PRODUCT DEFINTION

Columbus M3 CE AP Automation Connector

Contact info:
Richard Lindqvist
richard.lindqvist@columbusglobal.com
+46 73 147 20 31

Prepared by:
Fredrika Ståhl

Version history

Version	Date	Created by	Description
1.0	2021-01-08	Fredrika Ståhl	First version
2.0	2021-02-10	Fredrika Ståhl	Master data imported through Data lake
3.0	2021-05-05	Fredrika Ståhl	Data lake for initial load and APIs for delta data
4.0	2022-01-17	Moa Sandström	Updated M3 limitations
5.0	2022-02-09	Moa Sandström	Updated required tables M3
6.0	2022-03-24	Moa Sandström	Updated required tables M3
7.0	2022-09-20	Moa Sandström	Updated business limitations and added custom field mapping (4.2)
8.0	2022-12-05	Moa Sandström	Added tables needed for integration and new value for custom field mapping (4.2)
9.0	2023-02-10	Moa Sandström	Additional tables needed for integration and business limitations updated.
10.0	2023-08-17	Moa Sandström	Additional tables needed for integration, business limitations as well as master data frequency updated.
11.0	2024-02-08	Monica Pettersson	Additional tables needed for integration. (2.1)
12.0	2024-02-08	Monica Pettersson	IDM – Invoice document export from Medius (2.4)

Contents

1. Introduction

This document covers the Medius Accounts Payable Automation Multi-Tenant Cloud M3 integration interface. This document will give an overview of the integration as well as some technical descriptions. All descriptions are based on the integration for Infor M3 Cloud/Infor OS.

1.1. System overview

This chapter will give a brief description of the system on a high level.

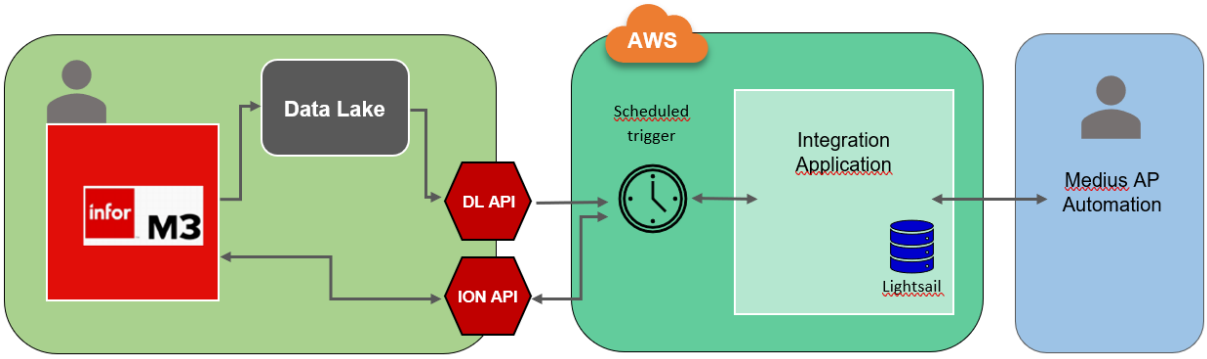


Image 1. System overview

The integration service runs within Amazon Web Services (AWS) on an EC2 server. This virtual server ensures scalability and security. The integration service is always active however, all the data flows are run by a schedule set in the service.

2. Solution overview

The integration service consists of three interfaces in Amazon web services (AWS). The communication is done in both directions but objects in the M3 standard are never changed. Invoices are updated through APS450MI.

A short description of the interfaces:

- Master data, sends formatted data from M3 to Medius AP Automation (Medius APA).
- Invoice posting, sends invoice data from Medius APA to M3.
- Invoice verification, sends verification in the form of the voucher number from M3 to Medius APA of the posted invoice.

2.1. Master data

This interface is responsible for synchronizing master data from M3 to Medius APA. The initial load is done through Data Lake. The synchronization is performed in a set order: Units, Currencies, Currency rates, Dimension values, Dimension restrictions, VAT codes, Payment terms, Suppliers, Items and Purchase orders. This is done to make sure that the later objects, for instance purchase orders, have all the data needed to function. Below entities are processed through this interface.

Once all data has been loaded, a flag is changed in the configuration enabling retrieval of delta data and new objects through EXPORTMI and standard APIs (specified in section 3.5). The interface is run by a schedule with a frequency interval based on type of master data.

Beside importing master data, the integration makes controls of invoice statuses, for example to know when to set the invoice to approved for payment. This requires access to the affected tables, even though it's not master data. These (specified below) are only required to access with EXPORTMI and not Data Lake.

