



Infor M3 Integration for Shopify

Version 2.0

Copyright © 2023 Infor

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident, or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material, and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade, or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor M3 Integration for Shopify 2.0

Publication date: June 2025

Contents

About this guide	5
Intended audience	5
Related documents.....	5
Disclaimer	5
Privacy policy.....	5
Overview	6
Terminology	6
Contacting Infor	6
Integration checklist.....	7
Integration of solution components.....	7
Business processes the integration supports.....	8
Features in this integration	10
Data mapping	11
Configuring Infor M3.....	11
BOD Processor Administration.....	11
Configuring InventoryCountReport	11
Configuring Customer.....	13
Configuring PriceList	12
Configuring SalesOrder.....	12
Configuring Shopify in API Gateway	15
Registering Shopify API Suite in API Gateway.....	15
Configuring an authorized application	15
Configuring Shopify	16
Configuring the location in the Shopify Store	16
Configuring the connector app in the Shopify store	16
Pending Transactions	17
Connecting to ION.....	18
Downloading the solution zip file	19
Importing Old Connection Point Solution	20
Importing object schemas.....	20
Importing and Approving Scripts	20
Importing Dataflows.....	21
Configuring AccountingEntity Filter in data flows.....	21
Configuring TenantID in data flows	22
Configuring the service account.....	22
Verifying the Infor M3 connection points	22
Updating the Infor M3 connection points	23
Updating the Shopify connection points	24
Updating the Shopify API Gateway connection points	24

Verifying the data flows between Infor M3 and Shopify	25
Activating the data flows	25
Importing New Connection point solution	25
Importing object schemas	25
Importing the data flow	26
Configuring AccountingEntity Filter in data flows	26
Configuring TenantID in data flows	27
Approving script documents	27
Configuring the service account	28
Verifying the Infor M3 connection points	28
Updating the Infor M3 connection points	29
Updating the Shopify connection points	29
Updating the Shopify API Gateway connection points	30
Verifying the data flows between Infor M3 and Shopify	30
Activating the data flows	31
Shopify Integration - Discount Handling	32
Publishing BODs	24
Starting applications and services	24
Initial data load	24
Sending the initial data load from Infor M3	24
Synchronizing items	25
Synchronizing inventory	25
Synchronizing price list	25
Processing sales order	26
Synchronizing shipment	27
Processing customer return	27
Verifying the integration	28
Verifying the data flow	28
Verifying business tasks	28
Performing ongoing business processes	28
Troubleshooting	29
Unsupported features	33
Net change history	32

About this guide

This document describes the integration of Infor M3 and Shopify. It contains the integration requirements, configuration tasks, and troubleshooting information.

Intended audience

This guide is intended for the system administrator or consultant who configures the integration between Infor M3 and Shopify.

Related documents

You can find these documents on docs.infor.com.

- Infor M3 Cloud Configuration Guide
- Infor M3 Outbound BOD Mapping and Descriptions
- Infor M3 Inbound BOD Mapping and Descriptions
- Infor M3 to Shopify Cross BOD Mapping and Descriptions

Disclaimer

Infor offers this product as complimentary content. As such, it does not fall under standard Infor Support and is self-supported.

Privacy policy

By using the Infor eCommerce Connector, customer data may be transferred cross-border. The relevant terms and conditions set forth in the customer's agreements with Shopify and Infor shall apply.

Overview

This chapter provides an overview of the integration of Infor M3 with Shopify using Infor ION.

Terminology

The terminology between Infor M3 and Shopify differs in some areas. To explain the process and communicate steps clearly, sometimes the BOD terminology is used. This table shows an overview of the Infor M3, Shopify, and BOD terminology.

Infor M3 term	Shopify	BOD term
Customer order	Orders	SalesOrder
Item	Products	ItemMaster
Shipment	Fulfillment	Shipment
Customer return	Return	CustomerReturn
Physical Inventory	Inventory	InventoryCount

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at <https://concierge.infor.com/> and create a support incident.

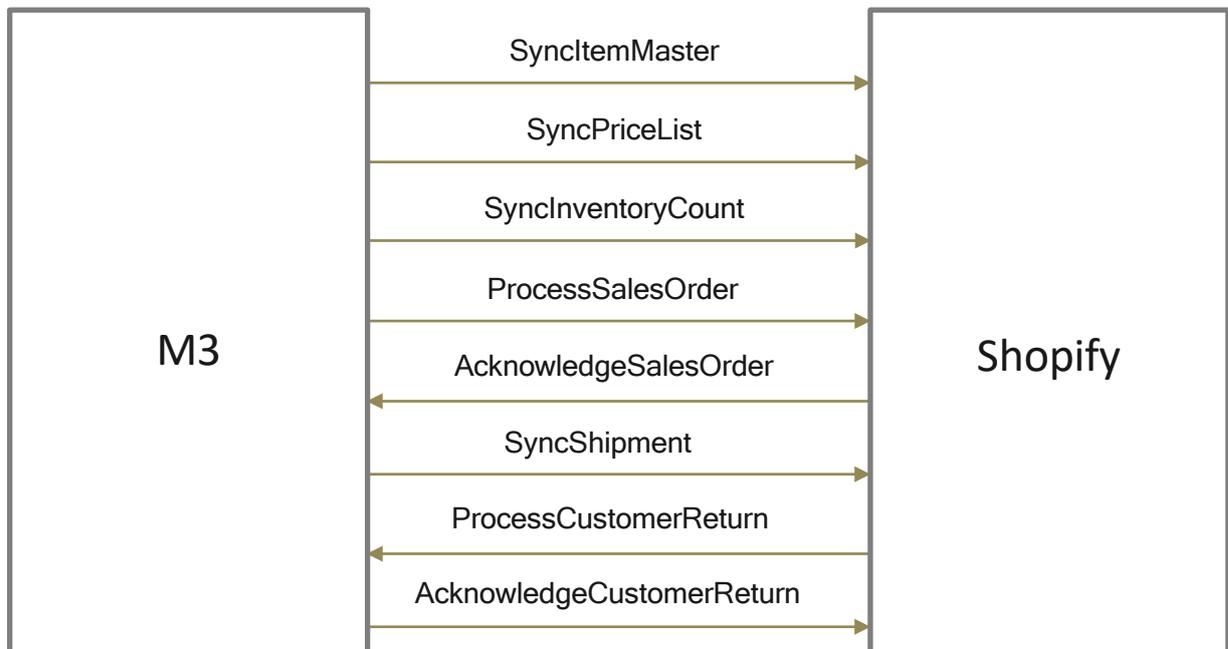
Integration checklist

Follow this checklist to integrate the products.

Complete?	Task	Reference
<input type="checkbox"/>	Collect the documents that are listed in the next column from docs.infor.com. These documents are required for this integration.	<i>Infor M3 Cloud Configuration Guide</i>
<input type="checkbox"/>	Ensure that all requirements and prerequisites are met.	<i>Requirements</i>
<input type="checkbox"/>	Review the description of this integration.	<i>Overview</i>
<input type="checkbox"/>	Verify that the Infor M3 instances you plan to use are connected to Infor ION. If they have not been configured, configure the connection.	<i>Infor M3 Cloud Configuration Guide</i>
<input type="checkbox"/>	Verify that the Shopify instances you plan to use are connected to Infor ION. If they have not been configured, then configure the connection.	<i>Configuring Shopify</i>
<input type="checkbox"/>	Perform additional configuration of Infor M3 specific to this integration.	<i>Configuring M3</i>
<input type="checkbox"/>	Perform additional configuration of Shopify specific to this integration.	<i>Configuring Shopify</i>
<input type="checkbox"/>	Publish BODs.	<i>Publishing BODs</i>
<input type="checkbox"/>	Verify the integration.	<i>Verifying the integration</i>

Integration of solution components

This diagram shows the flow of BODs between the products:



Business processes the integration supports

Shopify Inc. is a Canadian multinational e-commerce company with headquarters in Ottawa, Ontario. It is also the name of its proprietary e-commerce platform for online stores and retail point-of-sale systems. Shopify Inc. do have several product offerings, but the standard integration with Infor M3 is only tested against the base-offering known as Shopify.

