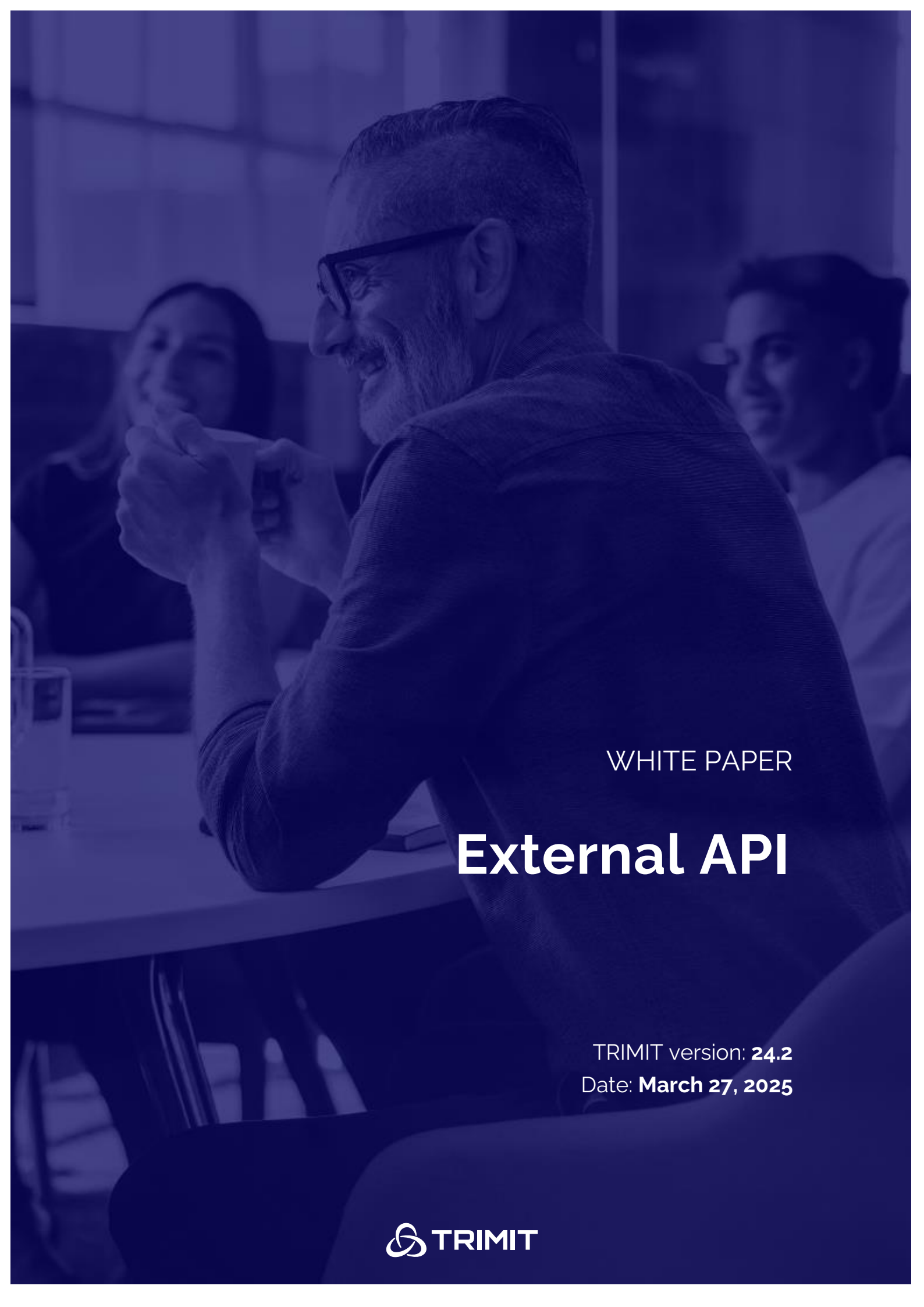




WHITE PAPER

External API

TRIMIT version: **24.2**

Date: **March 27, 2025**

Table of Contents

External API	1
Introduction.....	9
Components.....	10
Integration Setup.....	14
FastTab General.....	15
FastTab Products	16
FastTab Inventory	16
FastTab Customers.....	17
FastTab Sales Prices.....	18
FastTab Sales Documents.....	18
General Setup.....	19
Customer Setup	20
Sales Quote Setup.....	21
Sales Order Setup.....	22
Sales Invoice setup.....	23
Sales Return Order Setup	24
Sales Credit Memo Setup	25
Reversal Setup	25
Not in the Page, but can be added via Personalize:	26
FastTab Posted Sales Documents.....	28
Actions, Products.....	28
Edit Category Lines.....	28
Category Management.....	28
Refresh Products.....	28
Exported Products.....	29
Category List.....	29
Actions, Documents	29
Create Sales Posting Job Queue Job	29
Sales Documents	30
Posted Sales Documents	33
Related on the Integration Setup	35
Language Setup.....	35
Currency Setup.....	35
Location Setup	36
Customer Setup	36
Customer Template Setup.....	37

Item Setup	37
Table Data Setup	37
TRIMIT's Postman.json.....	39
Documentation in Postman Collection.....	39
Item References	40
Prices	41
VarDim Order	42
Requests	43
Request Setup	43
Requests Running	44
Request Log	44
Actions, Requests, Create Request Job Queue Job	44
Related, Request	45
Appendix 1: Codeunit 6036860	46
ExportProductList()	46
ExportProductList()	46
ExportProductList()	47
ExportProduct()	47
ExportProduct()	47
ExportProduct()	48
ExportProduct()	48
ExportProduct()	48
TrackExportedProducts()	49
TrackExportedProduct()	49
TrackExportedProduct()	49
GetCategories()	50
ExportInventories()	50
ExportInventories()	50
ExportInventories()	50
ExportInventoryProduct()	51
ExportInventoryProduct()	51
ExportInventoryProduct()	51
ExportItemInventories()	52
ExportItemInventories()	52
ExportPrices()	52
ExportPrices()	52
ExportPrices()	53
ExportPricesProduct()	53

ExportCustomers()	53
ExportCustomers()	54
ExportCustomers()	54
ExportCustomer()	54
ExportCustomer()	54
ImportSalesOrders()	55
ImportSalesOrders()	55
ImportSalesOrders()	55
ImportSalesInvoices()	56
ImportSalesInvoices()	56
ImportSalesInvoices()	56
ImportSalesReturnOrders()	56
ImportSalesReturnOrders()	57
ImportSalesReturnOrders()	57
ImportSalesCreditMemos()	57
ImportSalesCreditMemos()	57
ImportSalesCreditMemos()	58
ClearTempSalesDocuments()	58
ClearTempPostedSalesDocuments()	58
ExportSalesShipments()	58
ExportSalesShipments()	59
ExportSalesShipments()	59
ExportSalesShipments()	59
ExportSalesInvoices()	60
ExportSalesInvoices()	60
ExportSalesInvoices()	60
ExportSalesInvoices()	60
ExportSalesReturnReceipts()	61
ExportSalesReturnReceipts()	61
ExportSalesReturnReceipts()	61
ExportSalesReturnReceipts()	62
ExportSalesCreditMemos()	62
ExportSalesCreditMemos()	62
ExportSalesCreditMemos()	62
ExportSalesCreditMemos()	63
GetDefaultLocationCode()	63
GetLocationCodeWithCheck()	63
GetProductDescriptions()	64

GetProductDescriptions()	64
GetProductDimensions()	64
GetProductDimensions()	64
GetProductStatisticsGroups()	65
GetProductStatisticsGroups()	65
GetProductVarDims()	65
GetProductVarDims()	65
GetProductVarDimOrderChoices()	66
GetProductVarDimOrderChoices()	66
GetProductCompositions()	66
GetProductCompositions()	66
GetProductCarelables()	67
GetProductCarelables()	67
GetProductAdditionalInfo()	67
GetProductAdditionalInfo()	67
GetProductAdditionalTags()	68
GetProductAdditionalTags()	68
GetProductSustainability()	68
GetProductSustainability()	69
GetProductMeasurementCharts()	69
GetProductMeasurementCharts()	69
GetProductCollections()	69
GetProductCollections()	70
GetProductPrices()	70
GetProductPrices()	70
GetProductItems()	70
GetProductItems()	71
GetProductItemAttributes()	71
GetProductItemAttributes()	71
GetProductImages()	71
GetProductImages()	72
GetTableData()	72
GetIntegrationProductFromFilter()	72
GetIntegrationProductFromFilter()	72
CopyIntegrationProduct()	73
CheckProductExportAllowed()	73
CheckInventoryExportAllowed()	73
CheckCustomerExportAllowed()	73

CheckSalesQuoteImportAllowed()	73
CheckSalesOrderImportAllowed()	73
CheckSalesInvoiceImportAllowed()	73
CheckSalesCreditMemoImportAllowed()	74
CheckSalesReturnOrderImportAllowed()	74
CheckPostedSalesShipmentExportAllowed()	74
CheckPostedSalesInvoiceExportAllowed()	74
CheckPostedSalesReturnShipmentExportAllowed()	74
CheckPostedSalesCreditMemoExportAllowed()	74
GetCode10FromText20()	74
GetCode10FromCode20()	75
GetCode10FromCode40()	75
GetCode20FromCode30()	75
GetCode20FromCode35()	76
FindIntegrationSetupCode()	76
SetIntegrationSetup()	76
pf_OnAfterIsSaleAllowedCheck()	76
pf_OnAfterGenerateProductList()	77
pf_OnAfterGetProductList()	77
pf_OnAfterGetProductDetails()	77
pf_OnAfterGetExtendedDescription()	77
pf_OnAfterGetDescription()	78
pf_OnAfterGetPrice()	78
pf_OnAfterGetPriceWithFilters()	78
pf_OnAfterGetEAN()	79
pf_OnAfterGetGTIN()	79
pf_OnAfterGetReferences()	79
pf_OnAfterGetInventories()	79
pf_OnAfterGetInventory()	79
pf_OnAfterGetItemInventories()	80
pf_OnAfterCalculateInventory()	80
Appendix 2: Codeunit 6036862	81
ProcessDocumentWithSetup()	81
ReverseDocumentWithCheck()	81
ReverseDocument()	81
ReverseDocumentLines()	82
ProcessDocument()	82
ProcessDocument()	82

ProcessDocument()	83
ProcessDocument()	83
ProcessDocument()	83
CreateDocument()	84
SetDocumentNoAndSeries()	84
CreateJournal()	84
CreateDocJournal()	84
ReleaseDocument()	84
CreatePickDocFromOrder()	85
CreateOrderFromQuote()	85
CreateInvoiceFromQuote()	85
CreatePayment()	85
PostDocument()	86
PostDocumentJournal()	86
PostDocWithJournal()	86
SetInvoiceExportedStatus()	86
SetSalesInvoiceExportedStatus()	86
SetCreditMemoExportedStatus()	87
SetSalesCreditMemoExportedStatus()	87
SetShipmentExportedStatus()	87
SetSalesShipmentExportedStatus()	87
SetReturnReceiptExportedStatus()	88
SetSalesReturnReceiptExportedStatus()	88
IsDocumentNoUsed()	88
IsDocumentCreated()	89
HasSellToInfo()	89
HasBillToInfo()	89
HasShipToInfo()	89
CreateOrderExtraInfo()	90
CreateOrderExtraInfo()	90
GetIntegrationItemWithCheck()	90
GetIntegrationItem()	91
GetShipments()	91
GetShipments()	91
GetInvoices()	92
GetInvoices()	92
GetReturnReceipts()	92
GetReturnReceipts()	93