Master data entity	M3 table	Purpose	Frequency
Units	CSYTAB	Get units	One time/day
Currencies	CCURRA	Get currencies	One time/day
Currency rates	CCURRA	Get currency rates	One time/day
Dimension values	FCHACC	Get dimensions	One time/hour

Dimension restrictions	FCHACC	Get restrictions	One time/hour
Dimension intervals	FCHCHK	Cross-account checking	One time/hour
Payment date	FPLEDG + FAPIBH	Get payment date	One time/hour
VAT code incl. rates	CSYTAB + CVATPC	Get vat codes incl rates	One time/day
Payment terms	CSYTAB	Get payment terms	One time/day
Suppliers	CIDMAS	Get suppliers	One time/hour
Suppliers	CSUDIV	Get supplier exceptions	One time/hour
Suppliers	CIDVEN	Get supplier finance	One time/hour
Suppliers	CIDREF	Get supplier reference	One time/hour
Suppliers	CSYTAB	Get system table	One time/hour
Suppliers	CBANAC	Get bank account information	One time/hour
Suppliers	CIDADR	Get supplier address	One time/hour
Suppliers	CPAALI	Get partner alias	One time/day
Items	MITMAS	Get item	One time/hour
Items	MITPOP	Get supplier item	One time/hour
Purchase orders	MPHEAD	Get order heads	Every 15 minutes
Purchase orders	MPLINE	Get order lines	Every 15 minutes
Purchase orders	MPOEXP	Get order charge lines	Every 15 minutes
Purchase orders	MPCELE	Get costing element	Every 15 minutes
Purchase orders	FGRECL	Get delivery lines	Every 15 minutes
Purchase orders	FGRPCL	Get delivery charge lines	Every 15 minutes

Purchase orders	FGINLI	Get reversed and/or closed delivery lines	Every 15 minutes
Purchase orders	CACCST	Get coding lines	Every 15 minutes
Purchase orders	CMNUSR	Get user definition	Every 15 minutes
Purchase orders	MITAUN	Get alternative unit of measure	Every 15 minutes
Invoice posting (only EXPORTMI, not Data Lake)	FCR040	Control invoice is not stuck in GLS037	During invoice posting
Projects (only if activated in customer config upon request)	BPROJS	Import projects from project module	One time/hour
Project restrictions (only if activated in customer config upon request)	BPROJS	Import project restrictions from project module	One time/hour

Table 1. Tables where access in needed

2.2. Invoices

This flow contains two interfaces, Invoice posting and Invoice verification. Before this flow can be run Medius APA must have been populated with master data. Once the master data is in place an invoice is created in Medius APA. The invoice posting interface then formats and adds the data to M3 through APS450 via APIs.

2.3. Purchase Orders

The purpose of the purchase order flow is to sync purchase orders and changes that occur on the orders. These changes include changes on lines such as quantities, addition/removal of items as well as deliveries of goods on the purchase order.

The default behavior can be described as follows:
Initially, when the purchase order is created, it is not synced to Medius APA. It is only synced when there has been a recent delivery of goods on the order (status 70 to 80). The data from the calls will be joined and enriched before it is sent to Medius APA. If required, it is possible to transfer purchase orders and goods receipts with a lower status than the standard 70-80. This is in those cases set by the integration consultant in the customer configuration.

2.4. IDM - Invoice export from Medius

Upon request, Columbus can enable the invoice to be exported to IDM. Per default, the following attributes are being sent in the message.

```
attributeList.add(getAttribute( name: "BOD_SupplierInvoiceID", invoiceHead.getSupplierInvoiceNumber()));
attributeList.add(getAttribute( name: "BOD_SupplierPartyID", invoiceHead.getSupplierNumber()));
attributeList.add(getAttribute( name: "BOD_RemitToPartyID", invoiceHead.getSupplierNumber()));
attributeList.add(getAttribute( name: "BOD_AlternateDocumentID_1", invoiceHead.getDocumentId()));
attributeList.add(getAttribute( name: "M3_VoucherNumber", invoiceHead.getVoucherNumber()));
attributeList.add(getAttribute( name: "M3_InvoiceDate", DateUtils.getM3DateUtc(invoiceHead.getInvoiceDate())));
attributeList.add(getAttribute( name: "M3_AccountingDate", DateUtils.getM3DateUtc(invoiceHead.getAccountingDate())));
attributeList.add(getAttribute( name: "BOD_AccountingEntityID",  value: company + "_" + division));
attributeList.add(getAttribute( name: "M3_Company", String.valueOf(company)));
attributeList.add(getAttribute( name: "M3_Division", division));
attributes.setAttr(attributeList);
```

All above are invoice standard fields whereas t ‘invoiceHead.DocumentID’ is Medius serial number submitted as CORI in AddHead.

Following above, file name and invoice image can also be sent. The file name is inherited from invoice.imagePath including its full name and the ending ‘.pdf’. The invoice image is transcoded into base64-format in the integration which IDM translates back into the image.

