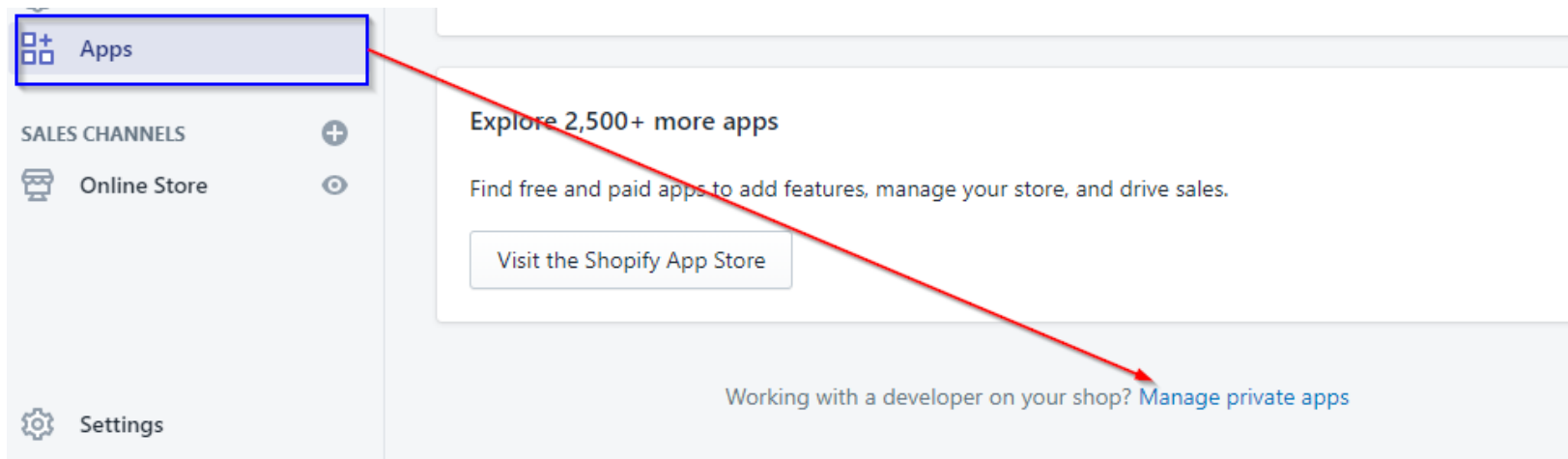
### Connection Setup | Required Settings

Since most of the configuration and setup for a Shopify integration is mainly in NetSuite, there is not much needed for the Shopify side of the Integration.

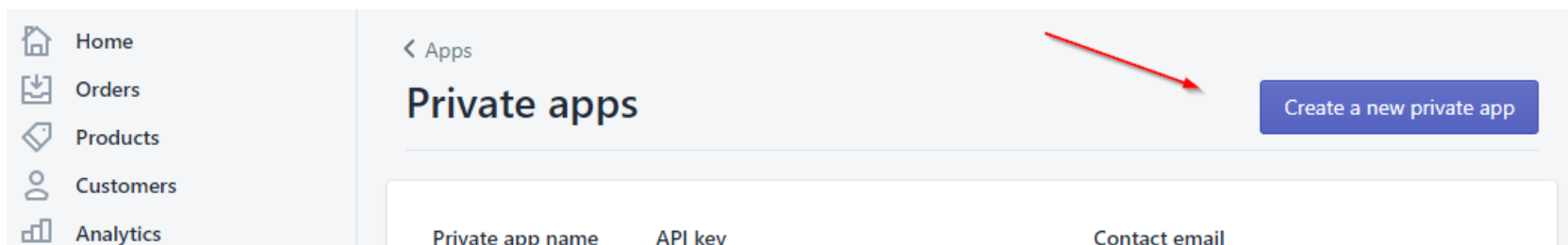
From Shopify we need Create a Private App and get API Token and Password along with copy the URL for the Shopify website location.

#### Creating a Private App

Go to Apps > *Manage Private Apps (Link towards the bottom of the Page)*



Click on Create a New Private App button:



Enter a Name for the App and email, then enable the following for permissions

## Create private app

### App details

These details can help you keep track of your private apps and stay informed about changes that affect your app.

[Learn more.](#)

Private app name

In8Sync NetSuite

Emergency developer email

support@in8sync.com

The email address used for developer communication regarding your app.

Private apps are subject to the [Shopify API License and Terms of Use.](#)

Permission	Permission Level
Assigned fulfillment orders	Read and Write
Customers	Read and Write
Discounts	Read and Write
Fulfillment services	Read and Write
Gift cards	Read and Write
Inventory	Read and Write
Locations	Read access
Merchant-managed fulfillment orders	Read and Write
Orders	Read and Write
Price Rules	Read and Write
Product Listings	Read and Write
Products	Read and Write

Shipping	Read and Write
Shopify Payments Payout	Read access
Store content	Read and Write
Third-party fulfillment orders	Read and Write

Once the Save button is clicked, copy the API Token and Password for use in NetSuite

### Admin API

These permissions determine what data this private app can access. You should enable only what is necessary for your app to work.

API key

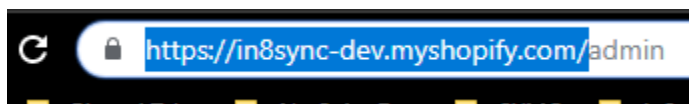
851[REDACTED]c37
📄

Password

5b[REDACTED]5a
Hide
📄

### Copy the Shopify URL Domain

Take note of the URL that you Vend runs on. In this case example we see “https://in8sync-dev.myshopify.com/”. Including the last slash



### Configure the NetSuite Setting

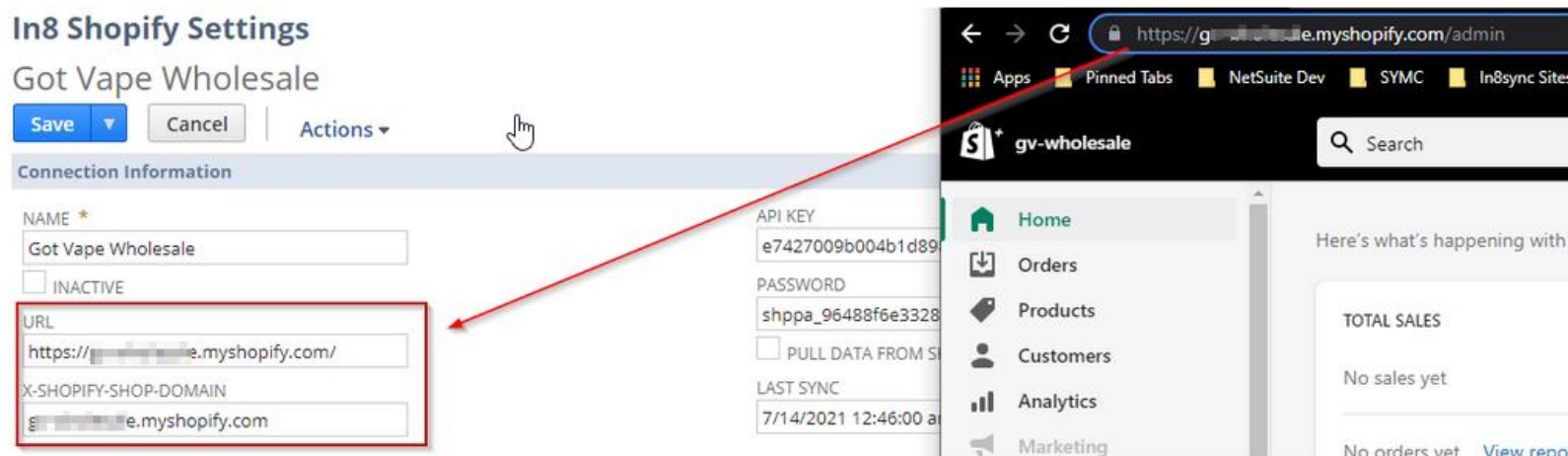
Under Setup Tab > In8Sync Shopify > Settings

Edit the Default Setting with Name “Shopify” (fresh Install) Or create a new one.

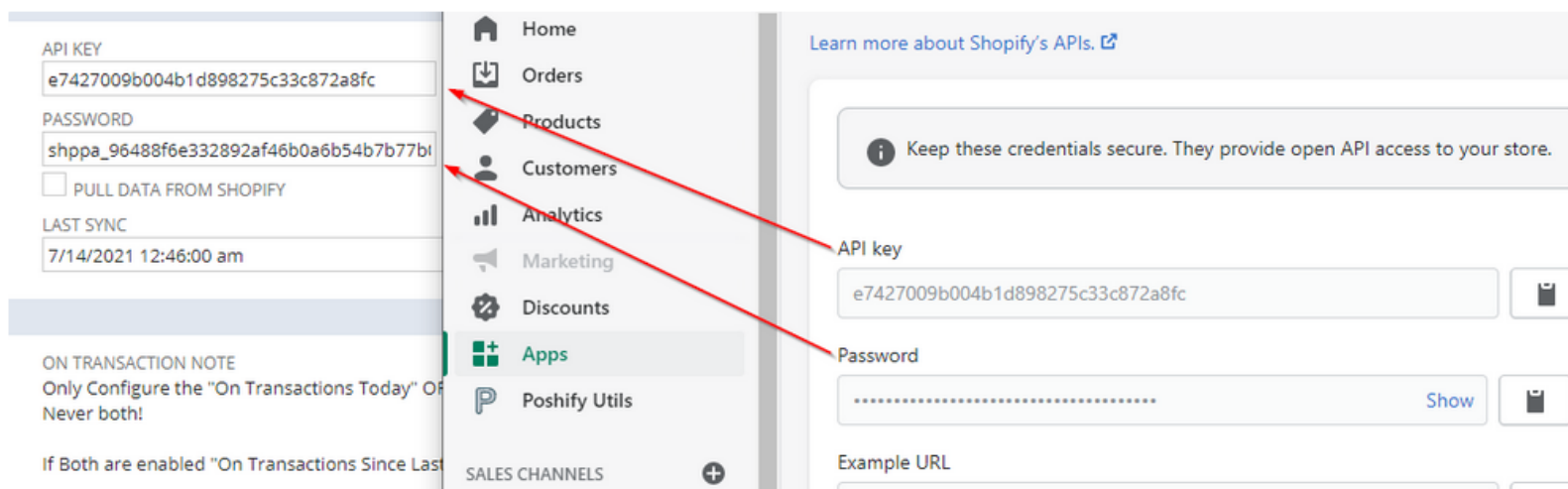
Setup	Customization	SuiteApps	Support
Setup Manager		Setup	
Company	>	Global Settings	>
Accounting	>	Settings	>
Sales	>	Mappings	>
Marketing	>	Shipping Methods	>
Support	>	Payment Methods	>
Intranet	>	Location Mappings	>
Import/Export	>	Users Mappings	>
In8Sync Shopify	>	Shopify Requests (Shopify to NS)	>
Records Catalog		Request Logs (Shopify to NS)	>

Set the URL, Domain, API Key and Password in NetSuite settings

Get the Admin URL from the Shopify Site and enter that into URL Field, without the “/admin” and the X-SHOPIFY-SHOP-DOMAIN field list the URL without the HTTPS:// and /



Then from the Private App creation above, take the Shopify API KEY and the Password and enter into the API Key and Password Fields



Before Saving, it is recommended to Create and Set the Required Fields

- Default Customer  
Used in the rare cases that the Shopify order Does not have any Customer assigned to it. Will use this customer to get the order into NetSuite



- Should be created on the Subsidiary that the Shopify Site will be running on in NetSuite
- Marked Taxable under the Financial SubTab if applicable
- First Name = Shopify
- Last Name = Default Customer
- Generic Product
  - Use in the Rare Case that the Line item on the order has NO Shopify Ids, like a Custom Item on the order from Draft order/Manual Order
  - Name = Shopify Generic Item
  - Set account as needed
  - Set Taxable
- Discount Item
  - Used for all Header and/or Line discount on the Order
  - Name = Shopify Discount
  - Set account as needed
  - Non-Taxable
- Shipping Discount Item
  - Used when there is a Shipping Discount and added as line item
  - Name = Shopify Shipping Discount
  - Set account as needed
  - Non-Taxable

Then Set the New Customer and Items in the required Fields section

Required Settings		
GENERIC PRODUCT *	DISCOUNT ITEM *	SHIPPING DISCOUNT ITEM *
Shopify Generic Product	Shopify Discount	Shopify Shipping Discount
DEFAULT CUSTOMER *		
CUST0000018 Shopify Default Customer		

Then Set the Subsidiary Internal ID (if One World) into the Subsidiary ID field in the general Settings

## General Settings

API VERSION

2020-10

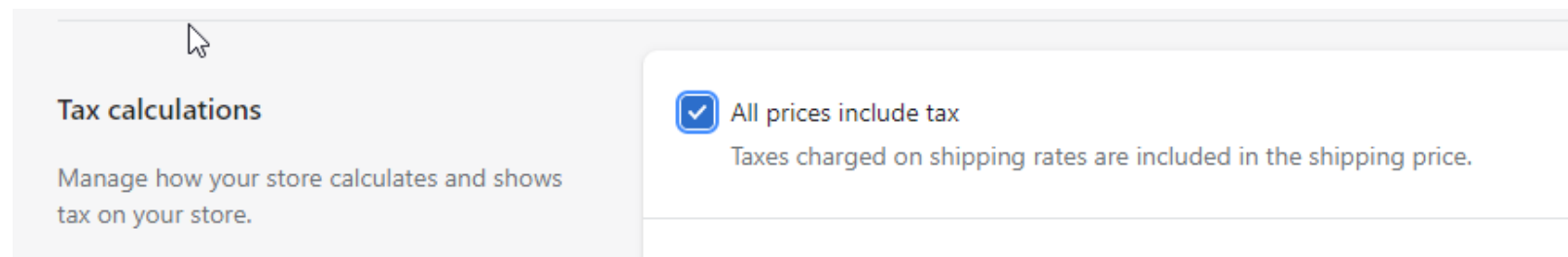
SUBSIDIARY ID

1

TAX ITEM

If your Shopify is Price Includes Tax, check the Price Includes Tax Checkbox in General Settings. This must align with Shopify for order totals to come in correctly.

Shopify, Under Settings > Taxes:



The screenshot shows the 'Tax calculations' section in Shopify. On the left, there is a sidebar with the title 'Tax calculations' and a description: 'Manage how your store calculates and shows tax on your store.' On the right, there is a main content area with a checked checkbox labeled 'All prices include tax' and a sub-description: 'Taxes charged on shipping rates are included in the shipping price.'

NetSuite settings:

## General Settings

API VERSION

2020-10

SUBSIDIARY ID

1

TAX ITEM

SHIPPING SET AS LINE ITEM

- ENABLE PRICE SYNC
- SYNC STOCK ON KIT ITEMS
- DO NOT SET TAX AMOUNT OVERRIDE
- PRICES INCLUDE TAX
- ENABLE SHOPIFY MULTI-CURRENCY SUPPORT

With Price Includes Tax, “Validate Taxes on CashSale/Invoice” Setting may not be needed.

VALIDATE TAXES ON CASHSALE/INVOICE

Then Save the Settings. This should be the basic Settings Setup needed to have the Integration working properly.

### Additional Settings

The following are the Other Setting Options that may be enabled by Default, Recommended, or Optional based on the Business Process needs.