The integration with M3 is a basic integration supporting a Business-to-Consumer (B2C) engagement model designed for capturing customer orders with short delivery timescales of standard products from stock. The integration uses Infor BODs as a communication mechanism, and a purpose-built Mediator app using Infor OS & AWS componentry that allows Shopify to easily integrate with Infor's BOD-based ecosystem. There is no release planned for M3 13.4 On- premises or single-tenant as much of the componentry used is not available outside the cloud.

M3 is the system of record for item and item-warehouse information. Basic information is transferred from M3 to Shopify through BOD. All items that are sent to Shopify are flagged as both sellable and returnable in Shopify, although items can be filtered using standard ION techniques. Within the standard integration, only normal items are supported. The process does not support styles and SKUs, configurable items, attribute-controlled or cores. Item images are retrieved from Infor Document Management using the item number as the key. It is possible to configure Shopify to utilize a PIM solution, but this is not part of the standard integration.

Shopify receives an updated stock file each day from M3 with net change indicating current stock balances. Inventory level checks are done against existing stock

with Shopify taking into account the consumption by customer orders placed in Shopify since the last file update. Shopify does not support interactive API capability and only on-hand stock can be sold as there is no concept of future available-to-promise. Shopify is unaware of customer orders placed through other channels. As such, a dedicated M3 warehouse is highly recommended to isolate stock to ensure customer orders are placed accurately, without over- committing stock.

The account ID is a parameter in Shopify at the tenant level. All Shopify B2C accounts are mapped to a single M3 CustomerID, using the Anonymous Customer philosophy in M3. One ID is used for all customer accounts, with the delivery address being tracked per customer order and not as part of the customer address master data. Addresses should not be saved.

Item prices and currency are retrieved using the Sync.PriceList BOD. Shopify does not support complex pricing models and does not have interactive pricing requests. Promotions, discounts, rebates, or charges are not supported by the integration.

All customer orders are deemed to be paid-in-full at order entry prior to the order being submitted to the ERP. The integration uses M3's pre-payment functionality so there is an automated revenue element to the transaction and no credit check is required on the order type. All settings that manipulate the order lines (supply model, joint delivery, etc.) or adjust dates should not be activated. Shopify does have the ability to specify the requested date at a line level only at header level. Shopify cannot consume external updates to the order. Any variations or updates made to the order line dates are not displayed in Shopify if the date was not originally requested. Order cancellations are not supported by the integration currently, as it is not possible to accurately control the timing of possible cancelations with the status control that Shopify has over the status changes in the M3 dispatch process.

In Shopify, the status of the customer order line is updated based on the Sync.Shipment BOD only at the point of delivery. This is called Fulfilment in Shopify.

Customer returns can be created in Shopify, and although synchronized to the ERP, they are flagged as immediately actioned in Shopify, and the customer credit is made instantaneously through Shopify. The credit is not synchronized to the ERP.

Shopify supports multiple options for credit card processing; however, it does not support CenPOS which is the Infor standard provider. The customer order that is passed from Shopify through the integration to M3 does not contain the reference numbers required for the credit card capture step or payment reconciliation. This work must be done externally to the CloudSuite.

As all orders are deemed to be paid in full in advance, all invoices produced by the ERP should be suppressed for the one customer ID used. There is no feedback on the invoice to Shopify as this is not required. But, as soon as the payment is received, the invoices should be closed using the M3-based processes.

Features in this integration

Feature	Description
Company	Integration supports only one company and one accounting entity(division).
Currency	Only one currency is supported.
Items	Only sellable items are synchronized from Infor M3 to Shopify.
Customer	All Shopify B2C accounts are mapped to a single Infor M3 customer ID, utilizing the <i>Anonymous Customer</i> philosophy in Infor M3.
Inventory	A dedicated Infor M3 warehouse is highly recommended to isolate stock to ensure customer orders are placed accurately, without over-committing stock.
Sales order	One ID is used for all customer accounts, the delivery address is tracked by order and not as part of the master data.
Customer return	Customer returns can be created in Shopify and synchronized to Infor M3. Customer credit is instantaneously made through Shopify. The credit is not synchronized to Infor M3.
Tracking Numbers	Tracking numbers are supported in the shipments. The tracking URL will be provided in a dummy format, which needs to be updated by end user.
Discounts	Order discounts, Product discounts and shipping discounts are supported from Shopify to M3. The reduced price will be synced to the M3.

Data mapping

You can use the *Infor M3 to Shopify Cross BOD Mapping and Descriptions* guide to determine these mapped items:

- The source location and target location of the integrated data
- Which conversions, if any, are performed between the source and target

Configuring Infor M3

In Infor M3, you must complete configuration tasks that are specific to this integration.

The *Infor M3 Cloud Configuration Guide* describes the required setup for BOD publishing. To integrate Infor M3 with Shopify, you must set up additional data and verify conditions.

BOD Processor Administration

If Master Accounting Entity (Master AE) is to be used for an existing integration, Infor recommends running an initial load of the master data BODs that support this functionality. Enabling a noun's Master AE affects how Infor M3 publishes the BOD of the respective noun, which may affect the integration. Activate this functionality with caution.

For more details on Master Accounting Entity, see the *Infor M3 CE Core Administration Guide*.

Configuring InventoryCountReport

These settings are required to trigger the SyncInventoryCountReport BOD:

Stock Message Partner

1. Open 'Stock Msg Partner. Open' (MMS865).
2. Specify this information on panel B:

Whs

Specify the warehouse.

Msg

Specify .

Partner

Specify BOD.

3. Select **Options > Add**.
4. Enter user ID in **Partner manager**.
5. Specify this information:

Field / Parameter	Value
Pln receipts	Select the checkbox
Picking list	Set to 1

Item master

Select the checkbox

Create MBM init	Select the checkbox
Download dely	Set to 1
Download instr	Check box should not be selected

Notes:

- Create a stock message partner for all warehouses that are used by the integration.
- For the warehouse setup, verify that the **Item master** field is set to 1 in MMS005 on panel H.

Standard Document

1. Open 'Standard Document. Open' (CRS027).
2. Specify E01 in the **Dno** field.
3. Select **Options > Add**.
4. Specify this information:

Field / Parameter	Value
Name	User-defined field
No. of copies	Set to 1
Doc category	Set to 1
Program	Set to MHS631

Standard Document Connect Media Control Object

1. Open **Std Document. Connect Media Ctrl Object** (CRS945).
2. Specify E01 in **Doc no**. Click **Apply**.
3. Specify the partner and warehouse you created in MMS865. Click **Options > Add**.
4. Enter the description and name. Click **Next**.
5. In **Related > Media**, specify MBMEVENT in **Media** and a sequence number in **Seq no**.
6. Click **Options > Add**.
7. Specify the warehouse in **Rcvr ref data 2**.
8. Specify BOD in **Rcvr ref data 3**. Click **Next**.

Note: Create a media control object for all warehouses used by the integration.

Configuring Customer

The integration is adopting the Anonymous Customer philosophy in M3.

