WHITE PAPER

TRIMIT EXTERNAL API 24.1

This document contains information about the External API functionality in TRIMIT 24.1 for Microsoft Dynamics 365 Business Central 2024, Wave 1 (W1).

Version: 24.1
Date: 30-08-2024

CONTENTS

INTRODUCTION

The **TRIMIT External API** (Application Programming Interface) is an extension on Microsoft's API and enables you to

- Export Master/Item Data
- Export Sales Prices
- Export Inventory/Availability
- Export Customer Data
- Import Sales Documents
- Export Posted Sales Invoices and Posted Credit Memos

This will allow third parties to create customized apps on top of this framework, which can be used for the integration of external systems from and to TRIMIT.
The Integration module, which consist of the **Integration Setup**, and related data, is found in the TRIMIT ERP, and APIs (codeunits) are available for exporting Master/Item data, exporting Inventory/Available Quantities of Items, and for importing Sales Documents and processing them within TRIMIT ERP, thus creating ordinary Sales Documents, Release them and even Post them if required.

Furthermore, exporting Posted Sales Documents is also available in the framework.
The objects are all in our **TRIMIT** App, which will give you the possibility to use OData APIs as an extension of the standard Microsoft APIs.
(See API(V2.0) for Dynamics 365 Business Central - Business Central | Microsoft Learn for more info)

As off this release, we can also give you a Postman collection (see Postman API Platform | Sign Up for Free for more info) that can be used as example to create your own Postman collection.
You can acquire this Postman collection via partnersupport@trimit.com.

In this White Paper, we also added the detailed documentation that you can find in the AL Code of the main two Codeunits that are used for our External API functionality:
Appendix 1: Codeunit 6036860 and Appendix 2: Codeunit 6036862.

The audience for this document is the Consultant or very skilled User.
It will only describe the TRIMIT fields/functionality and not the standard Business Central fields/functionality.
Be aware of not changing settings and parameters in a live database without consulting the implementing partner.

The functionality described is the functionality as of **TRIMIT 24.1 for Business Central W1 BC24.X**.
The pictures in this White Paper are based on the demo database for **TRIMIT 24.1,** which is based on Microsoft Dynamics 365 Business Central 2024 Release Wave 1, in the demo company **CRONUS TRIMIT W1 Ltd.** (based on CRONUS International Ltd. of Microsoft Dynamics 365 Business Central).

COMPONENTS

TRIMIT External API contains the following objects:

The Table	***6036835***	***trm Temp Integration Product***
	6036836	***trm Temp Integration Inventory***
	6036837	***trm Temp Integration SaleTrack***
	6036838	***trm Temp Integration Data***
	6036840	***trm Integration Setup***
	6036841	***trm Integration Setup Line***
	6036842	***trm Integration Exp. Product***
	6036843	***trm Integration Exp. Document***
	6037049	***trm Sales Comment Import***
The Pages	***6036201***	***trm Integration Exp. Products***
	6036870	***trm Integration Setup List***
	6036871	***trm Integration Setup***
	6036872	***trm Integration Customer Setup***
	6036873	***trm Integration Currency Setup***
	6036874	***trm Integration Language Setup***
	6036875	***trm Integration Location Setup***
	6036876	***trm Integration Item Setup***
	6036877	***trm Integration Cust. Template***
	6036879	***trm Integration Table Setup***
	6036881	***trm Integration Sales Docs***
	6036882	***trm Integration Sales Document***
	6036883	***trm Integration Sales Doc Sub***
	6037017 trm Sales Comment Import List	
The API Pages	***6037801***	***trm EAPI Attachments***
	6037802	***trm EAPI Campaigns***
	6037803	***trm EAPI Categories***
	6037804	***trm EAPI Configured Fields***
	6037805	***trm EAPI Customer Price Groups***
	6037806	***trm EAPI Default Dimensions***
	6037807	***trm EAPI Dim.Set Lines***
	6037808	***trm EAPI Item Attributes***
	6037809	***trm EAPI Locations***
	6037810	***trm EAPI PDF Document***
	6037811	***trm EAPI Pictures***
	6037812	***trm EAPI Price Group Params***
	6037813	***trm EAPI Salespersons***
	6037814	***trm EAPI Units Of Measure***
	6037815	***trm EAPI Collections***
	6037816	***trm EAPI Delivery Periods***

The API Pages	***6037817***	***trm EAPI Contacts***
	6037818	***trm EAPI Contacts Info***
	6037819	***trm EAPI Cust Finance Dtls***
	6037820	***trm EAPI Customers***
	6037821	***trm EAPI Return Order Lines***
	6037822	***trm EAPI Return Orders***
	6037823	***trm EAPI Sales Credit Memos***
	6037824	***trm EAPI Sales Cr.Memo Lines***
	6037825	***trm EAPI Sales Document Lines***
	6037826	***trm EAPI Sales Document***
	6037827	***trm EAPI Sales Invoice Lines***
	6037828	***trm EAPI Sales Invoices***
	6037829	***trm EAPI Sales Order Lines***
	6037830	***trm EAPI Sales Orders***
	6037831	***trm EAPI Sales Quote Lines***
	6037832	***trm EAPI Sales Quotes***
	6037833	***trm EAPI Inventories***
	6037834	***trm EAPI Item Additional Info***
	6037835	***trm EAPI Item Additional Tags***
	6037836	***trm EAPI Item Attribute***
	6037837	***trm EAPI Item Carelabel***
	6037838	***trm EAPI Item Composition***
	6037839	***trm EAPI Item Description***
	6037840	***trm EAPI Item Inventory***
	6037841	***trm EAPI Item Price***
	6037842	***trm EAPI Items***
	6037843	***trm EAPI Item Stat. Group***
	6037844	***trm EAPI Item Sustainability***
	6037845	***trm EAPI Master Add. Info***
	6037846	***trm EAPI Master Add. Tags***
	6037847	***trm EAPI Master Attributes***
	6037848	***trm EAPI Master Carelabel***
	6037849	***trm EAPI Master Collection***
	6037850	***trm EAPI Master Composition***
	6037851	***trm EAPI Master Description***
	6037852	***trm EAPI Master Dimension***
	6037853	***trm EAPI Master Inventories***
	6037854	***trm EAPI Master Item***
	6037855	***trm EAPI Master Measure Chart***
	6037856	***trm EAPI Master Price***
	6037857	***trm EAPI Masters***
	6037858	***trm EAPI Master Stat. Group***
	6037859	***trm EAPI Master Sustainability***
	6037860	***trm EAPI Master VarDim***

The API Pages	***6037861***	***trm EAPI Exported Documents***
	6037862	***trm EAPI SalesPostCreditMemos***
	6037863	***trm EAPI SalesPostCrMemoLines***
	6037864	***trm EAPI SalesPostDocLines***
	6037865	***trm EAPI SalesPostDocuments***
	6037866	***trm EAPI SalesPostInvLines***
	6037867	***trm EAPI SalesPostInvoices***
	6037868	***trm EAPI SalesPostRetRcptLines***
	6037869	***trm EAPI SalesPostReturnRcpts***
	6037870	***trm EAPI SalesPostShipLines***
	6037871	***trm EAPI SalesPostShipments***
	6037872	***trm EAPI Sales Tracking***
	6037873	***trm EAPI Product Add. Info***
	6037874	***trm EAPI Product Add. Tags***
	6037875	***trm EAPI Product Attributes***
	6037876	***trm EAPI Product Carelabel***
	6037877	***trm EAPI Product Collection***
	6037878	***trm EAPI Product Composition***
	6037879	***trm EAPI Product Description***
	6037880	***trm EAPI Product Dimension***
	6037881	***trm EAPI Product Item***
	6037882	***trm EAPI Product Measure Chart***
	6037883	***trm EAPI Product Price***
	6037884	***trm EAPI Products***
	6037885	***trm EAPI Product Stat. Group***
	6037886	***trm EAPI Product Sust.ability***
	6037887	***trm EAPI Product VarDim***
	6037888	***trm EAPI VarDim Types***
	6037889	***trm EAPI VarDim Options***
	6037890	***trm EAPI Item Images***
	6037892	***trm EAPI Master Images***
	6037894	***trm EAPI Product Images***
	6037895	***trm EAPI SalesQuote Comments***
	6037896	***trm EAPI SalesQrder Comments***
	6037897	***trm EAPI SalesCrMemo Comments***
	6037898	***trm EAPI SalesInvoice Comments***
	6037899	***trm EAPI ReturnOrder Comments***
	6037900	***trm EAPI SalesDoc Comments***
The Codeunits	***6036860***	***trm Integration Handling***
	6036862	***trm Integration Sales Handling***
	6036863	***trm Integration Handling Impl.***
	6036864	***trm Integration Proc. SalesDoc***
	6036865	***trm Integration Post SalesDoc***

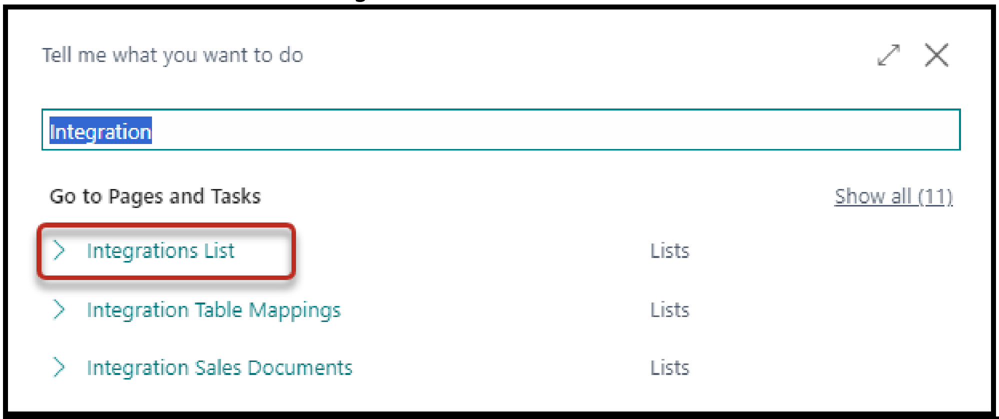
6036866 trm Integration Jnl Posting
6036870 trm Integration Data Container
6036872 trm Integration Agent
6037801 trm EAPI Handling

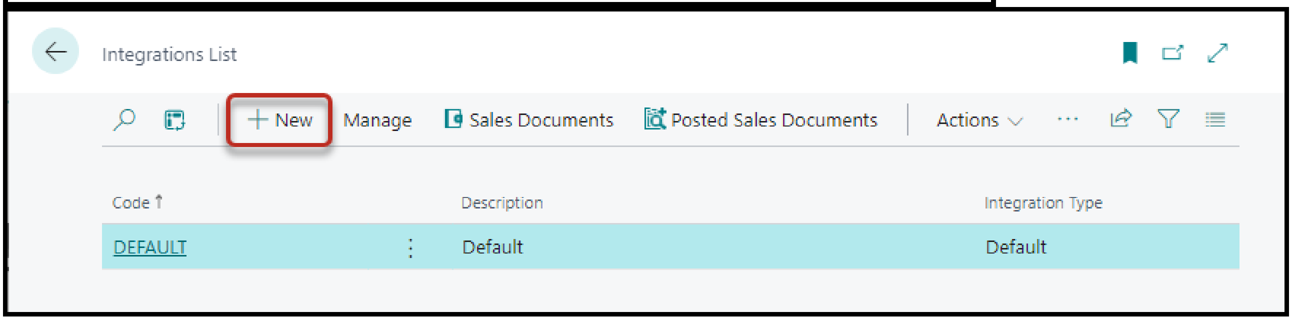
The Enum **6036840 trm Integration Type** Implementation**:** Codeunit **trm Integration Agent**
6036841 trm Integration Data Type
6036842 trm Integration Setup Line 1
6036843 trm Integration Setup Line 2
6036844 trm Integration Setup LineType
6036845 trm Integration Cust. Handling
6036846 trm Integration Product Type
6036847 trm Integration Sales Doc Type

NOTE
In the **TRIMIT** App, you can also see objects, which start with *trm API,* but they are part of the Integration between TRIMIT ERP and TRIMIT Portals.

INTEGRATION SETUP

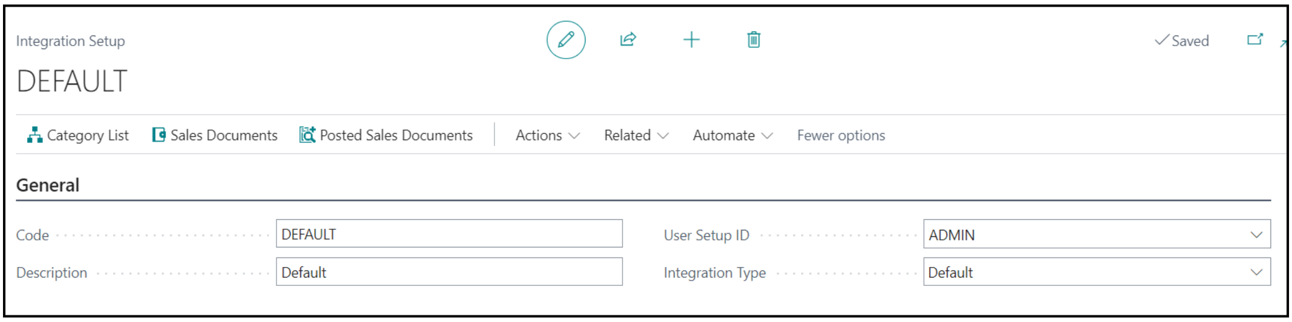
For the "export/import" of data based on the **TRIMIT External API**, you need to fill the **Integration Setup**. You can find it via ***Search*** *Integration*.





By clicking on the **Code,** you can open an existing Card or click ***+ New*** to create a new Setup.

FASTTAB GENERAL



CODE

The unique Code for this Integration Setup.

You are able to create several Codes, but you need to realize, that the User Setup ID must comply with the **Username** that is used in the Authorization of your Postman collection or system that integrates with TRIMIT.

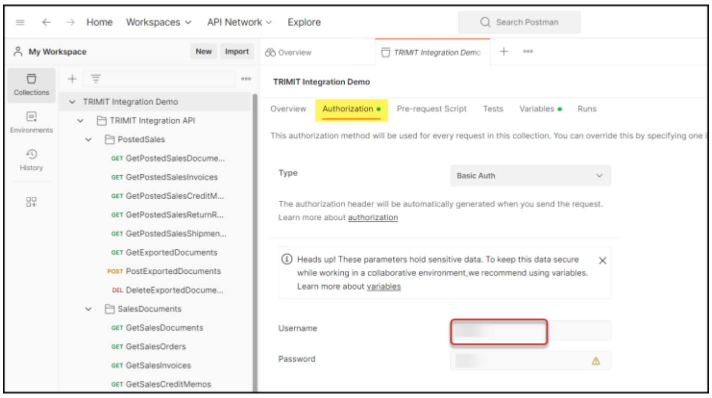
DESCRIPTION

Description for this specific **Code**.

USER SETUP ID

Here you need to enter a **User ID** from the **User Setups.**

This **User Setup ID** must be the same as the **Username** that is used in the Authorization of your **Postman collection**, which allows you to exchange data by using the TRIMIT External API.

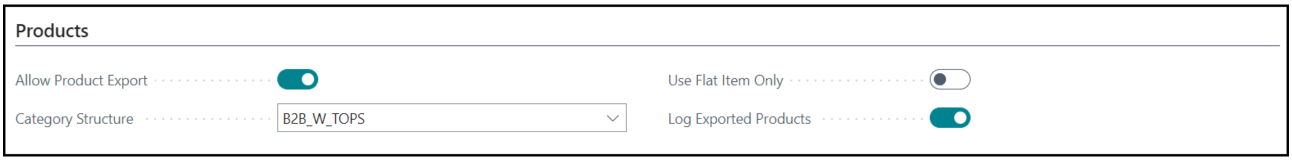


If you do not use Postman, you need to make sure that the **User Setup ID** is the same as the Username/Login of the system that allows you to export/import Data from/to TRIMIT.

INTEGRATION TYPE

At this point, we only have the option *Default*.
The **Integration Type** can be extended (it is an extendable Enum) by implementing the interface "trm Integration Agent".
Our *Default* implementation can be found in the Codeunit ***6036872 trm Integration Agent***.

FASTTAB PRODUCTS



ALLOW PRODUCT EXPORT

Is it allowed to export Product Data based on the Masters/Items that are in the **Category Structure**.
To be able to filter the export of Item/Master data, we created this special **Product Export**, which will be based on the **Category Structure** and the **Use Flat Item Only**.

> **NOTE**
> It will always be possible to export all the Items and Masters via the TRIMIT API services, which is an extension of the BC standard APIs with TRIMIT fields.
> And of course, you will be able to set filters in the app you use for showing the Data (like i.e., Postman).

CATEGORY STRUCTURE

You can use the **Category Structure** (Portal Categories in TRIMIT ERP) to filter the Masters/Items for which you want to export the Data if **Allow Product Export** is set to *Yes.*
For more details, see ***White Paper – TRIMIT Category Management*.**

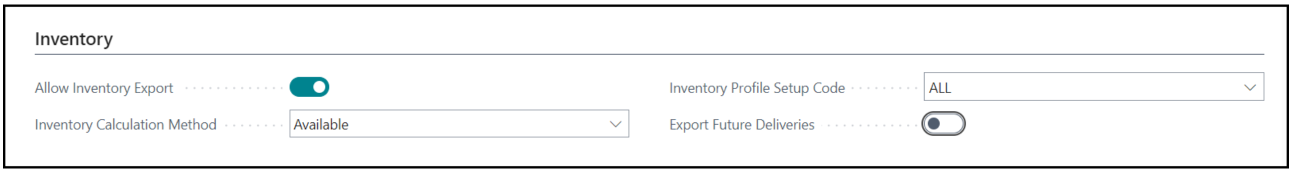
USER FLAT ITEM ONLY

You can determine if you only want to export the Items and not the Masters based on the **Allow Product Export** and **Category Structure.**

LOG EXPORTED PRODUCTS

You can determine if you want to log all the exported Products (based on the **Category Structure** and **Use Flat Item Only**).
In that case, after exporting the Products, you will be able to see the exported Products via ***Related, Products, Exported Products***.

FASTTAB INVENTORY



ALLOW INVENTORY EXPORT

You can determine if you should be able to export Inventory/Available Quantities.

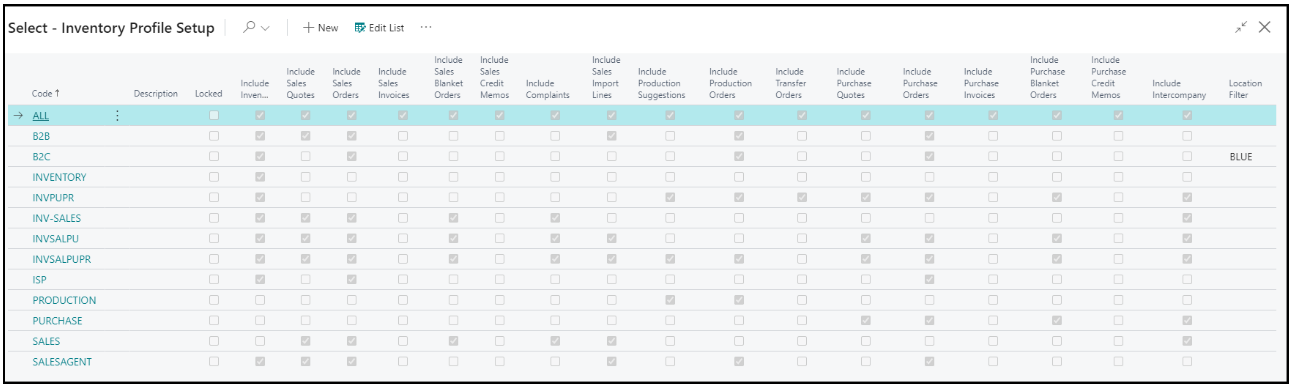
INVENTORY CALCULATION METHOD

If **Allow Inventory Export** is set to *Yes,* you are also able to determine which Quantities should be exported.
You can choose between the following three options:

None	No Quantities will be exported.
Inventory	The Inventory Quantities will be exported at the moment of the export.
Available	The Available Quantities will be exported based on the **Inventory Profile Setup Code**.

INVENTORY PROFILE SETUP CODE

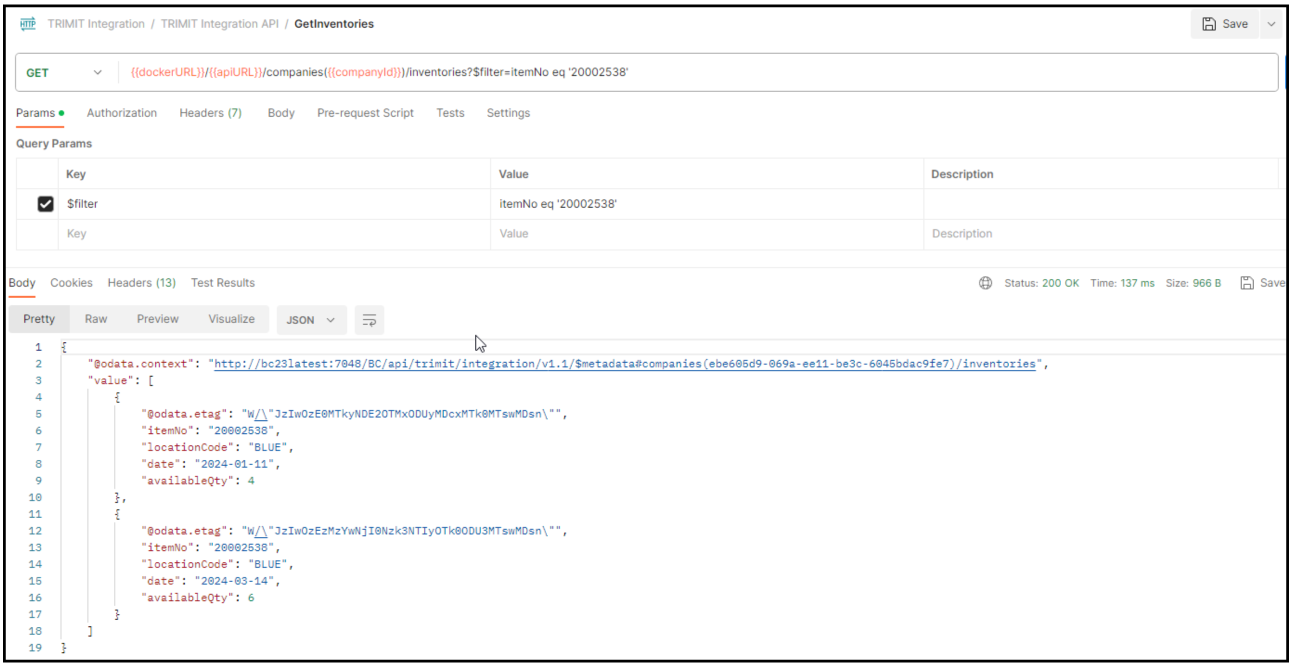
If **Inventory Calculation Method** is set to *Available,* you need to determine which Quantities should be included in the calculation of the Available Quantities based on this **Inventory Profile Setup Code**.
The **Inventory Profile Setup Code** determines from which documents the Quantities should be included in the Available Quantities.