**It is Recommended that if not Sure what a Setting Ultimately does or how it could negatively impact an Integration to please reach out to In8Sync Support.** [See Section 9.1 Support](#)

Setting	Details
Last Sync	Listing in UTC Time, is the Last time we Pulled Data Payloads from Shopify into the “Shopify Requests (Shopify to NS)” Queue; for processing
<p>LAST SYNC</p> <input type="text" value="09/29/2021 2:02:00 am"/>	
Created At (On or After)	Used to Stop the Data Puller from Pulling Orders in that were Created at Date was On or after the date of go live, or desired date. So not Past orders not wanting before then to come in and skew the NetSuite Accounting that may already have been complete by other order import or Journal entry and accounting adjustments
<p>CREATED AT (ON OR AFTER)</p> <input type="text"/>	
Last Order Sync (High Volume, Temp Sync Time)	Used for high Volume where each Page of orders we are pulling in we set the Date Time from Successful set of orders pulled, In case of any error to know where to pick up new time and not Pull orders over and over again due to any error.
<p>LAST ORDER SYNC (HIGH VOLUME, TEMP SYNC TIME)</p> <input type="text"/>	

<p>Last Customer Sync (High Volume, Temp Sync Time)</p>	<p>Used for high Volume where each Page of Customer we are pulling in we set the Date Time from Successful set of Customer pulled, in case of any error to know where to pick up new time and not Pull Customer over and over again due to any error.</p>
<p>LAST CUSTOMER SYNC (HIGH VOLUME, TEMP SYNC TIME)</p> <input data-bbox="212 347 772 396" type="text"/>	
<p>Last Product Sync (Updates)</p>	<p>Used when the setting “Pull Product Updates from Shopify” Is configured and enabled. To request only the New Products from Shopify Updated since Last time we pulled Product Data Updates</p>
<p>LAST PRODUCT SYNC (UPDATES)</p> <input data-bbox="212 568 779 617" type="text"/>	
<p>Stock Sync</p>	<p>Required to be enabled for Stock to Sync at all for Shopify Connected Items with Inventory Item Id; when Items Sync, Scheduled Update, Inventory Adjustments, or Shopify Stock Sync Button is hit.</p>
<p><input data-bbox="212 769 254 802" type="checkbox"/> STOCK SYNC</p>	
<p>Item on Transaction Today</p>	<p>Used for the Scheduled Stock Update Script. Limits the Items Qualified to Update Stock to Shopify, Based on if they were seen on ANY transaction Today.</p>
<p><input data-bbox="212 935 254 967" type="checkbox"/> ITEMS ON TRANSACTIONS TODAY</p>	
<p>On Transaction Since Last Sync</p>	<p>Two fields work in Tandem. When this is enabled with Scheduled Stock Update. Each time the stock is updated, the Last Sync on Transaction Date/Time is used to know the last time stock was update. The Stock Update would only then update stock for Items that were Seen on a Transaction Since the Last time the Stock Update Ran.</p>
<p><input data-bbox="233 1166 275 1198" type="checkbox"/> ON TRANSACTION SINCE LAST SYNC</p> <p>LAST STOCK SYNC ON TRANSACTION</p> <input data-bbox="233 1256 848 1305" type="text"/>	

Disable Pull Orders	By Default, when the Bundle is installed, the Puller will get any order Payload from Shopify and list in our “Shopify Request (Shopify to NS)” Queue. If the Mappings for “Customer” & “Sales Order” are inactive (inactive by default) the Payloads will not get processed and no orders are created. This setting When Live, enabled a quick setting to stop the pulling of new payloads for any reason under unchecked.
<input type="checkbox"/> DISABLE PULL ORDERS (COMING SOON)	
Order Item SKU Match, If no Ids	<p>When enabled, During Order Syncing to create in NetSuite, If the Item cannot be found by Shopify Ids, the sync will try to search NetSuite by the Item SKU, in order to find the item to put on the Order in NetSuite.</p> <p>Required to also List the Field ID that the SKU should be checked against from Item Record; "ORDER ITEM SKU MATCHING FIELD ID"</p> <p>If there is a “Order Item Specific Match” it will trump over this setting function.</p>
<input type="checkbox"/> ORDER ITEM SKU MATCH, IF NO IDS ORDER ITEM SKU MATCHING FIELD <input type="text"/>	
Entity New Address	
<input type="checkbox"/> ENTITY NEW ADDRESS	
Pull Customer Data from Shopify	When Enabled, Customers Only from Shopify are pulled to create/update with netsuite outside of Orders Pull.
<input type="checkbox"/> PULL CUSTOMER DATA FROM SHOPIFY	
Lead or Prospect on Create	<p>Requires the “Pull Customer Data from Shopify” to Be enabled. One or the Other can be used, but not both.</p> <p>When set, Customer sync from Shopify into NetSuite on Create of the Customer record, would set the Stage of the Customer to LEAD or PROSPECT.</p> <p>Once an order comes in the Customer in LEAD or PROSPECT stage, NetSuite would Automatically change the Stage to Customer</p> <p>Requires Parameter Mapping “lead” or “prospect” copied from Customer for use with this setting</p>

SET LEAD STATUS ON CREATE <input type="text"/>	
SET PROSPECT STATUS ON CREATE <input type="text"/>	
Create Fulfillment from Shopify	<p>This Setting when enabled will Pull Fulfillment from Shopify Order and create them on the corresponding Shopify Order in NetSuite.</p> <p>Used a Search in the Bundle to only request Fulfillment payload for Shopify orders that are not yet Fulfilled</p> <p>Used the Shopify Location Mapping to check what location to set for the Lines on the Fulfillment</p> <p>There is NO Serial Number Support as Shopify does not default have any details in orders for Serial #s</p> <p>Limited Lot# support</p>
<input type="checkbox"/> CREATE FULFILLMENT FROM SHOPIFY	
Enabled Lot# Inventory Details FIFO	When enabled, and we are creating fulfillments from Shopify, this will choose the Lot # record for the inventory Details with the lowest Internal ID. First in First Out (FIFO)
<input type="checkbox"/> ENABLE LOT # INVENTORY DETAILS FIFO	
Disable NS to Shopify Fulfillment Update	When Setting enabled the Item Fulfillment Sync back to Shopify will Stop. Used mainly in Conjunction with the "Create Fulfillment from Shopify" setting when we are creating NS fulfillments from Shopify
<input type="checkbox"/> DISABLE NS TO SHOPIFY FULFILLMENT UPDATE	
Pull Fulfilled Orders From Channels	<p>Since In8Sync Only pulls fulfilled orders from POS Channel, Other Channels that would have specific condition to need to be fulfilled before it is pulled AND the Setting "Create Fulfillment from Shopify" Is <b>not</b> Enabled, we can list the "source_name" Channel Identifier here, Comma Separated. Shopify would then pull fulfilled orders from the listed Channel as well as POS sourced orders</p> <p>If there is an APP in Shopify Fulfilling all Order in Shopify that we need to pull and create NetSuite Fulfillments, Please see setting "Create Fulfillment From Shopify" That that setting would pull all the Fulfillments from Shopify and create them against the Shopify NetSuite order and Auto Bill.</p>

<p>PULL FULFILLED ORDER FROM CHANNELS</p> <input type="text"/>	
<p>Pull Product Updates from Shopify</p>	<p>When enabled, this can pull Updates ONLY for Shopify Item Data Mapped in the Item UPDATE Parameters, for items that were already Synced Item Update Parameter Required: itemupdate</p>
<p><input type="checkbox"/> PULL PRODUCT UPDATES FROM SHOPIFY</p>	
<p>Image Folder</p>	<p>This is the File Cabinet Folder to store Image updates from Shopify To NetSuite to update the Items for updates when "Pull Product Updates from Shopify" is enabled and the images_&lt;item position&gt; (i.e. images_1) is mapped in "itemupdate" Parameter</p> <p>When Image Sync is enabled, we can only set the Main Images. Child Variation images are not mappable in Shopify as they only reference a Parent Image.</p> <p>When Syncing Images to the Main area, references to those get removed that were listed on Child items and those child item images are removed.</p> <p>LIMITS Shopify API recommendation is that images are no larger than:</p> <ul style="list-style-type: none"> <li>• 20MB</li> <li>• 2048x2048 px</li> </ul> <p>Shopify Does not like accepting URLs from images from Google Drive, Dropbox, and most 3rd party CDN</p>
<p>IMAGES FOLDER</p> <input type="text"/>	
<p>Enable Contact Sync</p>	<p>When Enabled: "Send to Shopify" Button will show up on Contact Records to sync Contacts to Shopify</p>

	REQUIRES: Contacts deployed against the "In8 Shopify Customer UE"
<input type="checkbox"/> ENABLE CONTACT SYNC	
Generic Product	If the order contains what is referred to as a "Write in" Item that technically does not exist in either Shopify or NetSuite, the Product on the line has no Shopify or Variation Id, this product will be used and the Shopify Item name will be listed in the NetSuite Line Item Description
GENERIC PRODUCT * <input type="text"/>	
Default Customer	This is required in case the order has no customer data, it will use the default customer listed here
DEFAULT CUSTOMER * <input type="text"/>	
Discount Item	When listed, this Discount Item will be used to list the Discount per line level discount on Shopify orders on the NetSuite Orders. This should be a Posting Discount
DISCOUNT ITEM * <input type="text"/>	
Line Discount, Subtract from Line	By Default, Line level Discounts add additional Line with Discount under the Item Lines. When this is enabled it will Simply subtract the Discount from the Item line itself
<input type="checkbox"/> LINE DISCOUNTS, SUBTRACT FROM LINE	
Shipping Discount Item	This item is meant to be a Separate Discount Item used strictly when a Shopify Discount that is Shipping Discount is used and will be set on line as discount item that can have a separate account assigned to it for Shipping Discount
SHIPPING DISCOUNT ITEM * <input type="text"/>	



Subsidiary ID	<p>When Subsidiary is Listed here, it enables the Customer Search by email to also include the Subsidiary as part of the Search with email. To make sure we check for the Right customer on the right Subsidiary</p> <p>When Multi-Subsidiary Customer is enabled, we will only look for the Email and if customer Subsidiary is different that this, we will set this subsidiary in the Subsidiaries Sublist on the customer to make sure the order can be created in the right Subsidiary</p>
<p>SUBSIDIARY ID</p> <input data-bbox="216 461 772 509" type="text" value="1"/>	
Tax Item	<p>When enabled, the Tax Item listed, will be used to set the Total Tax at the Line Item Level. It will be 1 line item for the whole order. This will not use "taxamountoverride" as normal on orders since it will be setting per the 1 line item.</p>
<p>TAX ITEM</p> <input data-bbox="216 659 772 711" type="text"/>	
Shipping Set as Line Item	<p>When you list a Item here, it will be used on Orders, to Set Shipping Cost on the Line Item vs the Default Shipmethod and ShipCost.</p> <p>Description will host the name of the shipping method from Shopify</p>
<p>SHIPPING SET AS LINE ITEM</p> <input data-bbox="216 863 772 915" type="text"/>	
Option Attribute Fields for Product Variants	<p>Enter a json specifying the Fields and the Names for the Attributes on the Matrix items that will sync. i.e.</p> <pre data-bbox="793 1008 1381 1198">[ {"field": "custitem3", "name": "Fragrance"},   {"field": "custitem1", "name": "Size"},   {"field": "custitem2", "name": "Color"},   {"field": "custitem10", "name": "Lights"},   {"field": "custitem9", "name": "PottyPet Variation"} ]</pre> <p>Checking JSON against <a href="https://jshint.com">https://jshint.com</a> can help make sure that this is correct</p>

OPTION ATTRIBUTE FIELDS FOR PRODUCT VARIANTS	
<pre>[ {"field": "custitem41", "name": "color"}, {"field": "custitem19", "name": "size"} ]</pre>	
Overwrite Variants	<p>Not Recommended to enable.</p> <p>When enabled, When the variants are synced with NetSuite Parent product to Shopify during item sync, if the item is not sent to the Shopify product variants with latest NetSuite data then Shopify will Delete that Variant Item from Shopify.</p> <p>We will also remove the Variant Id listings from that Item from under the In8 Shopify Ids sublist from Custom SubTab in NetSuite, in order to allow for syncing again in the future.</p> <p>In Short, Unchecking the Sync to Checkbox on the Variant items will cause the Variant item to not sync to Shopify and also be deleted.</p>
<input type="checkbox"/> OVERWRITE VARIANTS	
Enable Auto Item Sync on Create/Edit	<p>When enabled, when NetSuite Items are Marked to Sync to Shopify on Create or Edit through the UI, the item will attempt to sync to Shopify automatically. Per conditions to Sync</p> <p>Currently in DEV to be added as Feature</p>
<input type="checkbox"/> ENABLE AUTO ITEM SYNC ON CREATE/EDIT (COMING SOON)	
Enable Price Sync	<p>Coming SOON,</p> <p>To enable the use of Pricing only sync scheduled update to Shopify</p> <p>Will be ending up in our Setup/Tools Page as new feature</p>
<input type="checkbox"/> ENABLE PRICE SYNC	
Stock Sync on Kit Items	<p>When enabled we will request QTY from all Components on the Kit items. Whatever Component has the Lowest Stock QTY, we will use that for the Stock level to Sync to the Item in Shopify during Stock Script updates. Setup Tab &gt; In8sync Shopify &gt; Tools &gt; Stock Sync</p> <p>By default when this is Disabled, On Item Sync, it sets the Shopify Item to Track Quantity =</p>

	False When this is enabled, On Item Sync, it will set the Shopify Item to Track Quantity = True
<input checked="" type="checkbox"/> SYNC STOCK ON KIT ITEMS	
Do Not Set Tax Amount Override	When enabled, we will not set Taxamountoverride to the NetSuite order like default. Price Includes Tax by default does not set Taxamountoverride
<input checked="" type="checkbox"/> DO NOT SET TAX AMOUNT OVERRIDE	
Price Includes Tax	This setting should only be enabled with the Shopify Setting is also enabled for Price includes Tax under the Tax Settings. This setting when enabled puts the amount of the item (inclusive tax) on the Gross Amount Field on the Line items, does not do tax amount override and will also put the Discounts in as Line items.
<input type="checkbox"/> PRICES INCLUDE TAX	
Enable Shopify Multi-Currency Support	When enabled, for those account in Shopify with Multi-Currency enabled, this setting checks the presentment_currency on the order payload and gets the presentment_currency level to Align the totals listed on NetSuite order the same as shown on the Shopify Order. Customer Parameter Mapping for NetSuite Currency from the order.presentment_currency & Sale Order Parameter for NetSuite Currency from presentment_currency should be mapped with JSON to set the Currencies in NS by Internal id with the Shopify Currency text
<input type="checkbox"/> ENABLE SHOPIFY MULTI-CURRENCY SUPPORT	
Create Customer Deposit	When enabled, the mapped payment method is ripped off the Sales order and applied to a Customer Deposit on the Created Sales Order to be auto applied on invoice when billing the Sales Order
<input type="checkbox"/> CREATE CUSTOMER DEPOSIT	
Auto Bill on Fulfillment	When enabled, the Shopify Synced order when Moving to Pending Billing after being Fully Fulfilled with Transform the order to bill and create the Invoice or Cash sale as needed  Default NetSuite: when Payment Method is listed on the Sales Order, Cash Sale is Generate. When no Payment Method listed on the Sales order, Invoice is generated

<input checked="" type="checkbox"/> <b>AUTO BILL ON FULFILLMENT</b>	
Validate Taxes on Cash Sale/Invoice	When enabled, When Cash Sale or Invoice is Generated, we will enforce the Tax Total with "taxamountoverride" to match what the full Sales Order Tax was.  This is a Full Tax from Sales Order
<input type="checkbox"/> <b>VALIDATE TAXES ON CASHSALE/INVOICE</b>	
Enabled Return Suitelet	Mainly Enabled if Advanced Shopify Portal for Customer initiated Returns directly to NetSuite to create RMA against the Sync Shopify Orders, is completed.
<input checked="" type="checkbox"/> <b>ENABLE RETURN SUITELET</b>	
Do Not Notify Customer on Shipment	If this check box is checked, globally it will tell Shopify to not send an email notification to the customer when item fulfillment is synced to the Shopify, on initial fulfillment Creation
<input type="checkbox"/> <b>DO NOT NOTIFY CUSTOMER ON SHIPMENT</b>	
Allow Fulfillment Updates to Shopify	When enabled, globally any item fulfillment updates (edit and save on fulfillments already sent) in NetSuite will update tracking and Carrier data in Shopify. Also Available Per Fulfillment on the Item Fulfillment Record
<input type="checkbox"/> <b>ALLOW FULFILLMENT UPDATES TO SHOPIFY</b>	
Do Not Notify on Shipment Updates	When enabled, globally any item fulfillment updates (edit and save on fulfillments already sent) in NetSuite will NOT notify the customer of the update in Shopify. Updates would notify the customer by Default Also Available Per Fulfillment on the Item Fulfillment Record
<input type="checkbox"/> <b>DO NOT NOTIFY ON SHIPMENT UPDATES</b>	
Do not Set Shipping Method on Update	By default, the integration pulls order updates to NetSuite. When enabled, the Shipping Method will not get update when order Updates come into NetSuite. Allowing, any changed to Shipping Method on the Created order to not change
<input checked="" type="checkbox"/> <b>DO NOT SET SHIPPING METHOD ON UPDATE</b>	
Location to Send QTY on Hand for stock	The locations that are selected, that are also configured in the integration mapping, will look to the Quantity on Hand for Stock level VS the normal Quantity Available Quantity Available is always recommend as it is the Stock level that transactions Use to Commit stock

	Typically used in Combination with Advanced Configuration for Special requests
<p>LOCATIONS TO SEND QTY ON HAND FOR STOCK</p> <ul style="list-style-type: none"> <li>Amazon Sellable</li> <li>Amazon Seller Central</li> <li>Amazon Seller Central W#1</li> <li>Damaged Warehouse</li> </ul>	
	When choosing a location that we have mapped to send, we will take the Backorder QTY level and send to Shopify as a Negative QTY Typically used in Combination with Advanced Configuration for Special requests
<p>LOCATIONS TO SEND BACKORDER QTY AS NEGATIVE</p> <ul style="list-style-type: none"> <li>Amazon Sellable</li> <li>Amazon Seller Central</li> <li>Amazon Seller Central W#1</li> </ul>	
Enable Image Sync	When enabled, NetSuite Field "Item Display Image" (storedisplayimage) will sync over as 1st image. This field will not have to be mapped Then, if listed, the additional "Item Image Field ID" images will be sent as well, in the order listed
<input type="checkbox"/> ENABLE IMAGE SYNC	
Item Image Fields	Enter comma separated, image field ids from item record type, If you have more than one url fields to sync.
<p>ITEM IMAGE FIELDS</p> <input type="text"/>	
POS Generic Customer	Used when enabling the POS feature. The Customer Listed, in the same Subsidiary as the Site, will be used IF the Shopify POS order has no Customer listed on it.