Configuring PriceList

These are required settings to trigger the SyncPriceList BOD:

Shop

1. Open 'Shop. Open' (OPS500).
2. In 'Whs. Click Options', click **Add** to specify the warehouse.
3. Specify this information:

Panel	Field / Parameter	Value
E	Subscriber	Warehouse
G	POS system itf	Select the check box
	Sales price itf	Select the check box
H	Price list	Enter one (1) price list specified in 'Sales Price List. Open Basic' (OIS021)
	Curr method	Select the check box

Note: For the warehouse setup, verify that **Shop information** on the E panel in (MMS005) is selected to indicate that shop information can be entered for the warehouse.

Configuring SalesOrder

These settings are required for processing sales order and cash desk:

Order Type. Open (OIS010)

Panel	Field	Value
E	Dsp dely addr	Check box should not be selected
	Create address	Check box should not be selected
F	New entry order	Check box should not be selected
	Close	Set to 0
	New entry lines	Check box should not be selected
	Upd OV wo risk	Check box should not be selected
G	Chk duplicates	Set to 0
	Upd cum weight	Check box should not be selected

Panel	Field	Value
H	Avail check	Set to 0
	Reasonably chk	Check box should not be selected
	Supply model	Leave blank
	Round order qty	Set to 0, 2, or 3
I	Sls price rule	Set to 1
	Promotion check	Set to 0 or 1
	Cost price rule	Set to 3, 4, or 5
	Quantity limit	Set to 0.00
	Price date chk	Leave blank
K	Adv invoicing	Set to 4
(OIS014/E)	Request dely tm	Drop-down box value is set to 0
F	Route	Leave blank
	Route departure	Leave blank
G	Customer's ord	*M
K	Joint delivery	Leave blank

'Settings - Batch Orders' (OIS278/E)

Panel	Field	Value
E	Source sls prc	Set to 1
	Complete orders	Check box should be selected
	Severity level	Set to 20

Settings - Customer Returns (OIS399)

Panel	Field	Value
E	CO check	Set to 2

Cash Desk. Open (OIS210)

Panel	Field	Value
E	Multi-users	Check box should be selected

Dispatch Policy. Open (MWS010)

Panel	Field	Value
E	Released for allocation	Check box should be selected
	Released for picking	Check box should be selected
F	Allow overissues	Set to 0
G	Closing point	Set to 2

Payment Term. Open (CRS075)

Panel	Field	Value
E	Cash Payment	Set to 2
	Payment prop	Check box should be selected

Configuring Shopify in API Gateway

In API Gateway, you must complete configuration tasks that are specific to this integration.

Registering Shopify API Suite in API Gateway

1. Access the API Gateway application in Infor OS.
2. Select **Available APIs** from the menu and click **Add**.
3. Select `Shopify` from the list of API suite templates.
4. Click **Add Deployment** in the **Deployment Information** section.
5. Specify this information:
 - a. **HostName: Specify the hostname**
 - i. <https://shpfy.shpfy.eu1.prd3.inforcloudsuite.com/api>
 - ii. **Authentication type:** Specify `OAuth2.0`.
 - b. **Token Endpoint:** Specify the token endpoint
 - i. <https://shpfy.shpfy.eu1.prd3.inforcloudsuite.com/api/auth>
 - c. **Client ID & Client Secret:** Specify the client id and client secret details provided by Shopify.
 - i. **Scope:** Specify `admin, user`.
 - ii. **Client Authentication:** Specify `Send as Auth Header`.
6. Click **Save**.

Configuring an authorized application

1. Select **Authorized Apps** in API Gateway.
2. Click **Add New App**.
3. Select **Backend service**.
4. Specify the name as `M3_Shopify_Backend_Service` and description as `M3 Shopify Backend service`.
5. Enable **Issue Refresh Token**.
6. Click **Save**.
7. Click **Download Credentials** and enable the **Service Account**.
8. Specify the user who has all administrator rights under **Full Name**.
9. Download the `M3_Shopify_Backend_Service.ionapi` file for further reference.

Configuring Shopify

In Shopify, you must complete configuration tasks that are specific to this integration.

Configuring the location in the Shopify Store

Prerequisite: Ensure that a store already exists in Shopify.

These are the required settings to create a location in the Shopify store:

1. Navigate to **Settings** in the Shopify store, click the **Locations** tab.
2. Click the **Add location** button.
3. In **Location Name**, provide the location name in the format that matches the M3 Facility_Warehouse value. Click **Save**.
4. Example: If the facility value is A01 and warehouse value is 001, provide location name as A01_001.
5. In **Address**, provide the address details of the warehouse in Shopify with reference to the M3 warehouse address. Click **Save**.
6. **Note:** Ensure that all the M3 warehouses with respect to its facility are added as locations in Shopify store in the similar way.
7. Click **Change default location** in the **Default location** section, select the required location from the drop down and click **Save**.
8. **Note:** This makes sure that the product will be linked to selected location by default.

Configuring the connector app in the Shopify store

Prerequisites: Before proceeding with the configuration, ensure that the location matching the M3 Facility Warehouse value is created in Shopify, and it should be set as the default location.

1. Click Apps and select All recommended apps option. Then, navigate to Shopify Apps store.
2. In the next screen, specify the app name as infor-ecommerce-connector in the search box.
3. Click the Install button.
4. If prompted for login details, provide the required information.
5. In the next screen, click the Open app button. The app is loaded into the store.
6. You can see the recently installed application in the Installed Apps section. Click the app and wait for a few seconds until the app is loaded.
7. Navigate to the dashboard page by clicking the Go to Dashboard button after the setup instructions. In the Configuration Setup, click Alter Setup.

Note: if the signed-in user is a Shopify owner, the user can view the dashboard page.

When a Shopify user signs in to the Shopify connector app, it checks the user credentials, such as whether the signed-in user is an account owner or staff member of the store. If the signed-in user is an account owner, the Shopify connector app provides access to the user to create or update the cross-reference between the Shopify store and the Infor ERP system, along with creating the API Gateway credentials. These credentials are used in the Infor API Gateway to establish the connection between Infor OS and the Shopify connector app to invoke Shopify APIs.

1. In the **Cross Reference** tab, click **Upload ION API file**, choose the ION file downloaded from Infor M3. Refer to Configuring an authorized application section for the download steps.
2. It should automatically populate the **Tenant ID** field picked from the uploaded file. Select the product ^{M3} from the dropdown.
3. Specify the accounting entity in the **Accounting Entity** field, the dummy customer ID in the **Dummy Customer ID** field, and the charge code in the **Charge Code**.
4. Check the checkbox if you want to send the data to IMS in JSON format. If unchecked, XML will be sent by default.

Note:

- i. Ensure that the dummy customer ID is created in M3.
 - ii. If a charge code is maintained, only one charge code can be defined per store, and this value must be created manually in M3. Otherwise, an error message will be displayed during integration.
 - iii. The checkbox option is not available from the next release onwards.
5. Click **Save Configuration** button. This should save the cross-reference information to the server, and this record is displayed in the **Infor Tenant Configurations** grid in the same page.
 6. Select the **ION API Credentials** tab in the connector app, click the **Generate new Credentials** button to generate the new client ID and client secret.
 7. Ensure you to update these as client ID and client secret in the registered Shopify API Suite within the API Gateway section of M3. See: Configuring Shopify in API Gateway.

Pending Transactions

If a webhook fails to reach IMS due to an issue, you can view the failed webhook data by clicking the List button under **Pending Transactions** section in **Dashboard** tab. After clicking it, you'll see the unreachable webhooks organized by topic, following our standard supported webhook topics such as **order/create** and **refund/create**.

From this view, you can:

- Re-trigger individual webhook messages by clicking the **Sync** button next to each entry.
- Re-trigger all messages under a specific category by clicking the **Sync All** button.