GetCreditMemos()	93
GetCreditMemos()	93
GetCustomerNo()	93
GetCustomerNoFromSetup()	94
GetCustomerNoFromTemplateSetup()	94
CreateCustomerFromTemplate()	94
CreateIntegrationParams()	94
ImportSalesDocuments()	95
SetIntegrationSetup()	95
Pf_OnAfterImportSalesDocuments()	95
OnAfterGetShipment()	95
Pf_OnAfterGetInvoice()	96
Pf_OnAfterGetReturnReceipt()	96
Pf_OnAfterGetCreditMemo()	96
Pf_OnGetCustomerNo()	97
Pf_OnAfterGetCustNoFromSetup()	97
Pf_OnAfterCreateDocument()	97
Pf_OnAfterCreateOrderFromQuote()	97
Pf_OnAfterCreateInvoiceFromQuote()	98
Pf_OnAfterCreatePick()	98
Pf_OnAfterAutoRelease()	98
Pf_OnAfterManualRelease()	98
Pf_OnBeforeReverseDocument()	98
Pf_OnAfterReverseDocument()	99
Pf_OnBeforeCreateCustomer()	99
Pf_OnAfterCreateCustomer()	100
Pf_OnAfterCreateCustFromTemplate()	100
Pf_OnBeforeCreateIntegrationDoc()	100
Pf_OnAfterCreateDocument()	101
Pf_OnAfterCreateIntegrationDoc	101
Pf_OnBeforeProcessDocumentWithSetup()	101
Pf_OnBeforeProcessIntegrationDocWithSetup()	102
Pf_OnAfterProcessDocumentWithSetup()	102
Pf_OnAfterProcessIntegrationDocWithSetup()	102
Pf_OnBeforeProcessDocument()	103
Pf_OnBeforeProcessIntegrationDoc()	103
Pf_OnAfterProcessDocument()	104
Pf_OnAfterProcessIntegrationDoc()	104

Introduction

The **TRIMIT External API** (Application Programming Interface) is a replacement for the Microsoft's API and extends it with the TRIMIT specific Data.

It enables you to

- ➔ Export Master/Item Data
- ➔ Export Sales Prices of Masters/Items
- ➔ Export Inventory/Availability
- ➔ Export Customer Data
- ➔ Import Sales Documents (including VarDim Order information)
- ➔ Export Posted Sales Invoices and Posted Credit Memos (including VarDim Order information)
- ➔ Create new Customers

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, exporting Sales-/Campaign Prices 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 this 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)

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, Developer, 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 24.2 for Business Central W1 BC25.X**.

The pictures in this White Paper are based on the demo data for **TRIMIT 24.2**, which is based on Microsoft Dynamics 365 Business Central 2024 Release Wave 2, 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	<i>trm Temp Integration Product</i>
	6036836	<i>trm Temp Integration Inventory</i>
	6036837	<i>trm Temp Integration SaleTrack</i>
	6036838	<i>trm Temp Integration Data</i>
	6036840	<i>trm Integration Setup</i>
	6036841	<i>trm Integration Setup Line</i>
	6036842	<i>trm Integration Exp. Product</i>
	6036843	<i>trm Integration Exp. Document</i>
	6036846	<i>trm EAPI Request Log</i>
	6036847	<i>trm EAPI Request</i>
	6037049	<i>trm Sales Comment Import</i>

The Pages	6036201	<i>trm Integration Exp. Products</i>
	6036870	<i>trm Integration Setup List</i>
	6036871	<i>trm Integration Setup</i>
	6036872	<i>trm Integration Customer Setup</i>
	6036873	<i>trm Integration Currency Setup</i>
	6036874	<i>trm Integration Language Setup</i>
	6036875	<i>trm Integration Location Setup</i>
	6036876	<i>trm Integration Item Setup</i>
	6036877	<i>trm Integration Cust. Template</i>
	6036879	<i>trm Integration Table Setup</i>
	6036880	<i>trm Integration Overview</i>
	6036881	<i>trm Integration Sales Docs</i>
	6036882	<i>trm Integration Sales Document</i>
	6036883	<i>trm Integration Sales Doc Sub</i>
	6036884	<i>trm Integration Overview FB</i>
	6036886	<i>trm Integration Exported Prod</i>
	6036890	<i>trm Integration EAPI Requests</i>
	6036891	<i>trm EAPI Requests Running</i>
	6036895	<i>trm EAPI Requests Log</i>
	6036896	<i>trm Integration EAPI Req Setup</i>
	6037017	<i>trm Sales Comment Import List</i>

The API Pages	6037801	<i>trm EAPI Attachments</i>
	6037802	<i>trm EAPI Campaigns</i>
	6037803	<i>trm EAPI Categories</i>
	6037804	<i>trm EAPI Configured Fields</i>
	6037805	<i>trm EAPI Customer Price Groups</i>
	6037806	<i>trm EAPI Default Dimensions</i>
	6037807	<i>trm EAPI Dim.Set Lines</i>
	6037808	<i>trm EAPI Item Attributes</i>
	6037809	<i>trm EAPI Locations</i>
	6037810	<i>trm EAPI PDF Document</i>
	6037811	<i>trm EAPI Pictures</i>
	6037812	<i>trm EAPI Price Group Params</i>
	6037813	<i>trm EAPI Salespersons</i>
	6037814	<i>trm EAPI Units Of Measure</i>

The API Pages

6037815	<i>trm EAPI Collections</i>
6037816	<i>trm EAPI Delivery Periods</i>
6037817	<i>trm EAPI Contacts</i>
6037818	<i>trm EAPI Contacts Info</i>
6037819	<i>trm EAPI Cust Finance Dtls</i>
6037820	<i>trm EAPI Customers</i>
6037821	<i>trm EAPI Return Order Lines</i>
6037822	<i>trm EAPI Return Orders</i>
6037823	<i>trm EAPI Sales Credit Memos</i>
6037824	<i>trm EAPI Sales Cr.Memo Lines</i>
6037825	<i>trm EAPI Sales Document Lines</i>
6037826	<i>trm EAPI Sales Document</i>
6037827	<i>trm EAPI Sales Invoice Lines</i>
6037828	<i>trm EAPI Sales Invoices</i>
6037829	<i>trm EAPI Sales Order Lines</i>
6037830	<i>trm EAPI Sales Orders</i>
6037831	<i>trm EAPI Sales Quote Lines</i>
6037832	<i>trm EAPI Sales Quotes</i>
6037833	<i>trm EAPI Inventories</i>
6037834	<i>trm EAPI Item Additional Info</i>
6037835	<i>trm EAPI Item Additional Tags</i>
6037836	<i>trm EAPI Item Attribute</i>
6037837	<i>trm EAPI Item Carelabel</i>
6037838	<i>trm EAPI Item Composition</i>
6037839	<i>trm EAPI Item Description</i>
6037840	<i>trm EAPI Item Inventory</i>
6037841	<i>trm EAPI Item Price</i>
6037842	<i>trm EAPI Items</i>
6037843	<i>trm EAPI Item Stat. Group</i>
6037844	<i>trm EAPI Item Sustainability</i>
6037845	<i>trm EAPI Master Add. Info</i>
6037846	<i>trm EAPI Master Add. Tags</i>
6037847	<i>trm EAPI Master Attributes</i>
6037848	<i>trm EAPI Master Carelabel</i>
6037849	<i>trm EAPI Master Collection</i>
6037850	<i>trm EAPI Master Composition</i>
6037851	<i>trm EAPI Master Description</i>
6037852	<i>trm EAPI Master Dimension</i>
6037853	<i>trm EAPI Master Inventories</i>
6037854	<i>trm EAPI Master Item</i>
6037855	<i>trm EAPI Master Measure Chart</i>
6037856	<i>trm EAPI Master Price</i>
6037857	<i>trm EAPI Masters</i>
6037858	<i>trm EAPI Master Stat. Group</i>
6037859	<i>trm EAPI Master Sustainability</i>
6037860	<i>trm EAPI Master VarDim</i>
6037861	<i>trm EAPI Exported Documents</i>
6037862	<i>trm EAPI SalesPostCreditMemos</i>
6037863	<i>trm EAPI SalesPostCrMemoLines</i>
6037864	<i>trm EAPI SalesPostDocLines</i>
6037865	<i>trm EAPI SalesPostDocuments</i>

The API Pages	6037866	<i>trm EAPI SalesPostInvLines</i>
	6037867	<i>trm EAPI SalesPostInvoices</i>
	6037868	<i>trm EAPI SalesPostRetRcptLines</i>
	6037869	<i>trm EAPI SalesPostReturnRcpts</i>
	6037870	<i>trm EAPI SalesPostShipLines</i>
	6037871	<i>trm EAPI SalesPostShipments</i>
	6037872	<i>trm EAPI Sales Tracking</i>
	6037873	<i>trm EAPI Product Add. Info</i>
	6037874	<i>trm EAPI Product Add. Tags</i>
	6037875	<i>trm EAPI Product Attributes</i>
	6037876	<i>trm EAPI Product Carelabel</i>
	6037877	<i>trm EAPI Product Collection</i>
	6037878	<i>trm EAPI Product Composition</i>
	6037879	<i>trm EAPI Product Description</i>
	6037880	<i>trm EAPI Product Dimension</i>
	6037881	<i>trm EAPI Product Item</i>
	6037882	<i>trm EAPI Product Measure Chart</i>
	6037883	<i>trm EAPI Product Price</i>
	6037884	<i>trm EAPI Products</i>
	6037885	<i>trm EAPI Product Stat. Group</i>
	6037886	<i>trm EAPI Product Sust.ability</i>
	6037887	<i>trm EAPI Product VarDim</i>
	6037888	<i>trm EAPI VarDim Types</i>
	6037889	<i>trm EAPI VarDim Options</i>
	6037890	<i>trm EAPI Item Images</i>
	6037892	<i>trm EAPI Master Images</i>
	6037894	<i>trm EAPI Product Images</i>
	6037895	<i>trm EAPI SalesQuote Comments</i>
	6037896	<i>trm EAPI SalesOrder Comments</i>
	6037897	<i>trm EAPI SalesCrMemo Comments</i>
	6037898	<i>trm EAPI SalesInvoice Comments</i>
	6037899	<i>trm EAPI ReturnOrder Comments</i>
	6037900	<i>trm EAPI SalesDoc Comments</i>
	6037901	<i>trm EAPI SalesQuote VDOs</i>
	6037902	<i>trm EAPI SalesOrder VDOs</i>
	6037903	<i>trm EAPI SalesInvoice VDOs</i>
	6037904	<i>trm EAPI Return Order VDOs</i>
	6037905	<i>trm EAPI SalesCrMemo VDOs</i>
	6037906	<i>trm EAPI SalesDocument VDOs</i>
	6037907	<i>trm EAPI Item References</i>
	6037908	<i>trm EAPI Master References</i>
	6037909	<i>trm EAPI Product References</i>
	6037910	<i>trm EAPI Item Choicelist</i>
	6037911	<i>trm EAPI Master Choicelist</i>
	6037912	<i>trm EAPI Product Choicelist</i>
	6037913	<i>trm EAPI VarDim Combinations</i>
	6037914	<i>trm EAPI Countries/Regions</i>
	6037915	<i>trm EAPI Positive Negative</i>

The Codeunits	6036860	<i>trm Integration Handling</i>
	6036862	<i>trm Integration Sales Handling</i>
	6036863	<i>trm Integration Handling Impl.</i>
	6036864	<i>trm Integration Proc. SalesDoc</i>
	6036865	<i>trm Integration Post SalesDoc</i>

The Codeunits	6036866	<i>trm Integration Jnl Posting</i>
	6036867	<i>trm Integration Job Runner</i>
	6036870	<i>trm Integration Data Container</i>
	6036871	<i>trm Integration Parameters</i>
	6036872	<i>trm Integration Agent</i>
	6036873	<i>trm EAPI Request Handling</i>
	6036874	<i>trm EAPI Request Task Handling</i>
	6036875	<i>trm EAPI Request Helper</i>
	6036876	<i>trm EAPI Request Default</i>
	6036877	<i>trm EAPI Request From Table</i>
	6036878	<i>trm EAPI Request Data Helper</i>
	6036879	<i>trm EAPI Request Task Helper</i>
	6036880	<i>trm EAPI Request Task Default</i>
	6036899	<i>trm EAPI Request Authenticator</i>
	6037801	<i>trm EAPI Handling</i>

The Enum	6036840	<i>trm Integration Type</i> Implementation: Codeunit trm Integration Agent
	6036841	<i>trm Integration Data Type</i>
	6036842	<i>trm Integration Setup Line 1</i>
	6036843	<i>trm Integration Setup Line 2</i>
	6036844	<i>trm Integration Setup LineType</i>
	6036845	<i>trm Integration Cust. Handling</i>
	6036846	<i>trm Integration Product Type</i>
	6036847	<i>trm Integration Sales Doc Type</i>
	6036850	<i>trm EAPI Request</i>
	6036851	<i>trm EAPI Request Auth Type</i>
	6036852	<i>trm EAPI Request Task</i>

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.