Select - Inventory Profile Setup

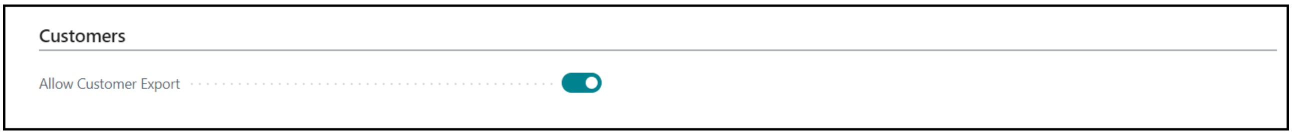
Code ↑	Description	Locked	Include Inven...	Include Sales Quotes	Include Sales Orders	Include Sales Invoices	Include Sales Blanket Orders	Include Sales Credit Memos	Include Complaints	Include Sales Import Lines	Include Production Suggestions	Include Production Orders	Include Transfer Orders	Include Purchase Quotes	Include Purchase Orders	Include Purchase Invoices	Include Purchase Blanket Orders	Include Purchase Credit Memos	Include Intercompany	Location Filter
ALL		☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	
B2B		☐	☑	☑	☑	☐	☐	☐	☐	☑	☐	☑	☐	☐	☑	☐	☐	☐	☐	
B2C		☐	☑	☐	☑	☐	☐	☐	☐	☐	☐	☑	☐	☐	☑	☐	☐	☐	☐	BLUE
INVENTORY		☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
INVPUPR		☐	☑	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑	☑	☑	☐	☑	☐	☑	
INV-SALES		☐	☑	☑	☑	☐	☑	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	
INVSALPU		☐	☑	☑	☑	☐	☑	☐	☑	☑	☐	☐	☐	☑	☑	☐	☑	☐	☑	
INVSALPUPR		☐	☑	☑	☑	☐	☑	☐	☑	☑	☑	☑	☐	☑	☑	☐	☑	☐	☑	
ISP		☐	☑	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	
PRODUCTION		☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☐	☐	☐	☐	
PURCHASE		☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☐	☑	☐	☑	
SALES		☐	☐	☑	☑	☐	☑	☐	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	
SALESAGENT		☐	☑	☑	☑	☐	☐	☐	☐	☑	☐	☑	☐	☐	☑	☐	☐	☐	☐	

EXPORT FUTURE DELIVERIES

Available Quantities will be calculated per Current Date, but if you also want to include Sales Lines with a Shipment Date > Current Date, you need to have a checkmark in **Export Future Deliveries**.
In Postmen, it might then look as follows:



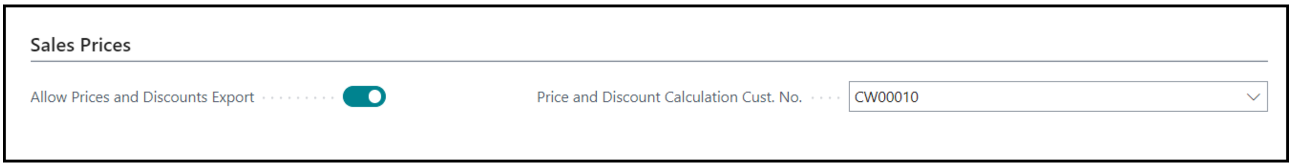
FASTTAB CUSTOMERS



ALLOW CUSTOMER EXPORT

You can determine if it is allowed to export Customer data for the TRIMIT Service Function.
(It is not applicable for the Customer API page in Postman.)

FASTTAB SALES PRICES

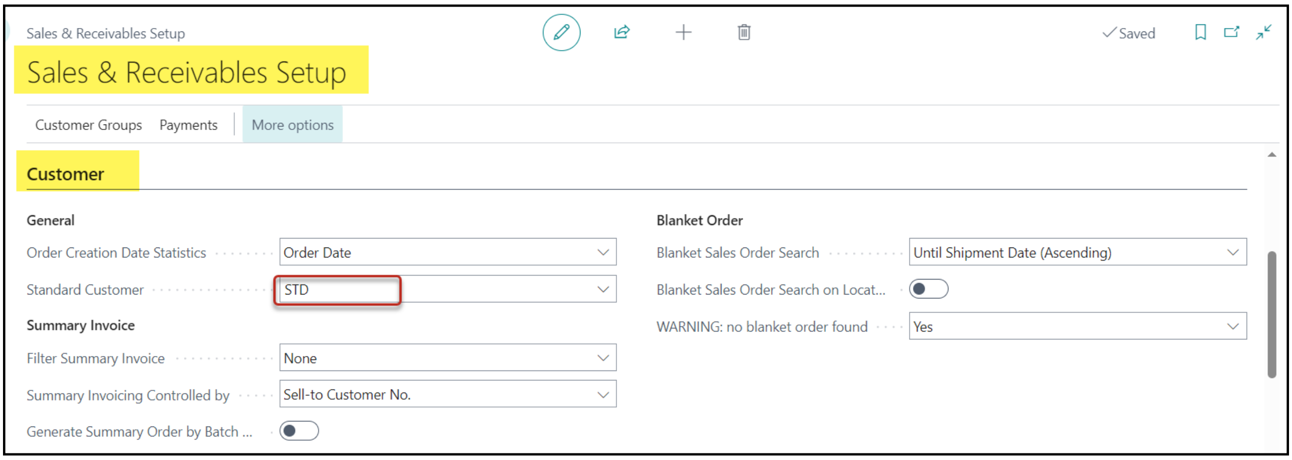


ALLOW PRICES AND DISCOUNTS EXPORT

You can determine if it is allowed to export Sales Prices.

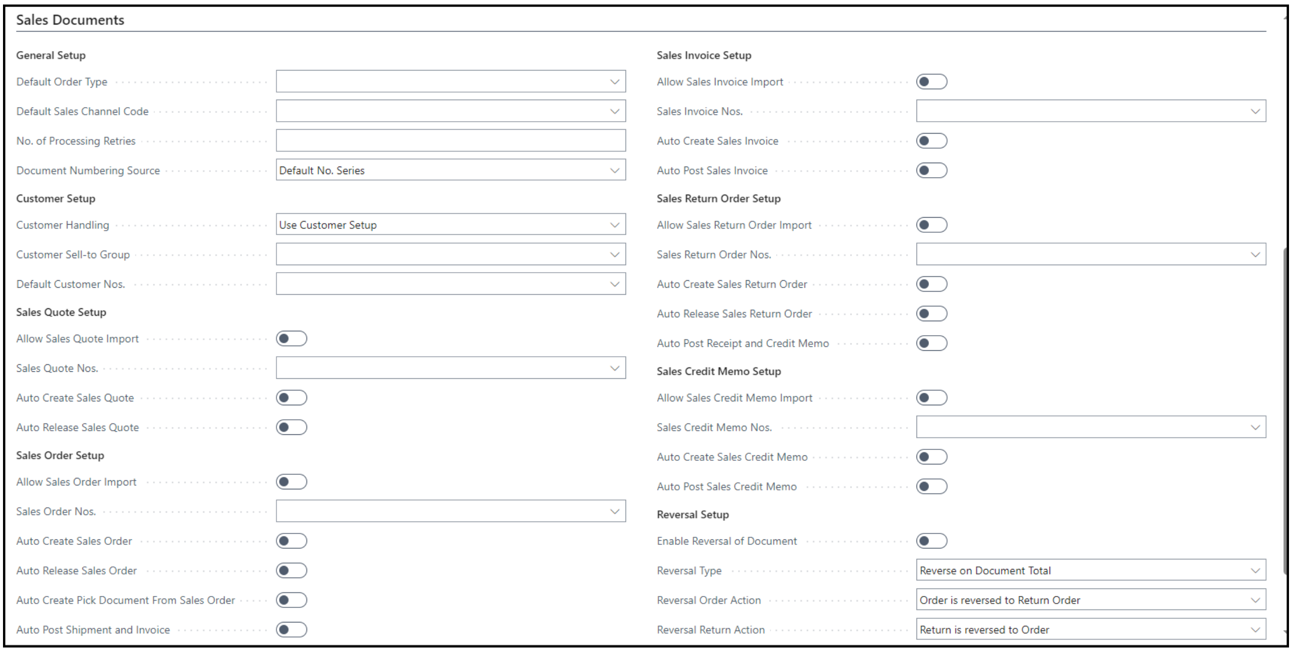
PRICE AND DISCOUNT CALCULATION CUST. NO.

If **Allow Prices and Discounts Export** is set to *Yes,* you also can determine for which Customer the Sales Price should be determined (based on the **Customer Price Group** of the Customer) that needs to be exported.
If no **Customer** has been filled, it will look at the Price in the Customer Price Group of the **Standard Customer** in the **Sales & Receivables Setup**.



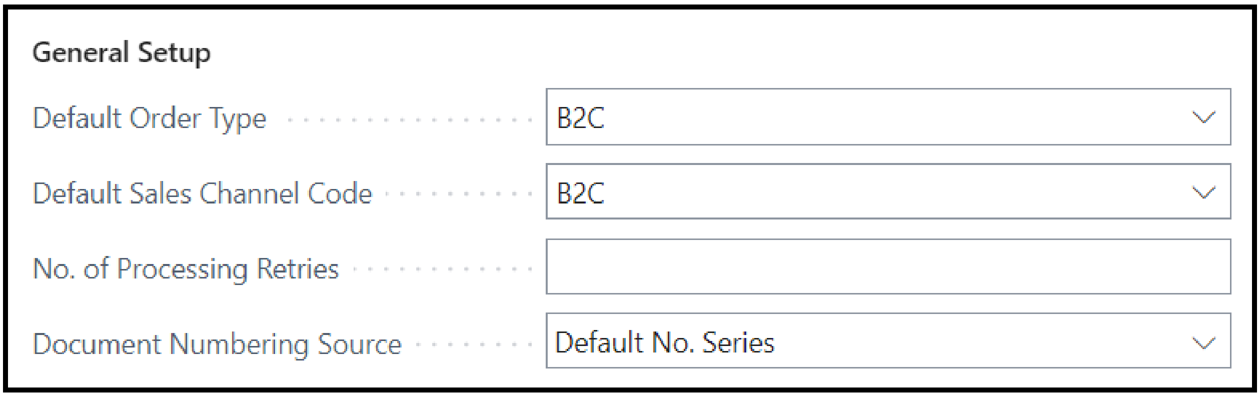
Unless you have filled the [Currency Setup](). So, you can leave the **Price and Discount Calculation Cust. No.** blank, and then the Sales Price will be determined based on the [Currency Setup]().

FASTTAB SALES DOCUMENTS



The FastTab ***Sales Documents*** is to determine how Integration Sales Documents should be handled.

GENERAL SETUP



DEFAULT ORDER TYPE

If you import Sales Documents, you can determine a specific **Order Type** for them in TRIMIT.

DEFAULT SALES CHANNEL CODE

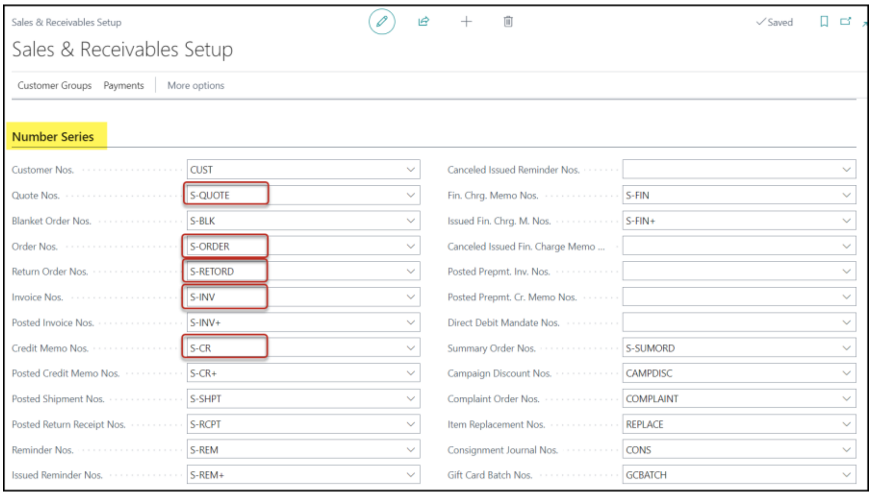
If you import Sales Documents, you can determine a specific **Sales Channel** for them in TRIMIT.

NO. OF PROCESSING RETRIES

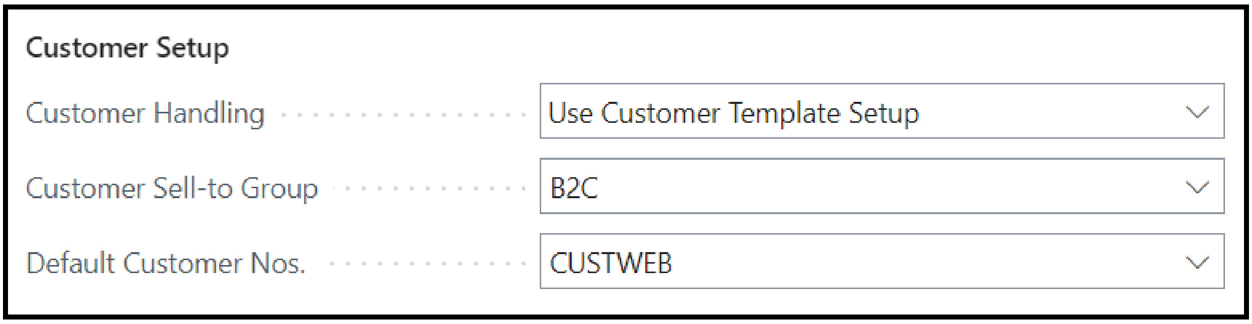
You could enter the number the import should be retried if anything disturbed the exchange of data.

DOCUMENT NUMBERING SOURCE

If you import Sales Documents, how should the **Sales Document Number** in TRIMIT be determined?
You can choose between the following three options:

<table>
<tr><td>Default No. Series</td><td>You can enter the No. Series in the Integration Setup for the different Sales Document Types.</td></tr>
<tr><td>Sales & Receivables No. Series</td><td>The No. Series filled in the Sales & Receivables Setup will be taken to determine the Number for the Sales Document in TRIMIT based on the Document Type.

</td></tr>
<tr><td>Integration Document No.</td><td>The Document No. in the Integration Sales Document determines the Number for the Sales Document in TRIMIT. This might lead to problems if this Document No. is inside the regular No. Series that have been setup in the Sales & Receivables Setup.</td></tr>
</table>

CUSTOMER SETUP



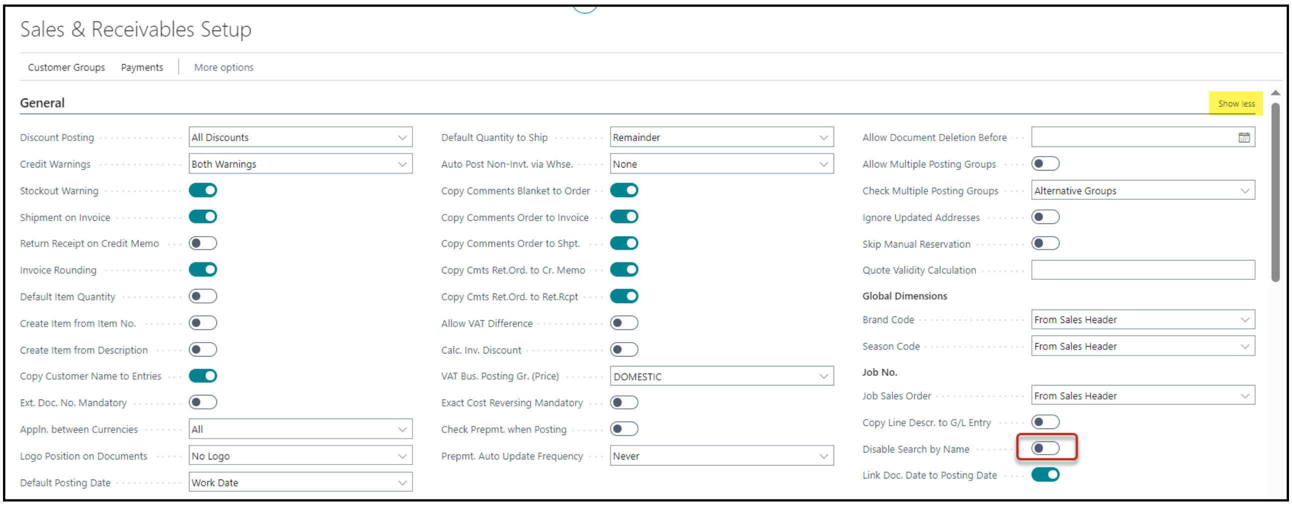
CUSTOMER HANDLING

This field determines how the Customer should be determined for an Integration Sales Document.
You can choose between the following three options:

Use Customer Setup	The **Sell-to Customer** in a regular Sales Document is determined by the **Customer Setup**. Based on the Customer Setup, the **Sell-to Customer** for the regular Sales Document in TRIMIT will be replaced, regardless the Customer No. that might be in the Integration Sales Document itself.
Use Customer Template Setup	The **Sell-to Customer** in a regular Sales Document is determined by the **Customer Template Setup**. Based on the Customer Template Setup a new Customer Number will be created using the appropriate Template and the **Default Customer Nos.** and this will replace the Customer No. that might be in the Integration Sales Document itself.
Apply Customer No. on Documents	The **Sell-to Customer** in a Sales Document is determined by the Customer No. in the Integration Sales Document.

NOTE
The third option might lead to errors if the **Customer No.** in the Integration Sales Document is not yet known in TRIMIT.

If the system should create new **Customer Nos.**, you must set the parameter **Disable Search by Name** in the Sales & Receivables Setup to *No*.



CUSTOMER SELL-TO GROUP

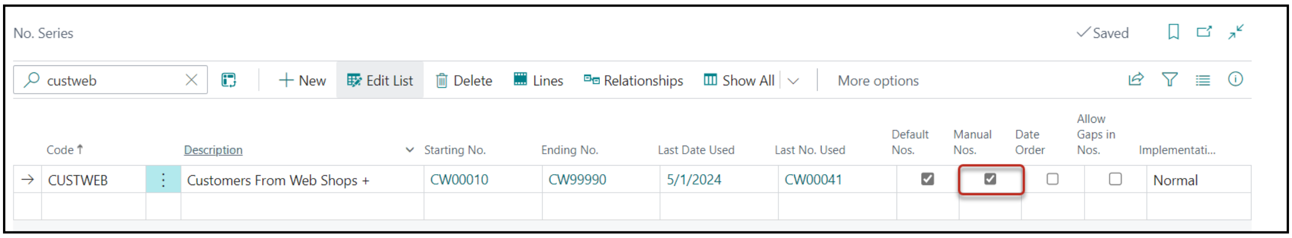
You can determine a default **Sell-to Group** for a new created Customer based on an Integration Sales Document.

DEFAULT CUSTOMER NOS.

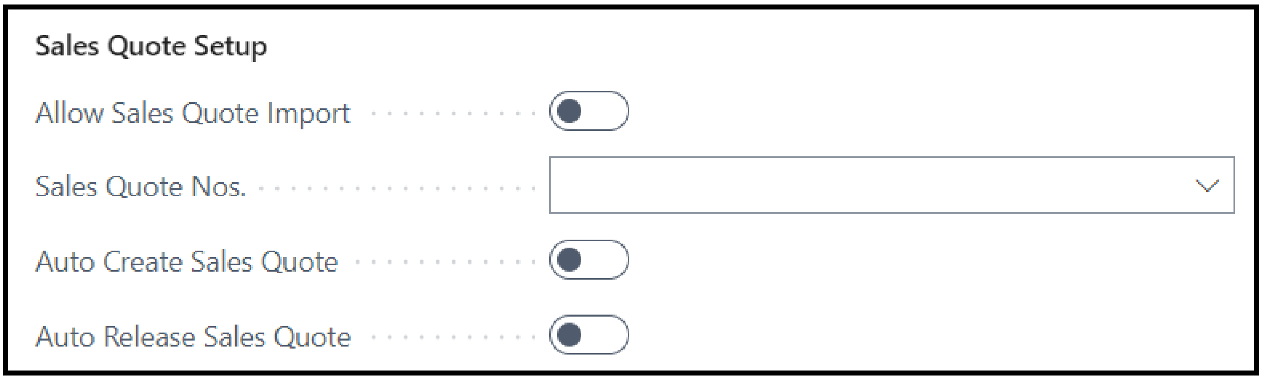
If **Customer Handling** is set to *Use Customer Template Setup* and a new Customer No. needs to be created, the system will determine that No. based on this **Default Customer Nos.** in which you can select a special No. Series. If this field is left blank, the regular **Customer Nos.** No. Series in the Sales & Receivables Setup will be taken.

NOTE

This **No. Series** need to have a checkmark in **Manual Nos**.