NOTE:

1. The retention period for stored order/return messages is 30 days from the moment of their first failure.
2. Modifying the current ION file setup will result in the deletion of the entire data table associated with that specific store.

Connecting to ION

This integration has a solution XML file that is used to set up a sample model. When you import the solution XML file in ION, the sample connection points and data flows for this integration are defined in ION.

In this integration, A new **configuration checkbox** has been added in the Shopify Connector to enable sending webhook payloads **directly in JSON format** to ION for Orders & Returns.

- If the check box is not selected, use the existing logic solution which receives XML as input from shopify.
- If a check box is selected, use the new logic solution which receives JSON payload from shopify, translate the same and send the BOD(XML) in ION to M3.

Before importing existing solution or new solution, it is mandatory to import monitor and distribution group.

Importing Monitor

1. Extract the **ZIP** package and locate the **Shopify_Error_Notification.xml** file **M3_Shopify_Monitor**
2. In **ION**, Navigate to **Monitors & Workflows -> Monitors**.
3. Click on the **Import** icon in the **Monitors** section.
4. Select the **Shopify_Error_Notification.xml** file from the local path.
5. Click **OK** to begin the import.
6. Once the import is completed, A confirmation popup will appear with the import results click **OK** to proceed.
7. **Activate the Monitor**.

Importing Distribution Group

1. Extract the **M3_Shopify_DistributionGroup.zip** from the solution package and locate the **DISTR1_1.CSV** file.
2. In **Security from OS**, Navigate to **Manage --> Groups**, Click on the **Import** button.
3. Select the **DISTR1_1.CSV** file from the local path.
4. Click **Import** and click **CLOSE** in a popup that appears.
5. Re-enter the **Groups** section, you can locate group with a name "shpygroup".
6. Click on the **details** button which is to the left side of the group name and add the required users to whom you like to receive error notification mails.
7. Once **Users** are added, click on **Save** button.

Note: The user should have below IFS roles [related to groups] to perform the above activity

Downloading the solution zip file

To maintain uninterrupted transactions, we are providing two solution zip files.

Please follow the import process as per the configuration selected on Shopify side:

- If the check box is **not selected**, please use the existing logic solution zip which receives **XML** as input from shopify uses the connection point of logical id "**infor.ims.shopify**".
- If the check box is **selected**, please use the new logic solution zip which receives JSON payload from shopify, translate the same and send the XML BOD to M3 uses the connection point of logical id "**infor.ims.shopifywebhook**".

Package Files:

The below .zip file consists of existing logic solution which receives XML as input from Shopify.

- m3ion-integration-to-shopify-1.x.yyyymmddhhmmss_OLD_CP.zip

The package below consists of new logic solution which receives JSON payload from Shopify, translate the same and sends the XML BOD to M3.

- m3ion-integration-to-shopify-1.x.yyyymmddhhmmss_NEW_CP.zip

Importing Old Connection Point Solution

Extract `m3ion-integration-to-shopify-1.x.yyyymmddhhmmss_OLD_CP.zip` to a local folder. These folders will be available within the extracted directory, containing their respective files.

- `M3-Shopify_ObjectSchemas.zip`
- `M3_Shopify_Scripts.zip`
- `M3_Shopify_Dataflows.zip`
- `M3_Shopify_Monitor.zip`
- `M3_Shopify_DistributionGroup.zip`

Importing object schemas

1. In ION, click **Data Catalog > Object Schema**.
2. Click **Import**.
3. Browse and select `M3-Shopify_ObjectSchemas`.
4. These object schemas are imported:
 - `ShopifyProduct`
 - `ShopifyInventoryCount`
 - `ShopifyFulfillment`
 - `ShopifySalesOrder`
 - `ShopifyReturn`
 - `ShopifyPriceList`
 - `M3OrderAPIRequest`

Importing and Approving Scripts

1. Extract the `M3_Shopify_Scripts.zip` from the solution package and locate the **Scripts.json** file.
2. In ION, Navigate to **Scripting -> Scripts**.
3. Click on **Import symbol** and select the **Scripts.json** file from your local [downloaded from package] and click **OK**.
4. In the **import** popup:
 - **Select the scripts** which will be appended. In our case we can see only below two scripts.
 - `ShopifyAckSalesOrder`
 - `ShopifyAckReturn`
 - Please select them.
 - Once the scripts are selected, the **“Skip Already Existing”** option will be greyed out.
5. Click **OK**.
6. A popup stating successful import of scripts will appear. Click **OK**.
7. The Scripts will appear in a **draft** state with the latest version.
8. **Open** the scripts and **approve** it.

Note: Before update or delete of scripts please make sure note down the custom changes you made.

Importing Dataflows

1. In ION, Navigate to **Connect → Dataflows**.
2. Delete the following existing dataflows, if present:
 - Shopify_M3_Orders
 - Shopify_M3_Return
3. Extract the ZIP package and locate the dataflows.xml file in **M3_Shopify_Dataflows.zip** folder.
4. Click on the **Import** icon in the Dataflows section.
5. Select the **dataflows.xml** file from the local path (under the ION package's old connection folder).
6. In the import popup:
 - For **Connection Points**, choose "**Merge Additional Documents**".
 - For **Document Flow Items**, leave the default selection as "**Skip Already Existing**".
7. Click **OK** to begin the import.
8. A confirmation popup will appear with the import results — click **OK** to proceed.
9. The imported dataflows will now be visible.
10. **Approve any related scripts** that are associated with the imported dataflows.

Note: Before deleting the dataflows, please make sure note down the custom changes you made it.

Configuring AccountingEntity Filter in data flows

You must update the Accounting Entity filters that are included in the dataflows.

Follow the steps below to configure the accounting entity:

- Select **Connect > Data Flows** in ION.
- Select the **M3_Shopify_ItemMaster** data flow.
- Highlight the decision filter.
- In the **AccEntityFilter** name, navigate to the **Conditions** tab in **Filter Properties**.
- Edit the **AE** record, remove the dummy value and add AccountingEntity value to it, for example, 770_AAA.
- Click **Save**

Note: Repeat the above steps for the data flows mentioned in this table with the respective **Filter names for AccountingEntity**.

Data Flow	Filter names for Accounting Entity
M3_Shopify_Fulfillment	AccEntityFilter
M3_Shopify_InventoryCount	AccEntityFilter
M3_Shopify_PriceList	AccEntityFilter
Shopify_M3_Orders	AccEntityFilter
Shopify_M3_Return	AE

Configuring TenantID in data flows

You must update the TenantID value in the dataflows. Follow below steps to configure the TenantID:

- Click **Connect > Data Flows** in ION.
- Select the **M3_Shopify_ItemMaster** document flow.
- Select the **ItemToProduct** value.
- Navigate to the **Input Parameters** tab and select `Value` from the input for the TenantID record in **Script Activity Properties**.
- Remove the dummy value and update the TenantID value.
- Click **Save**.

Note: Repeat the same process for the remaining data flows:

Data Flow	Script Activity for Accounting Entity
M3_Shopify_Fulfillment	ShopifyShipment
M3_Shopify_InventoryCount	InventoryCount
M3_Shopify_PriceList	M3toShopifyPriceList
Shopify_M3_Orders	Salesorder
Shopify_M3_Return	Return

Configuring the service account

1. Click the **User Menu**.
2. In **User Management**, select **Manage > Service Accounts**.
3. Click **Add New Item**.
4. Specify Shopify as **Description** and select a user who has administrator roles to call the API Gateway. Click **Save**.
5. Download `ServiceAccount.csv` to a local folder for further reference.

Note: The user must have the administrator rights to access **User Management**.