The IDM message is triggered by status 90 in APS450 and requires the passing of GLS037. It is included in the regular invoice verification. As such, if the invoice transaction gets stuck in APS450/GLS037 the IDM message is included in the integration’s queue system and will be sent once the issue has been resolved in APS450/GLS037.

3. Installation

3.1. Configuration in Medius APA

This chapter will describe the setup needed in Medius AP Automation. Once you have navigated to “Client application”. Fill out the form as stated in *Image* 2. M3 integration. Scope is set to Integration.Erp, see *Image 3*. Scope. To be set up by Medius Application consultant.

Image 2. M3 integration

Image 3. Scope

3.2. Configuration in Infor OS (M3)

This chapter will describe the setup needed in Infor OS. The general procedure is to set up an integration user (AD user) to run the integration. The integration user needs to be setup in MNS150 with access to APIs and tables listed below (3.5). With this user a backend service can be created in ION API (*Image 4*. Backend Service). When the service is created credentials can be downloaded,

creating a service account associated with the credentials (*Image 5*. Download Credentials). To be set up by M3 technical consultant.

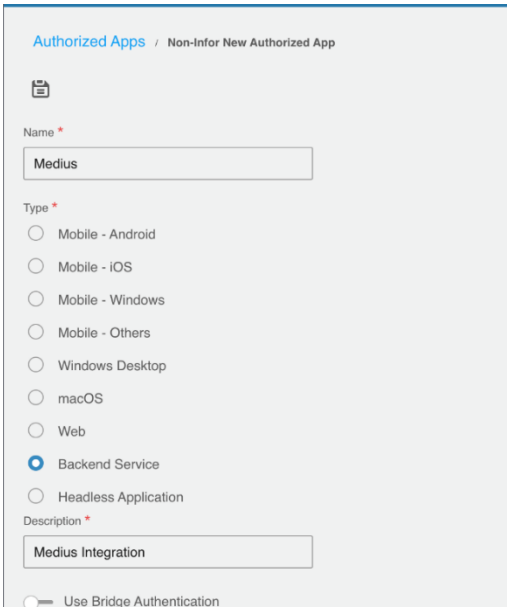


Image 4. Backend Service

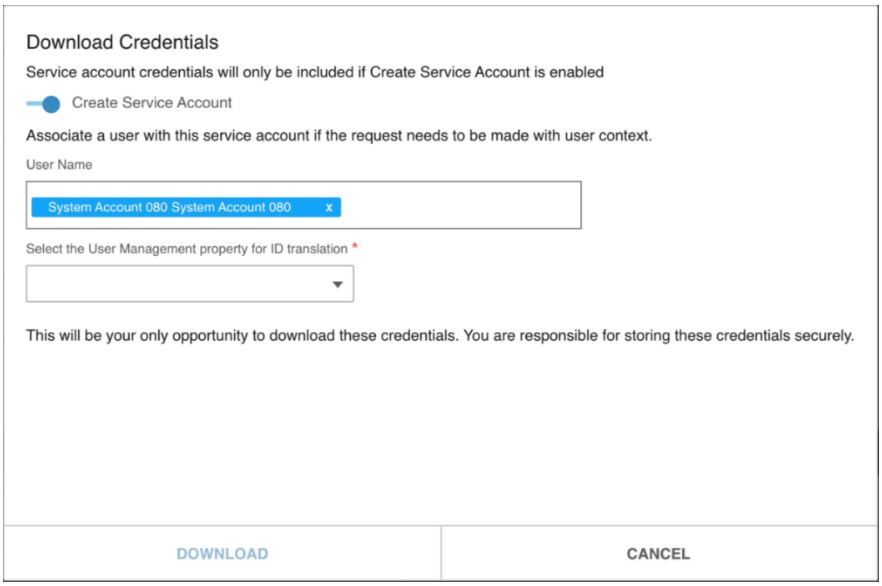


Image 5. Download Credentials

3.3. Integration configuration file

The integration service is hosted on an EC2 server within Amazon Web Services (AWS). A new company is added through an interface for the integration service. A new config file is then automatically created in the Nitrite database (embedded no-sql db). The configuration needs to include authentication information for both M3 (Infor OS) and Medius APA. Authentication is done via OAUTH2 and requires both an M3 user and a Medius user. When a new customer has been added, timestamps can be set in the config file to determine how far back we want to check for master and transactional data. For options regarding what can be set in the customer config, see separate document “Customer configuration”. To be set up by Integration consultant.