The **Yellow** highlighted objects are based on the Standard BC API objects but are extended with TRIMIT data fields.

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**.

Tell me what you want to do

Integration

Go to Pages and Tasks [Show all \(11\)](#)

- > Integrations List Lists
- > Integration Table Mappings Lists
- > Integration Sales Documents Lists

Integrations List

+ New Manage Sales Documents Posted Sales Documents Actions Automate Fewer options

Code ↑	Description	Integration Type
DEFAULT	Default B2C Integration	Default

Overview

Documents

1	0	0
Open Sales Documents	Sales Documents With Errors	Posted Sales Documents

Products

9	0	9
Exported Products	Exported Items	Exported Masters

Setup

3	3	2
Language Setups	Currency Setups	Customer Setups
4	2	0
Customer Template Setups	Location Setups	Integration Item Setups
0	0	
Additional Data Setups	External API Request Setups	

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

Integration Overview

DEFAULT

Category List Refresh Products Exported Products Setup Request Setup Actions Automate Fewer options

General

Code: DEFAULT User Setup ID: ADMIN

Description: Default B2C Integration Integration Type: Default

Allowed Services

Allow Product Export	Allow Posted Shipment Export
Allow Inventory Export	Allow Posted Sales Invoice Export
Allow Customer Export	Allow Posted Return Receipt Export
Allow Prices and Discounts Export	Allow Posted Sales Credit Memo Export
Allow Sales Quote Import	
Allow Sales Order Import	
Allow Sales Invoice Import	
Allow Sales Return Order Import	
Allow Sales Credit Memo Import	

Overview

Documents

1	0	0
Open Sales Documents	Sales Documents With Errors	Posted Sales Documents

Products

9	0	9
Exported Products	Exported Items	Exported Masters

Setup

3	3	2
Language Setups	Currency Setups	Customer Setups
4	2	0
Customer Template Setups	Location Setups	Integration Item Setups
0	0	
Additional Data Setups	External API Request Setups	

In this page, you will see the overview of the settings of the **Integration Setup**, which can be maintained by clicking **Setup**.

FastTab General

Integration Setup

DEFAULT

Category List Sales Documents Posted Sales Documents Actions Related Automate

General

Code DEFAULT User Setup ID ADMIN

Description Default B2C Integration Integration Type Default

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.

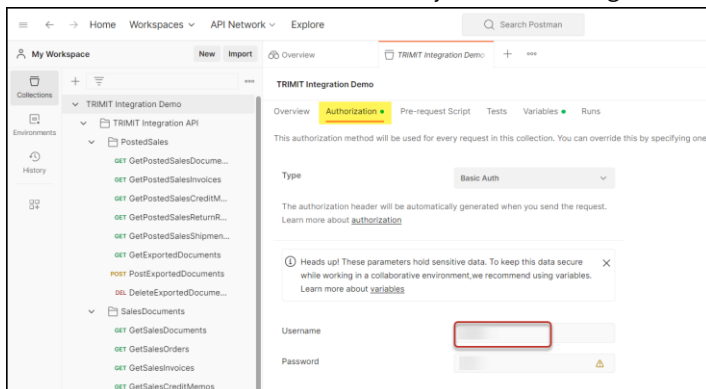
Description

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

Products

Allow Product Export

Category Structure

AGENT_W_TOPS

Use Flat Item Only

Log Exported 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 **Actions**, **Products**, [Exported Products](#).

FastTab Inventory

Inventory

Allow Inventory Export

Inventory Calculation Method

Available

Inventory Profile Setup Code

INV-SALES

Export Future Deliveries

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																							New	Edit List			
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		☐	☒	☒	☒	☐	☐	☐	☐	☒	☐	☒	☐	☐	☒	☐	☐	☐	☐	BLUE							
B2C		☐	☒	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐								
INVENTORY		☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐								
INVPUR		☐	☒	☐	☐	☐	☐	☐	☐	☐	☒	☐	☒	☒	☒	☐	☒	☐	☒								
INV-SALES		☐	☒	☒	☒	☐	☒	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐								
INVSALPU		☐	☒	☒	☒	☐	☒	☐	☒	☒	☐	☐	☐	☒	☒	☐	☒	☐	☐								
INVSALPUR		☐	☒	☒	☒	☐	☒	☐	☒	☒	☒	☒	☐	☒	☒	☐	☒	☐	☒								
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 Postman, it might then look as follows:

TRIMIT Integration / TRIMIT Integration API / GetInventories

GET {{dockerURL}}/({apiURL})/companies({companyid})/inventories?filter=itemNo eq '20002538'

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
☒ \$filter	itemNo eq '20002538'	
Key	Value	Description