Verifying the Infor M3 connection points

1. In ION, select **Connect > Connection Points**.
2. Verify that each of the provisioned Infor M3 connection points contain the documents listed in the table:

Document	Receive in application	Send from application
Sync.InventoryCount		X
Sync.ItemMaster		X
Sync.PriceList		X
Sync.Shipment		X
Process.CustomerReturn	X	
Process.SalesOrder	X	

Updating the Infor M3 connection points

You must update the connection points that are included in the imported data flows to use the existing Infor M3 connection point.

1. In ION, select **Connect > Data Flows**.
2. On the **Data Flows** page, select the **M3_Shopify_ItemMaster** flow for this integration and click **Details**.
3. The data flow has activity boxes that represent the connection points. The document icons between these activity boxes list the documents that flow between the connection points.
4. On the **Modeler** page, select the activity box that is labeled **M3**. This box represents your Infor M3 instance connection point in the data flow. This connection point sends documents to the Shopify connection point.
5. In the application task pane, click **Add**.
6. Select the Infor M3 connection point that is configured during the Infor M3 provisioning activity. Save the changes.
7. Repeat these steps for the document flows with the respective connection point activity:

Document Flow	Connection point activity
M3_Shopify_InventoryCount	M3
M3_Shopify_ItemMaster	M3
M3_Shopify_Fulfillment	M3
M3_Shopify_PriceList	M3
Shopify_M3_Orders	M3
Shopify_M3_Return	M3

Updating the Shopify connection points

Update the Shopify Infor Messaging Service (IMS) connection point that is included in the imported data flow to use the existing Shopify IMS connection point.

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the shopify connection point.
3. Remove the Dummy ION API Client ID and Specify the **ION API Client ID**. This ION API Client ID (ci JSON property value) can be found in “Configuring an authorized application”
4. Save the changes.

Updating the Shopify API Gateway connection points

Update the Shopify API Gateway connection point that is included in the imported document flows to use the existing Shopify API Gateway connection point.

M3_Shopify_Creation

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the `M3_Shopify_Creation` connection point.
3. Click **Import**.
4. Browse and select `ServiceAccount.csv` created in “Configuring *the service account*.”
5. Save the changes.

M3_Shopify_IDMImg

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the `M3_Shopify_IDMImg` connection point.
3. Click **Import**.
4. Browse and select `ServiceAccount.csv` created in “Configuring *the service account*.”
5. Save the changes.

Verifying the data flows between Infor M3 and Shopify

1. In ION, select **Connect > Data Flows**.
2. Verify that the solution data flows exist.
3. For the data flows, click the document icons in the flows and verify that they contain the expected documents, as shown in Integration of solution components.
4. Add any missing documents or messages.

Activating the data flows

1. In ION, select **Connect > Data Flows**.
2. On the **Data Flows** page, select the **M3_Shopify_ItemMaster** data flow and click **Activate**.
3. Repeat these steps for these data flows:
 - M3_Shopify_Fulfillment
 - M3_Shopify_InventoryCount
 - M3_Shopify_PriceList
 - Shopify_M3_Orders
 - Shopify_M3_Return

Importing New Connection point solution

Import this solution when the JSON checkbox is configured in Shopify.

Extract `m3ion-integration-to-shopify-1.x.yyyymmddhhmmss_New_CP.zip` to a local folder. These folders will be available within the extracted directory, containing their respective files.

- M3-Shopify_ObjectSchemas.zip
- M3_Shopify_Dataflows.zip
- M3_Shopify_Monitor.zip
- M3_Shopify_DistributionGroup.zip

Importing object schemas

1. In ION, click **Data Catalog > Object Schema**.
2. Click **Import**.
3. Browse and select `M3-Shopify_ObjectSchemas`.
4. These object schemas are imported:
 - a. ShopifyProduct
 - b. ShopifyInventoryCount
 - c. ShopifyFulfillment
 - d. shopifyreturns
 - e. ShopifyPriceList

-
- f. M3OrderAPIRequest
 - g. shpfysalesorder
 - h. shpfy_webhook_response
 - i. shpfy_webhook_finalresponse
 - j. shopify_refund_final
 - k. shopify_finalrefund
 - l. shopifygeterpid

Importing the data flow

1. Extract the ZIP package and locate the **dataflows.xml** file in **M3_Shopify_Dataflows.zip** folder.
2. In ION, click **Connect > Data Flows**.
3. Click **Import**.
4. Browse and select **Dataflows.xml**.
5. For any new version import, Rename or delete the existing dataflows and import the new dataflows.
6. If any custom dataflow is used, Select **Skip Already Existing**. Changes will not be affected in this case.

Note: Verify that the connections, data flows, scripts, and mappings are imported from the file.

7. These data flows are imported from the XML file.
 - M3_Shopify_Fulfillment
 - M3_Shopify_InventoryCount
 - M3_Shopify_ItemMaster
 - M3_Shopify_PriceList
 - Shopify-M3_ProcessJSON_OrderAndReturn

Configuring AccountingEntity Filter in data flows

You must update the Accounting Entity filters that are included in the M3_Shopify_ItemMaster data flow.

1. Select **Connect > Data Flows** in ION.
2. Select the **M3_Shopify_ItemMaster** data flow.
3. Highlight the decision filter.
4. In the **AccEntityFilter** name, navigate to the **Conditions** tab in **Filter Properties**.
5. Edit the AE record, Remove the dummy value and add the **AccountingEntity** value to, for example, **770_AAA**.
6. Click **Save**
7. Repeat the above steps for the data flows mentioned in this table with the respective **Filter names for AccountingEntity**.

Data Flow	Filter names for Accounting Entity
M3_Shopify_Fulfillment	AccEntityFilter
M3_Shopify_InventoryCount	AccEntityFilter
M3_Shopify_PriceList	AccEntityFilter
Shopify- M3_ProcessJSON_OrderAndReturn	AccEntityFilter AccEntityFilter

Configuring TenantID in data flows

You must update the TenantIDs that are included in the M3_Shopify_ItemMaster data flow.

- Click **Connect > Data Flows** in ION.
- Select the **M3_Shopify_ItemMaster** document flow.
- Select the **ItemToProduct** value.
- Navigate to the **Input Parameters** tab and select `value` from the input for the TenantID record in **Script Activity Properties**.
- Remove the dummy value and update the TenantID value.
- Click **Save**.
- Repeat the same process for the remaining data flows:

Data Flow	Script Activity for Accounting Entity
M3_Shopify_Fulfillment	ShopifyShipment
M3_Shopify_InventoryCount	InventoryCount
M3_Shopify_PriceList	M3toShopifyPriceList
Shopify- M3_ProcessJSON_OrderAndReturn	Ack_SalesOrder_Req SalesOrderReturn_Req

Approving script documents

1. In ION, click **Scripting > Scripts**.
2. Approve these script documents:
 - IDM_Image_Merge
 - IDM_Image_requestFormation
 - ShopifyFulfillment
 - ShopifyInventoryCount
 - ShopifyPriceList
 - ShopifyProduct
 - M3_OrderRequest
 - M3_Shopify_OrderID_Script
 - Shopify_JSON_Payload

- **Shopify-M3_ProcessSalesOrder**
 - M3-Shopify_SalesOrder_ACK
 - M3-Shopify_SalesOrder_ACK_Req
 - Shopify-M3_OrderReturn_ErpID
 - Shopify-M3_GetERPIID_Query
 - Shopify-M3_OrderReturn_JSON_Req
 - Shopify-M3_ProcessCustomerReturn
 - M3-Shopify_SalesOrderReturn_ACK
 - M3-Shopify_SalesOrderReturn_ACK_Req

Note: If the master accounting entity is enabled for ItemMaster, the company and division should be hardcoded in the ShopifyProduct script.