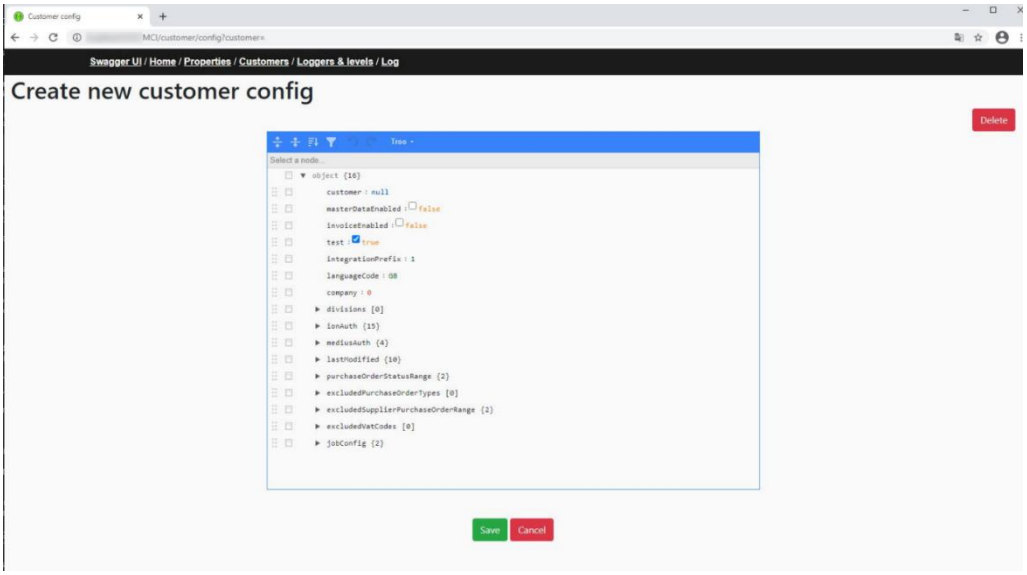


Image 6. Create a new customer config

3.4. Infor Data Lake – Initial load

Data Lake is Infor’s data warehouse exposing access to copies of M3 tables in csv format. Data access is provided by Data Lake’s own setup of rest APIs. Details of the setup can be found in Appendix 1: Data Lake setup guide. To be set up by M3 technical consultant.

3.5. The service user

The service user running the integration needs access to the tables listed under section 2.1 and the APIs below. To be set up by M3 technical consultant.

API
CRS075MI
APS450MI
APS455MI
APS110MI
CRS630MI
GLS200MI
GLS470MI
MMS200MI
EXPORTMI

Table 3. APIs where access is needed for the integration user

4. Prerequisites in M3

Standard Implementation Accelerator settings are not fully compatible with the required setup for Medius APA. The customer (in extension the company implementing the IA) is responsible for validating and adjusting setup in CRS630, CRS395 and APS905. Besides this, the following setup is required:

Setup	Comment
FAM function AP50 in CRS405	Setup specified in M3 config guide
APS900 Tolerance levels	Setup specified in M3 config guide
PPS280 Costing elements	As long as the workaround for manual coding line on purchase order based invoices is used (currently ongoing development). Only applicable for direct booking customers.
APS020 Accounting option 99	To handle small price rounding that can occur during purchase order invoice matching it is required to have APS020 option 99 setup to automatically handle such rounding.

Table 4. Prerequisites in M3

4.1. Known limitations

In the following chapter, known limitations are listed divided by technical/business limitations.

4.1.1 Technical limitations

Updates are handled synchronously, which means that there might be a small delay between updates in M3 or Medius APA and when they are sent to Medius APA or M3 respectively. This is by design to minimize the number of requests and reduce the risk of bottlenecks.

4.1.2 Business limitations

Updates and additions of new functionality are tightly connected with Infor's development of M3 programs and corresponding APIs. The rest of this section lists the known business limitations in the integration to this day.

Master data import	Supported	Current status
Payment information	Partly supported	BCIBAN + BCBACN + BCBAF4 used by default. FIAN, BAF1, BAF2, and BAF3 and prioritization order can be used through configuration.

Invoice transactions	Supported	Current status
Coding line on order based invoice (if preliminary booking is not used)	Not supported	Not supported by Infor
Booking of deviations when quantity deviations occur	Workaround in progress	Not supported by Infor
External charges	Supported with limitations	Charge must be in the same currency as the PO. Operator 90 (used for sales tax) is not supported. Alternative unit of measure on the item with external charge is only supported for direct booking customers.
Match invoices with orders in other currency	Not supported	Not supported by Medius
Fixed asset account booking when preliminary booking invoices	Not supported	Not supported by Infor
Two-way matching when preliminary booking invoices	Not supported	Not supported by Infor

Table 5. Known limitations in M3 regarding master data and invoice transactions

4.2. Medius custom field mapping

In the integration some of the fields on supplier invoice head level, line level and dimension line level are mapped against M3 and can only be used for these purposes. If field is left empty, M3 standard settings are used. If value is entered, this will overrule M3 standard.