SALES QUOTE SETUP



ALLOW SALES QUOTE IMPORT

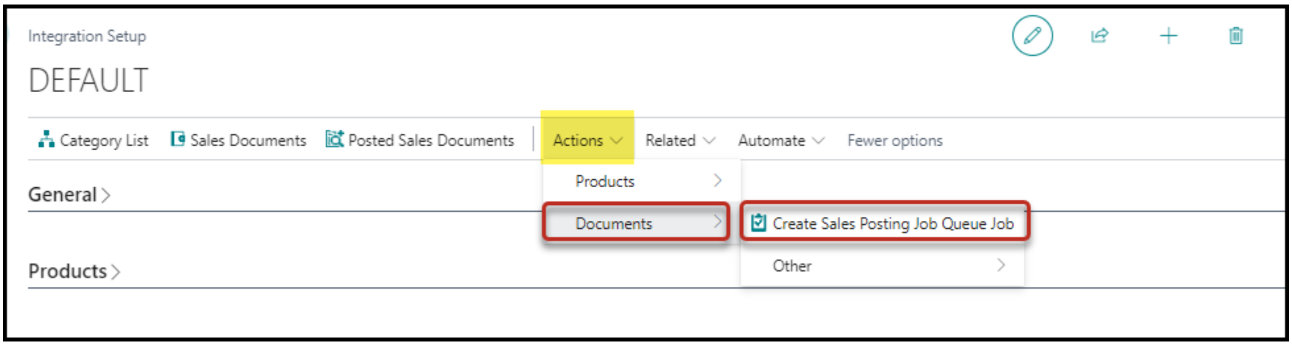
You can determine if it is allowed to import **Sales Quotes.**

SALES QUOTE NOS.

If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Quotes. Based on this **No. Series** the number for the regular **Sales Quote** will be determined. The Document Number of the Integration Sales Quote will then be placed in the **External Document Number** of the regular Sales Quote.

AUTO CREATE SALES QUOTE

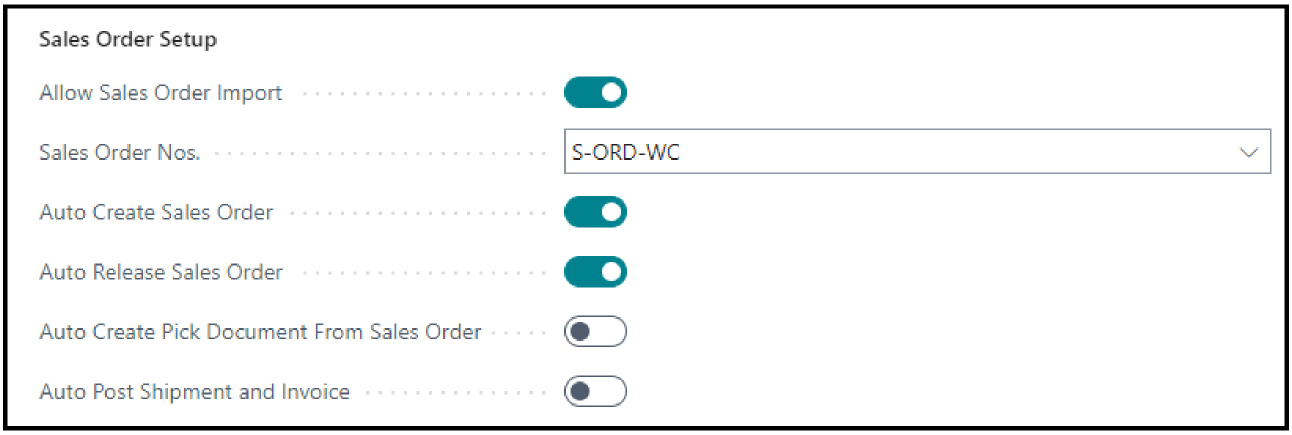
Based on the Integration Sales Quote should a regular Sales Quote be created based on a **Job Queue** or should the Integration Sales Quote remain in the **Integration Sales Documents**, from where it can be 'posted' manually to become a regular Sales Quote.
You can create the necessary **Job Queue** via ***Actions, Documents, Create Sales Posting Job Queue Job***.



AUTO RELEASE SALES QUOTE

If **Auto Create Sales Quote** is set to *Yes,* and a regular **Sales Quote** is created, you can determine if it also should be *Released* right away.

SALES ORDER SETUP



ALLOW SALES ORDER IMPORT

You can determine if it is allowed to import **Sales Orders**.

SALES ORDER NOS.

If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Orders. Based on this **No. Series** the number for the regular **Sales Order** will be determined.
The Document Number of the Integration Sales Order will then be placed in the **External Document Number** of the regular Sales Order.

AUTO CREATE SALES ORDER

Based on the Integration Sales Order should a regular Sales Order be created based on a **Job Queue** or should the Integration Sales Order remain in the **Integration Sales Documents**, from where it can be 'posted' manually to become a regular Sales Order.
You can create the necessary **Job Queue** via ***Actions, Documents, Create Sales Posting Job Queue Job*.**

AUTO RELEASE SALES ORDER

If **Auto Create Sales Order** is set to *Yes,* and a regular **Sales Order** is created, you can determine if it also should be *Released* right away.

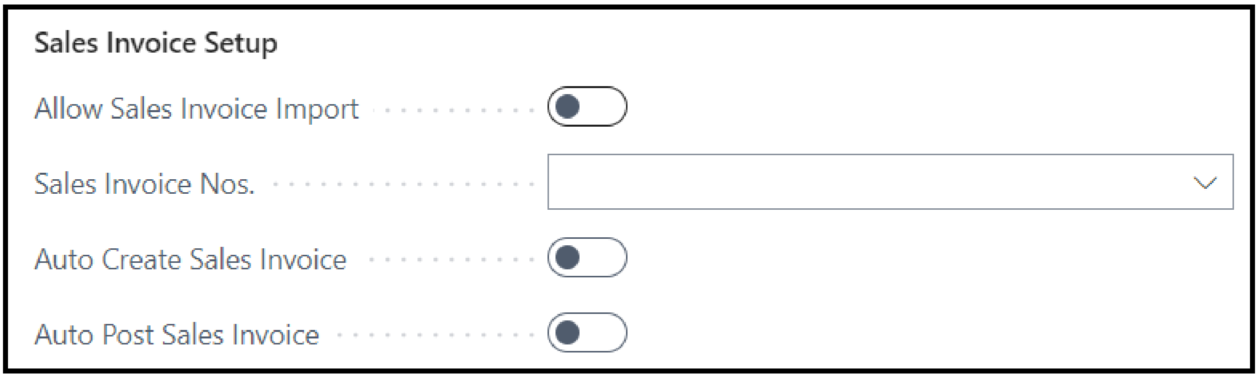
AUTO CREATE PICK DOCUMENT FROM SALES ORDER

If **Auto Create Sales Order** is set to *Yes,* and **Auto Release Sales Order** is set to *Yes,* and therefore, a regular **Sales Order** is created and released, you can determine if also a **TRIMIT Pick Document** should be created for that regular Sales Order immediately.

AUTO POST SHIPMENT AND INVOICE

If **Auto Create Sales Order** is set to *Yes,* and a regular **Sales Order** is created, you can also determine if that regular **Sales Order** should be posted as a *Shipment* and *Invoice* immediately.

SALES INVOICE SETUP



ALLOW SALES INVOICE IMPORT

You can determine if it is allowed to import **Sales Invoices**.

SALES INVOICE NOS.

If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Invoices. Based on this **No. Series** the number for the regular **Sales Invoice** will be determined.
The Document Number of the Integration Sales Invoice will then be placed in the **External Document Number** of the regular Sales Invoice.
Unless the Use Gen. Journal for Sales Invoice Posting is set to *Yes.*
In that case no regular Sales Invoice will be created, but a General Journal Line of **Document Type** *Invoice* and the **Document No.** is determined by this **Sales Invoice Nos.**, but you will lose the original Document Number of the Integration Sales Invoice.

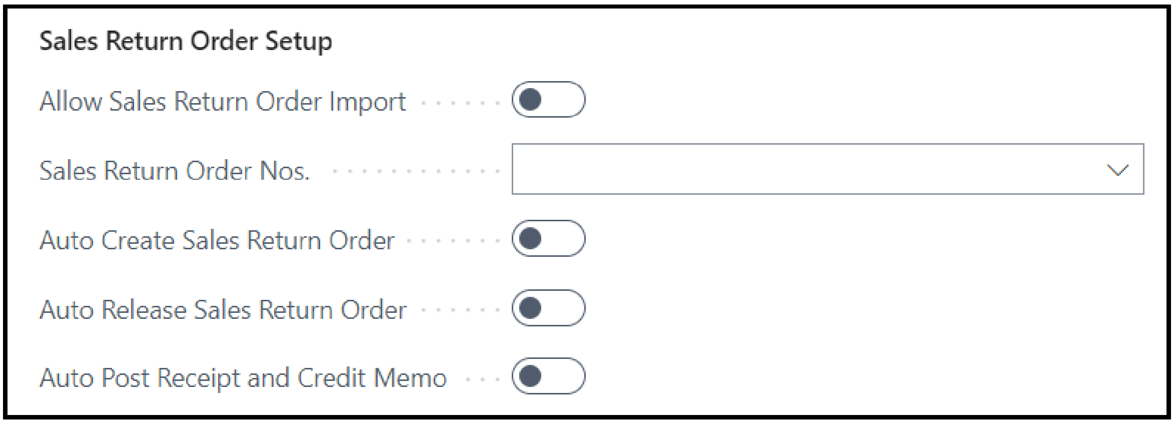
AUTO CREATE SALES INVOICE

Based on the Integration Sales Invoice should a regular Sales Invoice be created based on a **Job Queue** or should the Integration Sales Invoice remain in the **Integration Sales Documents**, from where it can be 'posted' manually to become a regular Sales Invoice.
You can create the necessary **Job Queue** via ***Actions, Documents, Create Sales Posting Job Queue Job*.**

AUTO POST SALES INVOICE

If **Auto Create Sales Invoice** is set to *Yes,* and a regular **Sales Invoice** is created, you can also determine if that regular **Sales Invoice** should be posted immediately.

SALES RETURN ORDER SETUP



ALLOW SALES RETURN ORDER IMPORT

You can determine if it is allowed to import **Sales Return Orders**.

SALES RETURN ORDER NOS.

If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Return Orders. Based on this **No. Series** the number for the regular **Sales Return Order** will be determined.
The Document Number of the Integration Sales Return Order will then be placed in the **External Document Number** of the regular Sales Return Order.

AUTO CREATE SALES RETURN ORDER

Based on the Integration Sales Return Order should a regular Sales Return Order be created based on a **Job Queue** or should the Integration Sales Return Order remain in the **Integration Sales Documents**, from where it can be 'posted' manually to become a regular Sales Return Order.
You can create the necessary **Job Queue** via ***Actions, Documents, Create Sales Posting Job Queue Job*.**

AUTO RELEASE SALES RETURN ORDER

If **Auto Create Sales Return Order** is set to *Yes,* and a regular **Sales Return Order** is created, you can determine if it also should be *Released* right away.

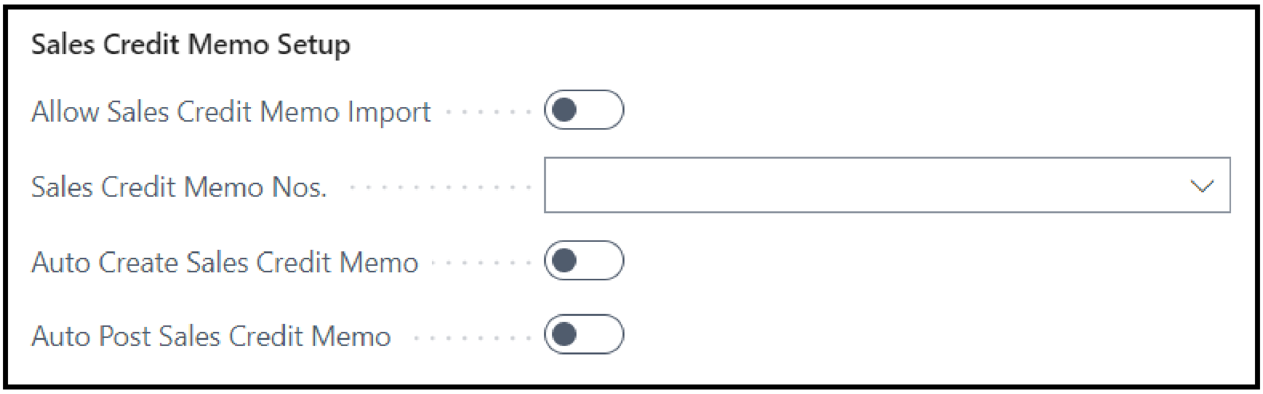
AUTO POST RECEIPT AND CREDIT MEMO

If **Auto Create Sales Return Order** is set to *Yes,* and a regular **Sales Return Order** is created, you can also determine if that regular **Sales Return Order** should be posted as a *Receipt* and *Credit Memo* immediately.

ALLOW SALES CREDIT MEMO IMPORT

You can determine if it is allowed to import **Sales Credit Memos**.

SALES CREDIT MEMO SETUP



SALES CREDIT MEMO NOS.

If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for Integration Sales Credit Memos. Based on this **No. Series** the number for the regular **Sales Credit Memo** will be determined.
The Document Number of the Integration Sales Credit Memo will then be placed in the **External Document Number** of the regular Sales Credit Memo.
Unless the Use Gen. Journal for Credit Memo Posting is set to *Yes.*

In that case no regular Sales Credit Memo is created, but a General Journal Line of **Document Type** *Credit Memo* and the **Document No.** is determined by this **Sales Credit Memo Nos.**, but you will lose the original Document Number of the Integration Sales Credit Memo.

AUTO CREATE SALES CREDIT MEMO

Based on the Integration Sales Credit Memo should a regular Sales Credit Memo be created based on a **Job Queue** or should the Integration Sales Credit Memo remain in the **Integration Sales Documents**, from where it can be 'posted' manually to become a regular Sales Credit Memo.
You can create the necessary **Job Queue** via ***Actions, Documents, Create Sales Posting Job Queue Job*.**

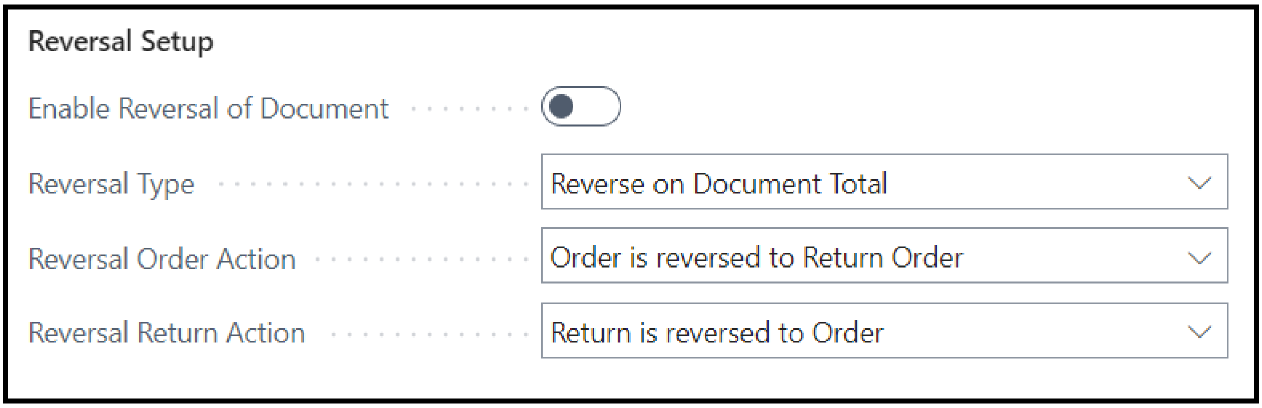
AUTO POST SALES CREDIT MEMO

If **Auto Create Sales Credit Memo** is set to *Yes,* and a regular **Sales Credit Memo** is created, you can also determine if that regular **Sales Credit Memo** should be posted immediately.

ENABLE REVERSAL OF DOCUMENT

You can determine if it should be possible to reverse an Integration Sales Document.
This could be applicable if the **Quantity** in the Integration Sales Document is negative.

REVERSAL SETUP



REVERSAL TYPE

At this point, the only option is *Reverse on Document Total*, so it is not possible only to reverse some of the Lines.

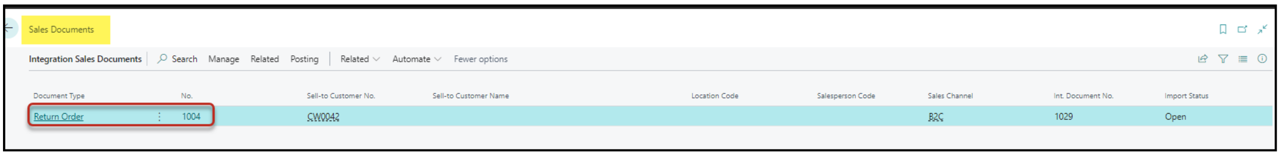
REVERSAL ORDER ACTION

If **Enable Reversal of Document** is set *Yes,* and the Integration Sales Document is a Sales Order and the Quantity is negative, you can determine how it should be reversed:
Order is reversed to Return Order or *Order is reversed to Credit Memo.*
This means, that although the Integration Sales Document has a Document Type *Order*, it will actually be created as a regular *Sales Return Order* or *Sales Credit Memo***.**

REVERSAL RETURN ACTION

If **Enable Reversal of Document** is set *Yes,* and the Integration Sales Document is a Sales Return Order, you can determine how it should be reversed:
Return is reversed to Order or *Return is reversed to Invoice.*

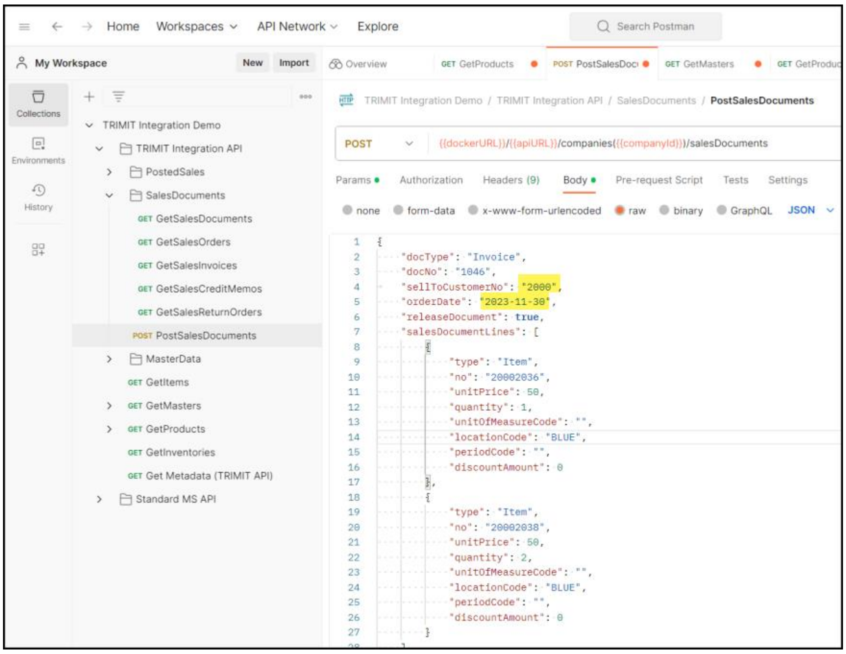
This means, that although the Integration Sales Document has a Document Type *Return Order*, it will actually be created as a regular *Sales Order* or *Sales Invoice*.
Keep in mind however that if it is reversed (when posting the **Integration Sales Document**) to an Order or Invoice, it will get the regular Document Number based on the **Document Numbering Source** and the **Sale Return Order Nos.**



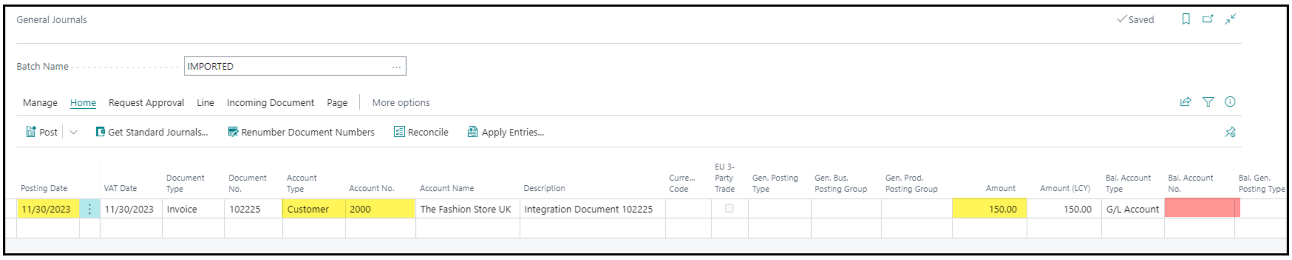
NOT IN PAGE, BUT CAN BE ADDED VIA PERSONALIZE:

USE GEN. JOURNAL FOR SALES INVOICE POSTING

Not shown in the page by default, but you can add it via ***Personalize.***
If you do not want to have a regular Sales Invoice created for an Integration Sales Invoice, you can set this field to *Yes.*
In that case, when 'posting' the Integration Sales Invoice, it will create a Journal Line in the **General Journal Template Name** and the **General Journal Batch Name.**
Keep in mind however, that only the **Customer** will be filled and an **Amount**, and not a **Balance Account** on the General Journal Line.
An Example:
If I would have the following data that should be imported:



I would have a total Amount of (1 x 50) + (2 x 50) = 150. This would result in a General Journal Line:



Therefore, you need to complete the Line, i.e., with an **Bal. Account No**. before posting.

As you can see, nothing will happen with the Items itself; so, do not use this function if Items are involved!!

USE GEN. JOURNAL FOR CREDIT MEMO POSTING

Not shown in the page by default, but you can add it via ***Personalize.***
If you do not want to have a regular Sales Credit Memo created for an Integration Sales Credit Memo, you can set this field to *Yes.*
In that case, when 'posting' the Integration Sales Credit Memo, it will create a Journal Line in the **General Journal Template Name** and the **General Journal Batch Name.**
Keep in mind however, that only the **Customer** will be filled and the **Amount**, and not a **Balance Account** on the General Journal Line.