Configuring the service account

1. Click the **User Menu**.
2. In **User Management**, select **Manage > Service Accounts**.
3. Click **Add New Item**.
4. Specify Shopify as **Description** and select a user who has administrator roles to call the API Gateway. Click **Save**.
5. Download `ServiceAccount.csv` to a local folder for further reference.

Note: The user must have the administrator rights to access **User Management**.

Verifying the Infor M3 connection points

1. In ION, select **Connect > Connection Points**.
2. Verify that each of the provisioned Infor M3 connection points contain the documents listed in the table:

Document	Receive in application	Send from application
Sync.InventoryCount		X
Sync.ItemMaster		X
Sync.PriceList		X
Sync.Shipment		X
Process.CustomerReturn	X	
Process.SalesOrder	X	

Updating the Infor M3 connection points

You must update the connection points that are included in the imported data flows to use the existing Infor M3 connection point.

1. In ION, select **Connect > Data Flows**.
2. On the **Data Flows** page, select the **M3_Shopify_ItemMaster** flow for this integration and click **Details**.

The data flow has activity boxes that represent the connection points. The document icons between these activity boxes list the documents that flow between the connection points.

3. On the **Modeler** page, select the activity box that is labeled **M3**. This box represents your Infor M3 instance connection point in the data flow. This connection point sends documents to the Shopify connection point.
4. In the application task pane, click **Add**.
5. Select the Infor M3 connection point that is configured during the Infor M3 provisioning activity.
6. Save the changes.
7. Repeat these steps for the document flows with the respective connection point activity:

Document Flow	Connection point activity
M3_Shopify_InventoryCount	M3
M3_Shopify_ItemMaster	M3
M3_Shopify_Fulfillment	M3
M3_Shopify_PriceList	M3
Shopify-M3_ProcessJSON_OrderAndReturn	M3

Updating the Shopify connection points

Update the Shopify Infor Messaging Service (IMS) connection point that is included in the imported data flow to use the existing Shopify IMS connection point.

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the shopify connection point.
3. Remove the Dummy ION API Client ID and Specify the **ION API Client ID**. This ION API Client ID (ci from JSON property value) can be found in “Configuring an authorized application”
4. Save the changes.

Updating the Shopify API Gateway connection points

Update the Shopify API Gateway connection point that is included in the imported document flows to use the existing Shopify API Gateway connection point.

M3_Shopify_Creation

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the `M3_Shopify_Creation` connection point.
3. Click **Import**.
4. Browse and select `ServiceAccount.csv` created in “Configuring the service account.”
5. Save the changes.

M3_Shopify_IDMImg

1. In ION, select **Connect > Connection Points**.
2. On the **Connection Points** page, select the `M3_Shopify_IDMImg` connection point.
3. Click **Import**.
4. Browse and select `ServiceAccount.csv` created in “Configuring the service account.”
5. Save the changes.

Verifying the data flows between Infor M3 and Shopify

1. In ION, select **Connect > Data Flows**.
2. Verify that the solution data flows exist.
3. For the data flows, click the document icons in the flows and verify that they contain the expected documents, as shown in Integration of solution components.
4. Add any missing documents or messages.

Activating the data flows

1. In ION, select **Connect > Data Flows**.
2. On the **Data Flows** page, select the **M3_Shopify_ItemMaster** data flow and click **Activate**.
3. Repeat these steps for these data flows:
 - M3_Shopify_Fulfillment
 - M3_Shopify_InventoryCount
 - M3_Shopify_PriceList
 - Shopify-M3_ProcessJSON_OrderAndReturn

Shopify Integration - Discount Handling

- **Supported Discounts**

Our Shopify integration supports the following types of discounts:

- Order Discounts
- Product Discounts
- Shipping Discounts

- **Product Discounts**

When a product discount is applied, the discounted amount is subtracted from the unit price of the product.

For example:

- If the actual product price is ₹100 and a ₹10 product discount is applied, the unit price sent will be ₹90.

- **Order Discounts**

- If the order-level discount is a fixed amount (e.g., ₹100 off) or a percentage (e.g., 10% off the order total), it is distributed among the line items proportionally based on their price.
- The discounted unit price (₹90) will be synced to M3 as item net price (OIS101). Based on the quantity the Line amount will be calculated.
- The net price value should be matched with the shopify line amount.

Fixed Amount Discount Example

- **Items in Cart:**

- Product A: \$20
- Product B: \$40
- Total Subtotal: \$60

- **Discount:** \$12 off the entire order

- **Calculation:**

- Product A Discount: $(\$20 / \$60) \times \$12 = \4
- Product B Discount: $(\$40 / \$60) \times \$12 = \8

Percentage Discount Example

- **Items in Cart:**

- Product A: \$20

-
- Product B: \$40
 - Total Subtotal: \$60
 - **Discount:** 20% off the entire order ($\$60 \times 0.20 = \12 total discount)
 - **Calculation:**
 - Product A Discount: $(\$20 / \$60) \times \$12 = \4
 - Product B Discount: $(\$40 / \$60) \times \$12 = \8

- **Shipping Discounts**

When a free shipping discount is applied:

- The shipping cost is set to zero. Shipping-related tags will be removed from the BOD.
- If there is no shipping charge, there is no need to send shipping-related tags in the BOD.

When No shipping discount is applied:

- The Shipping cost will be synced to M3 with the respective tags.
- The shipping amount will be synced to the charges in M3.

Note: The Charge code which is defined at store configuration should be maintained in M3 in OIS030.

This ensures accurate and optimized data transmission while maintaining proper discount calculations in the system.

Publishing BODs

After the ION connections are configured, BODs must be published for this integration.

Starting applications and services

Start these applications, services, and processes if they are not already running:

- Infor M3
- Shopify
- ION Service. Use the ION Grid Management user interface to verify that the ION Service is running. See the *Infor ION Grid Administration Guide*.

Initial data load

During normal processing, BODs are generated when a user makes a data change or completes a transaction. When you set up a new integration with Shopify, you must manually generate BODs that pass the current set of Infor M3 data to Shopify and vice versa, so that the base systems are synchronized. The BODs are placed in the message outbox, from which Infor ION retrieves them and passes them to the other application.

Sending the initial data load from Infor M3

Specify this information for ItemMaster:

BOD Verb

Specify `Sync`.

Table

Specify `MITMAS`.

Field

Specify `ITNO`.

Note: Use the BOD verb `Sync` when executing the initial load of ItemMaster.

After the initial data load, Infor M3 sends BOD updates when users change certain data or when certain transactions occur in Infor M3.

If data exists in M3 Business Engine, it can be published to the database using initial load sessions in M3 Business Engine.

See "Starting an initial data load from (EVS006)" in the *M3 CE Core Administration Guide* for how to perform an initial load of M3 data into BODs that are placed in the message outbox.

For a list of the events that generate BODs, see "Outbound BODs" in the *Infor M3 CE Core Administration Guide*.

Synchronizing items

1. The Infor M3 user adds or updates an item in 'Item. Open' (MMS001). Select Sales item in MMS001 panel G.
2. Infor M3 sends the SyncItemMaster BOD to ION.
3. As soon as the ItemMaster BOD is in ION, ION invokes the Shopify APIs to create or update the product in Shopify. The administrator monitors the appropriate message queues to identify and resolve the errors.

Notes:

- Only items with `Open` status and sales items from M3 are considered in Shopify.
- The item should be available in the default warehouse.
- Images can also be synchronized with Shopify, but first the image file must be uploaded to IDM, where the `id1` attribute should match the M3 item number. The Infor M3 user should update the item in order to display the image in Shopify.