Body Cookies Headers (13) Test Results

Status: 200 OK Time: 137 ms Size: 966 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "@odata.context": "http://bc23latest:7848/BC/api/trimit/integration/v1.1/$metadata#companies(ebe605d9-069a-ee11-be3c-6045bdac9fe7)/inventories",
3   "value": [
4     {
5       "@odata.etag": "W/\\\"JzIwOzE0MTkyNDE2OTMxODUyMDcxMTk0MTswMDsn\\\"",
6       "itemNo": "20002538",
7       "locationCode": "BLUE",
8       "date": "2024-01-11",
9       "availableQty": 4
10    },
11    {
12       "@odata.etag": "W/\\\"JzIwOzE0MTkyNDE2OTMxODUyMDcxMTk0MTswMDsn\\\"",
13       "itemNo": "20002538",
14       "locationCode": "BLUE",
15       "date": "2024-03-14",
16       "availableQty": 6
17    }
18  ]
19 }
```

FastTab Customers

Customers

Allow Customer Export ☒

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

Sales Prices

Allow Prices and Discounts Export ☒

Price and Discount Calculation Cust. No.

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**.

Sales & Receivables Setup

Sales & Receivables Setup

Customer Groups Payments More options

Customer

General

Order Creation Date Statistics

Standard Customer

Summary Invoice

Filter Summary Invoice

Summary Invoicing Controlled by

Generate Summary Order by Batch ☐

Blanket Order

Blanket Sales Order Search

Blanket Sales Order Search on Locat... ☐

WARNING: no blanket order found

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

Sales Documents

General Setup

Default Order Type

Default Sales Channel Code

No. of Processing Retries

Document Numbering Source

Freeze Unit Price ☒

Freeze Line Discount ☒

Customer Setup

Customer Handling

Use Customer From Incoming Documents ☐

Customer Sell-to Group

Default Customer Nos.

Sales Quote Setup

Allow Sales Quote Import ☒

Sales Quote Nos.

Auto Create Sales Quote ☒

Auto Release Sales Quote ☐

Sales Order Setup

Allow Sales Order Import ☒

Sales Order Nos.

Auto Create Sales Order ☒

Auto Release Sales Order ☒

Auto Create Pick Document From Sales Or... ☐

Auto Post Shipment and Invoice ☐

Sales Invoice Setup

Allow Sales Invoice Import ☐

Sales Invoice Nos.

Auto Create Sales Invoice ☐

Auto Post Sales Invoice ☐

Sales Return Order Setup

Allow Sales Return Order Import ☒

Sales Return Order Nos.

Auto Create Sales Return Order ☒

Auto Release Sales Return Order ☐

Auto Post Receipt and Credit Memo ☐

Sales Credit Memo Setup

Allow Sales Credit Memo Import ☐

Sales Credit Memo Nos.

Auto Create Sales Credit Memo ☐

Auto Post Sales Credit Memo ☐

Reversal Setup

Enable Reversal of Document ☐

Reversal Type

Reversal Order Action

Reversal Return Action

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

General Setup

General Setup

Default Order Type

B2C_FASH

▼

Default Sales Channel Code

B2C

▼

No. of Processing Retries

3

Document Numbering Source

Default No. Series

▼

Freeze Unit Price

Freeze Line Discount

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:

Default No. Series	You can enter the No. Series in the Integration Setup for the different Sales Document Types.
Sales & Receivables No. Series	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. Sales & Receivables Setup Customer Groups Payments More options Number Series Customer Nos. CUST ▼ Quote Nos. S-QUOTE ▼ Blanket Order Nos. S-BLK ▼ Order Nos. S-ORDER ▼ Return Order Nos. S-RETORD ▼ Invoice Nos. S-INV ▼ Posted Invoice Nos. S-INV+ ▼ Credit Memo Nos. S-CR ▼ Posted Credit Memo Nos. S-CR+ ▼ Posted Shipment Nos. S-SHPT ▼ Posted Return Receipt Nos. S-RCPT ▼ Reminder Nos. S-REM ▼ Issued Reminder Nos. S-REM+ ▼ Canceled Issued Reminder Nos. ▼ Fin. Chrg. Memo Nos. S-FIN ▼ Issued Fin. Chrg. M. Nos. S-FIN+ ▼ Canceled Issued Fin. Charge Memo ... ▼ Posted Preprmt. Inv. Nos. ▼ Posted Preprmt. Cr. Memo Nos. ▼ Direct Debit Mandate Nos. ▼ Summary Order Nos. S-SUMORD ▼ Campaign Discount Nos. CAMPDISC ▼ Complaint Order Nos. COMPLAINT ▼ Item Replacement Nos. REPLACE ▼ Consignment Journal Nos. CONS ▼ Gift Card Batch Nos. GCBATCH ▼
Integration Document No.	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.

Freeze Unit Price

If set to Yes, the field **Freeze Unit Price** in the Integration Sales Document Lines (Table **6037028 trm Sales Import Journal Line**) from an imported Sales Document will be set to Yes, and therefore also in the regular Sales Document that will be created when posting the Integration Sales Document, so you are not able to change the **Unit Price** from an imported Sales Document – unless you deliberately change the **Freeze Unit Price** in the created regular Sales Document Line manually.

Freeze Line Discount

If set to Yes, the field **Freeze Line Discount** in the Integration Sales Document Lines (Table **6037028 trm Sales Import Journal Line**) from an imported Sales Document will be set to Yes, and therefore also in the regular Sales Document that will be created when posting the Integration Sales Document, so you are not able to change the **Line Discount %** from an imported Sales Document – unless you deliberately change the **Freeze Line Discount** in the created regular Sales Document Line manually.

Customer Setup

Customer Setup

Customer Handling

Use Customer Setup

▼

Use Customer From Incoming Documents

☒

Customer Sell-to Group

▼

Default Customer Nos.

CUSTWEB

▼

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.

Sales & Receivables Setup

Customer Groups Payments More options

General Show less

Discount Posting All Discounts	Default Quantity to Ship Remainder	Allow Document Deletion Before
Credit Warnings Both Warnings	Auto Post Non-Invt. via Whse. None	Allow Multiple Posting Groups ☐
Stockout Warning ☒	Copy Comments Blanket to Order ☒	Check Multiple Posting Groups Alternative Groups
Shipment on Invoice ☒	Copy Comments Order to Invoice ☒	Ignore Updated Addresses ☐
Return Receipt on Credit Memo ☐	Copy Comments Order to Shpt. ☒	Skip Manual Reservation ☐
Invoice Rounding ☒	Copy Cmts Ret.Ord. to Cr. Memo ☒	Quote Validity Calculation
Default Item Quantity ☐	Copy Cmts Ret.Ord. to Ret.Rcpt ☒	Global Dimensions
Create Item from Item No. ☐	Allow VAT Difference ☐	Brand Code From Sales Header
Create Item from Description ☐	Calc. Inv. Discount ☐	Season Code From Sales Header
Copy Customer Name to Entries ☒	VAT Bus. Posting Gr. (Price) DOMESTIC	Job No.
Ext. Doc. No. Mandatory ☐	Exact Cost Reversing Mandatory ☐	Job Sales Order From Sales Header
Appln. between Currencies All	Check Prepmnt. when Posting ☐	Copy Line Descr. to G/L Entry ☐
Logo Position on Documents No Logo	Prepmnt. Auto Update Frequency Never	Disable Search by Name ☒
Default Posting Date Work Date		Link Doc. Date to Posting Date ☒

User Customer From Incoming Documents

Setting this field to Yes, means that for an imported Sales Document, the **Sell-to Customer No.** from the Import will be used if it exists. But if it does not exist already, it will create/use the Customer No. based on the **Customer Handling**.

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.**

No. Series ✓ Saved 🔖 🔍 🔗

Search: custweb ✕ + New Edit List Delete Lines Relationships Show All More options

Code ↑	Description	Starting No.	Ending No.	Last Date Used	Last No. Used	Default Nos.	Manual Nos.	Date Order	Allow Gaps in Nos.	Implementati...
→ CUSTWEB	Customers From Web Shops +	CW00010	CW99990	5/1/2024	CW00041	☒	☒	☐	☐	Normal

Sales Quote Setup

Sales Quote Setup

Allow Sales Quote Import ☐

Sales Quote Nos.

Auto Create Sales Quote ☐

Auto Release Sales Quote ☐

Allow Sales Quote Import

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

Sales Quote Nos.

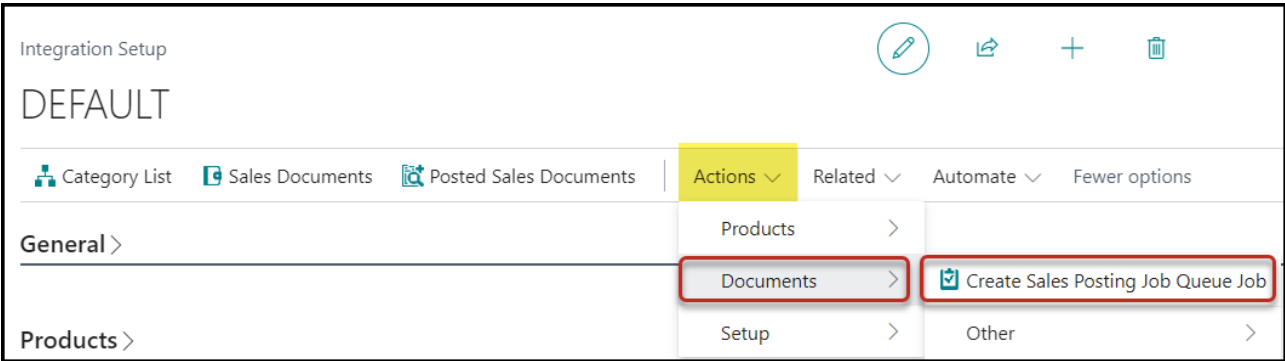
If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Quotes. Based on this **No. Series** the number for the regular **Sales Quote** will be determined.

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.

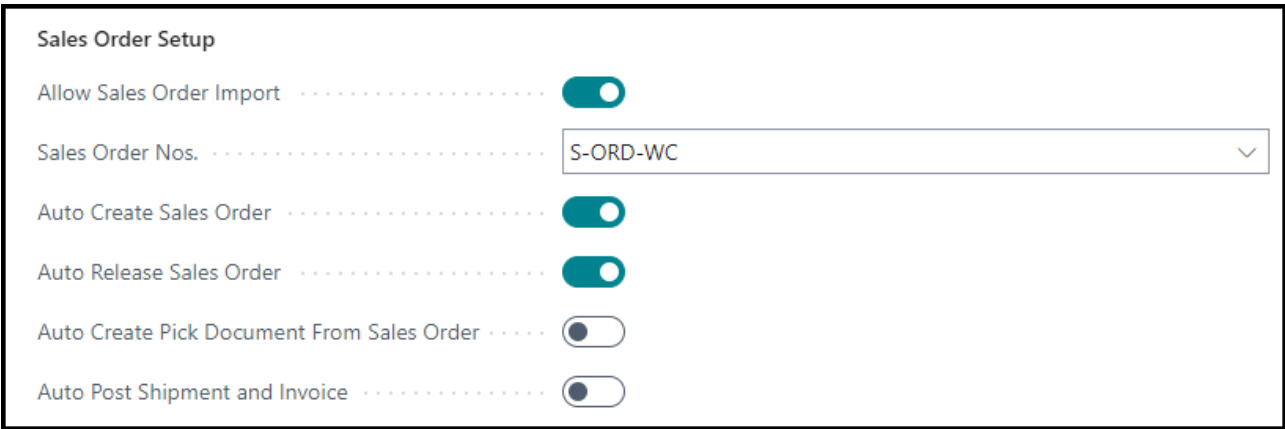
You can create the necessary **Job Queue** via **Actions, Documents, Create Sales Posting Job Queue Job**.



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

Sales Invoice Setup

Allow Sales Invoice Import

☐

Sales Invoice Nos.

Auto Create Sales Invoice

☐

Auto Post Sales Invoice

☐

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

Sales Return Order Setup

Allow Sales Return Order Import ☒

Sales Return Order Nos.

Auto Create Sales Return Order ☒

Auto Release Sales Return Order ☒

Auto Post Receipt and Credit Memo ... ☒

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.

Sales Credit Memo Setup

Sales Credit Memo Setup

Allow Sales Credit Memo Import ☒

Sales Credit Memo Nos.

Auto Create Sales Credit Memo ☒

Auto Post Sales Credit Memo ☒

Allow Sales Credit Memo Import

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

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.

Reversal Setup

Reversal Setup

Enable Reversal of Document ☒

Reversal Type

Reverse on Document Total

Reversal Order Action

Order is reversed to Return Order

Reversal Return Action

Return is reversed to Order

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 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.**

The image shows two screenshots of SAP tables. The top screenshot is the 'Sales Documents' table, showing a single row for a 'Return Order' with document number 1004, sell-to customer CW0042, and import status 'Open'. The bottom screenshot is the 'Posted Documents' table, showing a list of documents. The row for an 'Invoice' with document number 1004 and sell-to customer CW0042 is highlighted with a red box.

Document Type	No.	Sell-to Customer No.	Sell-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.	Import Status
Return Order	1004	CW0042				B2C	1029	Open

Document Type	No.	Sell-to Customer No.	Sell-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.
Order	0100013	CW0042				B2C	1027
Order	1002	CW0042				B2C	1025
Order	1003	CW0042				B2C	1028
Invoice	1004	CW0042				B2C	1029
Invoice	102222	CW0042				B2C	1022
Invoice	102223	CW0042				B2C	1023
Invoice	102224	CW0042				B2C	1024

Not in the 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**.

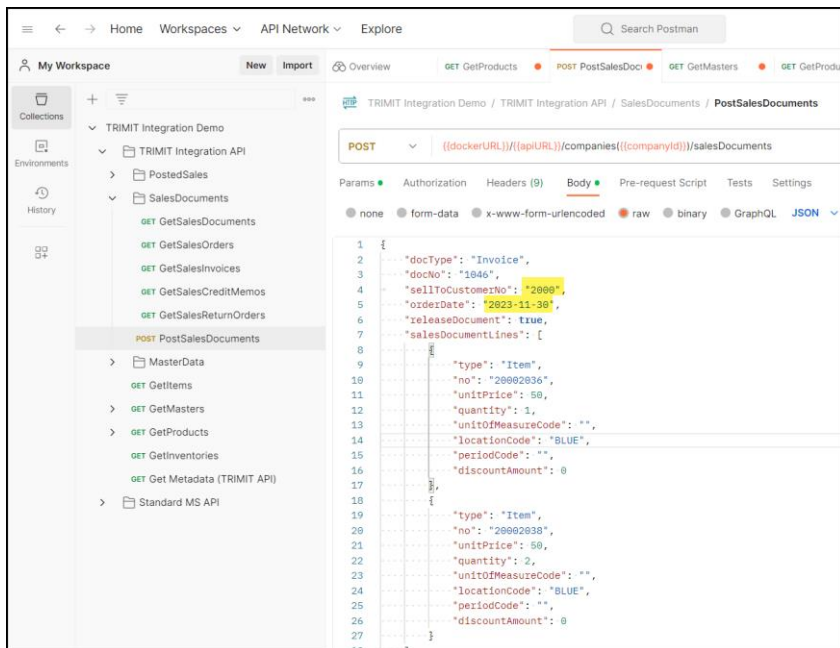
If you do not want to have a regular Sales Invoice created for an Integration Sales Invoice, you can set this field to Yes.

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 $(1 \times 50) + (2 \times 50) = 150$. This would result in a General Journal Line:

General Journals															
Batch Name: IMPORTED															
Manage Home Request Approval Line Incoming Document Page More options															
Post Get Standard Journals... Renumbr Document Numbers Reconcile Apply Entries...															
Posting Date	VAT Date	Document Type	Document No.	Account Type	Account No.	Account Name	Description	Curre. Code	EU 3-Party Trade	Gen. Posting Type	Gen. Bus. Posting Group	Gen. Prod. Posting Group	Amount	Amount (LCY)	Bal. Account No.
11/30/2023	11/30/2023	Invoice	102225	Customer	2000	The Fashion Store UK	Integration Document 102225						150.00	150.00	G/L Account