GENERAL JOURNAL TEMPLATE NAME

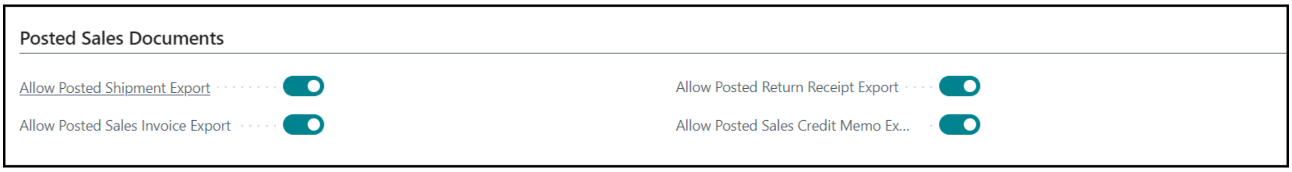
Not shown in the page by default, but you can add it via ***Personalize.***
In case you have set the **Use Gen. Journal for Sales Invoice Posting** and/or the **Use Gen. Journal for Credit Memo Posting** set to *Yes,* you also need to fill the **General Journal Template Name** and the **General Journal Batch Name.**

GENERAL JOURNAL BATCH NAME

Not shown in the page by default, but you can add it via ***Personalize.***

In case you have set the **Use Gen. Journal for Sales Invoice Posting** and/or the **Use Gen. Journal for Credit Memo Posting** set to *Yes,* you also need to fill the **General Journal Template Name** and the **General Journal Batch Name.**

FASTTAB POSTED SALES DOCUMENTS



ALLOW POSTED SHIPMENT EXPORT

You can determine if it is allowed to export **Posted Shipments**.

ALLOW POSTED SALES INVOICE EXPORT

You can determine if it is allowed to export **Posted Sales Invoices**.

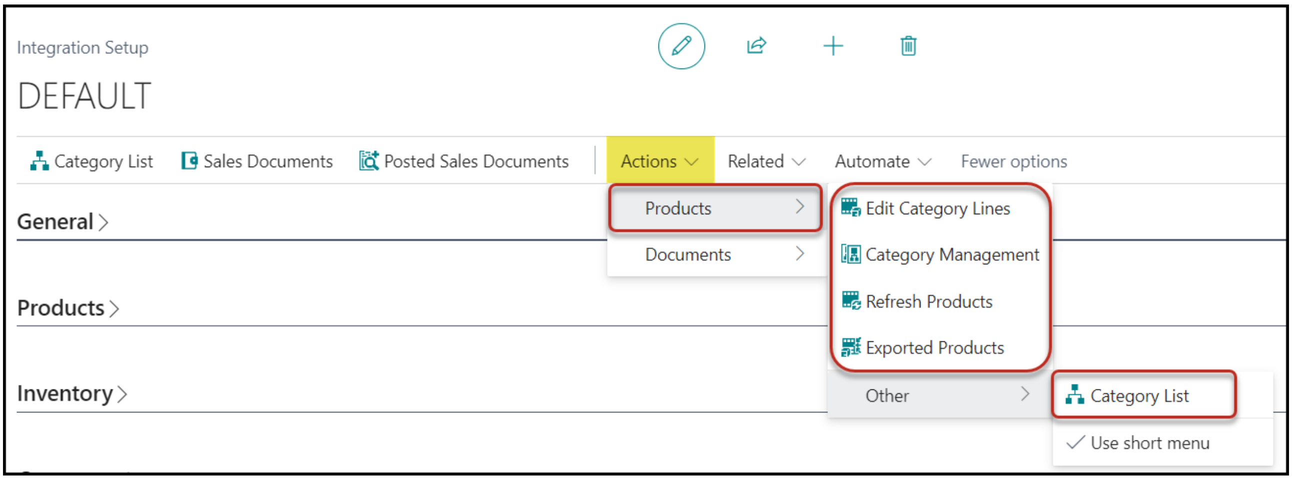
ALLOW POSTED RETURN RECEIPT EXPORT

You can determine if it is allowed to export **Posted Sales Return Receipts**.

ALLOW POSTED SALES CREDIT MEMO EXPORT

You can determine if it is allowed to export **Posted Sales Credit Memos**.

ACTIONS, PRODUCTS



EDIT CATEGORY LINES

The system will show the **Category Lines** of the Category Structure and you are able to add/delete/change the Lines; keep in mind however, that if you also use the same Category Structure for the TRIMIT Portals, this will also influence your TRIMIT Portals.

CATEGORY MANAGEMENT

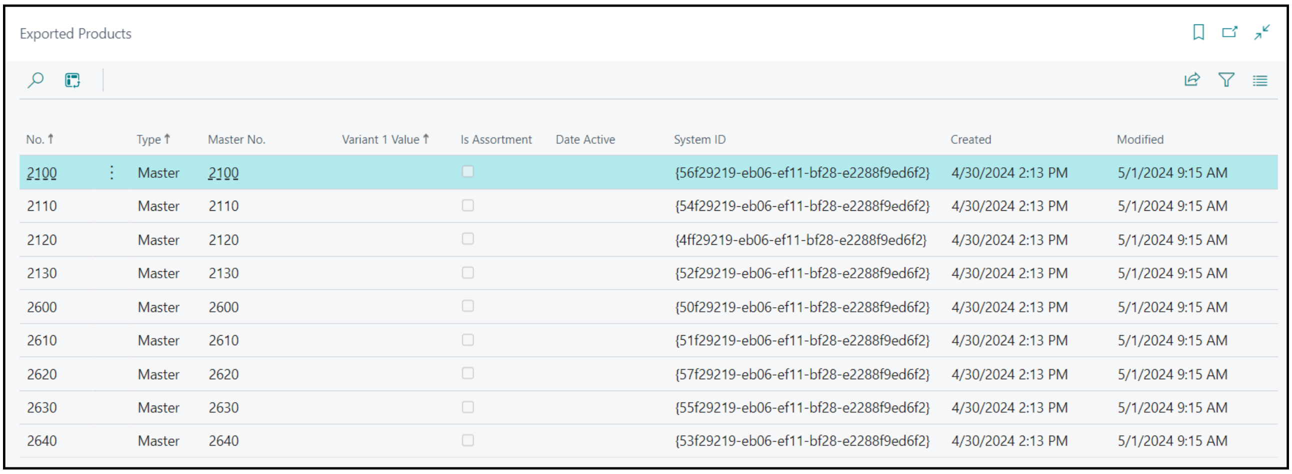
This will show you the Items and in which Category they might be.
It is also possible to attach Items to Categories.
For more details, see ***White Paper – TRIMIT Category Management***.

REFRESH PRODUCTS

This will refresh the Category Buffer. So in case you have changed the content of the **Category Structure** it is advised to use this Action, to make sure that the Category Buffer is up-to-date.

EXPORTED PRODUCTS

This will show a list of all the Products (Masters/Items) that have been exported if the field **Log Exported Products** is set to *Yes* and only when you used the **Get Products** functionality to export Products.
The exported Products will then be written in Table ***6036842 trm Integration Exp. Product***:

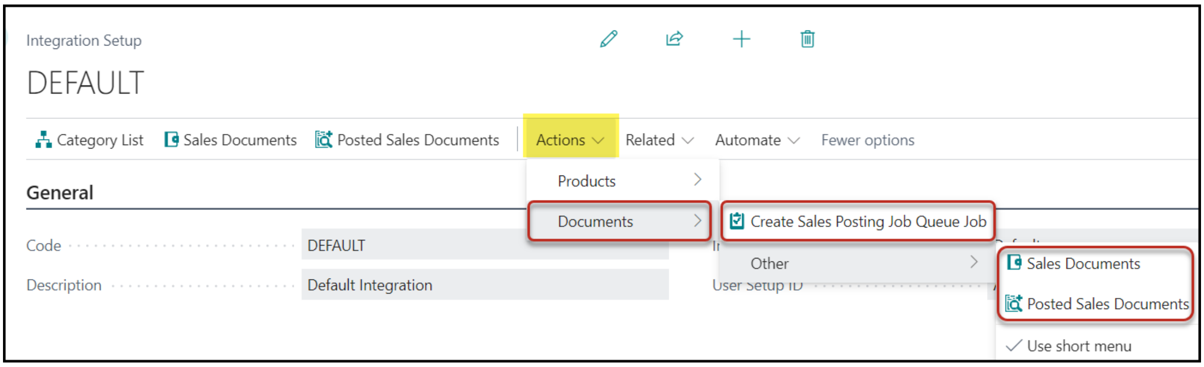

Exported Products

No.	Type	Master No.	Variant 1 Value	Is Assortment	Date Active	System ID	Created	Modified
2100	Master	2100		☐		{56f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2110	Master	2110		☐		{54f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2120	Master	2120		☐		{4ff29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2130	Master	2130		☐		{52f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2600	Master	2600		☐		{50f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2610	Master	2610		☐		{51f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2620	Master	2620		☐		{57f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2630	Master	2630		☐		{55f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM
2640	Master	2640		☐		{53f29219-eb06-ef11-bf28-e2288f9ed6f2}	4/30/2024 2:13 PM	5/1/2024 9:15 AM

CATEGORY LIST

This will show the **Category Tree View** of the content of the **Category Structure**.

ACTIONS, DOCUMENTS



CREATE SALES POSTING JOB QUEUE JOB

This Action will create a **Job Queue** for you for **Codeunit *6036865 trm Integration Post SalesDoc***, after which you can change the details.

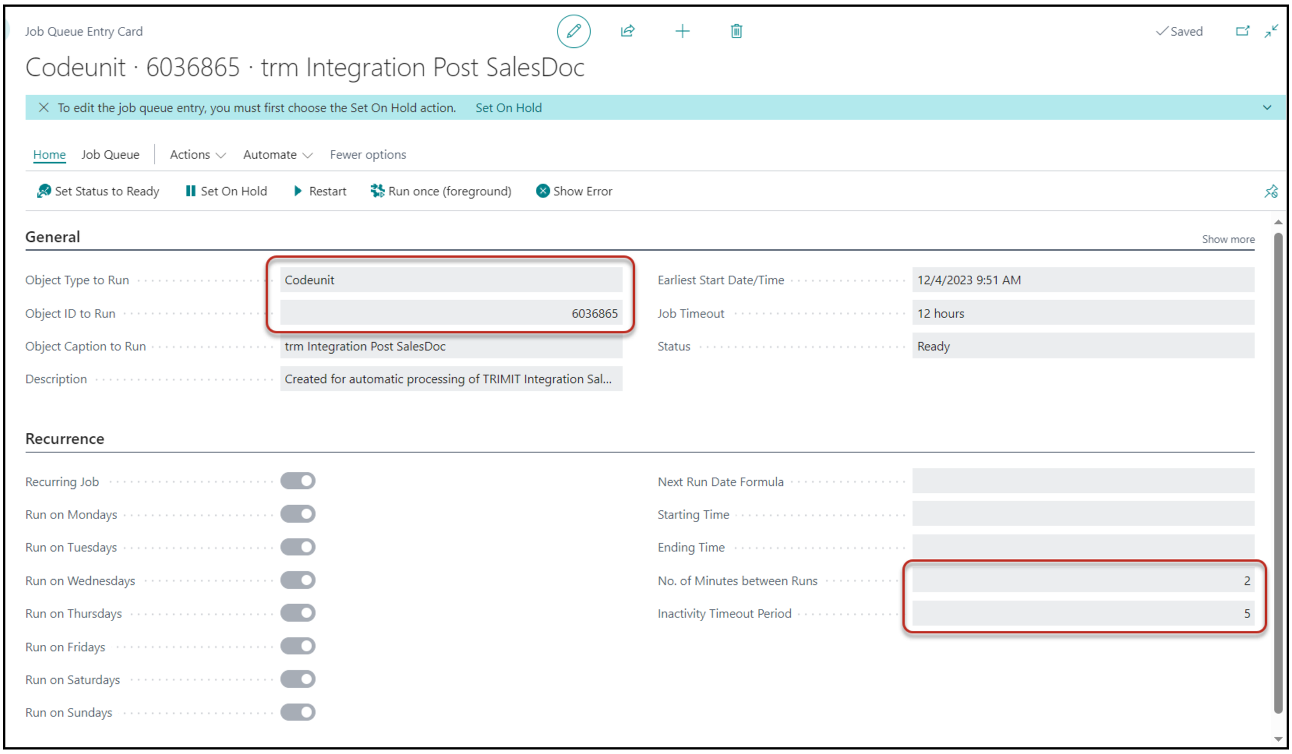
NOTE

This is NOT possible to create by an "Delegated Administrator". It can only be created by a "Normal Administrator".

This **Job Queue** will make sure that the Integration Sales Documents will be 'posted' after they have been imported, so you do not have to do this manually.

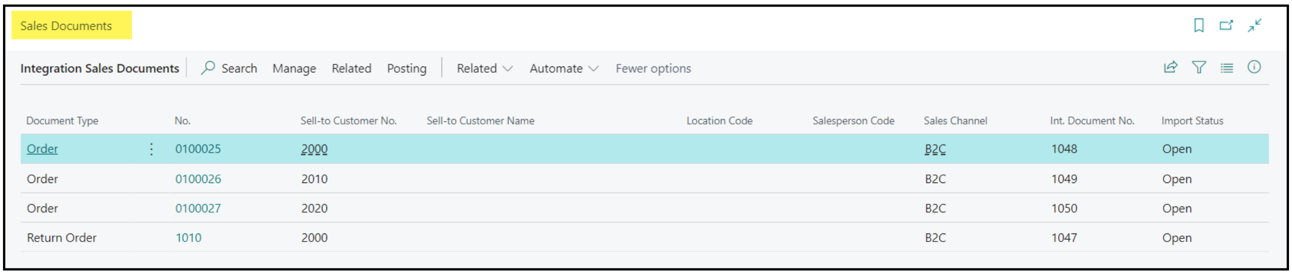
This **Job Queue** is necessary if you have set one or more of the following fields to *Yes* to actually create the regular Sales Document 'automatically'.

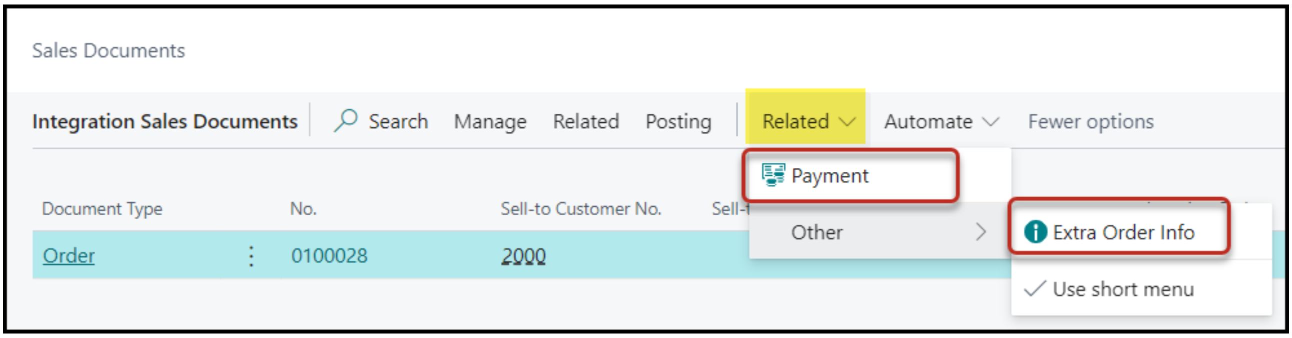
- Auto Create Sales Quote
- Auto Create Sales Order
- Auto Create Sales Return Order
- Auto Create Sales Invoice
- Auto Create Sales Credit Memo



SALES DOCUMENTS

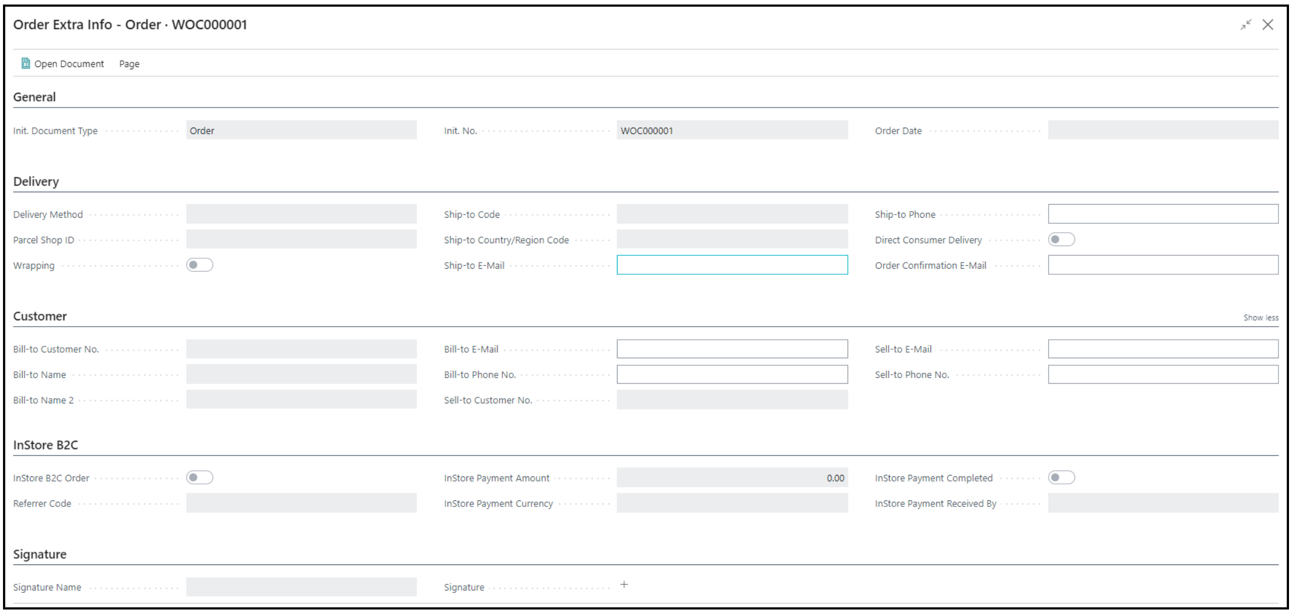
This will open the page with the Integration Sales Documents, which will show you only the data that was imported.



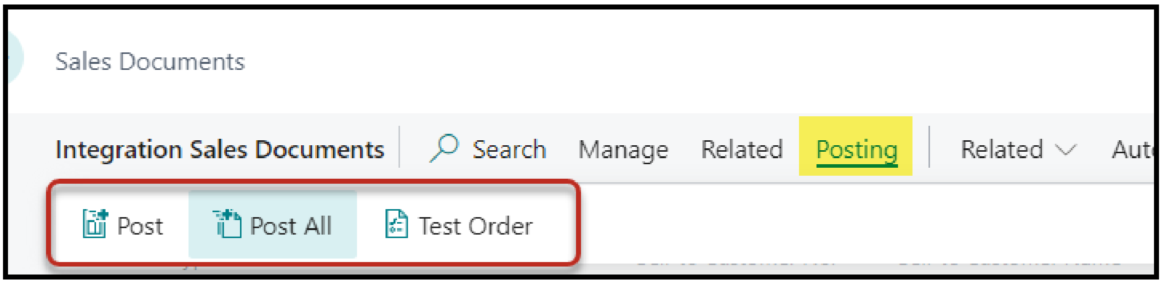


With ***Related, Payment,*** it will show you the Payment that was also imported.

With ***Related, Extra Order Info***, it will show you the Extra Order Info that was also imported.



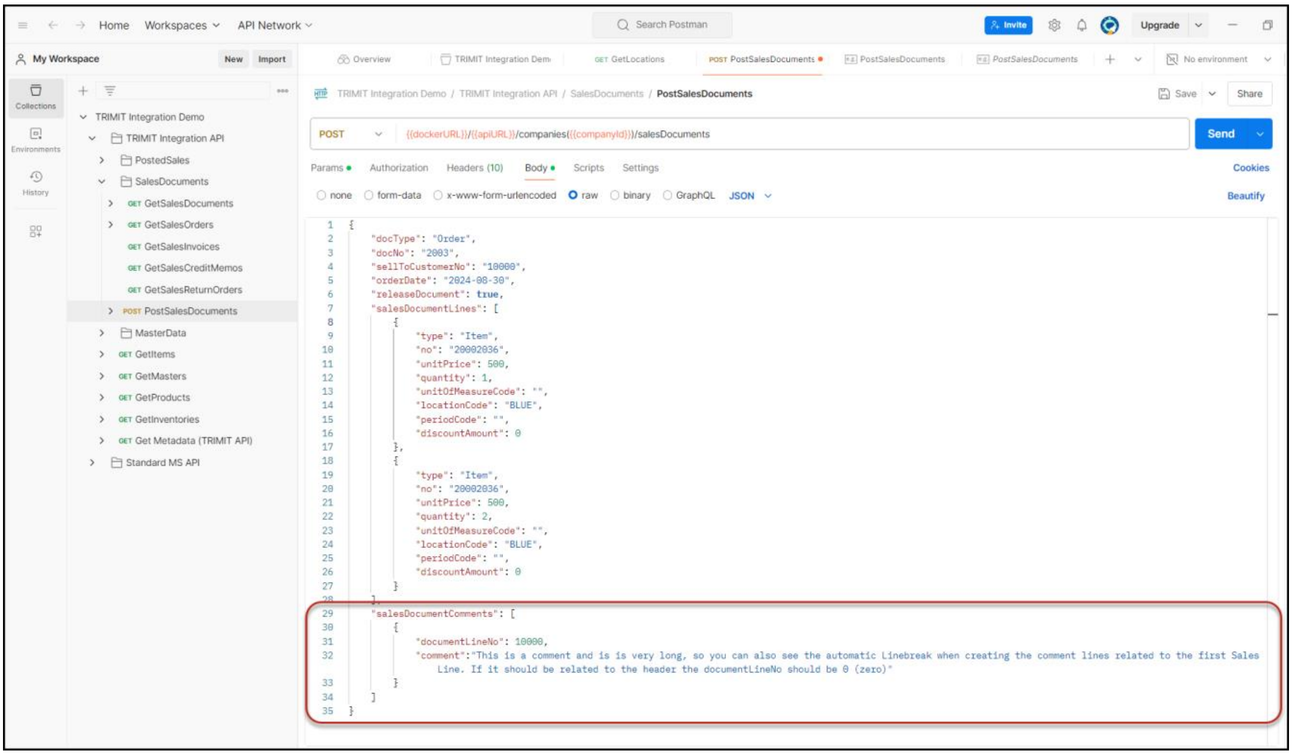
Both are based on the functionality we already have for our regular TRIMIT Portals.