POS GENERIC CUSTOMER <input type="text"/>	
POS Auto Fulfill	Should be enabled if Pulling Shopify POS orders
<input type="checkbox"/> POS AUTO FULFILL	
POS Lot Fulfillment FIFO	When enabled, and we are creating fulfillments from Shopify for POS orders, this will choose the Lot # record for the inventory Details with the lowest Internal ID. First in First Out (FIFO)
<input type="checkbox"/> POS LOT FULFILMENT FIFO	
No Fulfill IF Shipping	When Enabled, For POS order Listing Shipping Method (resulting with Shipping Lines in Order Payload) we will not Auto Fulfill. Allowing Fulfillment and Shipping Manually in NetSuite for the POS Order
<input type="checkbox"/> NO FULFILL IF SHIPPING	
POS Auto Bill	Should be enabled is Pulling Shopify POS orders, to auto bill the NetSuite Order Due to "Cash and Carry orders"
<input type="checkbox"/> POS AUTO BILL	
Enable NON-POS Refunds (NS to Shopify)	When enabled (enabled by default), for Orders Not Sourced from POS Shopify orders. This allows the Refunded Amount on Customer Refunds and Cash Refund, Generated from Shopify Orders, to trigger the Shopify Gateway to refund the amount seen on the Customer Refund or Cash Refund If this is disabled, but you want to send the Customer Refund or Cash Refund amount to Shopify, that is not handled with other integration for returns like Loop or Returnly, there is a Field per record "In8 Send Refund Amount To Shopify" on the Customer Refund and Cash Refund that can be checked to send only that Records Refund amount to Shopify.
<input checked="" type="checkbox"/> ENABLE NON-POS REFUNDS (NS TO SHOPIFY)	
Refund Reason Field ID	The Field ID listed, that exists on the RMA/Credit Memo Record or the Cash Refund Record Created from a Shopify Order, will update the Value from the Field to the Return For Reason on the Refund to Shopify; for NON-POS Refunds

REFUND REASON FIELD ID	
<input type="text" value="custbody_so_notes"/>	
Enable NON-POS Refunds (Shopify to NS)	When enabled, we will pull Refunded & Partially Refunded Status Orders, to adjust the Sales Order line when Per-Fulfillment, and perform Refund against POST Fulfilled Orders.  Recommended to have "Auto Bill on Fulfillment" enabled with this Feature
<input checked="" type="checkbox"/> <b>ENABLE NON-POS REFUNDS (SHOPIFY TO NS)</b>	
NON-POS Do not Refund Billed Orders	Not Recommended to be enabled, If enabled and the NetSuite order is billed, it does not create the Refund against the order.
<input type="checkbox"/> <b>NON-POS DO NOT REFUND BILLED ORDERS</b>	
Shopify > NS General Refund Item	This item is required when the setting " ENABLE NON-POS REFUNDS (SHOPIFY TO NS)" is enabled. If the Shopify Refund is not Refunding an Item, but is a General Amount Refund, this Item is used on the Refund Record for the General Refund. If the Item being Refunded it "NON-Restocking" it will also be used to make sure the Item does not Restock in NetSuite. Item Type Support is "Other Charge For Sale", Preference on this Item should be "Can Be Fulfilled = F" and Taxable = F or Tax schedule "Non Taxable"
SHOPIFY > NS GENERAL REFUND ITEM	
<input type="text" value="OthChargeForSale"/>	
Enable Cancelled Order (Shopify to NS)	When enabled without 'Enable NON-POS Refunds (Shopify to NS)' order cancelled that are not Refunded, it will close the NetSuite Sales Order if not fulfilled or billed.
<input checked="" type="checkbox"/> <b>ENABLE CANCELLED ORDERS (SHOPIFY TO NS)</b>	
Enable POS Refunds	Requires POS Auto Fulfill and POS auto Bill to be enabled. When Enabled, Orders that are POS source and with statuses Refund and Partial Refund can be pulled into NetSuite When a Refund is pulled in: -We will create a Cash Refund against the Cash Sale, Or Credit Memo and Customer Refund If the Original Order generated an Invoice When a Exchange, indicated in Shopify Payload by exchange-credit payment gateway -We will generate Return Authorization, Item Receipt, and Credit memo --If customer getting money back we will also generate the Customer Refund for that


	<p>amount</p> <p>-For the money used for the item being purchased in exchange, the Credit Memo under the In8Sync will have indication to what order the credit memo is from and what order the Credit memo should be applied to.,</p>
<p><input checked="" type="checkbox"/> ENABLE POS REFUNDS</p>	
Promotion Return	<p>Listing a Template here will enable the Promotion Return Function to Close the RA in question and create a NetSuite promotion that will sync to Shopify a Coupon Code for use for the Customer on the RA. Email will be sent from NetSuite and Listed on the RA customer Communications.</p> <p>On the RA if you list a Different email under the IN8sync SubTab for the Promotion Email it will send to that email.</p> <p>Requires the "Promotion Discount Item" &amp; "Promotion Return From Email" User</p> <p>Request to In8Sync Support for Full setup and Testing Recommended</p>
<p>PROMOTION RETURN TEMPLATE</p> <p>In8 Shopify Promotion Email Template ▼</p> <p>PROMOTION DISCOUNT ITEM</p> <p>DIS00000 ▼</p> <p>PROMOTION RETURN FROM EMAIL</p> <p>George Lasater ▼</p>	
Cancel Shopify Order   Sales Order CLOSED	<p>Global Setting</p> <p>When enabled and a Sales Order is Closed, it triggers a Status update to Cancel the Shopify Order. For NON-POS orders.</p> <p>This is not Recommended to be enabled Globally, unless fully aware of function.</p> <p>If Shopify to NetSuite Refunds is enabled, where some lines being closed would change status of NetSuite order to CLOSED even when all Items are not Closed, could cause issues.</p> <p>There is a Per Sales Order Option Under the In8Sync SubTab for individual use.</p>
<p><input type="checkbox"/> CANCEL SHOPIFY ORDER   SALES ORDER CLOSED</p>	
Cancel Fulfillment with Order Cancel	<p>Global Setting</p> <p>When enabled: This will add process to Cancel the order Fulfillment in order to Cancel the Order in Shopify when a NetSuite Order is Moved to CLOSED Status.</p>



	<p>Otherwise, we can only Cancel Shopify Orders that are Unfulfilled.</p> <p>There is a script on Sales Order Records, that when the Sales Order Moves to CLOSED, it will trigger to Cancel the Shopify Order.</p> <p>If Shopify to NetSuite Refunds is enabled, where some lines being closed would change status of NetSuite order to CLOSED even when all Items are not Closed, could cause issues.</p> <p>There is a Per Sales Order Option Under the In8Sync SubTab for individual use</p>
<input type="checkbox"/> CANCEL FULFILLMENT WITH ORDER CANCEL	
Refund On Cancelled	Currently not function in Bundle as Feature
<input type="checkbox"/> REFUND ON CANCELLED (CURRENTLY IN DEV)	
100% Match Customer (No Search/Create)	<p>When this setting lists a Customer record, it disables our Customer Search/Match and Create Functions. Always Matching this Customer during the Order Sync Process in NetSuite. So, all Orders Match and Sync Under This Customer.</p> <p>Only Addresses would update to the Address List, if the setting "100% Match Customer New Addresses" is enabled</p>
100% MATCH CUSTOMER (NO SEARCH/CREATE) <input type="text" value=""/>	
<input type="checkbox"/> 100% MATCH CUSTOMER NEW ADDRESSES	
Pull Manually Create Gift Card	<p>Check this box if you want to pull manually created gift cards from Shopify .You can view the pulled Shopify Gift Cards in Setup &gt; In8Sync Shopify &gt; Manual Gift Cards Queue.</p>
<input checked="" type="checkbox"/> PULL MANUALLY CREATED GIFT CARDS	
GC Last Sync	<p>This field is automatically updated by the Shopify bundle. Value here indicates the last time manually created gift cards were pulled from Shopify.</p> <p>This is listing Time on the NetSuite account Timezone</p>
GC LAST SYNC <input type="text" value="2/26/2021 12:58:39 pm"/>	

Gift Card Item	Select the Gift Card Item (Type should be Gift Certificate) you want to use when creating Cash Sales for Shopify Gift Cards.
<p>GIFT CARD ITEM</p> <p>Shopify Gift Card ▼</p>	
GC Discount Item	<p>Select the Gift Card Discount Item (Type should be Discount) you want to use when creating Cash Sales for Shopify Gift Cards.</p> <p>The selected discount item will be applied with discount rate of 100%. Otherwise, if not set, no discount item will be applied to the Cash Sale.</p>
<p>GC DISCOUNT ITEM</p> <p>DIS00000 ▼</p>	
Returnly GC Discount Item	<p>When Shopify has Returnly APP installed and issuing Manual Gift Cards, this Discount Items will be used when configured to allow for Different account for those. If not Configured the "GC Discount Item" will be used</p>
<p>RETURNLy GC DISCOUNT ITEM</p> <p>▼</p>	
Enable Shopify Payment Reconciliation	<p>Check this box if you want to pull Shopify Payments for reconciliation. You can view the pulled Shopify Payments Reconciliation in Setup &gt; In8Sync Shopify &gt; Payment Reconciliation.</p>
<p><input checked="" type="checkbox"/> ENABLE SHOPIFY PAYMENT RECONCILIATION</p>	
NO. of Business Days	<p>Enter the number of business days to pull payouts from. Example: Today is Monday 2 - Pull payouts made inclusively after the last 2 business days (Thursday last week).</p>
<p>NO. OF BUSINESS DAYS</p> <p>2</p>	
Payment Method and Clearing Account	<p>This is the payment method used for Shopify Payments. Shopify bundle automatically sets this value based on the mapped payment method for Shopify Payments.</p> <p>This is the Account used in the Shopify Payment Method. Shopify bundle automatically sets this value based on the G/L Account associated with your Shopify Payment Method.</p>

	Also Set on the generated BILL Payment Record Account
<p><b>PAYMENT METHOD</b>          Visa (Shopify Visa) ▼</p> <p><b>CLEARING ACCOUNT</b>          1007 Shopify Payments (Clearing Account)</p>	
<b>Fees Account</b>	Select the Account you want to use when creating Vendor Bills for Shopify Payment fees.
<p><b>FEES ACCOUNT</b>          6130 Miscellaneous Expense ▼</p>	
<b>Bank Account</b>	Select the Account you want to transfer the billed fees amount to. This will be used in the Journal Entry created after processing Shopify Payment Reconciliation.
<p><b>BANK ACCOUNT</b>          1000 Checking ▼</p>	
<b>Do Not Tax Shipping</b>	This setting is for use if SuiteTax is enabled and we need to stop Taxing Shipping Globally per this Site setting
<input type="checkbox"/> <b>DO NOT TAX SHIPPING</b>	
<b>On Account Gateway Name</b>	<p>When Adding the On Account Payment Gateway name from Shopify under the Payment Providers (typically is a configured Manual Payment Gateway), this enables the option for NetSuite to Pull "Payment Pending" Orders that have the Gateway Listed as the "On Account Payment Gateway Name".</p> <p>This also supports possible Multiple On Account Payment Gateway Name. Separated in the field by Comma.</p> <p>If the Payment Gateway name does not exist on the Payment Pending Shopify order, we do support using "null" as an option to list</p>
<p><b>ON ACCOUNT GATEWAY NAME</b>          On Account</p>	
<b>On Account Terms</b>	<p>When Listing the Terms for On Account here, it will set this Terms on all Order.</p> <p>If this left blank, the Term on the Order will be listed based on NetSuite Default Behavior to pull from the Customer if listed on the Customer Record, under the Financial SubTab</p>

ON ACCOUNT TERMS	
Due on receipt 	
Pull Authorized Orders	When enabled, we will also pull orders in Authorized status. Additionally, when the Orders in NetSuite are Billed we will send back through API, a trigger to Manually Capture the Orders in Shopify to complete the payment. WARNING: Shopify has authorization expirations that should be considered
<input checked="" type="checkbox"/> PULL AUTHORIZED ORDERS	
Order Risk	When Enabled, We will make an additional Call on Order Risk API to pull the Order Risk data and list is in the Sublist "In8 Shopify Order Risk" Found under the Sales Order Custom SubTab Enabled by Default, for reference only
<input checked="" type="checkbox"/> ORDER RISK	
Shopify Fraud Max Risk Score	The closer the score is to 1.0, the more likely it is that the order is fraudulent. Enter the maximum (between 0.0 and 1.0) to determine if the order should be imported up to the set Risk Score set. This Should only be configured to work in Unison with Shopify's Fraud setting to when the Shopify order is Auto Cancelled, so we do not create the same orders Shopify will Auto Cancel
SHOPIFY FRAUD MAX RISK SCORE	
<input type="text"/>	
Kount Accepted Statuses	This option gets enabled when you list the Status that are acceptable to Pull into NetSuite. Please list the Status that we should pull into NetSuite (comma Separated) Available Statuses: accept (Approved) investigate (Review) cancel (Decline) escalate (Escalate)  Shopify Kount APP should operate the same way to Cancel the orders that we are not creating in NetSuite

KOUNT ACCEPTED STATUSES	
<input type="text"/>	
Pricing Settings	These are Advanced Customization requiring Scoping to install Shopify Custom App from In8Sync and Customizations to Shopify Theme to control Pricing changes on Product, collections, Cart and Checkout to give pricing from NetSuite
<input checked="" type="checkbox"/> ENABLE ITEM PRICING <input checked="" type="checkbox"/> ENABLE GROUP PRICING <input checked="" type="checkbox"/> ENABLE WHOLESALE QUANTITY PRICING <input checked="" type="checkbox"/> ENABLE WHOLESALE PRICING <input checked="" type="checkbox"/> ENABLE QTY PRICING FOR EVERYONE	
Price Levels to Sync	To get the Price for the Price Levels, The Customers in NS with the Price Levels Configured must be synced to Shopify to get a Price Level Tag specific to their Price Level i.e. price_1 These Price Levels are the Approved Price Levels to Send over on Items
PRICE LEVELS TO SYNC <input type="checkbox"/> 10% Discount Level <input type="checkbox"/> 100% Sample Pricing <input type="checkbox"/> 15% Discount Level <input type="checkbox"/> 5% Discount Level <input type="checkbox"/> Base Price	

<p>Order Item Specific Matching</p>	<p><b>ORDER ITEM SPECIFIC MATCH NOTE</b>          When configuring the Order Item Match Sublist, the function is used when the order Payload is being processed to check if any SKU in the Shopify Payload should be skipped from Check for Shopify Ids and Just use the NetSuite Item Listed for to use instead of IDs matching.          Example:</p> <ul style="list-style-type: none"> <li>• Operator = contains</li> <li>• Shopify SKU = Routeins (Not Case Sensitive)</li> <li>• NetSuite Item = &lt;The NetSuite Item to be used&gt;</li> </ul> <p>In this case is the Order Item Line Contains a SKU with Routeins, we would use the Listed NetSuite Item Instead of Checking for a NetSuite Item by Shopify Ids</p>	
-------------------------------------	--	--

In8 Shopify Order Item Specific Match In8 Narvar Settings • System Notes •

New In8 Shopify Order Item Specific Match

NETSUITE OPERATOR FOR SHOPIFY SKU	SHOPIFY ITEM SKU	NETSUITE ITEM TO USE
<div style="border: 1px solid #ccc; padding: 2px;"> <input type="text" value="contains"/> </div> <ul style="list-style-type: none"> <li>- New -</li> <li>is</li> <li>contains</li> </ul>	<input type="text" value=""/>	<input type="text" value="&lt;Type then tab&gt;"/>

Insert Remove Change ID Actions

## 5.5 Mappings

Parameters for Mapping Are required for Things to Operate. Orders created, Customer created, fulfillments sent etc...