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

Posted Sales Documents	
Allow Posted Shipment Export	☒
Allow Posted Sales Invoice Export	☒
Allow Posted Return Receipt Export	☒
Allow Posted Sales Credit Memo Ex...	☒

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

The screenshot shows the 'Integration Setup' interface. At the top, there are icons for edit, share, add, and delete. Below this is a 'DEFAULT' section. A navigation bar contains 'Category List', 'Sales Documents', 'Posted Sales Documents', 'Actions', 'Related', 'Automate', and 'Fewer options'. The 'Actions' menu is open, showing a list of options: 'Products', 'Documents', 'Setup', 'Edit Category Lines', 'Category Management', 'Refresh Products', 'Exported Products', and 'Other'. The 'Products' option is highlighted with a red box. The 'Edit Category Lines', 'Category Management', 'Refresh Products', and 'Exported Products' options are grouped together with a red box. The 'Other' option is also highlighted with a red box. At the bottom right, there is a 'Category List' button and a 'Use short menu' checkbox.

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. Therefore, 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.

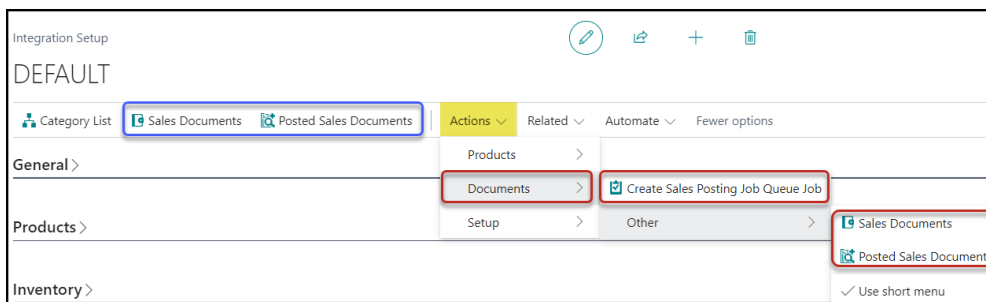
The exported Products will then be written in Table **6036842 trm Integration Exp. Product**:

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 the Job Queue 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

Job Queue Entry Card

Codeunit · 6036865 · trm Integration Post SalesDoc

To edit the job queue entry, you must first choose the Set On Hold action.

Set On Hold

HomeJob QueueActionsAutomateFewer options

Set Status to Ready

Set On Hold

Restart

Run once (foreground)

Show Error

General

Object Type to RunCodeunit

Object ID to Run6036865

Object Caption to Runtrm Integration Post SalesDoc

DescriptionCreated for automatic processing of TRIMIT Integration Sal...

Earliest Start Date/Time12/4/2023 9:51 AM

Job Timeout12 hours

StatusReady

Show more

Recurrence

Recurring Job

Run on Mondays

Run on Tuesdays

Run on Wednesdays

Run on Thursdays

Run on Fridays

Run on Saturdays

Run on Sundays

Next Run Date Formula

Starting Time

Ending Time

No. of Minutes between Runs2

Inactivity Timeout Period5

Sales Documents

Clicking **Sales Documents** in the **Integrations List** or **Integration Setup** will open the page with the Integration Sales Documents, which will show you only the data that was imported.

A screenshot of the 'Integrations List' header bar. It features a yellow 'Integrations List:' label, a dropdown menu set to 'All', and several icons for search, save, and actions. The 'Sales Documents' button, which includes a document icon, is highlighted with a red rectangular box. Other buttons include '+ New', 'Delete', 'Posted Sales Documents', and 'Actions'.

Integration Setup ✎ 🔗 + 🗑️

DEFAULT

🏠 Category List
📄 Sales Documents
📄 Posted Sales Documents
Actions ▾
Related ▾
Automate ▾

Sales Documents									
Integration Sales Documents									
Document Type	No.	Self-to Customer No.	Self-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.	Import Status	
Order	WOC000015	2000				B2C	KBI007	Open	
Order	WOC000016	2000				B2C	KBI008	Open	
Order	WOC000017	CW00010				B2C	KBI009	Open	
Return Order	1004	CW00010				B2C	KBI010	Open	

Clicking on the **No.** will open the **Integration Sales Document** Card.

Sales Documents			
Integration Sales Documents			
Document Type	No.	Sell-to Customer No.	Sell-to Customer No.
Order	WOC000015	2000	
Order	WOC000016	2000	
Order	WOC000017	CW00010	
Return Order	1004	CW00010	

Clicking **Related, Payment**, will show you the Payment that might also have been imported.
Clicking **Related, Extra Order Info**, will show you the Extra Order Info that was also imported.

Order Extra Info - Order - WOC000015

Open Document Page

General

Init. Document Type Order Init. No. WOC000015 Order Date 12/20/2024

Delivery

Delivery Method Parcel Shop ID Wrapping Ship-to Code Ship-to Country/Region Code GB Ship-to Phone Direct Consumer Delivery Order Confirmation E-Mail

Customer

Bill-to Customer No. Sell-to Customer No. Sell-to E-Mail Sell-to Phone No.

InStore B2C

InStore B2C Order InStore Payment Amount 0.00 InStore Payment Completed Referrer Code InStore Payment Currency InStore Payment Received By

Signature

Signature Name Signature

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

Sales Documents

Integration Sales Documents

Manage Related Posting

Related

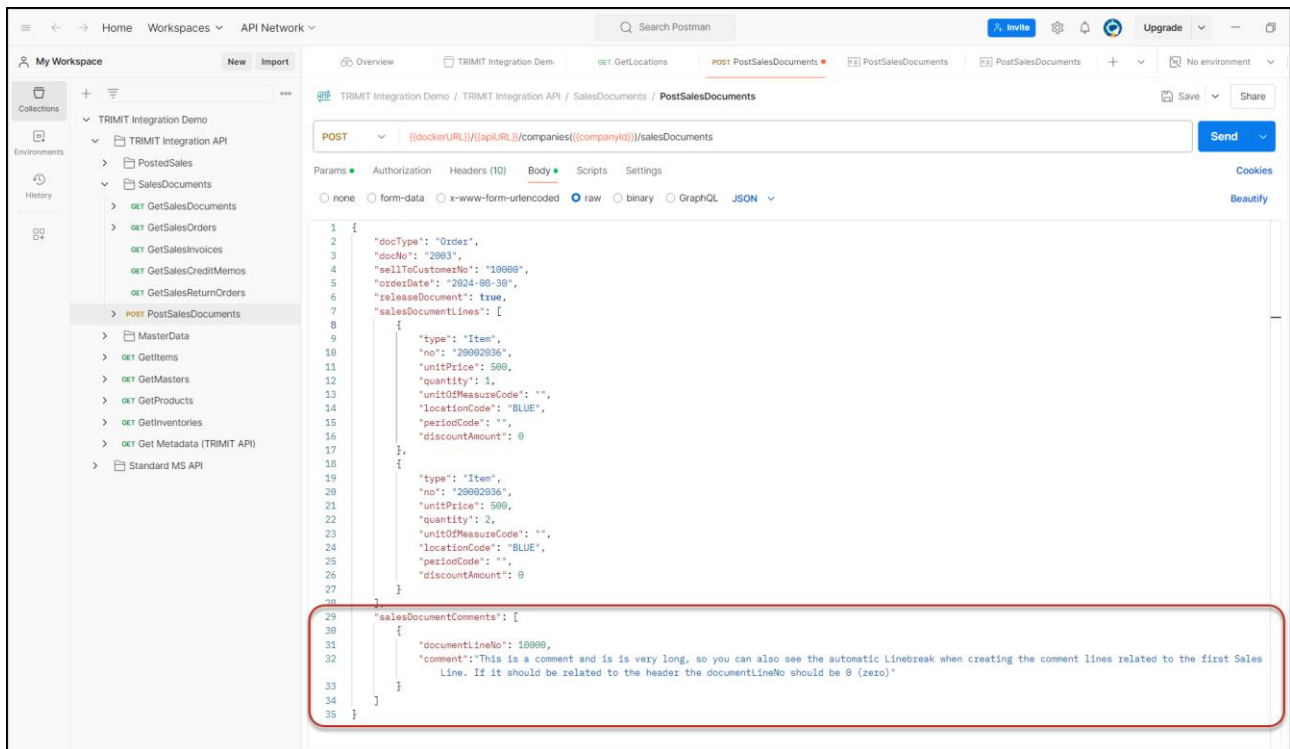
Post Post All Test Order

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

Integration Sales Document Card

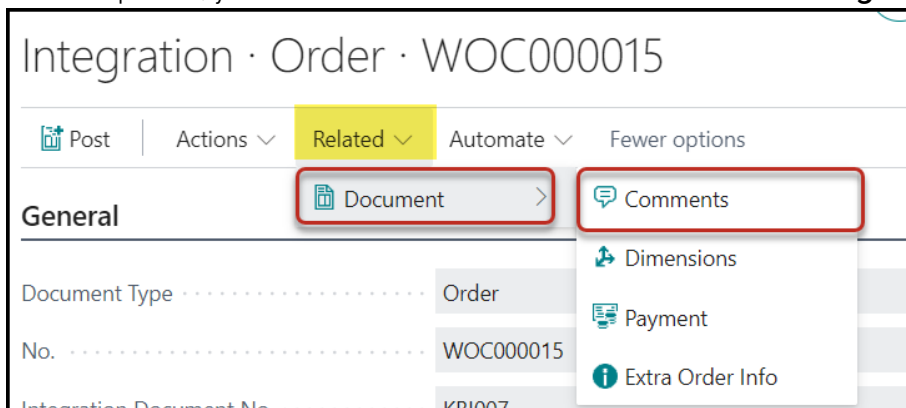
In the **Integration Sales Document** Card, you can see the details of the Integration Sales Document that was imported.

It is also possible to import **Comment** Lines for the Integration Sales Document Header, and for the Integration Sales Document Lines.

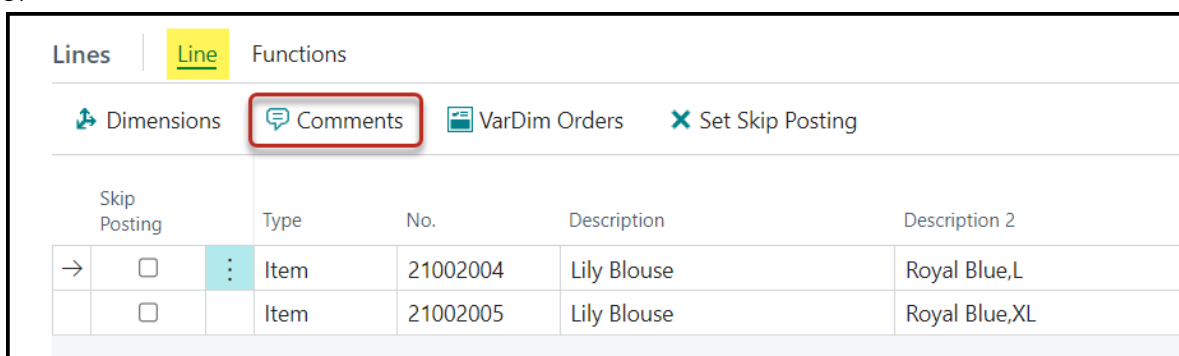


With **salesDocumentComments** you can add Comments to the Integration Sales Document Header or Integration Sales Document Line. If **documentLineNo** is 0 it will be Comments for the Header, if the **documentLineNo** <>0 it refers to the corresponding Line that you will be importing.

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



The **Comments**, **Dimensions** and **Payment** are not shown by default; you can add them via **Personalize**. Or



All the **Line** functions are not shown by default, you can add them via **Personalize**.

Sales Comment List - Integration · Order · WOC000015 · 10000 · 10000

Manage

Date	Comment
→	This is a comment and is is very long, so you can also see the automatic Linebreak when creating the comment lines related to the first Sales Line. If it should be related to the header the documentLineNo should be 0 (zero)

Close

These Comment Lines will also be copied to the actual Sales Document after Posting.
 For this example: in the Sales Order WOC000015 on the Item Line:

Sales Order

WOC000015 · The Fashion Store UK

×

Start validating data in documents and journals while you work. Messages are shown in the Document Check FactBox.

Enable this

Home

Prepare

Print/Send

Request Approval

Order

More options

Post...

Reopen

Create Warehouse Shipment

Create Inventory Put-away/Pick...

Archive Document

General

Lines

Manage

Line

Order

Functions

Item Availability by

Related Information

Expand Matrix	Type	No.	Description	Description
→	Matrix	2100	Lily Blouse	
→	Item	21002004	Lily Blouse	Royal
	Item	21002005	Lily Blouse	Royal

Master Card

Bottleneck Capacity

Order Line Picture

Reservation Entries

Item Tracking Lines

Select Item Substitution

Dimensions

Comments

Item Charge Assignment

Posted Sales Documents

After an Integration Sales Document has been posted (manually or by the Job Queue), you can see it in the page **Posted Sales Documents** by clicking **Posted Sales Documents** in the **Integrations List** or **Posted Integration Setup**, 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.**

WHITE PAPER: EXTERNAL API

Date: March 27, 2025

© TRIMIT Group. All rights reserved.

Page 33

Integrations List: All

+

New

🗑️>Delete

📄>Sales Documents

📄>Posted Sales Documents

Actions

Code ↑	Description	Integration Type
DEFAULT	⋮ Default B2C Integration	Default

Integration Setup

🔍🔗+

DEFAULT

📄>Category List

📄>Sales Documents

📄>Posted Sales Documents

Actions

Related

Automate

Posted Documents

Integration Sales Documents 🔍🔗ManageRelatedRelatedAutomateFewer options

Document Type	No. ↑	Sell-to Customer No.	Sell-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.
Order	⋮ WOC000008	CW000010				B2C	2003
Order	WOC000009	CW000010				B2C	KBI001
Order	WOC000010	CW000010				B2C	KBI002
Order	WOC000011	CW000010				B2C	KBI003
Order	WOC000012	CW000010				B2C	KBI004
Order	WOC000013	CW000010				B2C	KBI005
Order	WOC000014	2000				B2C	KBI006
Order	WOC000015	2000				B2C	KBI007

Posted Documents

Integration Sales Documents 🔍🔗ManageRelatedRelatedAutomateFewer options

Document Type	No. ↑	Sell-to Customer No.	Sell-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.
Order	⋮ WOC000008	CW000010				B2C	2003
Order	WOC000009	CW000010				B2C	KBI001
Order	WOC000010	CW000010				B2C	KBI002

Payment

Other

Extra Order Info

Use short menu

Clicking **Related, Payment**, will show you the Payment that was also imported and posted.
Clicking **Related, Extra Order Info**, will show you the Extra Order Info that was also imported and posted.
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 on the Integration Setup

Integration Setup

DEFAULT

Category List

Sales Documents

Posted Sales Documents

Actions

Related

Automate

Few

General

Code : DEFAULT

Description : Default B2C Integration

Products >

Inventory >

Language Setup

Currency Setup

Location Setup

Customer Setup

Customer Template Setup

Item Setup

Additional Data Setup

Language Setup

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

Language Setup

✓ Saved

Search + New Edit List Delete

Language Code ↑

Name

→ DAN Danish

DEU German

ENU English

NLD Dutch

NOR Norwegian

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 the **Customer Price Group** field is not

filled, it will take the **Customer Price Group** and **Currency Code** of the **Standard Customer** in the Sales & Receivables Setup.

Currency Setup

✓ Saved



 Search

 New

 Edit List

 Delete



	Currency Code ↑		Description	Customer Price Group ↑	Description
→				RETAIL	Retail Prices
	EUR		Euro	RETAIL	Retail Prices

Location Setup

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

Location Setup

✓ Saved



 Search

 New

 Edit List

 Delete



	Location Code ↑		Description	Default
→	BLUE		BLUE	☒
	CW		Central Warehouse	☐

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





 New

 Edit List

 Delete



	Country/Region Code ↑		Name	Customer No. ↑	Name
→	GB		Great Britain	2000	The Fashion Store UK
	DE		Germany	2010	The Shoe Store Europe
	US		USA	2040	Garment Company Ltd.
				CW00010	Linda Nielsen

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

+ New

Edit List

Delete

Template No. ↑	Description	Country/Region Code ↑	Name
→ CUSTB2C	⋮	Trimit B2C Domestic Customer Template	
CUSTB2CEU		Trimit B2C Domestic EU Template	DE Germany
CUSTB2CEU		Trimit B2C Domestic EU Template	DK Denmark
CUSTB2CFOR		Trimit B2C Domestic Foreign Custome...	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

✓ Saved

Search

+

New

Edit List

Delete

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

✓ Saved

Search

New

Edit List

Delete

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 Integration Demo / TRIMIT Integration API / GetProducts

GET {{dockerURL}}/api/trimit/integration/v1.1/\$metadata#companies(3109c411-da89-ee11-b72e-97916980265b)/products

Params

Key	Value	Description
☒ \$expand	configuredFields	
☐ \$filter	productType eq 'Master'	
☒ \$filter	productNo eq '2110'	

Body