4.2.1 Head level

Custom field head level	*Description*	*M3 field*
Text 1	Payee	SPYN
Text 2	Base Country	BSCD
Text 3	-	-
Text 4	Service Code	SERS
Text 5	VAT code	VTCD
Numeric 1	Cash discount base	TASD
Numeric 2	-	-
Numeric 3	-	-
Numeric 4	-	-
Numeric 5	-	-
List 1	Payment method	PYME
List 2	-	-
List 3	-	-
List 4	-	-
List 5	-	-
Date 1 (alternative to designated field VAT date)	Delivery date (VAT date)	DEDA

Payment details	Payment details	BKID
VAT date	Delivery date (VAT date)	DEDA

4.2.2 Line level

Custom field head level	*Description*	*M3 field*

Text 1	-	-
Text 2	-	-
Text 3	-	-
Text 4	Service Code	SERS
Text 5	-	-
Numeric 1	-	-
Numeric 2	-	-
Numeric 3	-	-
Numeric 4	-	-
Numeric 5	-	-
List 1	-	-
List 2	-	-
List 3	-	-
List 4	-	-
List 5	-	-

Freetext dimension 1	-	-
Freetext dimension 2	-	-
Freetext dimension 3	-	-
Freetext dimension 4	-	-
Freetext dimension 5	Voucher text	VTXT

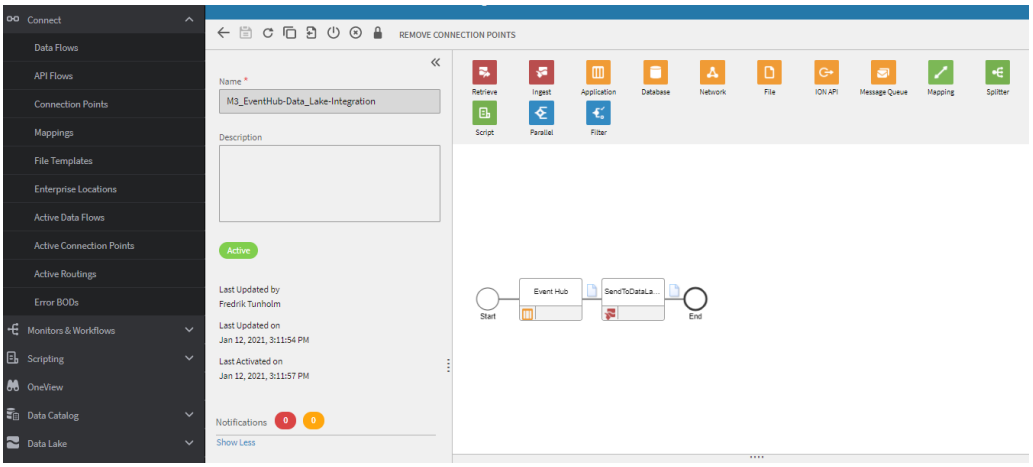
5. Appendix 1: Data Lake Setup Guide

5.1. Required Data Lake tables

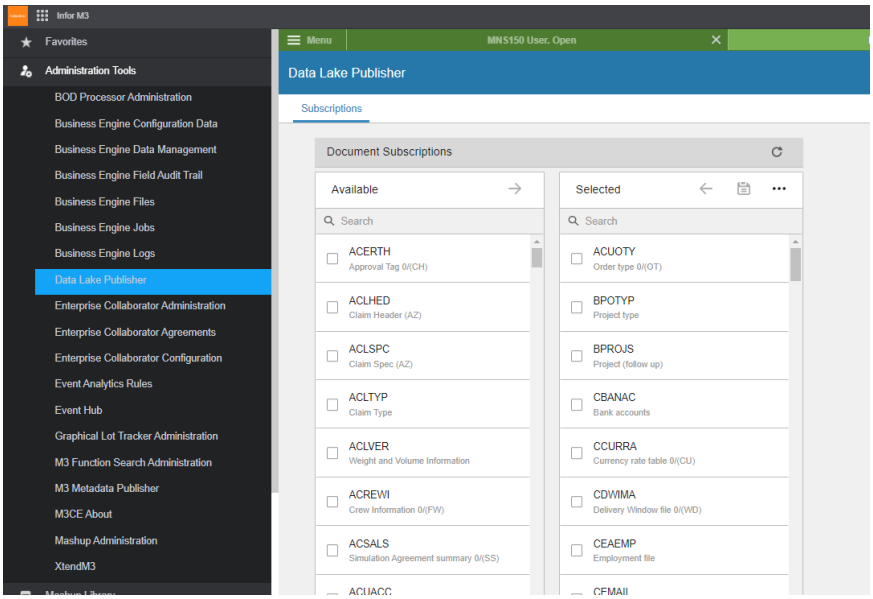
See full list of tables required under section 2.1.