With ***Posting, Post,*** you can create the regular Sales Document for the current Line you are standing on.
With ***Posting, Post All,*** you can create the regular Sales Documents for all the Lines at once.
With ***Posting, Test Order,*** you can test the data that was imported: Can the **Sales Document** be posted?

SALES DOCUMENT CARD

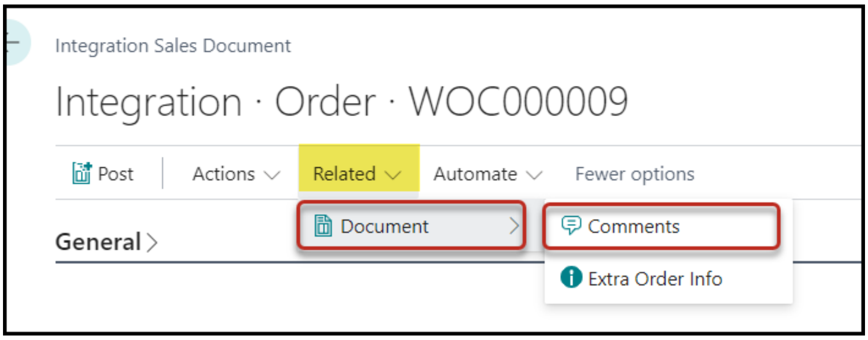
In the Sales Document Card, you can see the details of the Sales Document that was imported.
It is also possible to import Comment Lines on the Sales Document Header, and on the Sales Document Lines.



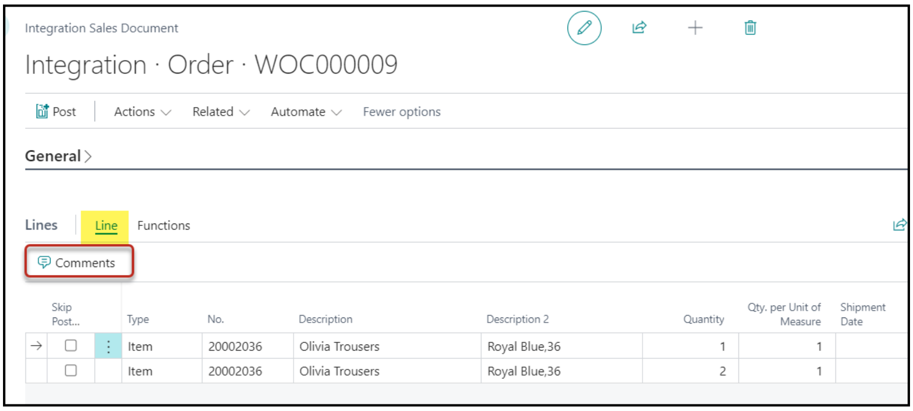
With **salesDocumentComments** you can add Comments to the Sales Document Header or Sales Document Line.

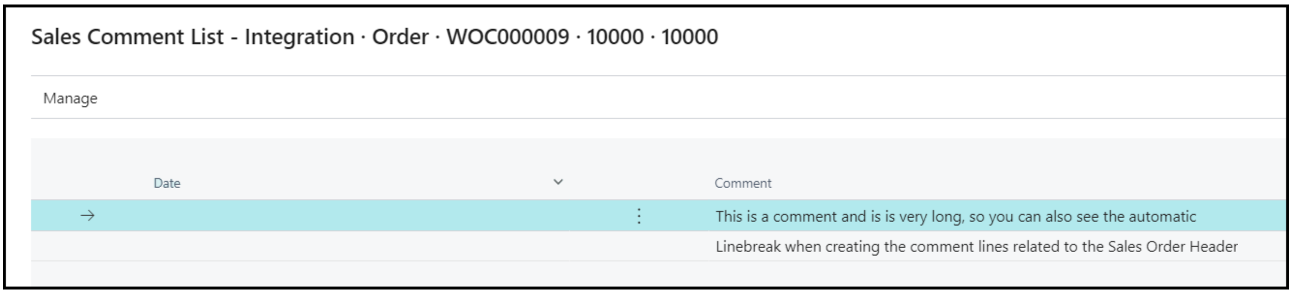
If **documentLineNo** is *0* it will be Comments for the Sales Document Header, if the **documentLineNo** <>*0* it refers to the corresponding Sales Document Line that you will be importing.

When imported, you can see those Comments in the Card of the **Sales Document** via:

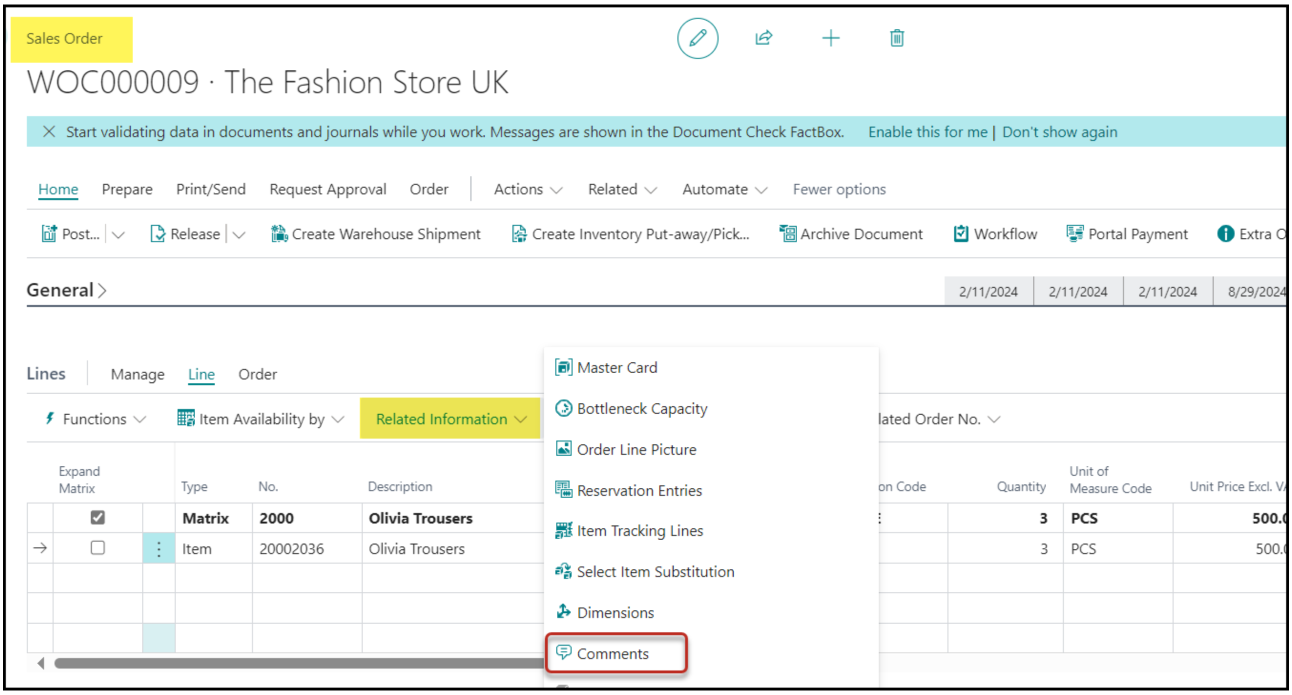


Or



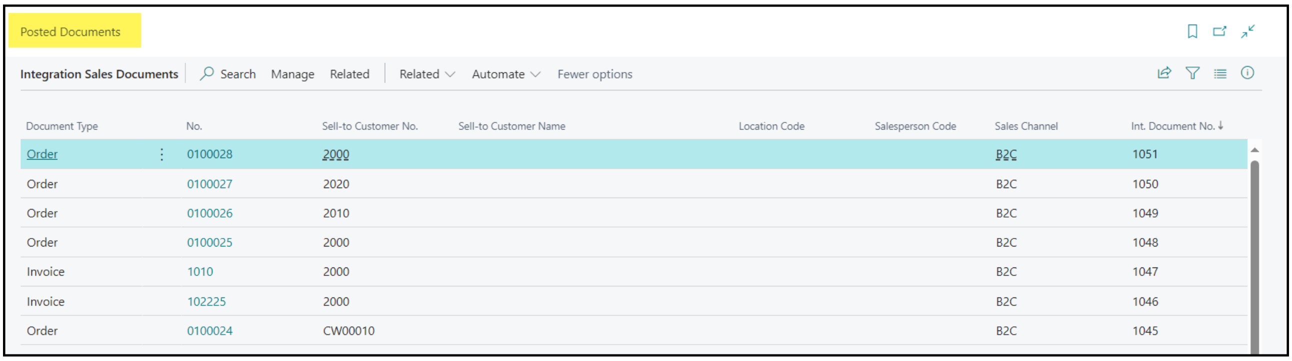


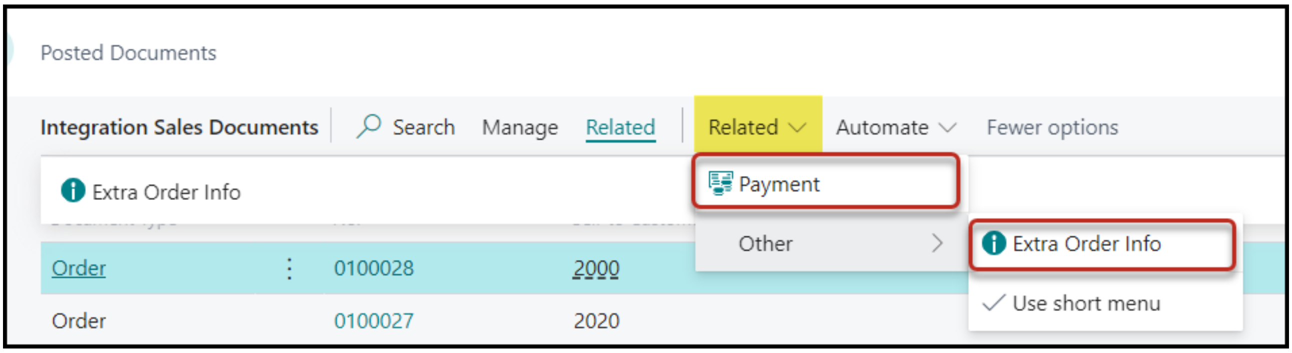
These Comment Lines will also be copied to the actual Sales Document after Posting.
For this example: in the Sales Order WOC000009 on the Item Line:



POSTED SALES DOCUMENTS

After an Integration Sales Document has been posted (manually or by the Job Queue), you will see it in the page **Posted Sales Documents**, but still only based on the data that has been imported, and the settings in the **Integration Setup** regarding the **No.** and the **Sell-to Customer No.**



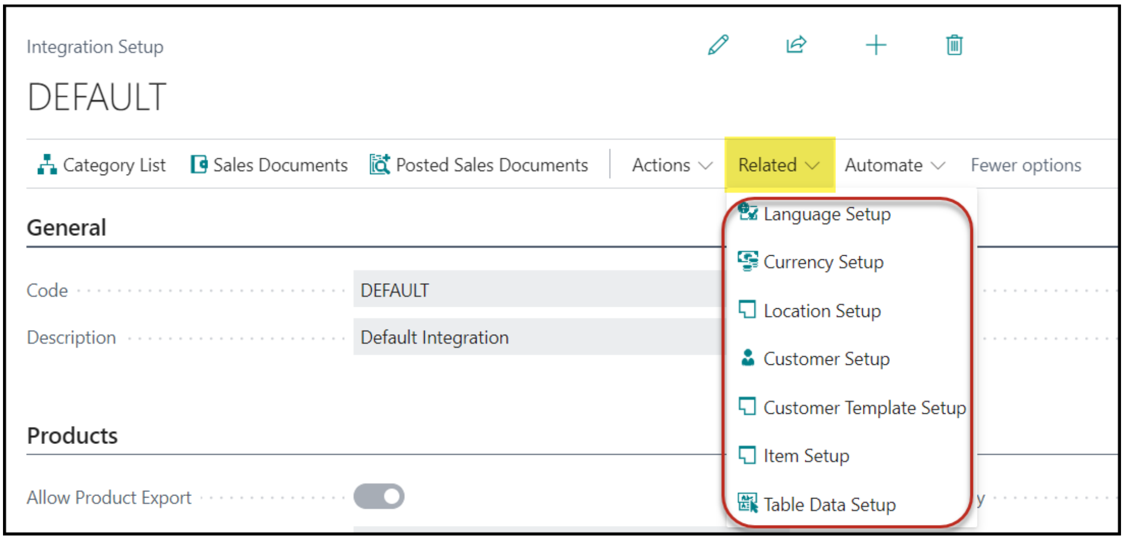


With ***Related, Payment,*** it will show you the Payment that was also imported and posted.
With ***Related, Extra Order Info***, it will show you the Extra Order Info that was also imported.
Both are based on the functionality we already have for our regular TRIMIT Portals.

POSTED SALES DOCUMENT CARD

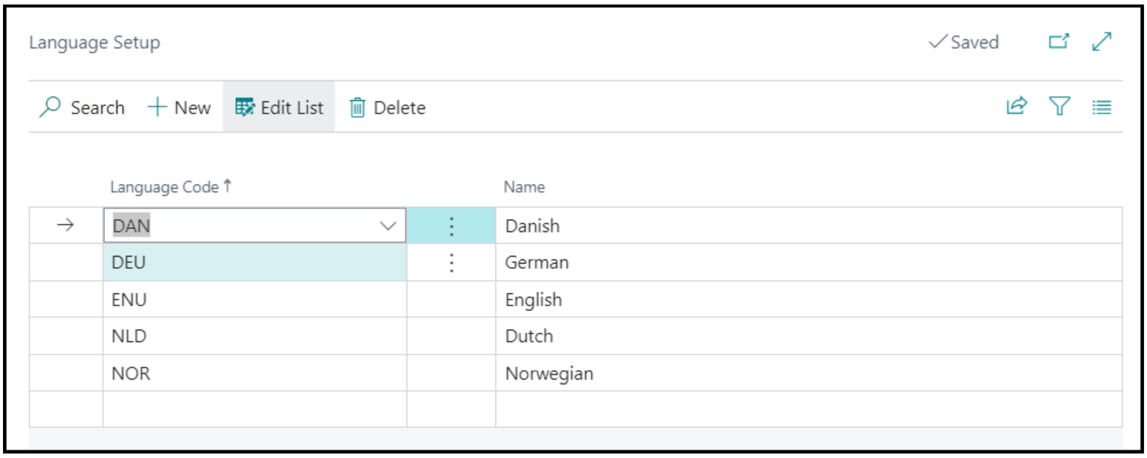
See [Sales Document Card](#) for the details.

RELATED



LANGUAGE SETUP

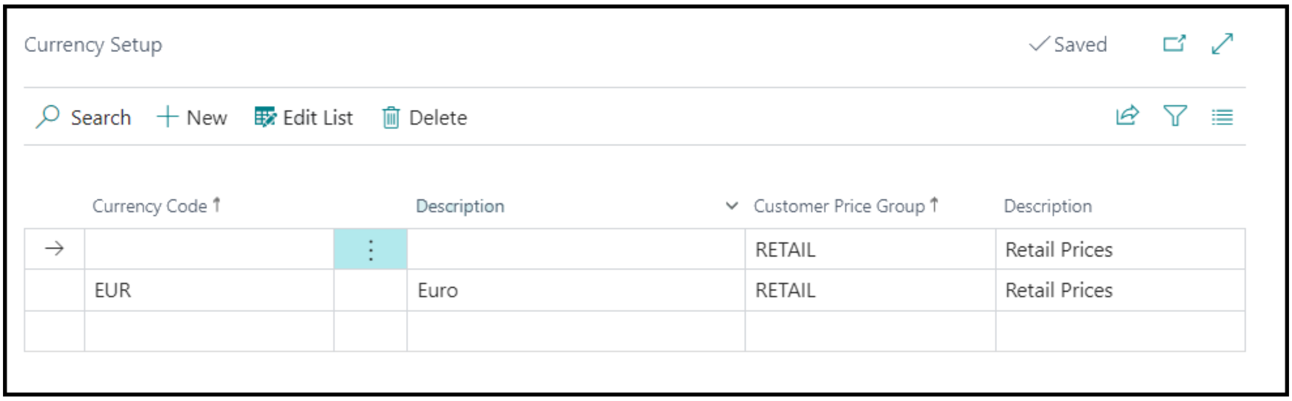
You can setup the **Language Codes** for which the Master (Item) Descriptions and Extended Text could be exported.



If you however have a checkmark in **Use Flat Item Only**, you will have to enter the Language related Descriptions on Item level; it will not construct it based on the Master Description Translations and the Variant Description Translations.
This is applicable for the following External API pages:
page 6037839 trm EAPI Item Description
page 6037851 trm EAPI Master Description
page 6037879 trm EAPI Product Description

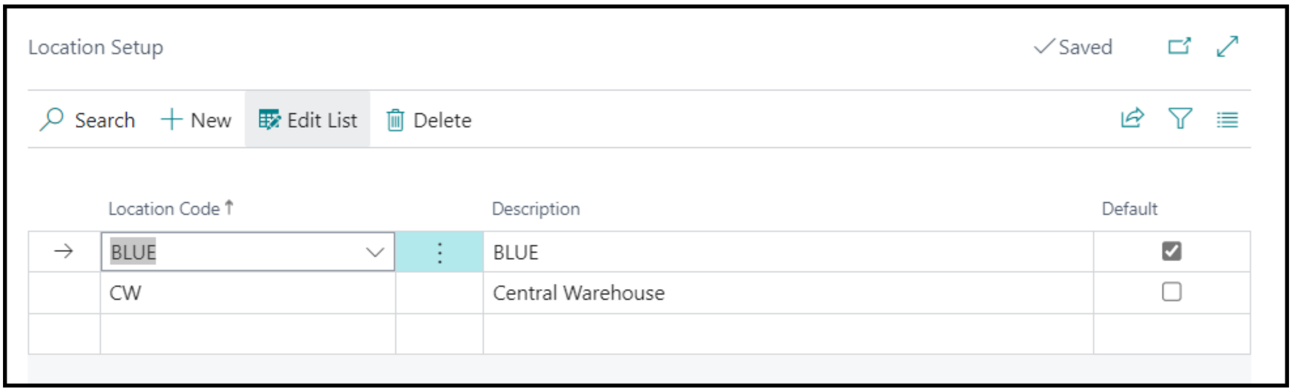
CURRENCY SETUP

For the export of Sales Prices, you can fill multiple combinations of **Currency Code** and **Customer Price Group,** to export several Sales Prices.
If the **Currency Setup** is not filled, it will export the Sales Price based on the **Customer Price Group** and **Currency Code** of the **Price and Discount Calculation Cust. No.** If this field is not filled, it will take the **Customer Price Group** and **Currency Code** of the **Standard Customer** in the Sales & Receivables Setup.



LOCATION SETUP

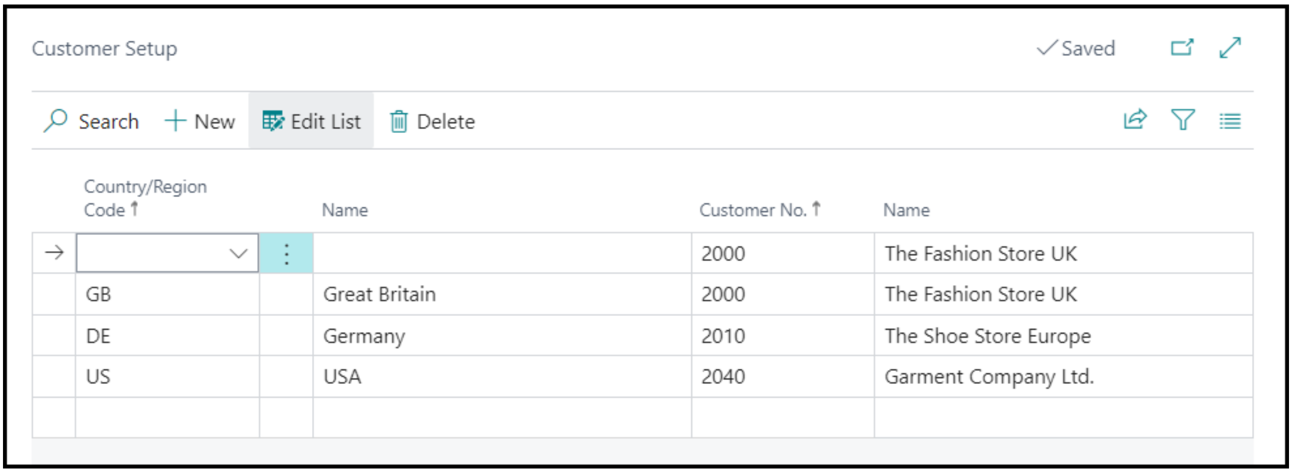
You need to setup the **Locations** that can be used in the imported Sales Documents:



If the imported Sales Document does not have a **Location,** it will be set to the **Location Code** in the **Location Setup** that has a checkmark in **Default.**
If the Integration Sales Document contains a **Location** that is not in the **Location Setup**, the import will fail.

CUSTOMER SETUP

Per **Country** you can setup the default **Customer No.** that should be taken in the Integration Sales Document if **Customer Handling** is set to *Use Customer Setup.*
The **Country/Region Code** will be checked with the **Ship-to Country/Region Code** from the Integration Sales Document.