```

1 {
2   "@odata.context": "http://docbc22w1:7048/BC/api/trimit/integration/v1.1/$metadata#companies(3109c411-da89-ee11-b72e-97916980265b)/products",
3   "value": [
4     {
5       "@odata.etag": "W/\"J2Iw0ZeyNTAAMzMAH2gZHTcxM0cIMjgSNTswMDsn\"",
6       "productId": "7d25c4db-3317-ee11-b6b3-c661fd0ee9e0",
7       "productType": "Master",
8       "productNo": "2110",
9       "categoryNode": "[A01AA77C-CB07-4B18-91B2-71A0DC4849B8]",
10      "lastModifiedDateTime": "2023-06-30T10:51:40.163Z",
11      "configuredFields": [
12        {
13          "@odata.etag": "W/\"J2Iw0ZE10ThZNU5N2zc4NjEzNjY5MTY1MTswMDsn\"",
14          "id": "d7e79ef0b-4e93-4919-b8ce-090c5eeb384c",
15          "name": "Net Weight",
16          "value": "0.5"
17        },
18        {
19          "@odata.etag": "W/\"J2E50zIzNTQzHzQ40Tk4NDMwNDgSNTcxOzAwOyc\"",
20          "id": "0964cc10-3279-4820-b2bd-8a1c58cd786a",
21          "name": "Gross Weight",
22          "value": "0.55"
23        }
24      ]
25    }
26  ]
27 }