Synchronizing inventory

1. The Infor M3 user executes 'Item Master Table. Export' (MHS630). Specify the **f Warehouse, Partner, Split output** - Checkbox is ticked, and Number of Records - 3000 (Min).
2. Infor M3 sends the SyncInventoryCountReport BOD to ION.
3. As soon as the Inventory BOD is in ION, ION invokes the Shopify APIs to create or update the inventory in Shopify. The administrator monitors the appropriate message queues to identify and resolve the errors.

Notes:

- The Synchronization of inventory data should be done daily. Alternatively, you can set a job schedule in Infor M3.
- Check whether the inventory location (Facility_Warehouse) is linked to an item in Shopify prior to submitting the job in M3. During creation, only default locations are linked to all products; the other locations will be added automatically.

Synchronizing price list

1. The Infor M3 user adds or updates an item in 'Item. Net Transfer' (OPS620). Specify the **Function Sel** and **Change date** to filter the records to be published.
2. Infor M3 sends the SyncPriceList BOD to ION.

3. As soon as the PriceList BOD is in ION, ION invokes the Shopify APIs to create or update the price list in Shopify. The administrator monitors the appropriate message queues to identify and resolve the errors.

Notes:

- Execute the item transfer for every price list ID.
- The price list ID should exist in 'Sales Price List. Open' (OIS017). Several items in (OIS021) can affect the processing of SyncPriceList. Processing can potentially be terminated due to high memory usage. You may consider splitting the sales price list in (OIS017) to send two or more BODs.

Processing sales order

1. Create an order in Shopify.
2. Select a customer. You have the option to change the shipping address.
3. Add one or more products then the order quantity.
4. Add the discounts* or shipping costs* if applicable.
5. Select the mode of payment as `Mark as paid` in **Collect payment**.
6. Shopify sends a ProcessSalesOrder BOD to Infor M3.
7. Infor M3 sends the AcknowledgmentSalesOrder with the `Accepted` status, then the order will be updated with the `erp_external_ID` to Shopify.
8. In case of failure, Infor M3 sends an acknowledgement with the `Reject` status, this order won't be updated in Shopify. The flow will stop at ION along with the complete error details in the BOD.
9. **Processing Shipping charges:**
 - If the shipping charges are applied, the respective charges will be synced to M3.
 - If the shipping charges are 0 or below, the charges will be excluded from syncing i.e. no shipping costs will be added.
10. **Processing Discounts:**
 - Only product discounts and shipping discounts are supported in this integration.
 - If the shipping cost is 0 after applying the shipping discount, then the charges will be excluded from syncing.

Note:

- Because the integration follows the Anonymous Customer philosophy, the customer order in M3 is always created for the customer number of the provided Dummy Customer ID value in Shopify.
- The charge code should be maintained in M3(OIS030) to synchronize from Shopify to M3. Else the BOD would fail saying the method does not exist.
- For transportation method (delivery method) to sync in M3, the first three characters should be maintained in M3(CRS070) or as an alternative transformations should be created in ION for each specific delivery method that is used in Shopify with respect to its value of delivery code of M3

Synchronizing shipment

1. The Infor M3 application user completes the delivery process of the SalesOrder BOD in 'Delivery.Open' (MWS410).
2. Ext tracking number can be defined to track the order.
3. Infor M3 sends the SyncShipment BOD to ION.
4. The administrator monitors the appropriate message queues to identify and resolve the errors.
 - **Note:** Only shipments with “**Shipped**” status in M3 are sent to Shopify. The tracking URL will be sent as a dummy URL (<https://trackinginfo.com/<trackingNumber>>) which needs to be modified at shopify As soon as the Shipment BOD is in ION, ION invokes the Shopify APIs and the order status gets updated to `Fulfilled` in Shopify end by clicking on edit tracking info at order level.

Processing customer return

1. Select **Order** in Shopify.
2. Select the order to be returned.
3. Select **Return**.
4. In **Return items**, specify the quantity, the reason for return, and the return shipment options. Then select **Create Return**.
5. Select **Order > Refund**.
6. Shopify sends a ProcessCustomerReturn BOD to Infor M3.
7. Infor M3 sends the AcknowledgmentCustomerReturn BOD with the `Accepted` status, then the order will be updated with the `erp_external_ID` to Shopify.
8. In case of failure, Infor M3 sends an acknowledgement with the `Reject` status, this order won't be updated in Shopify. The flow will stop at ION along with the complete error details in the BOD to Shopify.

Verifying the integration

After a successful integration, transactions in Infor M3 and Shopify generate BODs that send data through ION. This section describes how to verify that the integration sends and receives the expected data.

Verifying the data flow

To verify whether the initial data is flowing between the products, check the message inbox and outbox areas in both products to see if the Infor M3 data arrives, and if data is retrieved from Shopify.

If data is not flowing between the systems, see the troubleshooting section in the appropriate configuration guide.

Verifying business tasks

If multiple instances of the applications are sending BODs, verify that the expected data from each instance is available in the appropriate areas of the other application.

Performing ongoing business processes

You must coordinate any daily or scheduled activities between the integrated applications.

Troubleshooting

If data is not flowing between the systems, an error might exist in your configuration with ION. See the Infor OS configuration guides for the integrated applications.

If you receive error messages, see the error message descriptions for this integration.

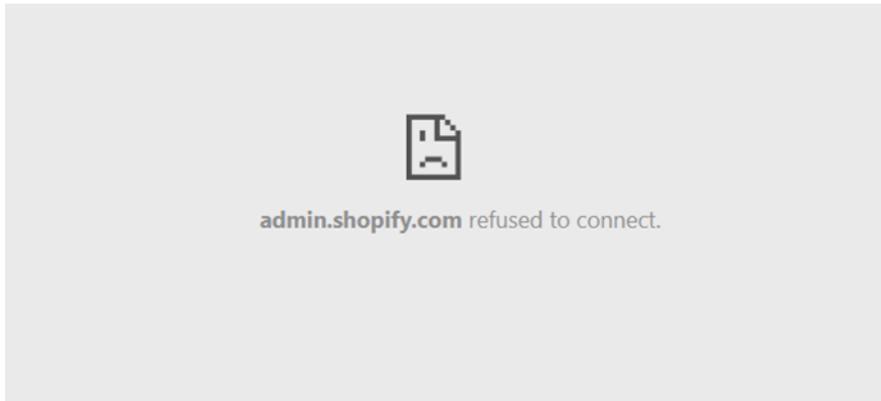
Problem	Possible Causes	Check	Possible Resolution
Update/Addition in Shopify is not displayed in Infor M3.	ION is not receiving the BOD.	If the BOD is not received, check the Queue Viewer in ION. If the routing is correct between Shopify and Infor M3, check the Routing Modeler in ION.	Change the routing and activate the routing or reactivate the routing.
ION received the BOD, but Infor M3 is not updated.	Check whether errors are logged in ION (Connect/Error BODs)	Resolve reported error.	
The update is rejected by Infor M3.	Check the BOD Integration setup in Infor M3. See the Infor M3 Configuration Guide.	Change the routing and activate the routing or reactivate the routing.	
Update/Addition in Infor M3 is not received by Shopify.	The BOD is not received by ION.	If the BOD is not received, check the queue Viewer in ION. If the routing is correct between Infor M3 and Shopify, check the Routing Modeler in ION.	Change the routing and activate the routing or only reactivate the routing.
The BOD is not received by Shopify, but published by ION.	Check the OneView Graphic modeler in ION for the respective business transaction.	Resolve the error specified in the API Gateway connection point.	
Account is created in Shopify not received in Infor M3	Customer is not created in Infor M3	Check the OneView Graphic modeler in ION for the respective	Acknowledge business transaction with rejected status should have clear information of the failure

Troubleshooting

Problem	Possible Causes	Check	Possible Resolution
		Acknowledge business transaction.	message. Resolve the error and resubmit the transaction through Connect -> Error BODs
Item is created in Infor M3 and not synced with Shopify.	Product not created in Shopify.	Check the OneView Graphic modeler in ION for the respective ItemMaster BOD	Fix the error that is displayed in the Confirm BOD and is triggered by the Shopify API.