5.2. ION Preparations

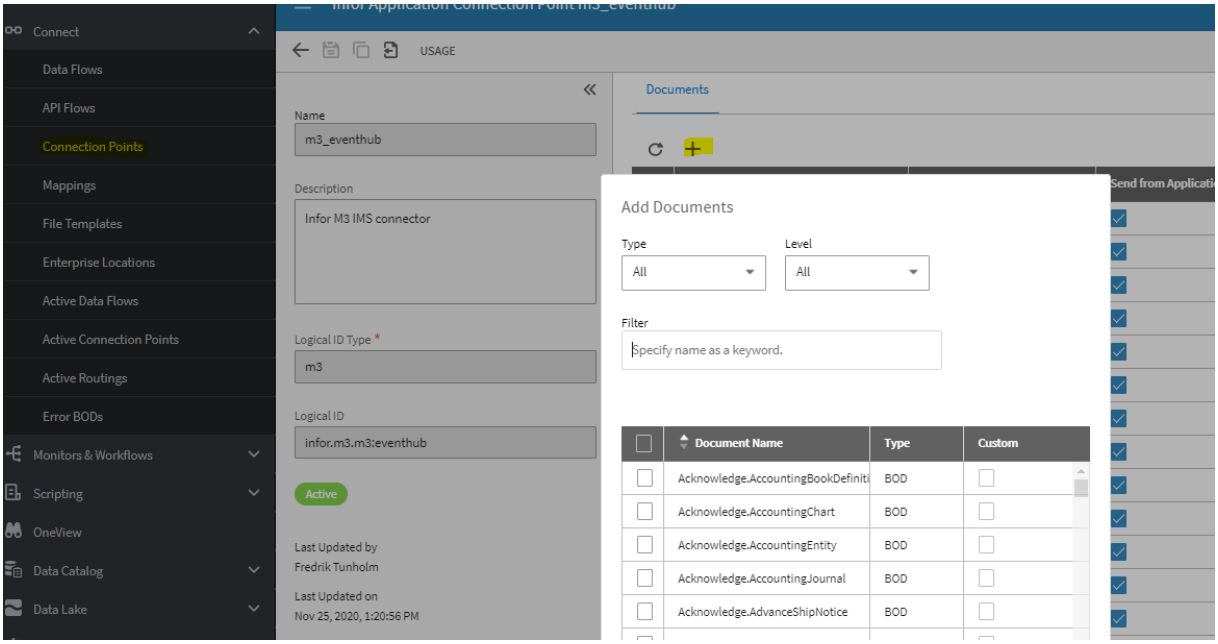
Create a Data Flow that will capture events from Event Hub and send them to Data Lake via the Ingest step.



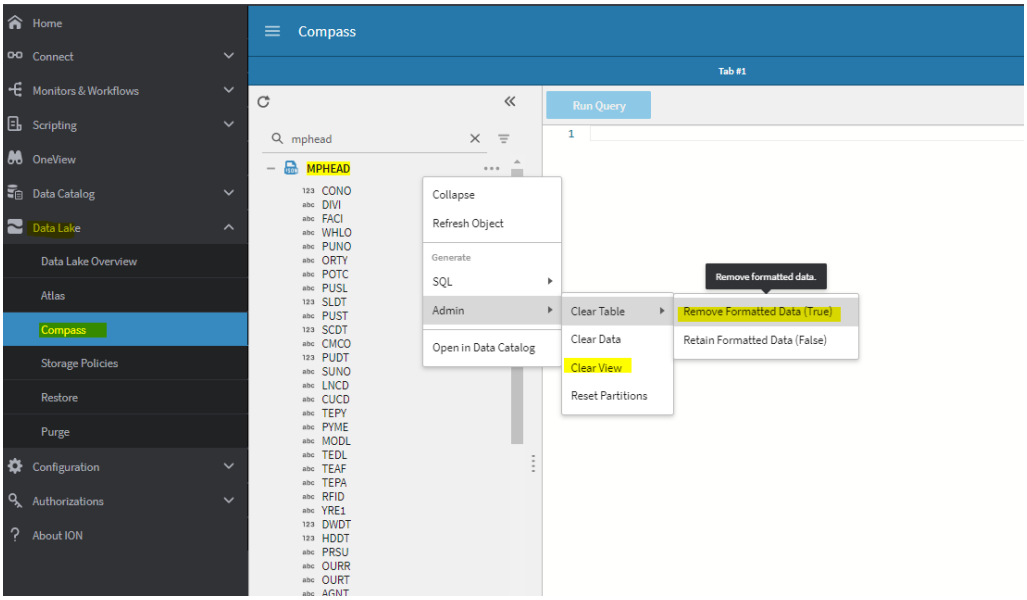
Select all required tables in the Data Lake Publisher



Add the tables in the Event Hub Connection Point. “Send from Application” should be checked.