Without the Parameters Active as needed, nothing is active to pull and create Orders, or sync Items or Customers

Customer and Sales Order are required to be Active to Create Orders Pulled from Shopify

Item Fulfillment is required to send Fulfillments to Shopify

The Following are Parameters that can be used in Shopify Integration:

- **customer (Listed by default, inactive)**
- **contact (listed by default, inactive, use when Setting “Enabled Contact Sync” is on to sync Contact records to Shopify)**
- lead (used with Setting “Set Lead Status on Create”, for specific mappings on create only for creating Lead)
- prospect (used with Setting “Set Prospect Status on Create”, for specific mappings on create only for creating Prospect)
- items (cannot sync item types if specific item type parameter does not exist)
  - **inventoryitem (Listed by default, inactive)**

- assemblyitem
- lotnumberedasassemblyitem
- serializedasassemblyitem
- kititem
- noninventoryitem
- lotnumberedinventoryitem
- serializedinventoryitem
- serivceitem
- otherchargeitem
- **itemfulfillment (Listed by default, inactive)**
- **salesorder (Listed by default, inactive)**
- orderitems - (used to map line level fields on sales order for items)
- Promotioncode - (used when wanting to sync NetSuite Standard Promotion to create discount codes in Shopify)
- **cashrefund - (needed for POS returns) (Listed by default, inactive)**
- **cashsale - (needed for Standalone POS returns) (Listed by default, inactive)**
- gccashsale - (used only for Manual Gift Card processing; Fixed Value support only)
- gcorderitems - (used only for Manual Gift Card processing; Fixed Value support only)
- itemupdate - (used only product update when enabled to what to map data from Shopify NS)
- p-out-bill - (Payout Reconciliation ONLY; Fixed Value only “Vendor Bill” Header)
- p-out-bill-expitems - (Payout Reconciliation ONLY; Fixed Value only “Vendor Bill” Lines)
- p-out-billpayment - (Payout Reconciliation ONLY; Fixed Value only “Vendor Bill Payment” Header)
- p-out-journal - (Payout Reconciliation ONLY; Fixed Value only “Journal Entry” Header)
- p-out-journal-items - (Payout Reconciliation ONLY; Fixed Value only “Journal Entry” Lines)

Below are the Parameter mappings that are installed and inactive by default. The parameters control the flows working when active.

Item (inventoryitem), Customer, Salesorder, and Item Fulfillment parameters, once configured and covering additional Mappings, are required to be enabled and active for the standard integration to operate. Otherwise, the Data Payloads from Shopify under the [Shopify Requests](#) coming into NetSuite will just error message that Customer and/or Sales Order Mappings are not enabled & Fulfillments created in NetSuite and Shipped will not Sync back to Shopify to fulfill the Shopify orders.

### cashrefund

cashrefund parameter mapping is created and inactive by default.

Used for POS refunds to create Cash Refunds against the original POS order. Typically has the same additional Mappings that the Sales Order Parameter would have been setup with.

### cashsale

cashsale parameter mapping is created and inactive by default

Used for POS refunds to create Cash Sale then the refund to record to the Order and Refund, when at the time, the original Sales order is not found. Typically, due to the Order and Refund being performed so close together when the Integration pulled the order it was already refunded.

### customer

Customer parameter mapping is created and inactive by default.

This parameter mapping is used for Syncing Customer records from NetSuite to Shopify and Connecting Shopify Customers to NetSuite for creation and update.

Default Sync updates the customer fields:

- Type = Individual (isperson = T)
  - Shopify Default Customer Record Type is Individual
- First and Last Name
  - NetSuite Requires both these Fields
- Email
- Phone
- Addresses
- Subsidiary
  - When OneWorld NetSuite account, to set the customer to the subsidiary necessary for placing the orders under the correct Subsidiary.
  - Confirmation on the Subsidiary to be mapped by Internal ID, Fixed Value, required



Field Mappings <span style="float: right;">Notes</span>						
In8 Shopify Mappings Fields • <span style="float: right;">Shopify Criteria</span>						
New In8 Shopify Mappings Fields						
NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE	GET FIELD VALUE	CUSTOM FIELD	ORDER
custentity_in8_shop_id		id				
email		email				
firstname		first_name				
lastname		last_name				
comments		note				
phone		phone				
subsidiary			1			2
isperson			T			
taxable			T			

Additional fields, i.e. Department, Class, Category etc... can be mapped per customer business flow requirements. Further Advanced mappings for business flows; please contact [In8sync Support](#)

For how the integration finds, searches and connects to NetSuite Customer on Create/Update, please see [Customer Flow](#)

Below are some Customer Mappings for required or option field as per integration needs.

#### [Customer Mappings from Order Payload Fields](#)

In order to map a Customer Mapping Value or JSON mapping to update a Customer during order sync and using the Order Fields from Payload, the "order." and be applied as prefix to the Shopify Field mapping to source the value in question from the Order Data and not the Customer Data values.

This example is sourcing the Order "source\_name" in order to set a Custom Segment for Sales Channel.

This example is also making use of the New Json Mappings to set the NetSuite Field ID: cseg1 internal id values based on the order\_source of the order

NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE	GET FIELD VALUE	CUSTOM FIELD	ORDER	JSON MAPPINGS
custentity_in8_shop_id		id					
email		email					
firstname		first_name					
lastname		last_name					
phone		phone					
subsidiary			2			2	
isperson			T				
taxable			T				
comments		note					
customform			87			1	
category			1				
cseg1		order.source_name					{ "ourCommonplace": "4", "verishop": "4", "default": "1" }

## inventoryitem

inventoryitem parameter mapping is created and inactive by default.

This Parameter and Mapping is the main Crux of Integration. Without NetSuite Item Setup done correctly and confirm for connection with Shopify Products, the rest of the integration for creating sales orders will not properly be able to find and set the correct item in NetSuite against the Sales orders from Shopify. The setup ensures that the Standard integration works properly in:

1. Syncing Items from NetSuite to Shopify
  1. Obtaining the Shopify Ids back to NetSuite on the Items under the In8Sync SubTab for the Items to be found when Order Processing occurs to put the correct NetSuite Items on the Sales Orders
2. Orders Pulled to NetSuite created correctly with the item in NetSuite.
  1. Shopify Order Items have Shopify ID and Variation ID that correspond to the Correct item being sold and would need to be on the NetSuite Product to know what item is being used on NetSuite Sales Order.

On Bundle installation, all default Mappings connect with the Shopify Standard Fields in our Bundle for Shopify Products, under the In8Sync SubTab on the NetSuite Item Record. These mappings MUST be confirmed for what NetSuite Fields are to be Kept and/or updated to be the source of the data in NetSuite that will be sent to the Shopify Product fields, for the Item Parameters to be enabled and Product Sync to work

properly. The Bundle Fields can be kept and used to load the needed Data for the Fields to Sync to Shopify or other NetSuite Fields can be used to be mapped for the Data.

## Field Mappings [Notes](#)

[In8 Shopify Mappings Fields](#) • [In8 Shopify Metafields Definition](#) • [Shopify Criteria](#) •

### New In8 Shopify Mappings Fields

NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE
custitem_in8_shop_id		product_id	
custitem_in8_shop_title		title	
custitem_in8_shop_id		id	
custitem_in8_shop_description		body_html	
custitem_in8_shopify_tags		tags	
custitem_in8_shop_weight		weight	
custitem_in8_shop_weighunit		weight_unit	
		published_scope	web
custitem_in8_publish_shop		published	
custitem_in8_shopify_vendor		vendor	
custitem_in8_shop_produ_type		product_type	
custitem_in8_shopify_vendor		vendor	
custitem_in8_shop_barcode		barcode	
custitem_in8_shop_sku		sku	

Field Mappings		
FIELD	OPERATOR	VALUE
ustitem_in8_sync_shopify	is	true

Locations mapped are the only ones approved to have stock updated to Shopify, to the Shopify location by Id in [Location Mappings](#).  
 Price Level mapping required to send and update the Regular Price to the Shopify Product.  
 Currency mapping required to know what currency the Price Level Price is sourced from.

### Item Information

#### LOCATION

- Austin Retail
- Factory
- Houston Retail

#### CURRENCY

#### PRICE LEVEL

Please see [Product Flow](#) for the Minimum required fields and optional fields to be mapped. Any uncertainty to how to map or source this data from NetSuite to Shopify, NetSuite as Master of Mapped Fields, please contact [In8sync Support](#)

What is Mapped to be managed in NetSuite is Master of NetSuite to Manage. What is not Mapped in NetSuite to be Managed will be Managed in Shopify Direct.

In8Sync, through Project Integration Setup would typically review and update the Item Forms to be organized and source the data that is syncing to Shopify under the In8Sync SubTab on the Item to better review and know what data is being managed through the Integration.

Item Sync by default makes use of a Checkbox, “Sync to Shopify” by Default, to allow which items to actually traverse the Data flow and reach Shopify. In many cases, like multiple Shopify site Integrations, there will be Custom Checkbox Fields Created to control the items going to which site.

Below are other outlined mappings that are required or optional for integration needs

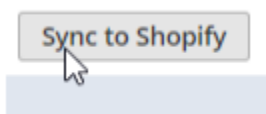
### Shopify Item Sync Triggers

For Item to Sync at All, their Configured Checkbox to “Sync to Shopify” (Default), must be checked. Depending on the Account, there might be New Checkbox to Sync to Shopify Per the account, especially if there are multiple Site. Usually Found under the In8Sync SubTab of the Item Records, per the Integration setup process.

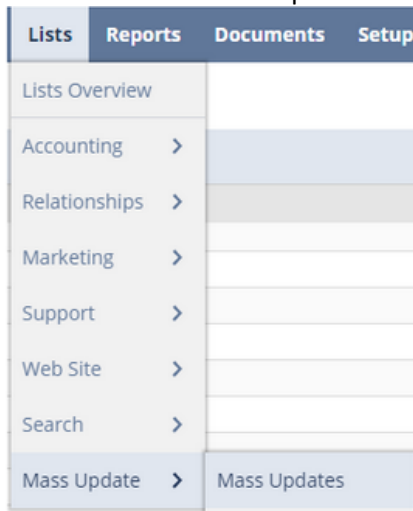
When Syncing Matrix Items, at least 1 Parent Matrix and 1 Child Item must have their Sync to Shopify Checkboxes marked = T. Of course all children that need to Sync should also have their Checkboxes = T.

Once Items are Marked to Sync the Following are Trigger to sync the Items to Shopify.

1. Directly Press “Sync to Shopify” on the item itself

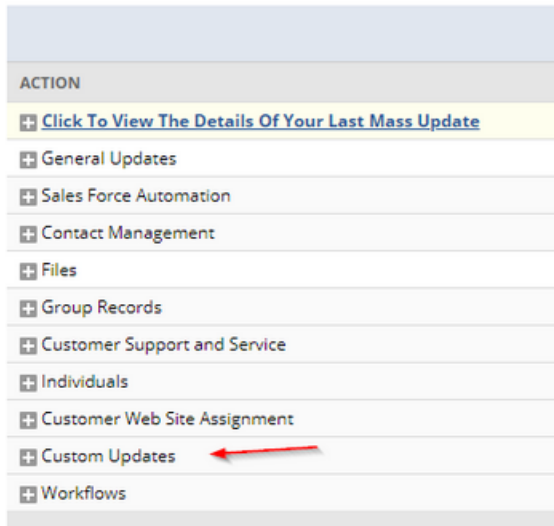


2. From Lists Tab > Mass Update > Mass Updates

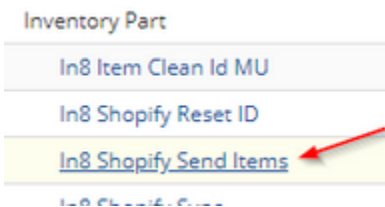


1. Under the menu section “Custom Updates”

## Mass Updates



2. Chose the “In8 Shopify Send Items” update



3. Filter Criteria of what to sync and preform the update (save for later if you wish)
  1. Mass Update Template for Finding New items with Sync to Shopify (differs per account), excluding Matrix Children (they will sync with the Parent), and has no Shopify IDs yet (Indicating they have not be synced to Shopify Yet)

### Mass Update

TITLE OF ACTION \*  
In8Sync Send Inventory Items to Shopify | NEW

TYPE  
Inventory Part

ACTION  
In8 Shopify - Send Items  
 PUBLIC

Criteria Results Audience Schedule Audit Trail

Use this tab to specify criteria that narrow down your search.

USE EXPRESSIONS

FILTER *	DESCRIPTION *	FORMU
Sync To Shopify (Custom)	is true	
Matrix Child Item	is false	
In8 Shopify Ids : Shopify Id	is empty	

Mass Update Template for Finding New items with excluding Matrix Children (they will sync with the Parent), and DOES HAVE Shopify IDs (Indicating they have be synced to Shopify Yet)

### Mass Update

In8Sync Send Inventory Items to Shopify | NEW

TITLE OF ACTION \*  
In8Sync Send Inventory Items to Shopify | UPDATE

OWNER  
In8Sync Support

TYPE  
Inventory Part

ACTION  
In8 Shopify - Send Items  
 PUBLIC

Criteria Results Audience Schedule Audit Trail

Use this tab to specify criteria that narrow down your search.

USE EXPRESSIONS

FILTER *	DESCRIPTION *
Matrix Child Item	is false
In8 Shopify Ids : Shopify Id	is not empty

## Matrix Option/Attribute Mapping

To Sync the Matrix Items you must map the Attributes in the settings for the Possible Attributes, By default Shopify allows only 3 total Attributes per item

On the Shopify Settings you will need to write an array of the fields and their Names:

Under Setup Tab > In8Sync Shopify > Settings

Example:

```
[  
{"field":"custitem1","name":"Size"},  
{"field":"custitem2","name":"Color"},  
{"field":"custitem10","name":"Lights"}  
]
```

<https://jshint.com> can be used to verify no errors in the JSON mapping.