Issue: App Not Loading in Shopify Store:

If your app is not working or unable to load in the Shopify store, and you encounter issues like the one described, follow these steps to resolve the problem.



Possible Cause:

This issue typically occurs when a new scope has been added to the app and is not yet authorized by the store.

Steps to Resolve:

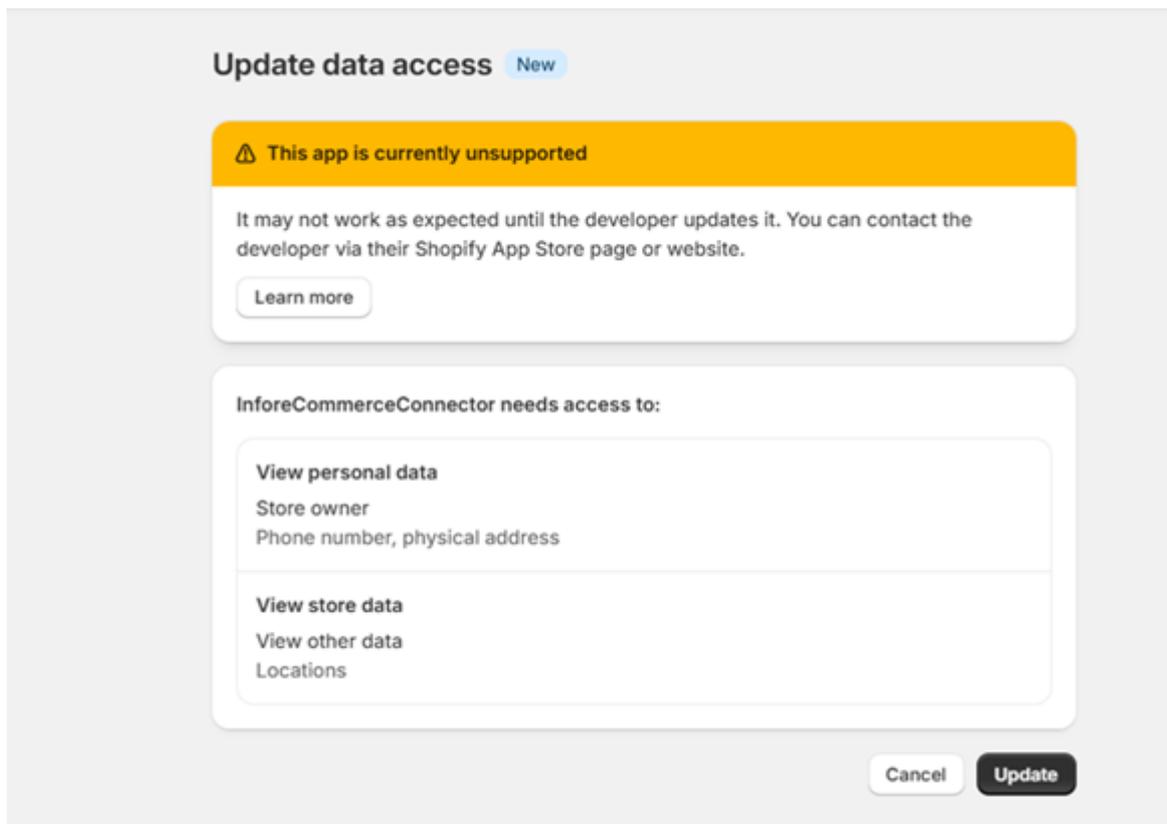
Open the Authorization Page

1. Open the link below in a new browser tab: [Authorization URL](#)



Please input the store address

2. Enter Your Store Address
 - a. In the text box, type your store address in the format: **store_name.myshopify.com**
 - b. Click the Submit button to proceed.
3. Update Data Access
 - a. You'll be redirected to the Shopify store page asking for data access authorization.
 - b. Look for a prompt with an Update button (as shown below).
 - c. Click the Update button.



4. Verify App Functionality
 - a. Once the update is completed, the app should load correctly.
 - b. Refresh your Shopify store page to ensure the issue is resolved.

Issue-2. Refused to Connect After Clicking Update App:

If you encounter a "Refused to Connect" error after clicking the Update App button in the app settings, follow these steps:

1. Navigate to the Dashboard Home
 - a. Locate the Home button on the left-hand side of the store.
 - b. Click on Home to return to the main store page.
2. Reload the App
 - a. Once you are on the Home page, reload the app from the Apps section. The issue should be resolved, and the app will load successfully.

Issue in ION

Troubleshooting and Resubmitting Confirm.BOD Errors in ION OneView

Overview

Whenever an error occurs in the integration flow—especially at the script level—a Confirm.BOD is automatically generated to capture the failure. This mechanism ensures traceability and helps prevent the loss of request payloads from upstream systems like Shopify.

A meaningful exception message is configured to appear in the error details, and an email notification is sent to the designated IFS user group. Once the issue is identified and fixed (typically at the script level), the following steps must be performed to resubmit the Confirm.BOD and allow the paused transaction to continue processing successfully.

Resubmitting Confirm.BOD in ION OneView – V2 Version

1. Log in to the ION Tenant.
2. Search for “**ION**” in the search box and open the **ION** application.
3. Click on **OneView** and navigate to the **Error BODs** tab.
4. By default, the **Error Messages** tab will be displayed, showing a grid of failed transactions.
5. Search for the relevant **Document Name** associated with the issue.
6. Once the error record appears, select the checkbox next to the row and click the “**RESUBMIT**” button (blue button).
7. The system will automatically resubmit the paused transaction, allowing it to proceed through the flow without data loss.

Unsupported features

This table shows the features that are not supported in the integration:

Feature	Description
Delete	Delete functionality is not part of the integration. If any record is manually deleted by the user, it is not synced to the target application.
SalesOrder	An update of the sales order in Infor M3 including cancellation is not supported in Shopify.
Sales price	Shopify does not support complex pricing models and does not have interactive pricing requests. Promotion, Discounts, Rebates, or Charges are not supported by the integration. The sales price is net of discount and tax.
Item	The integration does not support styles and variants, configurable items, attribute-controlled items, core items, or kit items.
Inventory	Shopify does not support interactive API capability, and only on-hand stock can be sold as there is no concept for future Available to Promise.
Taxes	Taxes are not supported in this integration. Only the item price and discounts are synced.

Net change history

Date	Version	Description
2024-05-29	1.0	Initial document
2024-11-19	1.0	Updated several setups
2025-01-24	1.1	Added the charge code at store configuration level.
2025-02-04	1.2	1. Removed Shipping tags if shipping charges are 0. 2. Few deprecated APIs are updated to Graph QL
2025-04-04	1.3	1. Handled different Discounts. Refer to the new section - Shopify Integration - Discount handling. 2. Product meta field type in Shopify is changed from unsupported "string" to "single_line_text_field". This allows customers to edit the meta field value. Download the latest ION package from the Market place. 3. Handled the sales order creation when external id meta field is not in first 50 fields of API response.
2025-06-30	2.0	1. Direct Webhook Payload Delivery (JSON) without BOD Creation (Order & Return) - Shopify Connector. 2. Retry Mechanism for Failed Webhook Payloads to IMS - Shopify connector . 3. Migration from REST API to GraphQL & Shopify Version Update. 4. Improved Error Handling in ION - Email Notifications for Failed Transactions.