Customer Setup

✓Saved

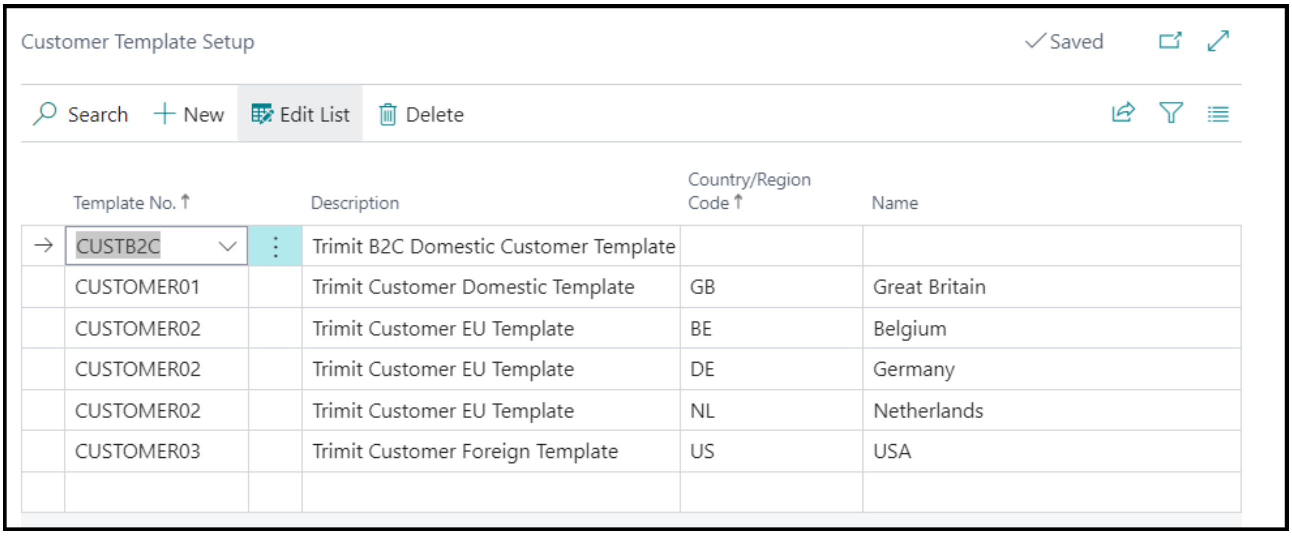
Search | New | Edit List | Delete

Country/Region Code ↑	Name	Customer No. ↑	Name
		2000	The Fashion Store UK
GB	Great Britain	2000	The Fashion Store UK
DE	Germany	2010	The Shoe Store Europe
US	USA	2040	Garment Company Ltd.

This will overrule any **Customer No.** that might be in the Integration Sales Document, and you will not see that **Customer No.** anywhere anymore in the regular Sales Document.

CUSTOMER TEMPLATE SETUP

Per **Country** you can setup the default **Customer Template Setup** that should be used to create a new Customer for an imported Sales Document if **Customer Handling** is set to *Use Customer Template Setup.*
The **Country/Region Code** will be checked with the **Sell-to Country/Region Code** from the Integration Sales Document.



Customer Template Setup

✓Saved

Search | New | Edit List | Delete

Template No. ↑	Description	Country/Region Code ↑	Name
CUSTB2C	Trimit B2C Domestic Customer Template		
CUSTOMER01	Trimit Customer Domestic Template	GB	Great Britain
CUSTOMER02	Trimit Customer EU Template	BE	Belgium
CUSTOMER02	Trimit Customer EU Template	DE	Germany
CUSTOMER02	Trimit Customer EU Template	NL	Netherlands
CUSTOMER03	Trimit Customer Foreign Template	US	USA

This will only happen if there is not an existing **Customer No.** with the same **Sell-to Type** and **E-Mail**. In that case the existing **Customer No.** will be taken.

ITEM SETUP

You can determine that if specific Item Numbers come in from the Integration Sales Documents, that it should be changed into another Item Number or a G/L Account.

Item Setup

Integration Item No.	Country/Region Code	Country/Region Name	Type	No.	Description
1234			Item	20002040	Olivia Trousers
TRANSPORT			G/L Account	9996820	Freight Webshop +

TABLE DATA SETUP

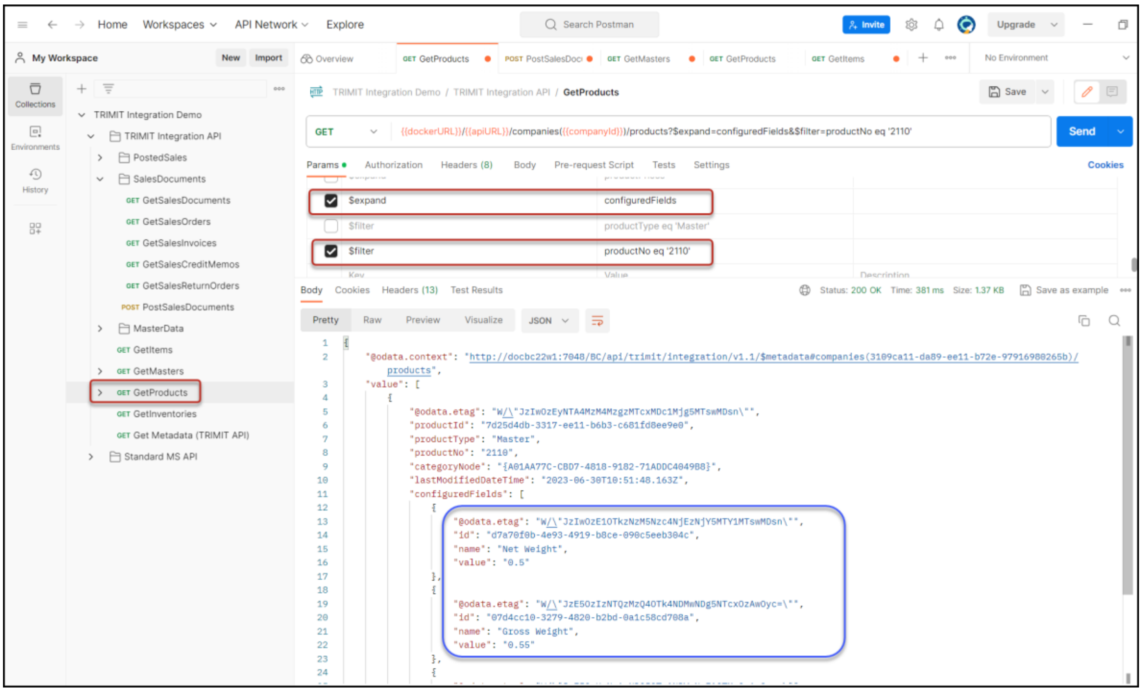
In **Table Data Setup**, you are able to add Fields from the Customer, Item or Master, which should be shown if you **Expand** the Product Data or Customer Data for the **ConfiguredFields.**

This can easily be used for Customized fields in your Customer, Item or Master to be exported.

Table Data Setup

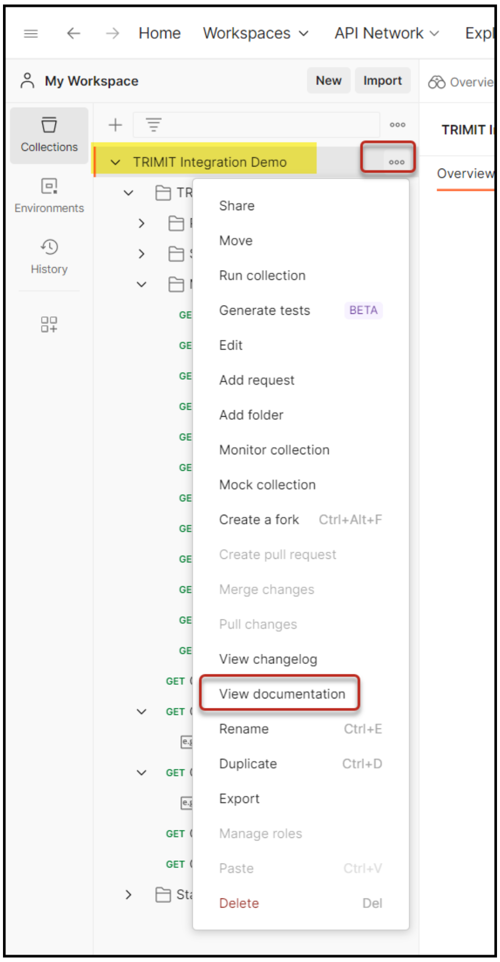
Table Number	Table Name	Field Number	Field Name
6037103	trm Master	420	Gross Weight
6037103	trm Master	421	Net Weight
18	Customer	18	Chain Name
27	Item	37	Alternative Item No.

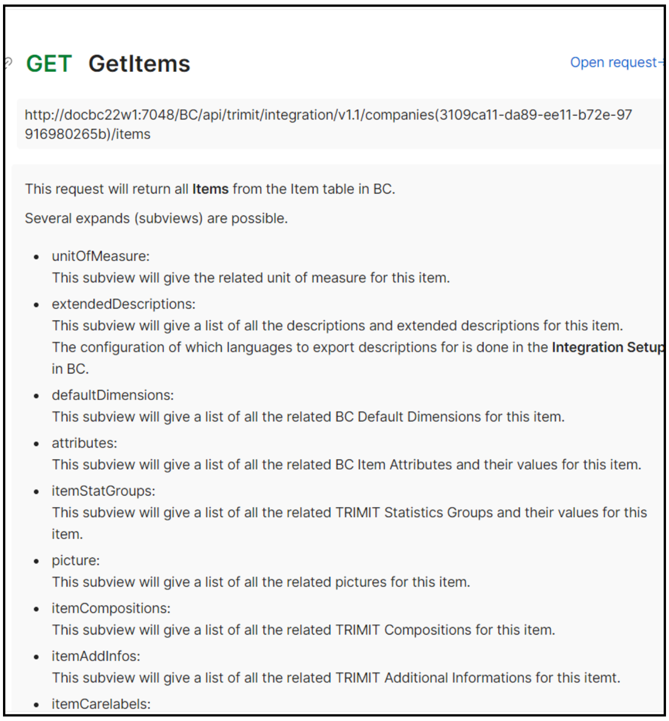
I.e.:



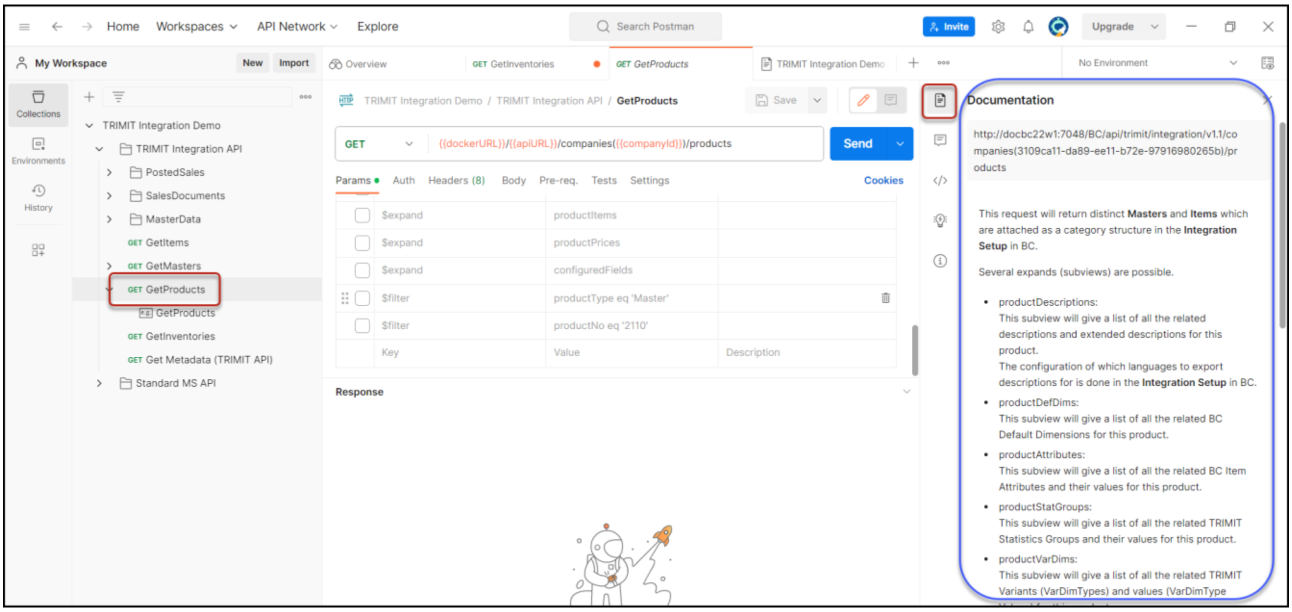
TRIMIT'S POSTMAN.JSON

In our **Postman collection** .json-file we also added documentation. Via the three dots, **View Documentation**, you can find all the documentation we embedded in the **Postman collection** .json-file**.**





It is also possible to see the Documentation for a specific export/import:



APPENDIX 1: CODEUNIT 6036860

You can also find Documentation to help you in the AL Code itself of Codeunit ***6036860 trm Integration Handling*** before all the Procedures. I.e.,

AL *IntegrationHandling.Codeunit.al* ×

src > src > Integration > Base > AL IntegrationHandling.Codeunit.al > Codeunit 6036860 "trm Integration Handling"

```
/// <summary>
/// Export a list of products with or without full product details.
/// </summary>
/// <param name="IntegrationSetupCode_par">The Integration Setup Code.</param>
/// <param name="IncludeDetails_par">Include or not include product details.</param>
0 references
procedure ExportProductList(IntegrationSetupCode_par: Code[20]; IncludeDetails_par: Boolean)
var
    TempProductData_loc: Record "trm Temp Integration Product" temporary;
begin
    ExportProductList(IntegrationSetupCode_par, IncludeDetails_par, TempProductData_loc);
end;
```

The following of this **Appendix 1** is an overview of all the documentation in this Codeunit.

EXPORTPRODUCTLIST()

Export a list of products with or without full product details.

Syntax

```
ExportProductList(IntegrationSetupCode_par: Code[20], IncludeDetails_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
IncludeDetails_par
Type: Boolean
Include or not include product details.

EXPORTPRODUCTLIST()

Export a list of products with or without full product details using a Data Container.

Syntax

```
ExportProductList(IntegrationSetupCode_par: Code[20], IncludeDetails_par: Boolean, var DataContainer_par: Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
IncludeDetails_par
Type: Boolean
Include or not include product details.
DataContainer_par
Type: Codeunit "trm Integration Data Container"

The Data Container is a single instance data cache codeunit, where the exported data is persisted.

EXPORTPRODUCTLIST()

Export a list of products with or without full product details returning the data.

Syntax

```
ExportProductList(IntegrationSetupCode_par: Code[20], IncludeDetails_par: Boolean,
var TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
IncludeDetails_par
Type: Boolean
Include or not include product details.
TempProductData_par
Type: Record "trm Temp Integration Product"
The Product Data in a temporary table

EXPORTPRODUCT()

Export product data for a specific master or item.

Syntax

```
ExportProduct(IntegrationSetupCode_par: Code[20], ProductType_par: Enum "trm
Integration Product Type", ProductNo_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ProductType_par
Type: Enum "trm Integration Product Type"
The product type of the specific product.
ProductNo_par
Type: Code[20]
The product no. of the specific product.

EXPORTPRODUCT()

Export product data for a specific master or item using a Data Container

Syntax

```
ExportProduct(IntegrationSetupCode_par: Code[20], ProductType_par: Enum "trm
Integration Product Type", ProductNo_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ProductType_par

Type: Enum "trm Integration Product Type"
The product type of the specific product.
ProductNo_par
Type: Code[20]
The product no. of the specific product.
DataContainer_par
Type: Codeunit "trm Integration Data Container"
The Data Container is a single instance data cache codeunit, where the exported data is persisted.

EXPORTPRODUCT()

Export product data for a specific master or item returning the data.

Syntax

```
ExportProduct(IntegrationSetupCode_par: Code[20], ProductType_par: Enum "trm
Integration Product Type", ProductNo_par: Code[20], var TempProductData_par:
Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ProductType_par
Type: Enum "trm Integration Product Type"
The product type of the specific product.
ProductNo_par
Type: Code[20]
The product no. of the specific product.
TempProductData_par
Type: Record "trm Temp Integration Product"
The Product Data in a temporary table

EXPORTPRODUCT()

Export product data for a specific master or item.

Syntax

```
ExportProduct(IntegrationSetupCode_par: Code[20], SystemId_par: Guid)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
SystemId_par
Type: Guid
The SystemId of the specific product.

EXPORTPRODUCT()

Export product data for a specific master or item returning the data.

Syntax

```
ExportProduct(IntegrationSetupCode_par: Code[20], SystemId_par: Guid, var
TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
SystemId_par
Type: Guid
The SystemId of the specific product.
TempProductData_par
Type: Record "trm Temp Integration Product"
The Product Data in a temporary table

TRACKEXPORTEDPRODUCTS()

Track Exported Products in Exported Products table.

Syntax

```
TrackExportedProducts(IntegrationSetupCode_par: Code[20], var TempProductData_par:
Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

TRACKEXPORTEDPRODUCT()

Track Exported Product in Exported Products table.

Syntax

```
TrackExportedProduct(IntegrationSetupCode_par: Code[20], ProductType_par: Enum
"trm Integration Product Type", ProductNo_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]

TRACKEXPORTEDPRODUCT()

Track Exported Product from SystemId in Exported Products table.

Syntax

```
TrackExportedProduct(IntegrationSetupCode_par: Code[20], SystemId_par: Guid)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
SystemId_par
Type: Guid

GETCATEGORIES()

Get a list of categories with the option of including products.

Syntax

```
GetCategories(IncludeProducts_par: Boolean, var ProductBufferLine_par: Record "trm CPO Category Buffer Line" temporary)
```

Parameters

IncludeProducts_par
Type: Boolean
Include or not include product data.
ProductBufferLine_par
Type: Record "trm CPO Category Buffer Line"
A temporary table containing the category data (VAR declared)

EXPORTINVENTORIES()

Export all inventories.

Syntax

```
ExportInventories(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

EXPORTINVENTORIES()

Export all inventories into a Data Container.

Syntax

```
ExportInventories(IntegrationSetupCode_par: Code[20], var DataContainer_par: Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTINVENTORIES()

Export all inventories into the temporary Inventory table.

Syntax

```
ExportInventories(IntegrationSetupCode_par: Code[20], var TempInventoryData_par: Record "trm Temp Integration Inventory" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempInventoryData_par
Type: Record "trm Temp Integration Inventory"

EXPORTINVENTORYPRODUCT()

Export inventory for a specific product and location code.

Syntax

```
ExportInventoryProduct(IntegrationSetupCode_par: Code[20], ItemNo_par: Code[20],
LocationCode_par: Code[10])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ItemNo_par
Type: Code[20]
LocationCode_par
Type: Code[10]

EXPORTINVENTORYPRODUCT()

Export inventory for a specific product and location code into a Data Container

Syntax

```
ExportInventoryProduct(IntegrationSetupCode_par: Code[20], ItemNo_par: Code[20],
LocationCode_par: Code[10], var DataContainer_par: Codeunit "trm Integration Data
Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ItemNo_par
Type: Code[20]
LocationCode_par
Type: Code[10]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTINVENTORYPRODUCT()

Export inventory for a specific product and location code into the temporary Inventory table.

Syntax

```
ExportInventoryProduct(IntegrationSetupCode_par: Code[20], ItemNo_par: Code[20],
LocationCode_par: Code[10], TempInventoryData_par: Record "trm Temp Integration
Inventory" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ItemNo_par
Type: Code[20]
LocationCode_par
Type: Code[10]
TempInventoryData_par
Type: Record "trm Temp Integration Inventory"

EXPORTITEMINVENTORIES()

Export non product-based inventories (filtered) into the temporary Inventory table.

Syntax

```
ExportItemInventories(IntegrationSetupCode_par: Code[20], var
TempItemInventoryData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempItemInventoryData_par
Type: Record "trm Temp Integration Product"

EXPORTITEMINVENTORIES()

Export inventory for a specific product and location code into the temporary Inventory table.

Syntax

```
ExportItemInventories(IntegrationSetupCode_par: Code[20], ItemNo_par: Code[20],
LocationCode_par: Code[10], TempItemInventoryData_par: Record "trm Temp
Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ItemNo_par
Type: Code[20]
LocationCode_par
Type: Code[10]
TempItemInventoryData_par
Type: Record "trm Temp Integration Product"

EXPORTPRICES()

Export a list of prices for the products.

Syntax

```
ExportPrices(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.

EXPORTPRICES()

Export a list of prices for the products into a Data Container.

Syntax

```
ExportPrices(IntegrationSetupCode_par: Code[20], var DataContainer_par: Codeunit
"trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par

Type: Code[20]
The Integration Setup Code.
DataContainer_par
Type: Codeunit "trm Integration Data Container"
The Data Container is a single instance data cache codeunit, where the exported data is persisted.

EXPORTPRICES()

Export a list of prices for the products returning the data in the temporary Product table.

Syntax

```
ExportPrices(IntegrationSetupCode_par: Code[20], var TempProductData_par: Record
"trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
TempProductData_par
Type: Record "trm Temp Integration Product"
The Product Data in a temporary table

EXPORTPRICESPRODUCT()

Export a list of prices for one product returning the data in the temporary Product table.

Syntax

```
ExportPricesProduct(IntegrationSetupCode_par: Code[20], ProductType_par: Enum "trm
Integration Product Type", ProductNo_par: Code[20], var TempProductData_par:
Record "trm Temp Integration Product" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"
The Product Data in a temporary table

EXPORTCUSTOMERS()

Export a list of Customers.

Syntax

```
ExportCustomers(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.

EXPORTCUSTOMERS()

Export a list of Customers using a Data Container.

Syntax

```
ExportCustomers(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
DataContainer_par
Type: Codeunit "trm Integration Data Container"
The Data Container is a single instance data cache codeunit, where the exported data is persisted.

EXPORTCUSTOMERS()

Export a list of Customers returning the data into a temporary Customer table.

Syntax

```
ExportCustomers(IntegrationSetupCode_par: Code[20], var TempCustomerData_par:
Record "Customer" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
TempCustomerData_par
Type: Record "Customer"
The Customer Data in a temporary table

EXPORTCUSTOMER()

Export a Customer using a Data Container.