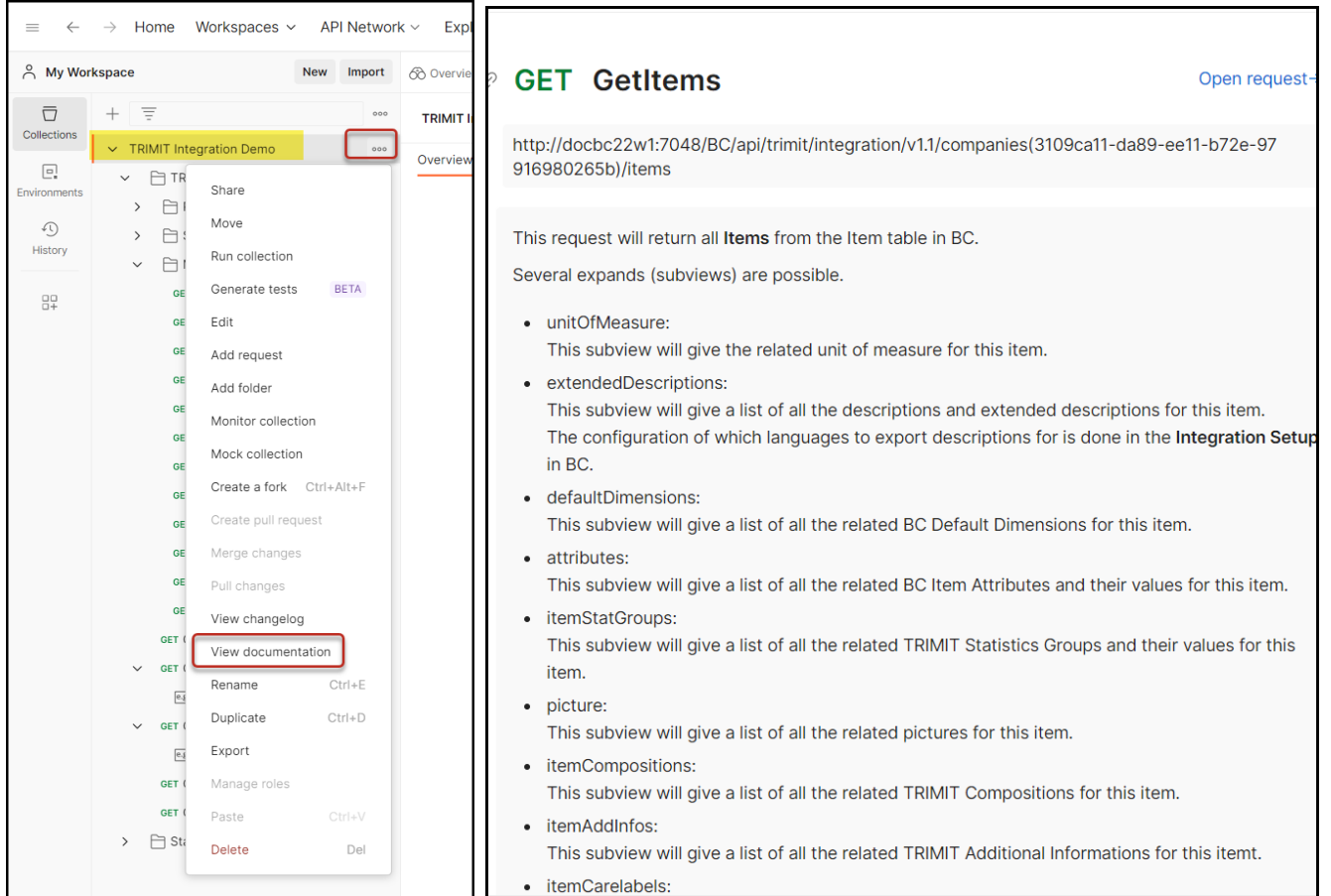
```

TRIMIT's Postman.json

We created a Postman Collection .json file for testing results of importing/exporting data.

Documentation in Postman Collection

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.



The screenshot shows the Postman interface with the 'TRIMIT Integration Demo' collection selected. A context menu is open for the 'GET GetItems' request, and the 'View documentation' option is highlighted. The right pane displays the documentation for the 'GET GetItems' request, which includes the URL, a description, and a list of expandable subviews.

GET GetItems

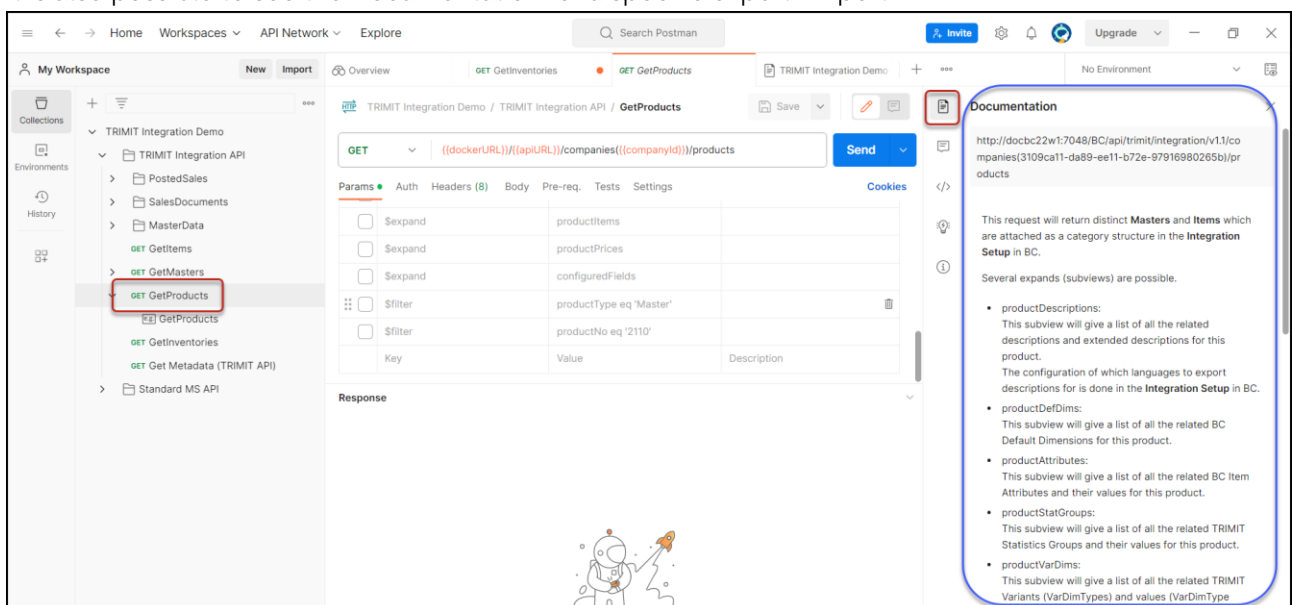
[http://docbc22w1:7048/BC/api/trimit/integration/v1.1/companies\(3109ca11-da89-ee11-b72e-97916980265b\)/items](http://docbc22w1:7048/BC/api/trimit/integration/v1.1/companies(3109ca11-da89-ee11-b72e-97916980265b)/items)

This request will return all **Items** from the Item table in BC.

Several expands (subviews) are possible.

- unitOfMeasure:**
This subview will give the related unit of measure for this item.
- extendedDescriptions:**
This subview will give a list of all the descriptions and extended descriptions for this item. The configuration of which languages to export descriptions for is done in the **Integration Setup** in BC.
- defaultDimensions:**
This subview will give a list of all the related BC Default Dimensions for this item.
- attributes:**
This subview will give a list of all the related BC Item Attributes and their values for this item.
- itemStatGroups:**
This subview will give a list of all the related TRIMIT Statistics Groups and their values for this item.
- picture:**
This subview will give a list of all the related pictures for this item.
- itemCompositions:**
This subview will give a list of all the related TRIMIT Compositions for this item.
- itemAddInfos:**
This subview will give a list of all the related TRIMIT Additional Informations for this item.
- itemCarelables:**

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



The screenshot shows the Postman interface with the 'TRIMIT Integration Demo' collection selected. A context menu is open for the 'GET GetProducts' request, and the 'View documentation' option is highlighted. The right pane displays the documentation for the 'GET GetProducts' request, which includes the URL, a description, and a list of expandable subviews.

GET GetProducts

[http://docbc22w1:7048/BC/api/trimit/integration/v1.1/companies\(3109ca11-da89-ee11-b72e-97916980265b\)/products](http://docbc22w1:7048/BC/api/trimit/integration/v1.1/companies(3109ca11-da89-ee11-b72e-97916980265b)/products)

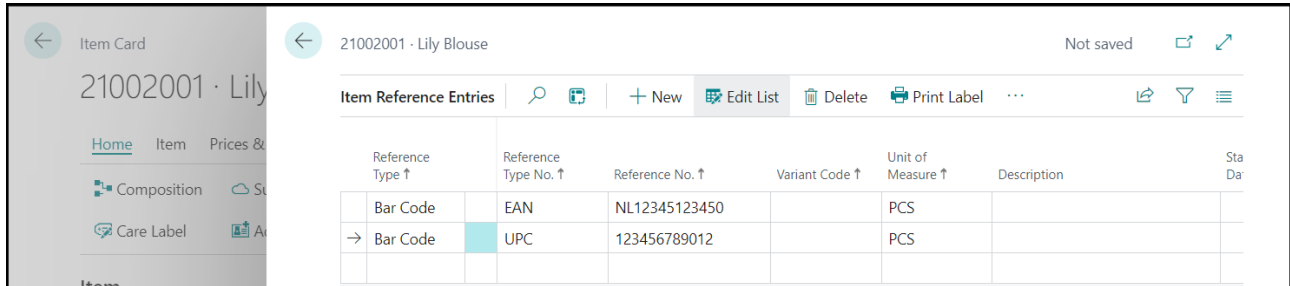
This request will return distinct **Masters** and **Items** which are attached as a category structure in the **Integration Setup** in BC.

Several expands (subviews) are possible.

- productDescriptions:**
This subview will give a list of all the related descriptions and extended descriptions for this product. The configuration of which languages to export descriptions for is done in the **Integration Setup** in BC.
- productDefDims:**
This subview will give a list of all the related BC Default Dimensions for this product.
- productAttributes:**
This subview will give a list of all the related BC Item Attributes and their values for this product.
- productStatGroups:**
This subview will give a list of all the related TRIMIT Statistics Groups and their values for this product.
- productVarDims:**
This subview will give a list of all the related TRIMIT Variants (VarDimTypes) and values (VarDimType

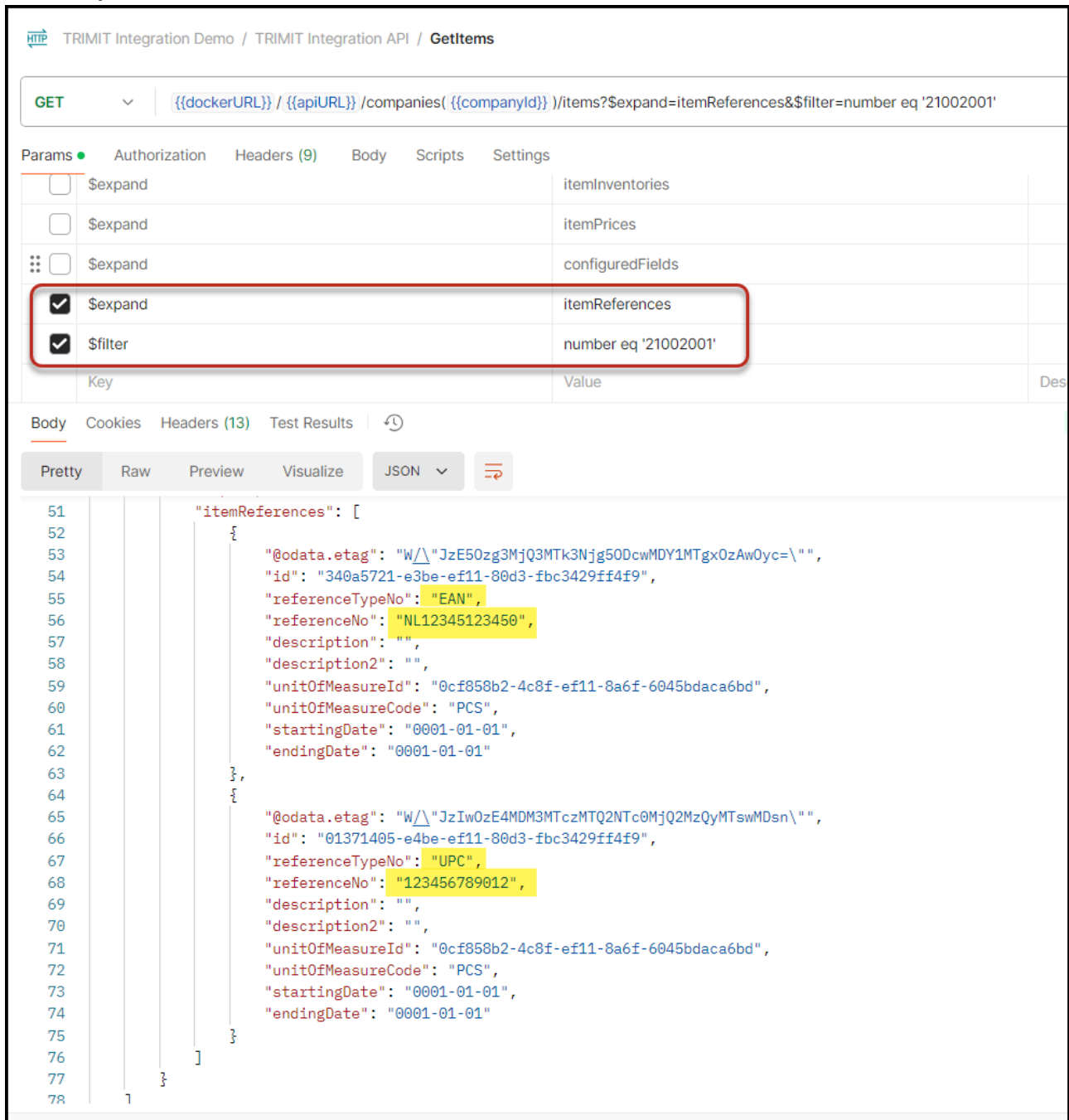
Item References

It is also possible to export the **Item References** of **Type Bar Code**. I.e.:



Reference Type ↑	Reference Type No. ↑	Reference No. ↑	Variant Code ↑	Unit of Measure ↑	Description	Sta Da
Bar Code	EAN	NL12345123450		PCS		
→ Bar Code	UPC	123456789012		PCS		

With **\$expand=itemReferences** this will result in:



TRIMIT Integration Demo / TRIMIT Integration API / GetItems

GET `{{dockerURL}} / {{apiURL}} /companies( {{companyId}} )/items?$expand=itemReferences&$filter=number eq '21002001'`

Params • Authorization Headers (9) Body Scripts Settings

☐	\$expand	itemInventories
☐	\$expand	itemPrices
☐	\$expand	configuredFields
☒	\$expand	itemReferences
☒	\$filter	number eq '21002001'

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize JSON