## Taxable Mapping

Shopify Values for the Item to be taxable or not is “true” and false”

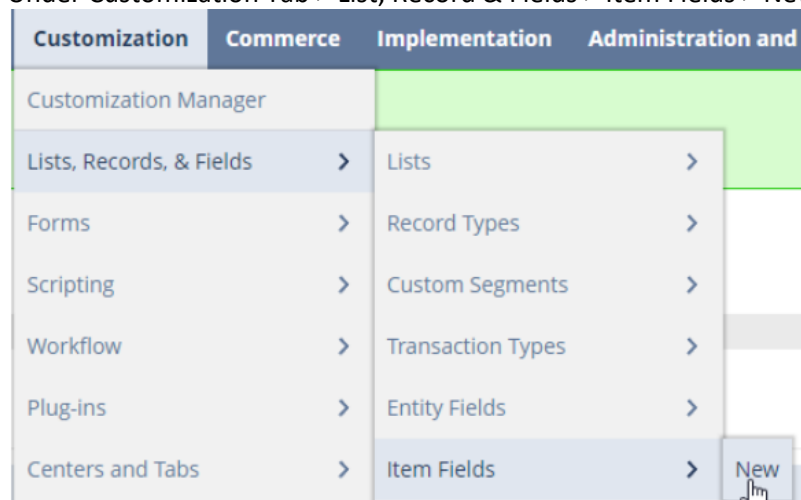
Charge tax on this product

```
"taxable": true,
```

Base on Example where the Items are Using Tax Schedule to Set on the the Items, with “Taxable” or “Non-Taxable”, we can create a Custom Field with Formula to list the Values need for Shopify (above) based on the Values of the Tax Schedules by Name.



Under Customization Tab > List, Record & Fields > Item Fields > New



Create the Item Field as Follows

Field	Value
Name	Shopify Taxable
Field ID	_shop_taxable
Type	Free-Form Text
Store Value	F
Applies To (SubTab)	All Item Types needed = T
Display (SubTab)	SubTab = In8Sync Display Type = Inline Text
Validation & Defaulting (SubTab) > Default Value	CASE WHEN {taxschedule} = 'Taxable' THEN 'true' WHEN {taxschedule} = 'Non-Taxable' THEN 'false' END

LABEL \*  
Shopify Taxable

ID  
custitem\_shop\_taxable

INTERNAL ID  
2528

OWNER  
In8Sync Integration

DESCRIPTION

MATRIX OPTION

TYPE  
Free-Form Text

LIST/RECORD

Applies To Display Valid

- INVENTORY ITEM
- NON-INVENTORY ITEM
- SERVICE
- OTHER CHARGE
- GROUP
- KIT/PACKAGE
- ASSEMBLY/BILL OF MATERIALS

Applies To Display Validation & Defaulting Sourcing & Filtering Access Translation History

Validation & Defaulting

INSERT BEFORE  
Unchanged

SUBTAB  
In8Sync

DISPLAY TYPE  
Inline Text

DISPLAY WIDTH

DEFAULT VALUE

```
CASE WHEN (taxschedule) = 'Taxable'  
THEN 'true'  
WHEN (taxschedule) = 'Non-Taxable'  
THEN 'false'  
END
```

FORMULA

Then Map the New NetSuite Field to the Shopify Field "taxable"

Field Mappings		Notes	
In8 Shopify Mappings Fields • In8 Shopify Metafields Definition • Shopify Criteria •			
New In8 Shopify Mappings Fields			
NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE
custitem_shop_status		status	
custitem_shop_status_date		status_date	
custitem_shop_status_date		status_date	
custitem_shop_weight		weight	
custitem_shop_weight_unit		weight_unit	
custitem_shop_barcode		barcode	
custitem_shop_sku		sku	
custitem_shop_taxable		taxable	

### itemfulfillment

Itemfulfillment parameter mapping is created and inactive by default.

This Parameter does not require additional mapping and by Default when NetSuite Sales Order' Item Fulfillments are completed, Shipped Status when Fulfillment status is enabled, will send the following data from the Item Fulfillment to the Shopify Order for Item Fulfillment.

1. Shipping Method
2. Tracking number/s, under Packages SubTab
3. Location, per the Item Fulfillment location for the Shopify Location ID references in the [Location Mappings](#)

There is the option to Source the Shipping Method, Tracking and Tracking URL from possible Custom Fields when hosted on the Item Fulfillment to send to Shopify different Values as needed.

Shopify Fields to send other Value from NetSuite are:

1. tracking\_company
2. tracking\_number
3. tracking\_url

### salesorder

Salesorder parameter mapping is created and inactive by default.

This parameter is used to create the Sales Orders from the Payloads pulled from Shopify. Since Customers are required to put orders under, the Customer Parameter must also be enabled for the integration to find, connect, create or update the Customer Prior to processing the order.

By Default, the integration will set the following:

1. Customer
2. Memo
  1. From Shopify Main Note field
3. PO#
  1. From the Shopify Order Number
4. Items
5. Addresses
6. Discounts
  1. Using Discount items from Settings for header or line discounts as seen from the Shopify order.
7. Tax Override
  1. When "Price Includes Tax" in Settings is Not enabled
8. Shipping Cost
9. Subsidiary
  1. Required to have the Subsidiary by Internal ID, Fixed Value, set in Mappings per the site setup
10. Location
  1. The 1 static location to Set on Shopify order to reduce Stock from
11. To Be Emailed

1. Set to Value = F to not email from NetSuite the Sales Order
12. Coupon Codes used
  1. Listed in the Sales Order Custom Field “In8Sync Coupon Codes Used” when Shopify order has Discount Code for discounts
13. External ID
  1. Prefixed per site on the mapping to ensure no Duplicate Sales Orders are possibly created
14. Order Status
  1. Default mapping for Pending Fulfillment

Field Mappings <span style="float: right;">Notes</span>						
In8 Shopify Mappings Fields • <span style="float: right;">Shopify Criteria</span>						
New In8 Shopify Mappings Fields						
NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE	GET FIELD VALUE	CUSTOM FIELD	ORDER
otherrefnum		name				
orderstatus			B			
externalid	DC	id				
custbody_in8_shop_id		id				
custbody_in8_used_coupon_code		discount_codes[0].code				
subsidiary			1			1
location			2			
tobeemailed			F			

Additional fields, i.e. Department, Class, Sales Channel etc... can be mapped per customer business flow requirements. Further Advanced mappings for business flows; please contact [In8sync Support](#)

Below are other outlined mappings that are required or optional for integration needs

#### Additional Details/Note Attributes Mapping

To Map the “Additional Details” Field Values, the following mapping examples for the Shopify JSON data needs to be done

In this example the Additional Details field for “Catalog code” will get mapped to the NetSuite Field “Source Code”:

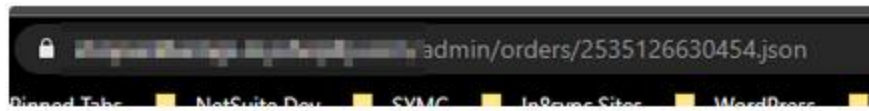
**ADDITIONAL DETAILS** [Edit](#)

Where did you hear about us?  
Catalog

Catalog code  
HF174B

The Shopify Payload listed this from a "note\_attributes" array:

If you have a JSON format viewer, then adding the ".json" to the end of the Shopify Order will get the JSON output that is pulled to NetSuite.  
Recommended Chrome Plugin = JSON Viewer Awesome



```
"note_attributes": [  
  {  
    "name": "Where did you hear about us?",  
    "value": "Catalog"  
  },  
  {  
    "name": "Catalog code",  
    "value": "HF174B"  
  }  
],
```

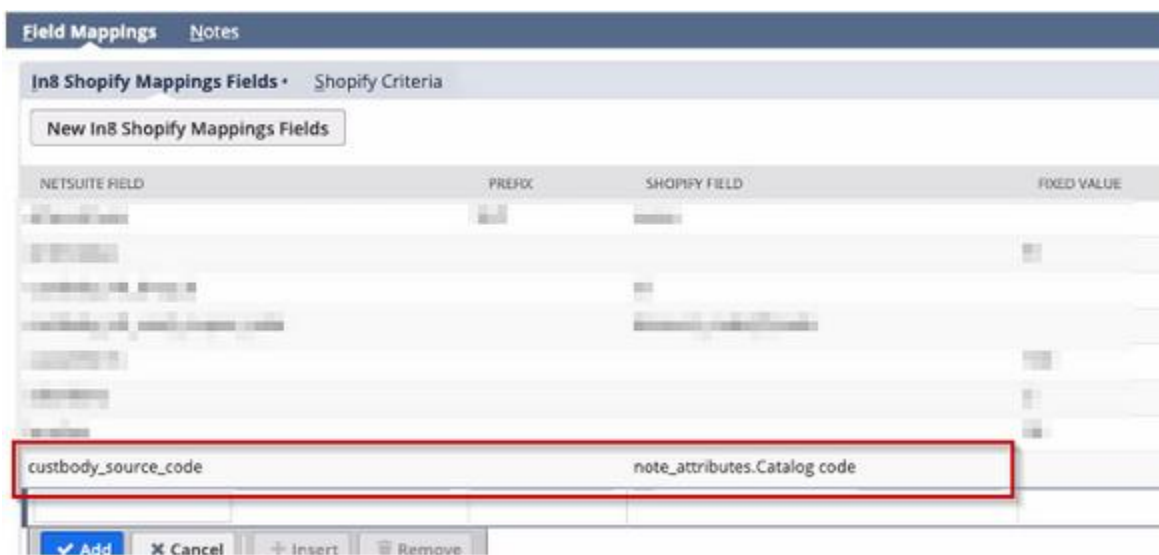
The Shopify Field Mapping should follow the following format:

note\_attributes.<the value of the name field exactly, including any punctuation>

In this case like the following:

note\_attributes.Catalog code

Then Map that to the NS field needed:



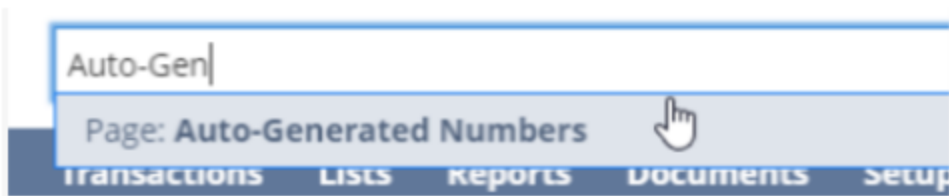
The screenshot shows the 'Field Mappings' interface for 'Notes'. It features a table with columns for 'NETSUITE FIELD', 'PREFIX', 'SHOPIFY FIELD', and 'FIXED VALUE'. A red box highlights a specific mapping row where the NetSuite field is 'custbody\_source\_code' and the Shopify field is 'note\_attributes.Catalog code'. Below the table are buttons for 'Add', 'Cancel', 'Insert', and 'Remove'.

NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VALUE
custbody_source_code		note_attributes.Catalog code	

Shopify Order# set in NetSuite Doc# (transid)

To override and be able to set the Shopify Order # into the Sales Order Default Document Number (tranid)

Go to "Auto-Generated Numbers"



Under "Document Numbers"



The "Allow Override" is the 2nd to Last Checkbox



### Set Up Auto-Generated Numbers

**Save** **Cancel** **Reset**

Record numbers for Entities, CRM, and Schedules cannot be reverted back to names after update. If you check Update, a number is assigned to each existing record of that type, and the name is stored in the Company Name field.

You can return to this page at any time to turn on or off numbering for new records. However, after you turn on numbering, records will be listed by number rather than name. It typically takes just a few minutes for 5000 records. While it is running, this page will not be available.

Entities • **Document Numbers** • CRM • Other

TYPE	PREFIX	SUFFIX	MINIMUM DIGITS	UPDATE INITIAL NUMBER	INITIAL NUMBER	USE SUBSIDIARY	USE LOCATION	CURRENT NUMBER	ALLOW OVERRIDE
Cash Refund				<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	3	<input type="checkbox"/>

Scroll Down to Sales Order and check "Allow Override"

Sales Order	SO			<input type="checkbox"/>	2	<input type="checkbox"/>	<input type="checkbox"/>	9	<input checked="" type="checkbox"/>
Statement Charge				<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

Then Save

Go to Setup Tab > In8Sync Shopify > Mappings > Edit Sales order > Change the Mapping for "Otherrefnum" to "trandid"

**Field Mappings** **Notes**

**In8 Shopify Mappings Fields** • Shopify Criteria

New In8 Shopify Mappings Fields

NETSUITE FIELD	PREFIX	SHOPIFY FIELD
otherrrefnum		name

**OK** **Cancel** **+ Insert** **Remove**

**Field Mappings** **Notes**

**In8 Shopify Mappings Fields** • Shopify Criteria

New In8 Shopify Mappings Fields

NETSUITE FIELD	PREFIX	SHOPIFY FIELD
trandid		name

**OK** **Cancel** **+ Insert** **Remove**

## Line-Item Properties Mapping

This one would be done under the “orderitems” Parameter for mapping.

To map the line item level Properties value to a NetSuite field the following way.

In a “orderitems” parameter:

- Set the NetSuite line field to which the Properties Value should be mapped to
- From the Payload for the lines identify when a Specific Properties “name” should be match to take its value to the line in NetSuite
  - In this case when the line has the Properties name as “Amazon Order Item ID”

```
▼ "line_items": [  
  ▼ {  
    "id": 10014755684530,  
    "admin_graphql_api_id": "gid://shopify/LineItem/10014755684530",  
    "fulfillable_quantity": 1,  
    "fulfillment_service": "manual",  
    "fulfillment_status": null,  
    "gift_card": false,  
    "grams": 0,  
    "name": "The Atomic Bear Tactical Pen - Self Defence Pen and Window Breaker - Used in Police Refill",  
    "price": "13.99",  
    ▼ "price_set": {  
      ▼ "shop_money": {  
        "amount": "13.99",  
        "currency_code": "USD"  
      },  
      ▼ "presentment_money": {  
        "amount": "13.99",  
        "currency_code": "USD"  
      }  
    },  
    "product_exists": true,  
    "product_id": 6132218527922,  
    ▼ "properties": [  
      ▼ {  
        "name": "Amazon Order Item ID",  
        "value": "49619999649962"  
      }  
    ],  
  },  
],
```

- The Value for the Properties > name “49619999649962” should be take and listed in the Mapping’s NetSuite Field
- the Shopify field mapping as **properties.{properties>name value}**
  - example = properties.Amazon Order Item ID (see below)

The screenshot displays the 'In8 Shopify Mappings' configuration page. At the top, there is a navigation bar with icons for home, star, and refresh, followed by menu items: Activities, SuiteSuccess, Payments, Transactions, Lists, Reports, and Analytics. Below this, the page title is 'In8 Shopify Mappings' and the specific mapping is identified as 'orderitems'. Action buttons include 'Save', 'Cancel', 'Change ID', and 'Actions'. The 'Primary Information' section contains:
 

- CUSTOM FORM**: 'In8 Shopify Customer Mapping Form' with a refresh icon.
- RECORD TYPE**: 'orderitems' with an 'INACTIVE' checkbox.
- SETTING**: 'The Atomic Bear' dropdown menu.
- COMMENTS**: An empty text area.

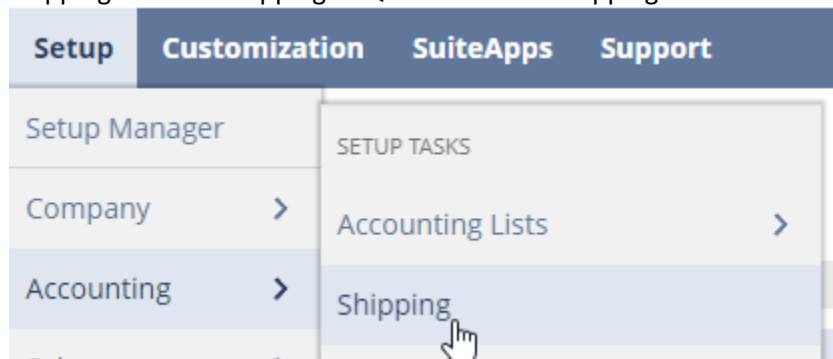
 The 'Field Mappings' section is titled 'In8 Shopify Mappings Fields' and includes a search icon and a 'New In8 Shopify Mappings Fields' button. A table lists the mappings:
 

NETSUITE FIELD	PREFIX	SHOPIFY FIELD	FIXED VA
custcol_in8_amazon_id		properties.Amazon Order Item ID	

 At the bottom of the field mappings section are buttons for 'Add', 'Cancel', 'Insert', and 'Remove'.