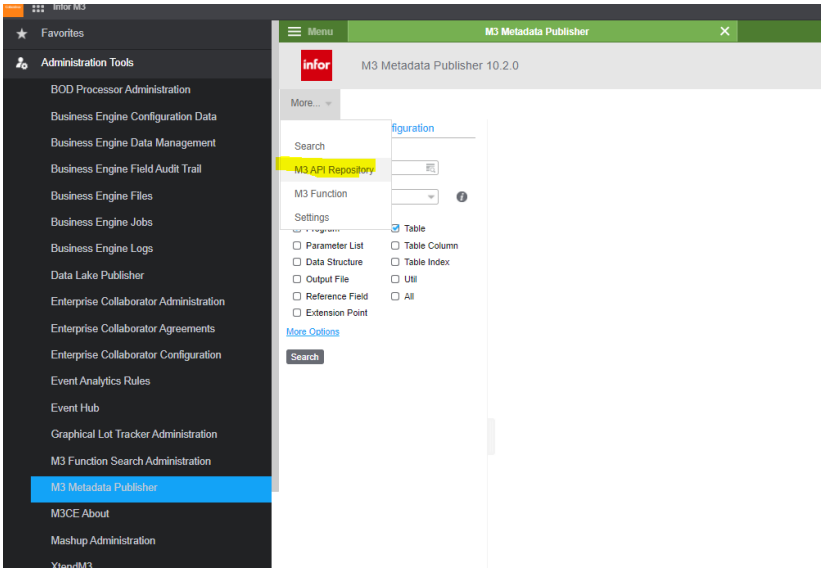


If Infor has introduced changes to the table metadata, then it will be necessary to trigger a metadata refresh. Run “Remove formatted data” and “Clear view” for the tables that require a metadata refresh.

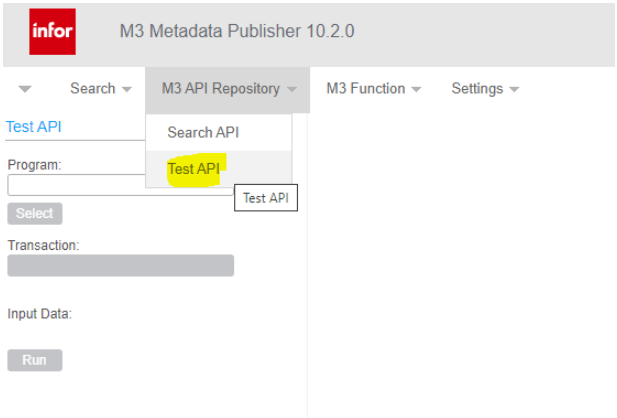


5.3. Load data from M3 to Data Lake (Initial Load)

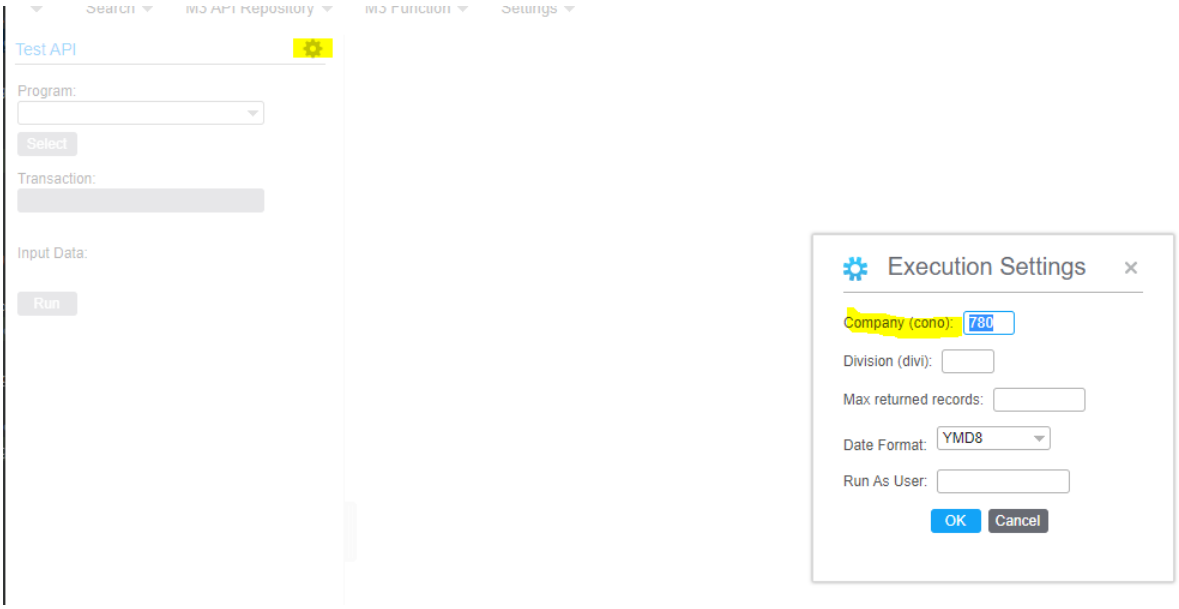
Go to the M3 Metadata Publisher, select “M3 API Repository” (opening this might require a right-click and then “open in new tab”).