Syntax

```
ExportCustomer(IntegrationSetupCode_par: Code[20], CustomerNo_par: Code[20], var
DataContainer_par: Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
CustomerNo_par
Type: Code[20]
The Customer No. to return.
DataContainer_par
Type: Codeunit "trm Integration Data Container"
The Data Container is a single instance data cache codeunit, where the exported data is persisted.

EXPORTCUSTOMER()

Export a Customer returning the data into a temporary Customer table.

Syntax

```
ExportCustomer(IntegrationSetupCode_par: Code[20], CustomerNo_par: Code[20], var
TempCustomerData_par: Record "Customer" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
CustomerNo_par
Type: Code[20]
The Customer No. to return.
TempCustomerData_par
Type: Record "Customer"
The Customer Data in a temporary table

IMPORTSALESORDERS()

Import Sales Orders.

Syntax

```
ImportSalesOrders(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

IMPORTSALESORDERS()

Import Sales Orders using a Data Container.

Syntax

```
ImportSalesOrders(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

IMPORTSALESORDERS()

Import Sales Orders using temporary Sales Document tables.

Syntax

```
ImportSalesOrders(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

IMPORTSALESINVOICES()

Import Sales Invoices.

Syntax

```
ImportSalesInvoices(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

IMPORTSALESINVOICES()

Import Sales Invoices using a Data Container.

Syntax

```
ImportSalesInvoices(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

IMPORTSALESINVOICES()

Import Sales Invoices using temporary Sales Document tables.

Syntax

```
ImportSalesInvoices(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

IMPORTSALESRETURNORDERS()

Import Sales Return Orders.

Syntax

```
ImportSalesReturnOrders(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

IMPORTSALESRETURNORDERS()

Import Sales Return Orders using a Data Container.

Syntax

```
ImportSalesReturnOrders(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

IMPORTSALESRETURNORDERS()

Import Sales Return Orders using temporary Sales Document tables.

Syntax

```
ImportSalesReturnOrders(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

IMPORTSALESCREDITMEMOS()

Import Sales Credit Memos.

Syntax

```
ImportSalesCreditMemos(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

IMPORTSALESCREDITMEMOS()

Import Sales Credit Memos using a Data Container.

Syntax

```
ImportSalesCreditMemos(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

IMPORTSALESCREDITMEMOS()

Import Sales Credit Memos using temporary Sales Document tables.

Syntax

```
ImportSalesCreditMemos(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

CLEARTEMPSALESDOCUMENTS()

Clears and empties the temporary tables for Sales Documents.

Syntax

```
ClearTempSalesDocuments(var tempSalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header" temporary, var tempSalesImportJnlLine_par: Record "trm Sales Import
Journal Line" temporary)
```

Parameters

tempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
tempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

CLEARTEMPPOSTEDSALESDOCUMENTS()

Clears and empties the temporary tables for Posted Sales Documents.

Syntax

```
ClearTempPostedSalesDocuments(var tempSalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header" temporary, var tempSalesImportJnlLine_par: Record "trm Sales
Import Journal Line" temporary, var tempIntegrationSalesTracking_par: Record "trm
Temp Integration SaleTrack" temporary)
```

Parameters

tempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
tempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
tempIntegrationSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

EXPORTSALESSHIPMENTS()

Export Sales Shipments.

Syntax

```
ExportSalesShipments(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

EXPORTSALESSHIPMENTS()

Export Posted Shipments using a Data Container.

Syntax

```
ExportSalesShipments(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTSALESSHIPMENTS()

Export Posted Shipments using the temporary Sales Document tables.

Syntax

```
ExportSalesShipments(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary, var
TempSalesTracking_par: Record "trm Temp Integration SaleTrack" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

EXPORTSALESSHIPMENTS()

Export Posted Shipments using temporary Sales Shipment tables.

Syntax

```
ExportSalesShipments(IntegrationSetupCode_par: Code[20], var
TempSalesShipmentHeader_loc: Record "Sales Shipment Header" temporary, var
TempSalesShipmentLine_loc: Record "Sales Shipment Line" temporary, var
TempPortalOrderExtraInfo_loc: Record "trm Portal Order Extra Info" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesShipmentHeader_loc
Type: Record "Sales Shipment Header"

TempSalesShipmentLine_loc
Type: Record "Sales Shipment Line"
TempPortalOrderExtraInfo_loc
Type: Record "trm Portal Order Extra Info"

EXPORTSALESINVOICES()

Export Posted Invoices.

Syntax

```
ExportSalesInvoices(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

EXPORTSALESINVOICES()

Export Posted Invoices using a Data Container.

Syntax

```
ExportSalesInvoices(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTSALESINVOICES()

Export Posted Invoices using the temporary Sales Document tables.

Syntax

```
ExportSalesInvoices(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary, var
TempSalesTracking_par: Record "trm Temp Integration SaleTrack" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

EXPORTSALESINVOICES()

Export Posted Invoices using temporary Sales Invoice tables.

Syntax

```
ExportSalesInvoices(IntegrationSetupCode_par: Code[20], var
TempSalesInvoiceHeader_loc: Record "Sales Invoice Header" temporary, var
TempSalesInvoiceLine_loc: Record "Sales Invoice Line" temporary, var
TempOrderExtraInfo_loc: Record "trm Portal Order Extra Info" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesInvoiceHeader_loc
Type: Record "Sales Invoice Header"
TempSalesInvoiceLine_loc
Type: Record "Sales Invoice Line"
TempOrderExtraInfo_loc
Type: Record "trm Portal Order Extra Info"

EXPORTSALESRETURNRECEIPTS()

Export Posted Return Receipts.

Syntax

```
ExportSalesReturnReceipts(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

EXPORTSALESRETURNRECEIPTS()

Export Posted Return Receipts using a Data Container.

Syntax

```
ExportSalesReturnReceipts(IntegrationSetupCode_par: Code[20], var
DataContainer_par: Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTSALESRETURNRECEIPTS()

Export Posted Return Receipts using the temporary Sales Document tables.

Syntax

```
ExportSalesReturnReceipts(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary, var
TempSalesTracking_par: Record "trm Temp Integration SaleTrack" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

EXPORTSALESRETURNRECEIPTS()

Export Posted Return Receipts using temporary Return Receipt tables.

Syntax

```
ExportSalesReturnReceipts(IntegrationSetupCode_par: Code[20], var
TempSalesReturnReceiptHeader_par: Record "Return Receipt Header" temporary, var
TempSalesReturnReceiptLine_par: Record "Return Receipt Line" temporary, var
TempPortalOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesReturnReceiptHeader_par
Type: Record "Return Receipt Header"
TempSalesReturnReceiptLine_par
Type: Record "Return Receipt Line"
TempPortalOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

EXPORTSALESCREDITMEMOS()

Export Posted Credit Memos.

Syntax

```
ExportSalesCreditMemos(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

EXPORTSALESCREDITMEMOS()

Export Posted Credit Memos using a Data Container.

Syntax

```
ExportSalesCreditMemos(IntegrationSetupCode_par: Code[20], var DataContainer_par:
Codeunit "trm Integration Data Container")
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
DataContainer_par
Type: Codeunit "trm Integration Data Container"

EXPORTSALESCREDITMEMOS()

Export Posted Credit Memos the temporary Sales Document tables.

Syntax

```
ExportSalesCreditMemos(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary, var
TempSalesTracking_par: Record "trm Temp Integration SaleTrack" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

EXPORTSALESCREDITMEMOS()

Export Posted Credit Memos using temporary Sales Credit Memo tables.

Syntax

```
ExportSalesCreditMemos(IntegrationSetupCode_par: Code[20], var
TempSalesCrMemoHeader_par: Record "Sales Cr.Memo Header" temporary, var
TempSalesCrMemoLine_par: Record "Sales Cr.Memo Line" temporary, var
TempPortalOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesCrMemoHeader_par
Type: Record "Sales Cr.Memo Header"
TempSalesCrMemoLine_par
Type: Record "Sales Cr.Memo Line"
TempPortalOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETDEFAULTLOCATIONCODE()

Gets the default location from the integration setup.

Syntax

```
[Code[10]] := GetDefaultLocationCode()
```

Return

Code[10]
LocationCode

GETLOCATIONCODEWITHCHECK()

Gets a location from the integration setup.

Syntax

```
[Code[10]] := GetLocationCodeWithCheck(SetLocationCode_par: Code[10])
```

Parameters

SetLocationCode_par

Type: Code[10]

Return

Code[10]

GETPRODUCTDESCRIPTIONS()

Get descriptions for a specific product.

Syntax

```
GetProductDescriptions(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTDESCRIPTIONS()

Get descriptions for a specific product.

Syntax

```
GetProductDescriptions(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTDIMENSIONS()

Get dimensions for a specific product.

Syntax

```
GetProductDimensions(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTDIMENSIONS()

Get dimensions for a specific product.

Syntax

```
GetProductDimensions(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTSTATISTICSGROUPS()

Get statistics groups for a specific product.

Syntax

```
GetProductStatisticsGroups(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTSTATISTICSGROUPS()

Get statistics groups for a specific product.

Syntax

```
GetProductStatisticsGroups(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTVARDIMS()

Get vardims for a specific product.

Syntax

```
GetProductVarDims(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTVARDIMS()

Get vardims for a specific product.

Syntax

```
GetProductVarDims(var TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCOMPOSITIONS()

Get compositions for a specific product.

Syntax

```
GetProductCompositions(ProductType_par: Enum "trm Integration Product Type", ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCOMPOSITIONS()

Get compositions for a specific product.

Syntax

```
GetProductCompositions(var TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCARELABELS()

Get care labels for a specific product.

Syntax

```
GetProductCarelabels(ProductType_par: Enum "trm Integration Product Type", ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCARELABELS()

Get care labels for a specific product.

Syntax

```
GetProductCarelabels(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTADDITIONALINFO()

Get additional information for a specific product.

Syntax

```
GetProductAdditionalInfo(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTADDITIONALINFO()

Get additional information for a specific product.

Syntax

```
GetProductAdditionalInfo(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTADDITIONALTAGS()

Get additional tags for a specific product.

Syntax

```
GetProductAdditionalTags(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par

Type: Record "trm Temp Integration Product"

GETPRODUCTADDITIONALTAGS()

Get additional tags for a specific product.

Syntax

```
GetProductAdditionalTags(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTSUSTAINABILITY()

Get sustainability information for a specific product.

Syntax

```
GetProductSustainability(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTSUSTAINABILITY()

Get sustainability information for a specific product.

Syntax

```
GetProductSustainability(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTMEASUREMENTCHARTS()

Get measurement charts for a specific product.

Syntax

```
GetProductMeasurementCharts(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par

Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTMEASUREMENTCHARTS()

Get measurement charts for a specific product.

Syntax

```
GetProductMeasurementCharts(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCOLLECTIONS()

Get related collections for a specific product.

Syntax

```
GetProductCollections(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTCOLLECTIONS()

Get related collections for a specific product.

Syntax

```
GetProductCollections(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTPRICES()

Get sales prices for a specific product.

Syntax

```
GetProductPrices(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par

Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTPRICES()

Get sales prices for a specific product.

Syntax

```
GetProductPrices(var TempProductData_par: Record "trm Temp Integration Product"
temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTITEMS()

Get related items for a specific product (only master).

Syntax

```
GetProductItems(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTITEMS()

Get related items for a specific product (only master).

Syntax

```
GetProductItems(var TempProductData_par: Record "trm Temp Integration Product"
temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTITEMATTRIBUTES()

Get item attributes for a specific product.

Syntax

```
GetProductItemAttributes(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTITEMATTRIBUTES()

Get item attributes for a specific product.

Syntax

```
GetProductItemAttributes(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTIMAGES()

Get related images for a specific product.

Syntax

```
GetProductImages(ProductType_par: Enum "trm Integration Product Type",
ProductNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]
TempProductData_par
Type: Record "trm Temp Integration Product"

GETPRODUCTIMAGES()

Get related images for a specific product.

Syntax

```
GetProductImages(var TempProductData_par: Record "trm Temp Integration Product"
temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

GETTABLEDATA()

Get table data for a specific product.

Syntax

```
GetTableData(var TempIntegrationData_par: Record "trm Temp Integration Data"
```

```
temporary)
```

Parameters

TempIntegrationData_par
Type: Record "trm Temp Integration Data"

GETINTEGRATIONPRODUCTFROMFILTER()

Get ProductType and ProductNo from a filter on Integration Product table.

Syntax

```
GetIntegrationProductFromFilter(var IntegrationProduct_par: Record "trm Temp
Integration Product", var ProductType_par: Enum "trm Integration Product Type",
var ProductNo_par: Code[20])
```

Parameters

IntegrationProduct_par
Type: Record "trm Temp Integration Product"
ProductType_par
Type: Enum "trm Integration Product Type"
ProductNo_par
Type: Code[20]

GETINTEGRATIONPRODUCTFROMFILTER()

return Product Type and Product No in the Integration Product table from a filter on the same table.

Syntax

```
GetIntegrationProductFromFilter(var IntegrationProduct_par: Record "trm Temp
Integration Product")
```

Parameters

IntegrationProduct_par
Type: Record "trm Temp Integration Product"

COPYINTEGRATIONPRODUCT()

Copies one or more Integration Product records.

Syntax

```
CopyIntegrationProduct(var FromIntegrationProduct_par: Record "trm Temp
Integration Product", var ToIntegrationProduct_par: Record "trm Temp Integration
Product", ClearTarget_par: Boolean)
```

Parameters

FromIntegrationProduct_par
Type: Record "trm Temp Integration Product"
ToIntegrationProduct_par
Type: Record "trm Temp Integration Product"
ClearTarget_par
Type: Boolean

CHECKPRODUCTEXPORTALLOWED()

Checks if product export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckProductExportAllowed()
```

CheckInventoryExportAllowed()

Checks if inventory export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckInventoryExportAllowed()
```

CHECKCUSTOMEREXPORTALLOWED()

Checks if Customer export is enabled on the Integration Setup. If not enabled, it will give an error.

Syntax

```
CheckCustomerExportAllowed()
```

CHECKSALESQUOTEIMPORTALLOWED()

Checks if sales quote import is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckSalesQuoteImportAllowed()
```

CHECKSALESORDERIMPORTALLOWED()

Checks if sales order import is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckSalesOrderImportAllowed()
```

CHECKSALESINVOICEIMPORTALLOWED()

Checks if sales invoice import is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckSalesInvoiceImportAllowed()
```

CHECKSALESCREDITMEMOIMPORTALLOWED()

Checks if sales credit memo import is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckSalesCreditMemoImportAllowed()
```

CHECKSALESRETURNORDERIMPORTALLOWED()

Checks if sales return order import is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

```
CheckSalesReturnOrderImportAllowed()
```

CHECKPOSTEDSALESSHIPMENTEXPORTALLOWED()

Checks if sales shipment export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

`CheckPostedSalesShipmentExportAllowed()`

CHECKPOSTEDSALESINVOICEEXPORTALLOWED()

Checks if sales invoice export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

`CheckPostedSalesInvoiceExportAllowed()`

CHECKPOSTEDSALESRETURNSHIPMENTEXPORTALLOWED()

Checks if sales return shipment export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

`CheckPostedSalesReturnShipmentExportAllowed()`

CheckPostedSalesCreditMemoExportAllowed()

Checks if sales credit memo export is allowed on the Integration Setup. If not allowed, it will give an error.

Syntax

`CheckPostedSalesCreditMemoExportAllowed()`

GETCODE10FROMTEXT20()

Validates and change the Value_par from Text[20] to Code[10] characters. Useable in TRIMIT because many
relations are based on Text[20] fields eventhough code only has a max length of 10.

Syntax

`[Code[10]] := GetCode10FromText20(Value_par: Text[20])`

Parameters

Value_par
Type: Text[20]
The text value from a Text[20] variable or field.

Return

Code[10]
Return value as a valid Code[10] value.

GETCODE10FROMCODE20()

Validates and change the Value_par from 20 to 10 characters. Useable in TRIMIT because various table relations are based on Code[20] fields eventhough the used code only has a max length of 10.

Syntax

`[Code[10]] := GetCode10FromCode20(Value_par: Code[20])`

Parameters

Value_par
Type: Code[20]
The code from a Code[20] variable or field.

Return

Code[10]
Return code as a valid Code[10] value.

GETCODE10FROMCODE40()

Validates and change the Value_par from 40 to 10 characters. Useable in TRIMIT because various table relations are based on Code[40] fields eventhough the used code only has a max length of 10.

Syntax

```
[Code[10]] := GetCode10FromCode40(Value_par: Code[40])
```

Parameters

Value_par
Type: Code[40]
The code from a Code[40] variable or field.

Return

Code[10]
Return code as a valid Code[10] value.

GETCODE20FROMCODE30()

Validates and change the Value_par from 30 to 20 characters.

Syntax

```
[Code[20]] := GetCode20FromCode30(Value_par: Code[30])
```

Parameters

Value_par
Type: Code[30]
The code from a Code[30] variable or field.

Return

Code[20]
Return code as a valid Code[20] value.

GETCODE20FROMCODE35()

Validates and change the Value_par from 35 to 20 characters.

Syntax

```
[Code[20]] := GetCode20FromCode35(Value_par: Code[35])
```

Parameters

Value_par
Type: Code[35]
The code from a Code[35] variable or field.

Return

Code[20]
Return code as a valid Code[20] value.

FINDINTEGRATIONSETUPCODE()

Finds the Integration Setup Code related to a specific User ID.

Syntax

```
[Code[20]] := FindIntegrationSetupCode(UserID_par: Code[50])
```

Parameters

UserID_par
Type: Code[50]
The User ID used to search for the Integration Setup Code. Use blank for Default.

Return

Code[20]
The Integration Setup Code.

SETINTEGRATIONSETUP()

Validates and sets the Integration Setup as a global variable used in this codeunits procedures.

Syntax

```
SetIntegrationSetup(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.

PF_ONAFTERISSALEALLOWEDCHECK()

Integration Event that can be used to alter the return value of the procedure IsSaleAllowedCheck(). This event
is triggered right before returning the value in the procedure IsSaleAllowedCheck().

Syntax

```
pf_OnAfterIsSaleAllowedCheck(IntegrationSetupCode_par: Code[20], var
ReturnValue_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ReturnValue_par
Type: Boolean
The value returned in the function IsSaleAllowedCheck.

PF_ONAFTERGENERATEPRODUCTLIST()

Integration Event that can be used to alter the product list generated in the procedure GenerateProductList().
This event is triggered as the last Line in the procedure GenerateProductList().

Syntax

```
pf_OnAfterGenerateProductList(IntegrationSetupCode_par: Code[20])
```

Parameters

IntegrationSetupCode_par
Type: Code[20]

The Integration Setup Code.

PF_ONAFTERGETPRODUCTLIST()

Syntax

```
pf_OnAfterGetProductList(IncludeDetails_par: Boolean, var TempProductData_par:
Record "trm Temp Integration Product" temporary)
```

Parameters

IncludeDetails_par
Type: Boolean
TempProductData_par
Type: Record "trm Temp Integration Product"

PF_ONAFTERGETPRODUCTDETAILS()

Syntax

```
pf_OnAfterGetProductDetails(var TempProductData_par: Record "trm Temp Integration
Product" temporary)
```

Parameters

TempProductData_par
Type: Record "trm Temp Integration Product"

PF_ONAFTERGETEXTENDEDDESCRIPTION()

Syntax

```
pf_OnAfterGetExtendedDescription(ItemNo_par: Code[20], LanguageCode_par: Code[10],
var ExtendedText_par: Text)
```

Parameters

ItemNo_par
Type: Code[20]
LanguageCode_par
Type: Code[10]
ExtendedText_par
Type: Text

PF_ONAFTERGETDESCRIPTION()

Syntax

```
pf_OnAfterGetDescription(ItemNo_par: Code[20], LanguageCode_par: Code[10], var
Description_par: Text[100], var Description2_par: Text[50])
```

Parameters

ItemNo_par
Type: Code[20]
LanguageCode_par
Type: Code[10]
Description_par
Type: Text[100]
Description2_par

Type: Text[50]

PF_ONAFTERGETPRICE()

Syntax

```
pf_OnAfterGetPrice(ItemNo_par: Code[20], CustomerPriceGroup_par: Code[10],
CurrencyCode_par: Code[10], var Price_par: Decimal, var PriceFound_par: Boolean)
```

Parameters

ItemNo_par
Type: Code[20]
CustomerPriceGroup_par
Type: Code[10]
CurrencyCode_par
Type: Code[10]
Price_par
Type: Decimal
PriceFound_par
Type: Boolean

PF_ONAFTERGETEAN()

Syntax

```
pf_OnAfterGetEAN(ItemNo_par: Code[20], var EAN_par: Code[50])
```

Parameters

ItemNo_par
Type: Code[20]
EAN_par
Type: Code[50]

PF_ONAFTERGETGTIN()

Syntax

```
pf_OnAfterGetGTIN(ItemNo_par: Code[20], var GTIN_par: Code[14])
```

Parameters

ItemNo_par
Type: Code[20]
GTIN_par
Type: Code[14]

PF_ONAFTERGETINVENTORIES()

Syntax

```
pf_OnAfterGetInventories(var TempInventoryData_par: Record "trm Temp Integration
Inventory" temporary)
```

Parameters

TempInventoryData_par
Type: Record "trm Temp Integration Inventory"

PF_ONAFTERGETINVENTORY()

Syntax

```
pf_OnAfterGetInventory(var TempInventoryData_par: Record "trm Temp Integration Inventory" temporary)
```

Parameters

TempInventoryData_par
Type: Record "trm Temp Integration Inventory"

PF_ONAFTERGETITEMINVENTORIES()

Syntax

```
pf_OnAfterGetItemInventories(var TempItemInventoryData_par: Record "trm Temp Integration Product" temporary)
```

Parameters

TempItemInventoryData_par
Type: Record "trm Temp Integration Product"

PF_ONAFTERCALCULATEINVENTORY()

Syntax

```
pf_OnAfterCalculateInventory(ItemNo_par: Code[20], LocationCode_par: Code[10], var AvailableQty_par: Decimal)
```

Parameters

ItemNo_par
Type: Code[20]
LocationCode_par
Type: Code[10]
AvailableQty_par
Type: Decimal

APPENDIX 2: CODEUNIT 6036862

You can also find Documentation to help you in the AL Code itself of
Codeunit ***6036862 trm Integration Sales Handling*** before all the Procedures**.** I.e.,

```
AL IntegrationSalesHandling.Codeunit.al ×
src > src > Integration > Documents > AL IntegrationSalesHandling.Codeunit.al

    /// <summary>
    /// Postprocess the document based on the current Integration Setup or the settings in the Sales Import Journal Header.
    /// </summary>
    /// <param name="SalesImportJnlHeader_par"></param>
    1 reference
    procedure ProcessDocumentWithSetup(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
    var
        SalesImportJnlHeader_loc: Record "trm Sales Import Jnl Header";
    begin
        SalesImportJnlHeader_loc.SetRange("Import Document Type", SalesImportJnlHeader_par."Import Document Type");
        SalesImportJnlHeader_loc.SetRange("Document Type", SalesImportJnlHeader_par."Document Type");
        SalesImportJnlHeader_loc.SetRange("No.", SalesImportJnlHeader_par."No.");
        SalesImportJnlHeader_loc.SetRange("Integration Setup Code", SalesImportJnlHeader_par."Integration Setup Code");
        SalesImportJnlHeader_loc.FindFirst();
```

The following of this **Appendix 2** is an overview of all the documentation in this Codeunit.