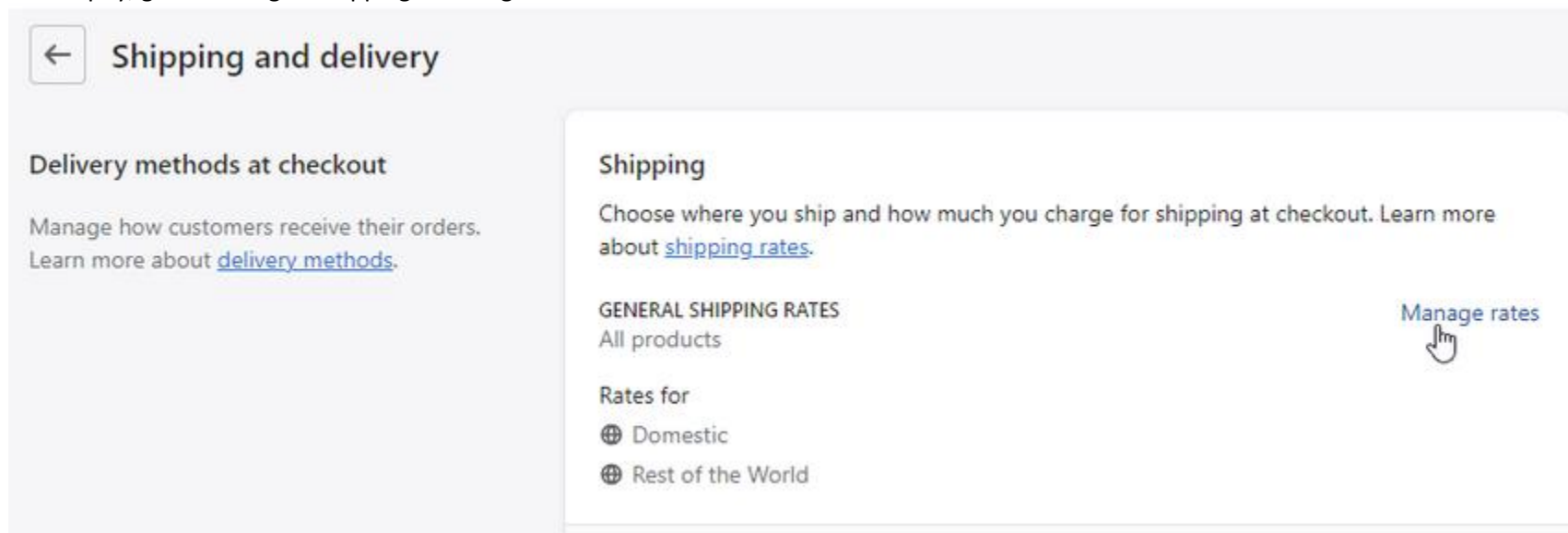
## 5.6 Shipping Mappings

Shipping Method Mapping REQUIRE NetSuite Shipping Is enable and NetSuite Shipping Items created.



There is the Option in [Settings](#) to use an Item for setting Shipping as Line item if Shopify Shipping it not going to be enabled.


On Shopify, go to Settings > Shipping > Manage Rates.



## Mappings will Use the Shopify Shipping Method Titles

Economy	0lb-5lb	5 to 8 business days	\$4.90	...
Economy	5lb-70lb	5 to 8 business days	\$19.90	...
Standard	0lb-1lb	3 to 4 business days	\$6.90	...
Standard	1lb-5lb	3 to 4 business days	\$9.90	...

Add rate

 Rest of the World Rest of World				...
Carrier name ▲	Handling fee	Shipping speed	Services	
DHL Express (Discounted rates from Shopify Shipping)	—	Calculated	Worldwide	...
USPS (Discounted rates from Shopify Shipping)	—	Calculated	3 Services ▲	...
			<ul style="list-style-type: none"> <li>First Class Package International</li> <li>Priority Mail Express International</li> <li>Priority Mail International</li> </ul>	

On Orders in Shopify would be the Title not including Weight

Subtotal	25 items	\$982.00
Shipping	Freight Calculated Later (20.12 lb)	\$0.00
<b>Total</b>		<b>\$982.00</b>
<hr/>		
Paid by customer		\$0.00

In NetSuite, to map the Shopify Title Code to the NS create Shipping Item  
Go to Setup > In8 Shopify > Shipping Methods

Setup	Customization	SuiteApps	Support
Setup Manager		Setup/Tools	
Company	>	Global Settings	>
Accounting	>	Settings	>
Sales	>	Mappings	>
Marketing	>	Shipping Methods	>
Support	>	Payment Methods	>
Intranet	>	Location Mappings	>
Import/Export	>	Users Mappings	>
In8Sync Shopify	>	Shopify Requests (Shopify to NS)	>
Records Catalog		Request Logs (Shopify to NS)	>

Add the Shopify Shipping Title in the Shopify Code Field, NetSuite Shipping Method to list on the NetSuite orders created & the Setting (site) the title is to be used with

### In8 Shopify Shipping List

VIEW Default Customize View New In8 Shopify Shipping

**FILTERS**

SHOW INACTIVES | EDIT

NEW	EDIT   VIEW	INTERNAL ID	SHOPIFY CODE ▲	SHIPPING METHOD	SETTING
	Edit   View	8	ARR GV WEST	Customer Pick-Up	Wholesale
	Edit   View	7	Freight Calculated Later	Freight Calculated Later	Wholesale



Shopify Local Delivery and Local Pickup  
Under Settings > Shipping and Delivery

### Local delivery

Deliver orders directly to local customers. Learn more about [local delivery](#).

#### MANAGE DELIVERY BY LOCATION



**ARR GV EAST** Doesn't offer delivery

8 Holmes Avenue, Jersey City New Jersey 07306, United States

[Manage](#)



**ARR GV WEST** Doesn't offer delivery

23865 Vía del Rio, Yorba Linda California 92887, United States

[Manage](#)

### Local pickup

Allow local customers to pick up their orders. Learn more about [local pickup](#).

#### MANAGE PICKUP BY LOCATION



**ARR GV WEST** Offers pickup

23865 Vía del Rio, Yorba Linda California 92887, United States

[Manage](#)



**ARR GV EAST** Doesn't offer pickup

8 Holmes Avenue, Jersey City New Jersey 07306, United States

[Manage](#)

Local Pickup - the Shopify Order will state the Local Pickup Name as the Shipping Title

The screenshot displays a Shopify order summary on the left and a local pickup configuration panel on the right. The order summary includes a 'Paid' status, a subtotal for '1 item', shipping for 'Belgian Delights (0.25 kg)', and tax for 'GST 10% (Included)'. The local pickup panel shows the pickup location as 'Belgian Delights' with the address '58 Export Drive, Molendinar Queensland 4214, Australia'. A green badge next to the location name indicates 'Offers pickup'.

Item	Quantity
Belgian Delights (0.25 kg)	1

**Local pickup**  
Allow local customers to pick up their orders. Learn more about [local pickup](#)

MANAGE PICKUP BY LOCATION

- Belgian Delights** Offers pickup  
58 Export Drive, Molendinar Queensland 4214, Australia

Local Delivery - Shopify Order will state the "Local Delivery" as the Shipping Title



Subtotal	31 items
Shipping	Local Delivery (6.47 kg)
Tax	GST 10% (Included)

SHOPIFY CODE ▲ 📌	SHIPPING METHOD 📌	SETTING 📌
🔒 Belgium (2) 📌	Belgium	Belgium-Delights
🔒 Belgium (2) 📌	Belgium	Belgium-Delights-📌
Belgian Delights	Local pickup	Belgium-Delights-📌
Belgian Delights	Local pickup	Belgium-Delights
Free Shipping	Free Rate	Belgium-Delights
Free Shipping	Free Rate	Belgium-Delights-📌
FreeRate	Free Rate	Belgium-Delights
FreeRate	Free Rate	Belgium-Delights-📌
FreeRate-📌	Free Rate	Belgium-Delights
FreeRate-📌	Free Rate	Belgium-Delights-📌
Free	Free	Belgium-Delights
Local delivery	Local delivery	Belgium-Delights-📌
Local Delivery	Local delivery	Belgium-Delights-📌

## 5.7 Payment Mappings

In NetSuite when mapping the Shopify Payment Methods to the NetSuite Payment Methods the following are the known Shopify ShortCodes that need to be mapped per the Method options:

**Always Map manual → Cash (Unless Used for Payment Pending Puller, per settings "On Account")**

**If mapping blank Shopify Shipping code to a NetSuite Payment Method, it would cover if there is no Payment method listed in the Shopify Order and Payload**

### Legend:

**Shopify Payment Name** = ShortCode to Map

Shopify Released Shop Pay Installments, seems this is included in Shopify Payout Reconciliation. Both look like they should be Mapped to the NetSuite Shopify Payments, Payment Method for reconciliation

- **Shopify Payments** = shopify\_payments
- **Shop Pay Installments** = shopify\_installments
  
- **PayPal Express Checkout** = paypal
- **AmazonPay with E-Merchant Solutions** = merchant\_e\_solutions
- **Amazon Pay** = amazon\_payments
- **Manual (created in Admin Backend)** = manual
- **Afterpay** = afterpay
- **Afterpay North America** = afterpay\_north\_america
- **Sezzle** - sezzle
- **CyberSource** = cyber\_source
- **Authorize.Net** = authorize\_net
- **CoinBase Commerce** = coinbase\_commerce\_
- **BitPay** = bit\_pay
- **Windcave PxPay** = pxpay
- **LayBuy** = laybuy
- **eWAY Rapid 3.1** = eway\_rapid
- **Braintree** = braintree
- **Stripe** = Stripe

- **PayPal Payflow Pro** = payflow
- **QuadPay** = quadpay
- **TrustCommerce** = trust\_commerce

## In8 Shopify Payment Method Mapping List

VIEW Default ▾ Customize View | New In8 Shopify Payment Method Mapping

 FILTERS

EDIT   VIEW	INTERNAL ID	SHOPIFY PAYMENT METHOD	NETSUITE PAYMENT METHOD
<a href="#">Edit</a>   <a href="#">View</a>	2	manual	Cash
<a href="#">Edit</a>   <a href="#">View</a>	12	paypal	PayPal (Shopify)
<a href="#">Edit</a>   <a href="#">View</a>	23	shopify_installments	Credit Card (Shopify)
<a href="#">Edit</a>   <a href="#">View</a>	7	shopify_payments	Credit Card (Shopify)

### **Payment Pending Order (On Account)**

Payment Pending Orders, also referred to as “On Account” Orders, can be pulled in to Create Sales order without Payment Method to generate Invoice for Later Payment from NetSuite.

In Settings, the Payment Gateway Names for Payment Pending must be listed.

Customers may have Manual Gateway Setup and those gateway names should be listed in the Settings.

If there is not Payment Gateway Names in the Customer Payment Pending orders, the use of “null” can be used

In this example the customer would have two Manual Gateways setup in the Shopify Payment Settings, Net 30 and Net 60, plus other payment Pending order that have no Payment Gateway Names

Under Shopify > Settings > Payments

**Manual payment methods**

Payments that are processed outside your online store. When a customer makes a manual payment, you need to approve their order before fulfilling.

Manual payment methods ▾

Net 60 is active Edit

Net 30 is active Edit

```
"payment_gateway_names": [],
"phone": null,
```

```
"financial_status": "pending",
"fulfillment_status": null
```

In this case setting the Gateway Names of “null,Net 30,Net 60” would cover all 3 in the Setup Tab > In8Sync Shopify > Settings

**On Account Payment Settings**

ON ACCOUNT GATEWAY NAME

null,Net 30,Net 60

Authorized Payment Settings

## 5.8 Location Mappings

Locations Mappings under Setup Tab > In8Sync Shopify > Location Mappings, has a Three Part Function:

- **NetSuite to Shopify Stock updates for updating QTY Available**
  - Requires Item Parameter Mapping for what NS location/s are approved to sync Stock Updates to Shopify for that site
    - The Item type Specific Parameters are referenced in conjunction to match with the Location Mappings Location ID Configured for the Stock to be Sent which Shopify location.
  - Stock “QTY available” from NetSuite Items Location mapped to sync to Shopify Locations CANNOT be sent from NetSuite to Shopify If the Site Item Type Specific Mappings do not list the location.
- **NetSuite to Shopify Fulfillment is needed to Successfully Update the Shopify Order Fulfillment (location ID required)**
  - Fulfillment from NetSuite to Shopify, to update the order, requires a Shopify Location ID
- **Shopify to NetSuite Item Fulfillment Creation**
  - [Additional Setting](#) “Create Fulfillment from Shopify” would have to be enabled.
  - Uses the Location Mapping to Check the Location ID in the Shopify Fulfillment Payload to find the NetSuite location to use on the NetSuite Fulfillment we would create.

### NetSuite

- Locations must be configured on Item Type Specific mappings for Stock levels to sync to Shopify.
  - Configuration on Shopify Locations settings page is also required (below).
  - Setup->Shopify->Mappings->Item Mapping (by product type)
  - The locations mapped here are checked during Stock Sync, to know what locations are Approved to pull stock from to update Shopify Item Location with.



## In8 Shopify Mappings

inventoryitem

Save

Cancel

Change ID

Actions

### Primary Information

CUSTOM FORM \*

In8 Shopify Item Mapping Form

RECORD TYPE \*

inventoryitem

INACTIVE

### Item Information

LOCATION

Amazon Sellable

Amazon Seller Central

Amazon Seller Central W#1

Damaged Warehouse

location for subsidiary stagin6

- Locations then get mapped to Shopify locations
  - These locations are used for Shopify location ids on Stock Sync and Item Fulfillment Sync
  - Setup->Shopify->Locations

## In8 Shopify Location Mappings

← → List Search Customiz

Save Cancel Change ID Actions

### Primary Data

CUSTOM FORM \*

In8 Shopify Location Mappings Form

DATE CREATED  
11/25/2020 6:38 am

LAST MODIFIED BY  
4/23/2021 1:32 pm Marcel In8Sync

INACTIVE

### Location Mapping Data

#### INSTRUCTIONS

Please do not use both Single and Multiple Location Fields. Only use one or the other based on need.

- If Mapping 1 NetSuite Location to 1 Shopify Location, please use the Single Select NetSuite locations Field
- If needing to SUM the stock from multiple NetSuite locations to 1 Shopify Location, Please use the Multi-Select Field.