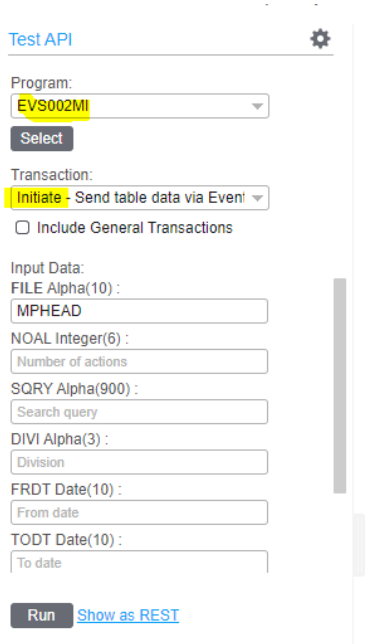
Go to “Test API”



Configure the Test API with the company to do the initial load for (division is optional, leaving it empty will load data for all divisions on selected company).

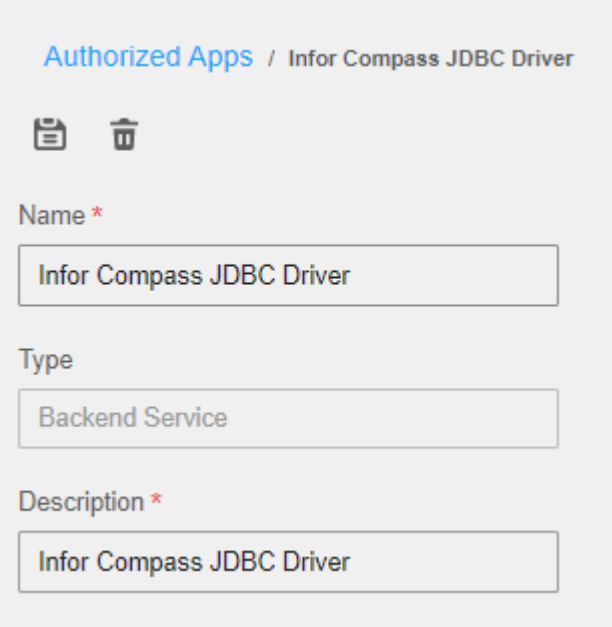


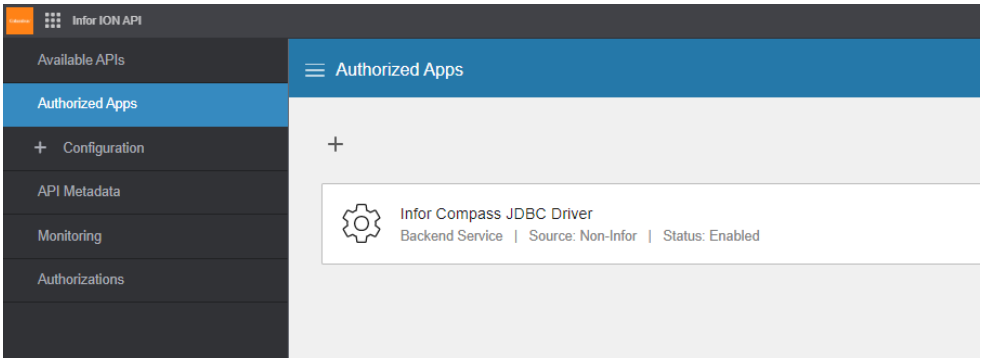
Select the program “EVS002MI” and transaction “Initiate”. Input the table to do the initial load on (FILE). Make sure the field “DTLK” is set to ‘1’ (this means that the data will be sent to Data Lake). Repeat this step for all the required tables.



5.4. Data Lake Authorization

Create an authorized app in Infor ION API. The naming of this authorization is important if the usage of external SQL query editors is to be used to query this Data Lake, which might be good for debugging purposes. Make sure “Issue Refresh Tokens” is enabled for this configuration.





After the authorized app is created, download the credentials file. It is important that this set of credentials is setup with the service account to be used for the integration. The generated file is the credentials that will be used by the integration when communicating with both Data Lake and M3 API.