PROCESSDOCUMENTWITHSETUP()

Postprocess the document based on the current Integration Setup or the settings in the Sales Import Journal
Header.

Syntax

```
ProcessDocumentWithSetup(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

REVERSEDOCUMENTWITHCHECK()

Reverse the document based on the current Integration Setup and return it in the temporary tables. If the document has been reversed, the temporary tables will contain the reversed document otherwise they will be empty. If the Integration Setup is not set for reversal an error will be raised if the document has wrong sign on total.

Syntax

```
ReverseDocumentWithCheck(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header", var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header" temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import
Journal Line" temporary)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par

Type: Record "trm Sales Import Journal Line"

REVERSEDOCUMENT()

Syntax

```
ReverseDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", NewDocumentType_par: Enum "Sales Document Type", DocumentAmount_par:
Decimal, var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"
temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line"
temporary)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
NewDocumentType_par
Type: Enum "Sales Document Type"
DocumentAmount_par
Type: Decimal
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

REVERSEDOCUMENTLINES()

Syntax

```
ReverseDocumentLines(SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", NewDocumentType_par: Enum "Sales Document Type", DocumentAmount_par:
Decimal, var TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line"
temporary)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
NewDocumentType_par
Type: Enum "Sales Document Type"
DocumentAmount_par
Type: Decimal
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

PROCESSDOCUMENT()

Postprocess the document.

Syntax

```
ProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

PROCESSDOCUMENT()

Postprocess the document based on the function parameter.

Syntax

```
ProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", ReleaseDocument_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDocument_par
Type: Boolean

PROCESSDOCUMENT()

Postprocess the document based on the function parameters.

Syntax

```
ProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", ReleaseDocument_par: Boolean, PostDocument_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDocument_par
Type: Boolean
PostDocument_par
Type: Boolean

PROCESSDOCUMENT()

Postprocess the document based on the function parameters.

Syntax

```
ProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", ReleaseDocument_par: Boolean, CreatePayment_par: Boolean,
PostDocument_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDocument_par
Type: Boolean
CreatePayment_par
Type: Boolean
PostDocument_par
Type: Boolean

PROCESSDOCUMENT()

Postprocess the document based on the function parameters.

Syntax

```
ProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
```

```
Header", ReleaseDocument_par: Boolean, CreatePickDoc_par: Boolean,
CreatePayment_par: Boolean, PostDocument_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDocument_par
Type: Boolean
CreatePickDoc_par
Type: Boolean
CreatePayment_par
Type: Boolean
PostDocument_par
Type: Boolean

CREATEDOCUMENT()

Creates a document from the Sales Import Journal Header tables.

Syntax

```
CreateDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

SETDOCUMENTNOANDSERIES()

Sets the No., Internal No., and the No. Series (if applicable) in the Sales Import Journal Header table.

Syntax

```
SetDocumentNoAndSeries(var salesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

salesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEJOURNAL()

Creates a General Journal from the Sales Import Journal Header table.

Syntax

```
CreateJournal(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEDOCJOURNAL()

Syntax

```
CreateDocJournal(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

RELEASEDOCUMENT()

Releases the document referred in the Sales Import Journal Header.

Syntax

```
ReleaseDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEPICKDOCFROMORDER()

Creates a Pick Document based on the Sales Order referred in the Sales Import Journal Header.

Syntax

```
CreatePickDocFromOrder(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEORDERFROMQUOTE()

Creates a Sales Order from the Sales Quote referred in the Sales Import Journal Header.

Syntax

```
CreateOrderFromQuote(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEINVOICEFROMQUOTE()

Creates a Sales Invoice from the Sales Quote referred in the Sales Import Journal Header.

Syntax

```
CreateInvoiceFromQuote(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEPAYMENT()

Creates a Payment on the document referred in the Sales Import Journal Header.

Syntax

```
CreatePayment(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

POSTDOCUMENT()

Post the document referred in the Sales Import Journal Header.

Syntax

```
PostDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

POSTDOCUMENTJOURNAL()

Post the General Journal line referred in the Sales Import Journal Header.

Syntax

```
PostDocumentJournal(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

POSTDOCWITHJOURNAL()

Syntax

```
PostDocWithJournal(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

SETINVOICEEXPORTEDSTATUS()

Syntax

```
SetInvoiceExportedStatus(InvoiceNo_par: Code[20], Value_par: Boolean)
```

Parameters

InvoiceNo_par
Type: Code[20]
Value_par
Type: Boolean

SETSALESINVOICEEXPORTEDSTATUS()

Set the Exported flag on the Invoice with a confirmation dialog if ConfirmDialog is set.

Syntax

```
SetSalesInvoiceExportedStatus(IntegrationSetupCode_par: Code[20], InvoiceNo_par:
Code[20], Value_par: Boolean, ConfirmDialog_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
InvoiceNo_par
Type: Code[20]
Value_par
Type: Boolean
ConfirmDialog_par
Type: Boolean

SETCREDITMEMOEXPORTEDSTATUS()

Syntax

```
SetCreditMemoExportedStatus(CreditMemoNo_par: Code[20], Value_par: Boolean)
```

Parameters

CreditMemoNo_par
Type: Code[20]
Value_par
Type: Boolean

SETSALESCREDITMEMOEXPORTEDSTATUS()

Set the Exported flag on the Credit Memo with a confirmation dialog if ConfirmDialog is set.

Syntax

```
SetSalesCreditMemoExportedStatus(IntegrationSetupCode_par: Code[20],
CreditMemoNo_par: Code[20], Value_par: Boolean, ConfirmDialog_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
CreditMemoNo_par
Type: Code[20]
Value_par
Type: Boolean
ConfirmDialog_par
Type: Boolean

SETSHIPMENTEXPORTEDSTATUS()

Syntax

```
SetShipmentExportedStatus(ShipmentNo_par: Code[20], Value_par: Boolean)
```

Parameters

ShipmentNo_par
Type: Code[20]
Value_par

Type: Boolean

SETSALESSHIPMENTEXPORTEDSTATUS()

Set the Exported flag on the Shipment with a confirmation dialog if ConfirmDialog is set.

Syntax

```
SetSalesShipmentExportedStatus(IntegrationSetupCode_par: Code[20], ShipmentNo_par: Code[20], Value_par: Boolean, ConfirmDialog_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ShipmentNo_par
Type: Code[20]
Value_par
Type: Boolean
ConfirmDialog_par
Type: Boolean

SETRETURNRECEIPTEXPORTEDSTATUS()

Syntax

```
SetReturnReceiptExportedStatus(ReturnShipmentNo_par: Code[20], Value_par: Boolean)
```

Parameters

ReturnShipmentNo_par
Type: Code[20]
Value_par
Type: Boolean

SETSALESRETURNRECEIPTEXPORTEDSTATUS()

Set the Exported flag on the Return Shipment with a confirmation dialog if ConfirmDialog is set.

Syntax

```
SetSalesReturnReceiptExportedStatus(IntegrationSetupCode_par: Code[20], ReturnShipmentNo_par: Code[20], Value_par: Boolean, ConfirmDialog_par: Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
ReturnShipmentNo_par
Type: Code[20]
Value_par
Type: Boolean
ConfirmDialog_par
Type: Boolean

ISDOCUMENTNOUSED()

Check to see if the current Document No. has already been used.

Syntax

```
[Boolean] := IsDocumentNoUsed(var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header", Silent_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
Silent_par
Type: Boolean

Return

Boolean

ISDOCUMENTCREATED()

Check whether the document has already been imported and return true if it is imported, otherwise false. The
function will raise an error based on the ThrowError flag.

Syntax

```
[Boolean] := IsDocumentCreated(var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header", Silent_par: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
Silent_par
Type: Boolean

Return

Boolean

HASSELLTOINFO()

Returns true if the document has Seller customer and address information filled.

Syntax

```
[Boolean] := HasSellToInfo(SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

Return

Boolean

HASBILLTOINFO()

Returns true if the document has Billing customer and address information filled.

Syntax

```
[Boolean] := HasBillToInfo(SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Return

Boolean

HASSHIPTOINFO()

Returns true if the document has Shipping customer and address information filled.

Syntax

```
[Boolean] := HasShipToInfo(SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

Return

Boolean

CREATEORDEREXTRAINFO()

Create Extra Order Information from the referred Sales Import Journal Header.

Syntax

```
CreateOrderExtraInfo(SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEORDEREXTRAINFO()

Create Extra Order Information from the referred Sales Import Journal Header and return it in a temporary
OrderExtraInfo record.

Syntax

```
CreateOrderExtraInfo(var salesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", var orderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

salesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
orderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETINTEGRATIONITEMWITHCHECK()

Finds the Type and No. of an Integration Item. Raises an error if the Integration Item does not exist.

Syntax

```
GetIntegrationItemWithCheck(IntegrationItemNo_par: Code[20],
CountryRegionCode_par: Code[10], var Type_par: Enum "trm Integration Setup
LineType", var No_par: Code[20])
```

Parameters

IntegrationItemNo_par
Type: Code[20]
CountryRegionCode_par
Type: Code[10]
Type_par
Type: Enum "trm Integration Setup LineType"
No_par
Type: Code[20]

GETINTEGRATIONITEM()

Finds the Type and No. of an Integration Item. Returns true if the Integration Item exist.

Syntax

```
[Boolean] := GetIntegrationItem(IntegrationItemNo_par: Code[20],
CountryRegionCode_par: Code[10], var Type_par: Enum "trm Integration Setup
LineType", var No_par: Code[20])
```

Parameters

IntegrationItemNo_par
Type: Code[20]
CountryRegionCode_par
Type: Code[10]
Type_par
Type: Enum "trm Integration Setup LineType"
No_par
Type: Code[20]

Return

Boolean

GETSHIPMENTS()

Export Posted Shipments using temporary Sales Shipment tables.

Syntax

```
GetShipments(var TempSalesShipmentHeader_par: Record "Sales Shipment Header"
temporary, var TempSalesShipmentLine_par: Record "Sales Shipment Line" temporary,
var TempPortalOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempSalesShipmentHeader_par
Type: Record "Sales Shipment Header"
TempSalesShipmentLine_par
Type: Record "Sales Shipment Line"
TempPortalOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETSHIPMENTS()

Export Posted Shipments using the temporary Sales Document tables.

Syntax

```
GetShipments(var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header" temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import
Journal Line" temporary, var TempSalesTracking_par: Record "trm Temp Integration
SaleTrack" temporary)
```

Parameters

TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

GETINVOICES()

Export Posted Invoices using temporary Sales Invoice tables.

Syntax

```
GetInvoices(var TempSalesInvoiceHeader_par: Record "Sales Invoice Header"
temporary, var TempSalesInvoiceLine_par: Record "Sales Invoice Line" temporary,
var TempPortalOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempSalesInvoiceHeader_par
Type: Record "Sales Invoice Header"
TempSalesInvoiceLine_par
Type: Record "Sales Invoice Line"
TempPortalOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETINVOICES()

Export Posted Invoices using the temporary Sales Document tables.

Syntax

```
GetInvoices(var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"
temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line"
temporary, var TempSalesTracking_par: Record "trm Temp Integration SaleTrack"
temporary)
```

Parameters

TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

GETRETURNRECEIPTS()

Export Posted Return Receipts using temporary Return Receipt tables.

Syntax

```
GetReturnReceipts(var TempReturnReceiptHeader_par: Record "Return Receipt Header"
temporary, var TempReturnReceiptLine_par: Record "Return Receipt Line" temporary,
var TempOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempReturnReceiptHeader_par
Type: Record "Return Receipt Header"
TempReturnReceiptLine_par
Type: Record "Return Receipt Line"
TempOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETRETURNRECEIPTS()

Export Posted Return Receipts using the temporary Sales Document tables.

Syntax

```
GetReturnReceipts(var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header" temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import
Journal Line" temporary, var TempSalesTracking_par: Record "trm Temp Integration
SaleTrack" temporary)
```

Parameters

TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

GETCREDITMEMOS()

Export Posted Credit Memos using temporary Sales Credit Memo tables.

Syntax

```
GetCreditMemos(var TempSalesCrMemoHeader_par: Record "Sales Cr.Memo Header"
temporary, var TempSalesCrMemoLine_par: Record "Sales Cr.Memo Line" temporary, var
TempPortalOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempSalesCrMemoHeader_par
Type: Record "Sales Cr.Memo Header"
TempSalesCrMemoLine_par
Type: Record "Sales Cr.Memo Line"
TempPortalOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

GETCREDITMEMOS()

Export Posted Credit Memos the temporary Sales Document tables.

Syntax

```
GetCreditMemos(var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header" temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import
Journal Line" temporary, var TempSalesTracking_par: Record "trm Temp Integration
```

```
SaleTrack" temporary)
```

Parameters

TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
TempSalesTracking_par
Type: Record "trm Temp Integration SaleTrack"

GETCUSTOMERNO()

Returns the Customer No. from the Integration Setup. Raises an error if a problem is encountered.

Syntax

```
[Code[20]] := GetCustomerNo(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

Return

Code[20]

GETCUSTOMERNOFROMSETUP()

Syntax

```
[Code[20]] := GetCustomerNoFromSetup(var SalesImportJnlHeader_par: Record "trm
Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

GETCUSTOMERNOFROMTEMPLATESETUP()

Syntax

```
[Code[20]] := GetCustomerNoFromTemplateSetup(var SalesImportJnlHeader_par: Record
"trm Sales Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATECUSTOMERFROMTEMPLATE()

Syntax

```
[Code[20]] := CreateCustomerFromTemplate(CustTemplateNo: Code[20], var
SalesImportJournalHeader_par: Record "trm Sales Import Jnl Header")
```

Parameters

CustTemplateNo
Type: Code[20]

SalesImportJournalHeader_par
Type: Record "trm Sales Import Jnl Header"

CREATEINTEGRATIONPARAMS()

Syntax

```
CreateIntegrationParams(SellToInfoSet_par: Boolean, BillToInfoSet_par: Boolean,
ShipToInfoSet_par: Boolean)
```

Parameters

SellToInfoSet_par
Type: Boolean
BillToInfoSet_par
Type: Boolean
ShipToInfoSet_par
Type: Boolean

IMPORTSALESDOCUMENTS()

Syntax

```
ImportSalesDocuments(IntegrationSetupCode_par: Code[20], var
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

SETINTEGRATIONSETUP()

Sets the Integration Setup as a global variable used in this codeunits procedures.

Syntax

```
SetIntegrationSetup(IntegrationSetupCode_par: Code[20], ForceGetSetup_par:
Boolean)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
The Integration Setup Code.
ForceGetSetup_par
Type: Boolean
Defines if the Integration Setup should be forced retrieved from the database or not.

PF_ONAFTERIMPORTSALESDOCUMENTS()

Syntax

```
pf_OnAfterImportSalesDocuments(IntegrationSetupCode_par: Code[20], var
```

```
TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header" temporary, var
TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line" temporary)
```

Parameters

IntegrationSetupCode_par
Type: Code[20]
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"

ONAFTERGETSHIPMENT()

Syntax

```
OnAfterGetShipment(var TempSalesShipmentHeader_par: Record "Sales Shipment Header"
temporary, var TempSalesShipmentLine_par: Record "Sales Shipment Line" temporary,
var TempOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempSalesShipmentHeader_par
Type: Record "Sales Shipment Header"
TempSalesShipmentLine_par
Type: Record "Sales Shipment Line"
TempOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

PF_ONAFTERGETINVOICE()

Syntax

```
Pf_OnAfterGetInvoice(var TempSalesInvoiceHeader_par: Record "Sales Invoice Header"
temporary, var TempSalesInvoiceLine_par: Record "Sales Invoice Line" temporary,
var TempOrderExtraInfo_par: Record "trm Portal Order Extra Info" temporary)
```

Parameters

TempSalesInvoiceHeader_par
Type: Record "Sales Invoice Header"
TempSalesInvoiceLine_par
Type: Record "Sales Invoice Line"
TempOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

PF_ONAFTERGETRETURNRECEIPT()

Syntax

```
Pf_OnAfterGetReturnReceipt(var TempReturnReceiptHeader_par: Record "Return Receipt
Header" temporary, var TempReturnReceiptLine_par: Record "Return Receipt Line"
temporary, var TempOrderExtraInfo_par: Record "trm Portal Order Extra Info"
temporary)
```

Parameters

TempReturnReceiptHeader_par
Type: Record "Return Receipt Header"

TempReturnReceiptLine_par
Type: Record "Return Receipt Line"
TempOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

PF_ONAFTERGETCREDITMEMO()

Syntax

```
Pf_OnAfterGetCreditMemo(var TempSalesCrMemoHeader_par: Record "Sales Cr.Memo
Header" temporary, var TempSalesCrMemoLine_par: Record "Sales Cr.Memo Line"
temporary, var TempOrderExtraInfo_par: Record "trm Portal Order Extra Info")
```

Parameters

TempSalesCrMemoHeader_par
Type: Record "Sales Cr.Memo Header"
TempSalesCrMemoLine_par
Type: Record "Sales Cr.Memo Line"
TempOrderExtraInfo_par
Type: Record "trm Portal Order Extra Info"

PF_ONGETCUSTOMERNO()

Syntax

```
Pf_OnGetCustomerNo(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", var CustomerNo_par: Code[20])
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
CustomerNo_par
Type: Code[20]

PF_ONAFTERGETCUSTNOFROMSETUP()

Syntax

```
Pf_OnAfterGetCustNoFromSetup(IntegrationSetup_par: Record "trm Integration Setup",
var CustomerNo_par: Code[20], var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header")
```

Parameters

IntegrationSetup_par
Type: Record "trm Integration Setup"
CustomerNo_par
Type: Code[20]
SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

PF_ONAFTERCREATEDOCUMENT()

Syntax

```
Pf_OnAfterCreateDocument(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header", var SalesHeader_par: Record "Sales Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
SalesHeader_par
Type: Record "Sales Header"

PF_ONAFTERCREATEORDERFROMQUOTE()

Syntax

```
Pf_OnAfterCreateOrderFromQuote(var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

PF_ONAFTERCREATEINVOICEFROMQUOTE()

Syntax

```
Pf_OnAfterCreateInvoiceFromQuote(var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"

PF_ONAFTERCREATEPICK()

Syntax

```
Pf_OnAfterCreatePick(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", var SalesHeader_par: Record "Sales Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
SalesHeader_par
Type: Record "Sales Header"

PF_ONAFTERAUTORELEASE()

Syntax

```
Pf_OnAfterAutoRelease(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", var SalesHeader_par: Record "Sales Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
SalesHeader_par
Type: Record "Sales Header"

PF_ONAFTERMANUALRELEASE()

Syntax

```
Pf_OnAfterManualRelease(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl
Header", var SalesHeader_par: Record "Sales Header")
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
SalesHeader_par
Type: Record "Sales Header"

PF_ONBEFOREREVERSEDOCUMENT()

Syntax

```
Pf_OnBeforeReverseDocument(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header", NewDocumentType_par: Enum "Sales Document Type", DocumentAmount_par:
Decimal, var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"
temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line"
temporary, var IsHandled_loc: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
NewDocumentType_par
Type: Enum "Sales Document Type"
DocumentAmount_par
Type: Decimal
TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
IsHandled_loc
Type: Boolean

PF_ONAFTERREVERSEDOCUMENT()

Syntax

```
Pf_OnAfterReverseDocument(var SalesImportJnlHeader_par: Record "trm Sales Import
Jnl Header", NewDocumentType_par: Enum "Sales Document Type", DocumentAmount_par:
Decimal, var TempSalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"
temporary, var TempSalesImportJnlLine_par: Record "trm Sales Import Journal Line"
temporary, IsHandled_loc: Boolean)
```

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
NewDocumentType_par
Type: Enum "Sales Document Type"
DocumentAmount_par
Type: Decimal

TempSalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
TempSalesImportJnlLine_par
Type: Record "trm Sales Import Journal Line"
IsHandled_loc
Type: Boolean

PF_ONBEFORECREATECUSTOMER()

Syntax

```
Pf_OnBeforeCreateCustomer(IntegrationSetup_par: Record "trm Integration Setup",
CustTemplateNo_par: Code[20], var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header", var Customer_par: Record Customer, var Handled_loc: Boolean)
```

Parameters

IntegrationSetup_par
Type: Record "trm Integration Setup"
CustTemplateNo_par
Type: Code[20]
SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
Customer_par
Type: Record Customer
Handled_loc
Type: Boolean

PF_ONAFTERCREATECUSTOMER()

Syntax

```
Pf_OnAfterCreateCustomer(IntegrationSetup_par: Record "trm Integration Setup",
CustTemplateNo_par: Code[20], var SalesImportJnlHeader_par: Record "trm Sales
Import Jnl Header", var Customer_par: Record Customer, Handled_loc: Boolean)
```

Parameters

IntegrationSetup_par
Type: Record "trm Integration Setup"
CustTemplateNo_par
Type: Code[20]
SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
Customer_par
Type: Record Customer
Handled_loc
Type: Boolean