Make sure all NetSuite locations to each site mapped here, also align with all the locations mapped in Item Parameters

NETSUITE LOCATION

NETSUITE MULTI-LOCATIONS

Amazon Sellable  
Amazon Seller Central  
Amazon Seller Central W#1  
Damaged Warehouse  
location for subsidiary stagin6

SHOIFY LOCATION ID

33311260777

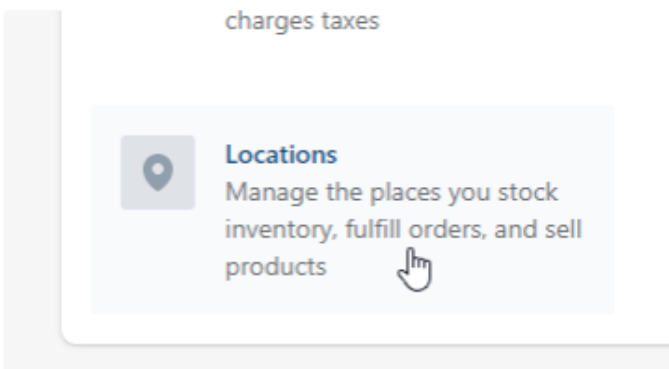
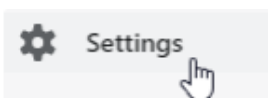
SETTING

Shopify Dev

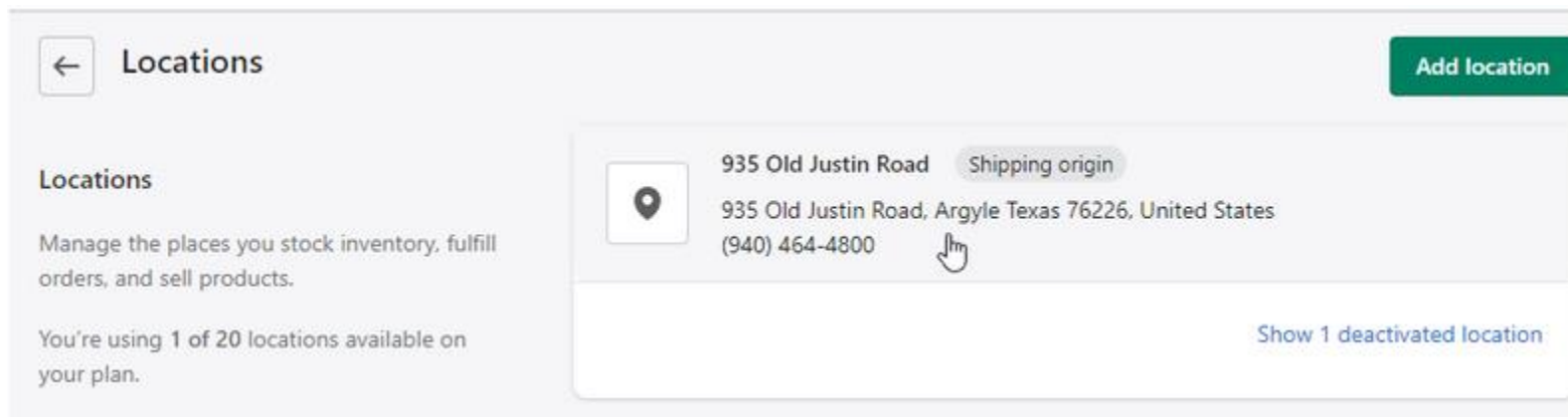
Notes Files Workflow

## Shopify

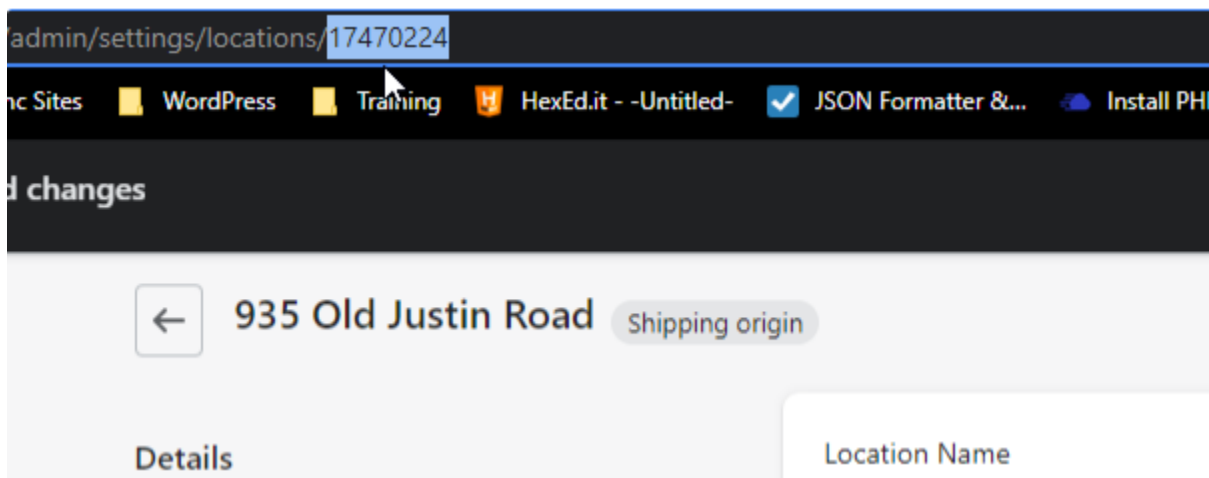
To get the Shopify Location Id from Shopify Locations for the mappings  
Go to Shopify Site > Settings (Bottom Left of Admin Console) > Locations



Click on the location in Shopify to get the id for that



In the URL is the ID



One to One location Mappings look like so:

## In8 Shopify Location Mappings List

VIEW Default ▾

Customize View

New In8 Shopify Location Mappings

+ FILTERS



SHOW INACTIVES

EDIT



EDIT   VIEW	INTERNAL ID	NETSUITE LOCATION ▲	SHOPIFY LOCATION ID	SETTING
Edit   View	1	Evobox	7859535972	Thread 2.0
Edit   View	2	Evobox	36284072072	Thread Custom
Edit   View	3	Evobox	7268728875	Thread Wholesale
Edit   View	4	Evobox	3232708	Thread LLC
Edit   View	5	Fashionplace Mall	36702486610	Thread Retail
Edit   View	6	Thread Wallets Office	35502620754	Thread Retail
Edit   View	7	University Mall	60198584491	Thread Retail
Edit   View	8	UTC Mall	63491571883	Thread Retail

### END: REQUIRED SECTIONS FOR SUCCESSFUL STANDARD SETUP

#### 5.9 User Mappings

#### 5.10 Shopify Requests (Shopify to NS)

There is nothing to setup in this section. This section lists the orders and other data pulled in from Shopify to get processed as well as Sales Order and other records Created in NetSuite. This is also where we can see any Order and other records creation Statuses/Errors.

If an order has failed and the order issue has been resolved, this is also where you can come to re-process an order that pulled in.

Transaction IDs can be seen here when the order was successfully Processed:

## In8 Shopify Requests List

VIEW Default ▾

Customize View

New In8 Shopify Requests

+ FILTERS



SHOW INACTIVES

EDIT



EDIT   VIEW	INTERNAL ID	TOPIC	DATE TIME	STATUS	SHOPIFY ID	INTERNAL ID	TRANSACTION ID ▲
Edit   View	1078	orders/paid	5/13/2019 3:26:31 pm	Completed	1103553036397	160847	SLS00002786

See Section [6.2 NetSuite Setup > Tools](#) for more information on how to clear the Queue logs or to Pull data from Shopify before the scheduled 15-minute interval.

### 5.11 Requests Logs (Shopify to NS)

This Section is a direct link to the Script for reviewing execution logs that In8Sync Support uses when review possible issues ad Bugs

The main script by default use is the “In8 Shopify Sync SS”. When certain customers are running “SuiteCloud Plus” and capable to run multiple Concurrent requests, the “In8 Shopify Sync SS” Deployments can be Undepoloyed, and the “In8 Shopify Sync MR” can be scheduled to process order with Concurrent Requests processing options.

NAME ▲
In8 Shopify Sync MR
In8 Shopify Sync SS

### 5.12 Shopify Synced Orders

This Section is the Save Search, that is Part of the Bundle, to Show all the NetSuite Sales Orders created by the Integration. All NetSuite Sales Orders created by the integration have the Following Fields listed with Values for the Shopify Order per site, under the In8Sync SubTab.

- In8 Shopify ID

- In8 Shopify Source
- In8 Shopify Site
- In8 Shopify Order Number

### 5.13 Sync Queue (NS to Shopify)

The “Sync Queue” is a Custom Record queue, used mainly for Syncing Matrix items to Shopify. It can also be used when the “[Global Settings](#)” are configured to send all Items through the Queue to Sync.

When Items are listed in the “Sync Queue” the Script that processes this Queue is automatically triggered to start Processing and keep re-triggering to process as long as there are NetSuite Item payloads in “Pending” Status.

### 5.14 Sync Queue Logs (NS to Shopify)

The “Sync Queue Logs” is the Script that triggers Automatically the Items in the “Sync Queue” for syncing the items to the Shopify.

### 5.15 Shopify Plus ONLY - Manual Gift Cards Pull

Shopify Locks the Gift Card API to Private APPs, unless Shopify Plan is at Shopify Plus.

#### 5.15.1 Manual Gift Card Queue

The Manual Gift Card Queue is a Custom Record used when “Shopify Plus” Customers ONLY, are setup to pull the Manually issued Gift Cards from Shopify through the Private App API Connection.

When pulled, the Integration will create Cash Sales for the Gift Cards pulled to create the NetSuite Gift Certificates needed when redeemed on Shopify Orders.

#### In8 Shopify Manual Gift Card List

VIEW Default | Customize View | New In8 Shopify Manual Gift Card

**FILTERS**

     SHOW INACTIVES | EDIT  

QUICK SORT ▼ | 3801 — 1790 ▼

EDIT   VIEW	INTERNAL ID ▼	NAME	SETTINGS	INITIAL VALUE	CURRENCY	LAST CHARACTERS	PROCESSING STATUS	CASH SALE TRANSACTION
<a href="#">Edit</a>   <a href="#">View</a>	3801	486872252547	Duck Camp	200.00	USD	xxx1	Success	Cash Sale #CS30678
<a href="#">Edit</a>   <a href="#">View</a>	3701	486812778627	Duck Camp	21.65	USD	5eac	Success	Cash Sale #CS27797
<a href="#">Edit</a>   <a href="#">View</a>	3601	486796984451	Duck Camp	200.00	USD	hlsz	Success	Cash Sale #CS27419
<a href="#">Edit</a>   <a href="#">View</a>	3501	486657523843	Duck Camp	1,000.00	USD	6f3f	Success	Cash Sale #CS25394
<a href="#">Edit</a>   <a href="#">View</a>	3401	486504366211	Duck Camp	117.99	USD	79aa	Success	Cash Sale #CS23524
<a href="#">Edit</a>   <a href="#">View</a>	3301	486461767811	Duck Camp	750.00	USD	c40f	Success	Cash Sale #CS22856
<a href="#">Edit</a>   <a href="#">View</a>	3201	486441255043	Duck Camp	2,000.00	USD	eh6f	Success	Cash Sale #CS22688

Is required to enable and setup in the In8Sync [Settings](#) and Set what Gift Card Item in NetSuite to be Used & the Discount Item to use to discount the Cash Sale 100%.

Manual Gift Cards | Shopify Plus ONLY

PULL MANUALLY CREATED GIFT CARDS

GC LAST SYNC  
11/11/2021 1:06:05 pm

GIFT CARD ITEM  
Manual Gift Card

GC DISCOUNT ITEM  
Shopify Discount

RETURNLY GC DISCOUNT ITEM

Any Questions on Setting up correctly and additional future options, please contact [In8Sync Support](#)

### 5.15.2 Manual Gift Card Logs

The Manual Gift Card Logs is the Script in NetSuite that Processes each Manual Gift Card Payload to create the Cash Sales.

Typically set to schedule of Daily and every 15 minutes

[Schedule](#) • [Parameters](#) • [Execution Log](#) • [System Notes](#)

SINGLE EVENT

DAILY EVENT  Repeat every  day(s)

WEEKLY EVENT  Repeat every weekday

MONTHLY EVENT

YEARLY EVENT

START DATE \*  START TIME

REPEAT

END BY

NO END DATE

### 5.15.3 Manual Gift Card Puller

The Manual Gift Card Puller is the script use to Pull the Manual Gift Card Payload into the Manual Gift Card Queue for Processing.

Typically set to schedule of Daily and every 15 minutes

**Schedule** • Parameters • Execution Log • System Notes

SINGLE EVENT

DAILY EVENT  Repeat every  day(s)

WEEKLY EVENT  Repeat every weekday

MONTHLY EVENT

YEARLY EVENT

START DATE \*  START TIME

REPEAT

END BY

NO END DATE

### 5.16 Other Transaction Info Results

The other Transaction Info Results is the Custom Record lists the general Transaction data available to the Shopify order. That each listing is also connected to the Sales Order Created by the Integration. To supply what is available per gateway used on the Shopify Order, i.e. Transaction ID, Gateway, Total and fees.

### 5.17 Shopify Payments Payout Reconciliation

Per Site, Shopify supplies Payouts Daily for Reconciliation of Fees that they take and the amounts that they Payout to the Shopify Owner. The Following Section are Requirements and setup for how the In8Sync Integration Pulls and Handles performing the Reconciliation in NetSuite to Create "Bills", "Payments" and Journal Entries to support handling the Accounting Reconciliation in NetSuite



### Requirements

- NetSuite Vendor Record
  - Used when creating the Vendor Bill for the Payout, for the Daily Payout Fees that need to be paid to the Shopify Vendor in NetSuite.
- Clearing Account
  - Configured on the NetSuite Payment Method “Shopify Payments” used in Payment Mapping from Shopify. Checks the In8Sync Payment Mapping to See what is Mapped for Shopify Payments for the Bank Type Account on the NetSuite Payment Method
- Fees Account
  - Account used when creating the Vendor Bill and Payment to put the Fees part of the daily Payout into.
- Banks Account
  - Account Used on the Journal Entry that transfers the Payout Amount (amount - fees) from Clearing account into this Account.

### Parameter Mappings

Possible Parameters for Mappings, that can be used to help map and set additional Fields (Header or Line) like Department, Class, and Account are:

NOTE: For assistance on mappings please contact [In8sync Support](#).

- **p-out-bill** (Payout Reconciliation ONLY; “Vendor Bill” Header)
- **p-out-bill-expitems** (Payout Reconciliation ONLY; “Vendor Bill” Lines)
- **p-out-billpayment** (Payout Reconciliation ONLY; “Vendor Bill Payment” Header)
- **p-out-journal** (Payout Reconciliation ONLY; “Journal Entry” Header)
- **p-out-journal-items** (Payout Reconciliation ONLY; “Journal Entry” Lines)

### Settings

From Setup Tab > In8Sync Shopify > Settings

- Enable Shopify Payment Reconciliation
- Enter No. of Business days in the past
- List Vendor
- List the Fee Account
- List the Bank account

- Edit the NetSuite Payment Method mapped for Shopify Payments and set the “Bank Type” account for the Shopify Clearing account as needed.

Payment Reconciliation		
<input type="checkbox"/> ENABLE SHOPIFY PAYMENT RECONCILIATION	PAYMENT METHOD Shopify Payment	FEES ACCOUNT [Dropdown]
NO. OF BUSINESS DAYS [Input]	CLEARING ACCOUNT 10500 Shopify Clearing	BANK ACCOUNT [Dropdown]
VENDOR [Dropdown] + [Link]		

For the Settings that are set and enabled the Following Section need to be confirmed Setup to properly schedule the needed scripts that Pull and Process the Daily Payouts.

- [Payout & Transactions Puller](#)
  - Required to be scheduled to Pull the Payload for the Daily Payouts and the Transactions data that are Part of that Payout
- [Payout Processor Logs](#)
  - Required to Process the Daily Payout payloads to create the Bill, Payment, and Journal Entry to reconcile the Fees and Amount minus the fees to the correct Accounts from the Shopify Clearing account
- [Payout Transaction Logs](#)
  - Required to Process the Transactions that are part of the Daily Payout to connect them to each order Synced in Netsuite and list the Transactions under the Payout Record under the “[Payout Reconciliation Results](#)” OR “[Payout Reconciliation Queue](#)” (both ultimately show the same Daily Payout list, Created Records and possible errors in processing.

For any Assistance needed please reach out to [In8Sync Support](#)

### 5.17.1 Payout Transactions Results

Payout Transaction Results lists the Transaction and type that is listed and processed.

**From Setup Tab > In8Sync Shopify > Payout Transaction Results**

The Filter on the Page has a Payout and Type Filter to sort and be able to export CSV for reconciliation purposes.

Types:

- Charge
- Refund
- Adjustment
- Dispute

### In8 Shopify Payout Transaction Results: Results

List Search Audit Trail

Return To Criteria Edit this Search

**FILTERS**

PAYOUT  TYPE  STYLE Normal ▾

Val\_4191049\_1218232\_732 — Val\_4191248\_1218232\_812 TOTAL: 2134

NEW	EDIT   VIEW	INTERNAL ID	TRANSACTION	PAYOUT	SHOPIFY TRANSACTION ID	SHOPIFY ORDER ID	TYPE	FEE	AMOUNT
	Edit   View	3401		68943249597	1640968519869	3706034618557	charge	0.56	10.17
	Edit   View	3402		68943249597	1641008857277	3706137903293	charge	1.41	42.84
	Edit   View	3403		68943249597	1641021210813	3706171064509	charge	0.80	19.05
	Edit   View	3404		68943249597	1641026486461	3706184630461	charge	1.84	59.20
	Edit   View	3405		68943249597	1641072623805	3706317537469	charge	1.89	60.97
	Edit   View	3406		68943249597	1641084125373	3706346832061	charge	1.48	45.20

Once Processed the Results also Show on the SubList “In8 Shopify Payout Transactions” under the In8Sync SubTab on the Transaction.

Items	Shipping	Billing	Accounting	Relationships	Communication	Related Records	System Information	Custom	EDI	HJ - EDI	PO Acknowledgement	Pacejet	In8Sync
IN8SYNC WOOCOMMERCE ORDER ID						<input type="checkbox"/> NARVAR RETURN							<input type="checkbox"/> IS POS REFUND/EXCHANGE
IN8SYNC WOOCOMMERCE SOURCE						<input type="checkbox"/> GIFT CARD RETURN							POS EXCHANGE ORDER ID
IN8SYNC PDF URL						<input type="checkbox"/> SHOPIFY REFUND RETURN							POS EXCHANGE ORDER NUMBER
IN8 SHOPIFY ID 3870004936893						<input type="checkbox"/> PAID RETURN LABEL							POS EXCHANGE NUMBER CREATE FROM
IN8 SHOPIFY SOURCE web						<input type="checkbox"/> IN8SYNC PROMOTION RETURN							<input type="checkbox"/> SHOPIFY CANCEL FULFILLMENT ON CANCEL
<input type="checkbox"/> IN8 SHOPIFY AUTHORIZED						IN8 SHOPIFY REFUND ID							IN8SYNC PROMOTION RETURN CODE
IN8 SHOPIFY TRANSACTION ID						IN8 SHOPIFY REFUND EMAIL							<input type="checkbox"/> IN8 SHOPIFY ORDER PAID WITH GC
IN8 SHOPIFY SETTING More Birds						<input type="checkbox"/> SHOPIFY REFUND SHIPPING							IN8 SHOPIFY RETURN REASON
IN8 SHOPIFY NUMBER #7752						<input type="checkbox"/> IN8 SHOPIFY ON ACCOUNT							<input type="checkbox"/> IN8 SEND REFUND AMOUNT TO SHOPIFY (CR)

In8Sync Ids **In8 Shopify Payout Transactions** • In8 Shopify Order Risk •

VIEW: IN8 SHOPIFY PAYOUT TRANSACTIONS

Default View [Dropdown] [Filter Icon] [Refresh Icon]

[New In8 Shopify Payout Transactions](#) [Attach](#) [Customize View](#)

EDIT	PAYOUT	SHOPIFY TRANSACTION ID	SHOPIFY ORDER ID	TYPE	FEE ▲	AMOUNT
Edit	71426113725	1684745224381	3870004936893	charge	0.89	22.87 F

They are also seen on the Payout Record  
**From Setup Tab > In8Sync Shopify > Payout Reconciliation Results > Clicking View on a Payload Result**

## In8 Shopify Payouts: Results

[Return To Criteria](#)

[Edit this Search](#)

**FILTERS**

STYLE

Normal

EDIT

NEW	EDIT   VIEW	INTERNAL ID	PAYOUT DATE
	Edit   View	4101	6/2/2021
	Edit   <b>View</b>	4001	6/1/2021
	Edit   View	3901	5/31/2021

Billed

**Request**

PAYLOAD

```
{
  "id": "71426113725",
  "status": "paid",
  "date": "2021-06-02",
  "currency": "USD",
  "amount": "9420.90",
  "summary": "In8 Shopify Payout",
  "adjustments_fee_amount": "0.00",
  "adjustments_gross_amount": "0.00",
  "charges_fee_amount": "312.90",
  "charges_gross_amount": "9733.80",
  "refunds_fee_amount": "0.00",
  "refunds_gross_amount": "0.00",
  "reserved_funds_fee_amount": "0.00",
  "reserved_funds_gross_amount": "0.00"
}
```

**Transactions**

VENDOR BILL  
Bill 6/1/2021

VENDOR PAYMENT  
Bill Payment #222601

JOURNAL ENTRY  
Journal #JOU8745

Notes Files Transactions

VIEW: IN8 SHOPIFY PAYOUT TRANSACTIONS

New In8 Shopify Payout Transactions Attach Customize View 1 to 25 of 189

EDIT	TRANSACTION	SHOPIFY TRANSACTION ID	SHOPIFY ORDER ID	TYPE	FEE	AMOUNT	REMOVE
Edit	Sales Order #SO921737	1682513756349	3863954948285	charge	0.56	9.82	Remove
Edit	Sales Order #SO922018	1684034650301	3868026405053	charge	0.58	10.75	Remove
Edit	Sales Order #SO921587	1681987010749	3862657532093	charge	0.59	11.14	Remove

### 5.17.2 Payout Transactions Logs

Payout Transaction Logs should already be scheduled, But should be checked

**From Setup Tab > In8Sync Shopify > Payout Transactions Logs**

Daily, Starting at Noon, Repeat every 6 hours

<u>Schedule</u> •	<u>Execution Log</u> •	<u>System Notes</u>	STATUS
			Scheduled
SINGLE EVENT			
X DAILY EVENT	X	Repeat every 1 day(s)	
WEEKLY EVENT		Repeat every weekday	
MONTHLY EVENT			
YEARLY EVENT			
START DATE	START TIME	REPEAT	
5/26/2021	noon	every 6 hours	
END BY			
<input checked="" type="checkbox"/> NO END DATE			

### 5.17.3 Payout Reconciliation Results

Show the Payouts daily and the Results of the Vendor Bill, Vendor Bill Payment, and Journal Entry

**From Setup Tab > In8Sync Shopify > Payout Reconciliation Results** (can also see main queue under Payout Reconciliation Queue)

**In8 Shopify Payouts: Results** List Search Audit Trail

[Return To Criteria](#) [Edit this Search](#)

**FILTERS**

STYLE  
Normal

TOTAL: 44

NEW	EDIT   VIEW	INTERNAL ID	PAYOUT DATE	STATUS	CURRENCY	CHARGES	REFUNDS	ADJUSTMENTS	FEES	TOTAL AMOUNT	PROCESSING STATUS	VENDOR BILL	VENDOR PAYMENT	JOURNAL ENTRY
	Edit   View	4101	6/2/2021	In Transit	USD	2,130.20	(194.99)	0.00	(68.65)	1,866.56	Pending			
	Edit   View	4001	6/1/2021	Paid	USD	9,733.80	0.00	0.00	(312.90)	9,420.90	Success	Bill 6/1/2021	Bill Payment #222601	Journal #JOU8745
	Edit   View	3901	5/31/2021	Paid	USD	819.18	(79.41)	0.00	(28.79)	710.98	Success	Bill 5/31/2021	Bill Payment #222600	Journal #JOU8744
	Edit   View	3801	5/27/2021	Paid	USD	773.85	(145.76)	0.00	(25.52)	602.57	Success	Bill 5/27/2021	Bill Payment #222596	Journal #JOU8734
	Edit   View	3701	5/26/2021	Paid	USD	1,763.31	(6.98)	0.00	(54.24)	1,702.09	Success	Bill 5/26/2021	Bill Payment #222593	Journal #JOU8727

### 5.17.4 Payout Reconciliation Queue

Show the Payouts daily and the Results of the Vendor Bill, Vendor Bill Payment, and Journal Entry

From Setup Tab > In8Sync Shopify > Payout Reconciliation Results (can also see main queue under Payout Reconciliation Queue)

**In8 Shopify Payouts: Results** List Search Audit Trail

[Return To Criteria](#) [Edit this Search](#)

**FILTERS**

STYLE  
Normal

TOTAL: 44

NEW	EDIT   VIEW	INTERNAL ID	PAYOUT DATE	STATUS	CURRENCY	CHARGES	REFUNDS	ADJUSTMENTS	FEES	TOTAL AMOUNT	PROCESSING STATUS	VENDOR BILL	VENDOR PAYMENT	JOURNAL ENTRY
	Edit   View	4101	6/2/2021	In Transit	USD	2,130.20	(194.99)	0.00	(68.65)	1,866.56	Pending			
	Edit   View	4001	6/1/2021	Paid	USD	9,733.80	0.00	0.00	(312.90)	9,420.90	Success	Bill 6/1/2021	Bill Payment #222601	Journal #JOU8745
	Edit   View	3901	5/31/2021	Paid	USD	819.18	(79.41)	0.00	(28.79)	710.98	Success	Bill 5/31/2021	Bill Payment #222600	Journal #JOU8744
	Edit   View	3801	5/27/2021	Paid	USD	773.85	(145.76)	0.00	(25.52)	602.57	Success	Bill 5/27/2021	Bill Payment #222596	Journal #JOU8734
	Edit   View	3701	5/26/2021	Paid	USD	1,763.31	(6.98)	0.00	(54.24)	1,702.09	Success	Bill 5/26/2021	Bill Payment #222593	Journal #JOU8727

### 5.17.5 Payout Processor logs

The Processor should be configured already on installation, but check to make sure.

**From Setup Tab > In8Sync Shopify > Payout Processor Logs**

Should be Scheduled Daily, Starting 4pm, Repeat every 4 hours

Schedule • Execution Log • System Notes			STATUS
			Scheduled
SINGLE EVENT			
X DAILY EVENT	Repeat every 1 day(s)		
WEEKLY EVENT	Repeat every weekday		
MONTHLY EVENT			
YEARLY EVENT			
START DATE	START TIME	REPEAT	
1/31/2021	4:00 pm	every 4 hours	
END BY			
<input checked="" type="checkbox"/>	NO END DATE		

### 5.17.6 Payout & Transactions Puller

This area is where the script schedule is confirmed to Pull the Daily payouts and transactions that are part of each Daily Payouts

The Puller gets the Payouts and the Transactions Payload we will process.

The Puller should be configured already on installation, but check to make sure.

**From Setup Tab > In8Sync Shopify > Payout Puller**

Should be Scheduled Daily, Starting 12am, Repeat every 6 hours



SINGLE EVENT		STATUS
X DAILY EVENT	X Repeat every 1 day(s)	Scheduled
WEEKLY EVENT	Repeat every weekday	
MONTHLY EVENT		
YEARLY EVENT		
START DATE	START TIME	REPEAT
1/30/2021	12:00 am	every 6 hours
END BY		
<input checked="" type="checkbox"/>	NO END DATE	

### 5.18 Stock Sync Queue

The “Stock Sync queue” is a Custom Record Queue, that Items in NetSuite with Shopify Ids are send to in order to have their stock updated more directly when they were seen on a NetSuite Inventory Adjustment record.

When the Inventory Adjustments in NetSuite are saved and the items are identified, they are added to the Queue. Then the “[Stock Sync](#)” Script automatically Triggers to process each item in the Queue in “Pending” Status, in order to send the New Stock level to the Shopify Item.

### 5.19 Stock Sync Logs

The “Stock Sync Logs” is the Script that triggers Automatically, the Items in the “Stock Sync Queue”, for updates to those items to the Shopify Items.

## 6.1 Testing Installation

It is recommended to have completely gone through this Guide.

### Test An order

Create an order in Shopify to see it get automatically pulled into NetSuite and created.

Please note: Order created in Shopify will get pulled into NetSuite upon the Next Pull of data into NetSuite. By default, this is every 15 minutes. If wanting to Pull data early you can use the Tools Section to Pull Data from Shopify immediately See Section [5.2 NetSuite Setup > Setup/Tools](#) for more information.

## Test Customers

Customers Create/Updated in Vend or in NetSuite will Sync to the other system.

[NetSuite to Shopify](#) Sync can be conditional if the conditions are set in the Customer Mappings. See Section [5.5 NetSuite Setup > Customer](#) for more information.

Either click on the “Sync to Shopify” button, to Sync Customer Record to Shopify based on the conditions set in the Configuration Mappings.

Please Note: See [Shopify Sync Triggers](#) for How to Sync Items and possible Customers

[Shopify to NetSuite](#) Sync will happen when a Customer in Shopify is Created/Updated(settings enabled to pull Customer Data only), or attached to an Order.

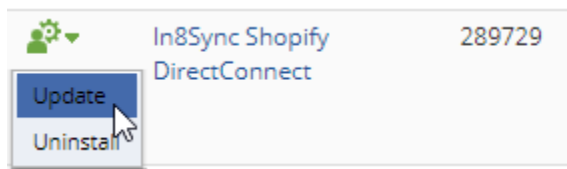
## 7.1 Updating the Bundle

The In8Sync Shopify Bundle can be updated at any time. There will be an indicator next the Bundle Version when a New Version has been released:



From Customization Tab > SiteBundler > Search & Install Bundles > List

Find the Bundle by Name > Hover over the Green Edit Icon > Click on Update



On the Next Page that loads, Confirm the update by Clicking on “Update Bundle”

## Preview Bundle Update



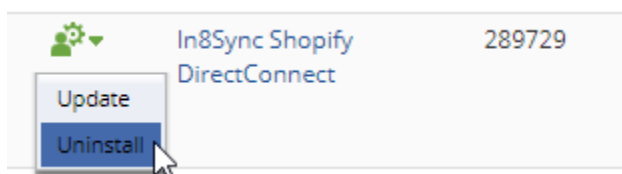
**Name:** In8Sync Shopify DirectConnect **Current Version:**

### 8.1 Uninstalling the Bundle

To uninstall the Bundle, please follow the instructions below:

From Customization Tab > SiteBundler > Search & Install Bundles > List

Find the Bundle by Name > Hover over the Green Edit Icon > Click on Uninstall



On the Next Page that loads, it will show the status of the uninstallation of the In8Sync Bundle:

Press Refresh until completed, then the Bundle will no longer be installed and the Menu and all items from that bundle removed from the NetSuite Account.

## 9.1 Support

### Troubleshooting

Any issues with orders or customer updates will show in the Processing Queue, See Section [5.10 NetSuite Setup > Shopify Requests](#) for more information. If needing support, please See Section [Support](#)

### Support Options

For Advanced Configuration issues or troubleshooting please contact Support with the Below Options.

### Contacting Support

<https://in8Sync.com/support>

Phone: +1-888-604-4120 OPT 3

Business Hours: 8am-6pm MST(MDT)

Estimated Response Time: Support response time is usually same day but could be as long as 48 hours.

### NetSuite Disclaimer

NetSuite does not test, approve, or support SuiteBundles developed by our partners or our customers.

NETSUITE MAKES NO WARRANTIES OF ANY KIND RELATED TO BUNDLES, INSTALLATION OF A BUNDLE IN A NETSUITE ACCOUNT OR ITS USE WITH A NETSUITE ACCOUNT, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND NETSUITE SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, NETSUITE DOES NOT REPRESENT OR WARRANT THAT USE OF THE BUNDLE WILL MEET YOUR REQUIREMENTS OR THAT YOUR USE OF NETSUITE SERVICE WITH THE BUNDLE WILL BE UNINTERRUPTED, TIMELY, SECURE OR FREE FROM ERROR.

IN NO EVENT SHALL NETSUITE HAVE ANY LIABILITY TO YOU OR ANY THIRD PARTY ARISING OUT OF OR RELATED TO YOUR USE OF THE BUNDLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, COVER OR PUNITIVE DAMAGES, HOWEVER CAUSED, WHETHER IN CONTRACT, TORT OR UNDER ANY OTHER THEORY OF LIABILITY, AND WHETHER OR NOT YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. WITHOUT LIMITING THE FOREGOING, NETSUITE SHALL HAVE NO LIABILITY FOR LOSS OF PROFITS, REVENUE OR DATA OR FOR INTERRUPTIONS IN SERVICE ARISING OUT OF OR RELATED TO YOUR USE OF THE BUNDLE. THE FOREGOING DISCLAIMER SHALL NOT APPLY TO THE EXTENT PROHIBITED BY APPLICABLE LAW.