```
51 "itemReferences": [  
52   {  
53     "@odata.etag": "W/\\\"JzE50zg3MjQ3MTk3Njg5ODcwMDY1MTgxOzAwOyc=\\\"",  
54     "id": "340a5721-e3be-ef11-80d3-fbc3429ff4f9",  
55     "referenceTypeNo": "EAN",  
56     "referenceNo": "NL12345123450",  
57     "description": "",  
58     "description2": "",  
59     "unitOfMeasureId": "0cf858b2-4c8f-ef11-8a6f-6045bdaca6bd",  
60     "unitOfMeasureCode": "PCS",  
61     "startingDate": "0001-01-01",  
62     "endingDate": "0001-01-01"  
63   },  
64   {  
65     "@odata.etag": "W/\\\"JzIwOzE4MDM3MTczMTQ2NTc0MjQ2MzQyMTswMDsn\\\"",  
66     "id": "01371405-e4be-ef11-80d3-fbc3429ff4f9",  
67     "referenceTypeNo": "UPC",  
68     "referenceNo": "123456789012",  
69     "description": "",  
70     "description2": "",  
71     "unitOfMeasureId": "0cf858b2-4c8f-ef11-8a6f-6045bdaca6bd",  
72     "unitOfMeasureCode": "PCS",  
73     "startingDate": "0001-01-01",  
74     "endingDate": "0001-01-01"  
75   }  
76 ]  
77 }  
78 }
```


Prices

It is now also possible to filter on specific fields of the Sales Prices, to determine which Sales Prices of Items should be exported. This is initiated by **\$Filter=calculatedPrice eq true** after which you can set the special filters on the subpage for prices.

So, if the **Price and Discount Calculation Cust. No.** would be 2000, and you would have:

Item 2100

Saved

Sales Prices

+ New

Edit List

Delete

Dimension Prices

Search Table Lines

Formula Lines

More options

General

Sales Type Filter: None

Item/Master No. Filter: 2100

Sales Code Filter:

Starting Date Filter:

Item Type Filter: Master

Currency Code Filter:

Sales Type ↑	Sales Code ↑	Item No. ↑	Unit of Measure Code ↑	Minimum Quantity ↑	Currency Code ↑	Unit Price	Price Includes VAT	Dimension Prices	Starting Date ↑	Ending Date	Search Expression	Search Table	Use Search Table Result	Formula
→ Customer Price Group	B2C_CAMP	2100	PCS	0		55.20	☒	0					Default	
Customer Price Group	B2C_CAMP	2100	PCS	0	DKK	460.00	☒	0					Default	
Customer Price Group	B2C_CAMP	2100	PCS	0	EUR	87.20	☒	0					Default	
Customer Price Group	B2C_CAMP	2100	PCS	0	NOK	503.20	☒	0					Default	
Customer Price Group	B2C_CAMP	2100	PCS	0	SEK	552.00	☒	0					Default	
Customer Price Group	B2C_CAMP	2100	PCS	0	USD	91.20	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0		69.00	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0	DKK	575.00	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0	EUR	109.00	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0	NOK	629.00	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0	SEK	690.00	☒	0					Default	
Customer Price Group	RETAIL	2100	PCS	0	USD	114.00	☒	0					Default	
All Customers		2100	PCS	0		35.99	☐	0					Default	
All Customers		2100	PCS	0	EUR	45.99	☐	0					Default	

You could get:

TRIMIT Integration Demo / TRIMIT Integration API / GetItems

Save

GET

{{dockerURL}} / {{apiURL}} / companies({{companyId}}) / items? \$expand= itemPrices(\$Filter= calculatedPrice eq true and currencyCode eq 'EUR') & \$filter= number eq '21002001'

Params

Authorization

Headers (9)

Body

Scripts

Settings

☐ \$expand

ItemInventories

☐ \$expand

configuredFields

☐ \$expand

itemReferences

☒ \$filter

number eq '21002001'

☒ \$expand

itemPrices(\$Filter= calculatedPrice eq true and currencyCode eq ...

Key

Value

Description

Body

Cookies

Headers (13)

Test Results

200 OK

93 ms

1.48 KB

Save

Pretty

Raw

Preview

Visualize

JSON

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

```
{
  "includeInMRP": true,
  "mmpMethod": "Net Requirement",
  "mmpPeriod": "",
  "mmpResponsible": "",
  "itemPrices": [
    {
      "@odata.etag": "W/\"JzE50zMyMDAxMDA1Mjk0NjAyODg0NDUxOzAwOyc=\"",
      "id": "d2879dab-ed43-4ea3-97cf-dc3cfaeb4909",
      "customerPriceGroupId": "00000000-0000-0000-0000-000000000000",
      "customerPriceGroup": "",
      "currencyId": "66f558b2-4c8f-ef11-8a6f-6045bdaca6bd",
      "currencyCode": "EUR",
      "customerId": "00000000-0000-0000-0000-000000000000",
      "customerNumber": "2000",
      "unitOfMeasureId": "0cf858b2-4c8f-ef11-8a6f-6045bdaca6bd",
      "unitOfMeasure": "PCS",
      "minimumQuantity": 0,
      "campaignId": "00000000-0000-0000-0000-000000000000",
      "campaignNo": "",
      "priceDate": "2024-12-20",
      "priceIncludesVAT": false,
      "price": 45.99,
      "discountPercentage": 0,
      "externalName": "Calculated price",
      "calculatedPrice": true
    }
  ]
}
```

VarDim Order

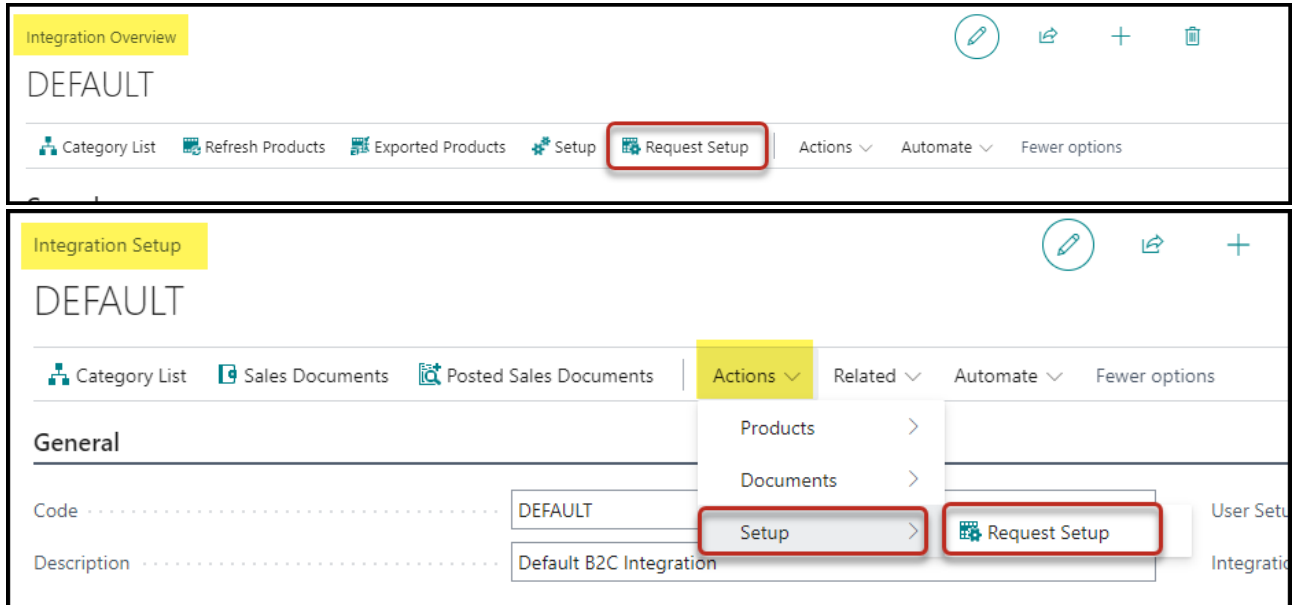
It is also possible to have **VarDim Order** Information imported for a Sales Line, by using **"salesDocumentVarDimOrders"** and then submitting the specific **documentLineNo** it belongs to with a specific **type**, **varDimType** and **Value**.

The screenshot shows the Postman interface for a POST request to the TRIMIT Integration API. The request is to the endpoint `POST {{dockerURL}} / {{apiURL}} / companies( {{companyId}} ) / salesDocuments`. The request body is a JSON object representing a sales document with three lines. Each line has a `documentLineNo` and a `salesDocumentVarDimOrders` array. The first line is an "Item" with a quantity of 2. The second line is a "Variant" with a value of "03". The third line is a "Variant" with a value of "22".

```
2  "docType": "Order",
3  "docNo": "KBI101",
4  "sellToCustomerNo": "3000",
5  "orderDate": "2025-01-10",
6  "releaseDocument": true,
7  "salesDocumentLines": [
8    {
9      "lineNo": 10000,
10     "type": "Item",
11     "no": "3100",
12     "unitPrice": 719,
13     "quantity": 2,
14     "unitOfMeasureCode": "",
15     "locationCode": "BLUE",
16     "periodCode": "",
17     "discountAmount": 0
18   },
19 ],
20
21   "salesDocumentVarDimOrders": [
22     {
23       "documentLineNo": 10000,
24       "type": "Variant",
25       "varDimType": "COLOR_LAM",
26       "value": "03"
27     },
28     {
29       "documentLineNo": 10000,
30       "type": "Variant",
31       "varDimType": "LEGS",
32       "value": "22"
33     },
34     {
35       "documentLineNo": 10000,
36       "type": "Variant",
37       "varDimType": "SIZE_TD",
38       "value": "23"
39     }
40   ]
41 }
```

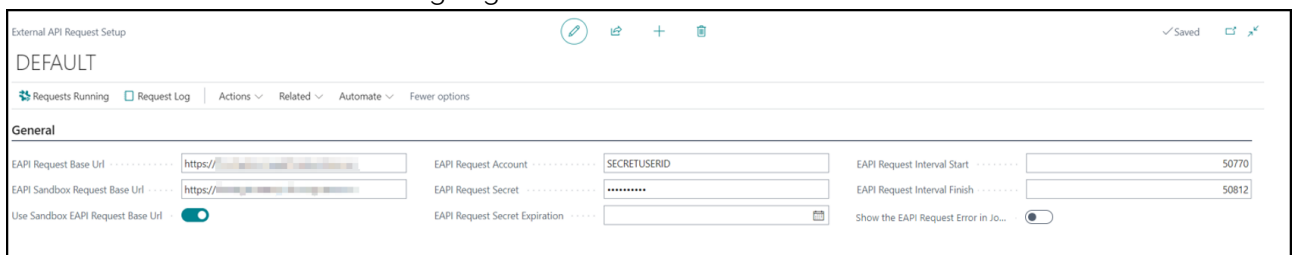
Requests

With the **Requests** you can determine to which system and what information you want to export from TRIMIT. Therefore, you need to create a **Request Setup**, which can be done via the **Integration Overview** or the **Integration Setup**:



Request Setup

With the **Request Setup**, you are able to fill the details for all the data requests to an external System with data from TRIMIT – so the outgoing Data from TRIMIT.



EAPI Request Base Url

Common path to the API that we want to call.

EAPI Sandbox Request Base Url

Common path to the API in a Sandbox that we want to call.

Use Sandbox EAPI Request Base Url

A checkmark makes sure, that the call takes place to a Sandbox of the external System via the **EAPI Sandbox Request Base Url**. Otherwise, the call takes place via the **EAPI Request Base Url**.

EAPI Request Account

The User ID for the connection: the *Client ID*.

EAPI Request Secret

The Password for the connection: the *Client Secret*.

EAPI Request Secret Expiration

Expiration Date and Time for the **EAPI Request Secret**.

EAPI Request Interval Start

Number of the first Request Enum for this integration.

EAPI Request Interval Finish

Number of the last Request Enum for this integration.

Show the EAPI Request Error in Job Queue

If there is an error in the EAPI Request, it will stop the Job Queue.

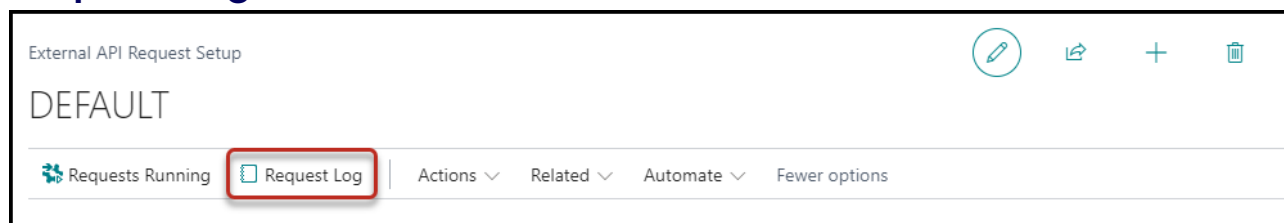
Without a checkmark, it will simply skip the error and continue to the next Request.

Requests Running



This will show a list of all running Requests.

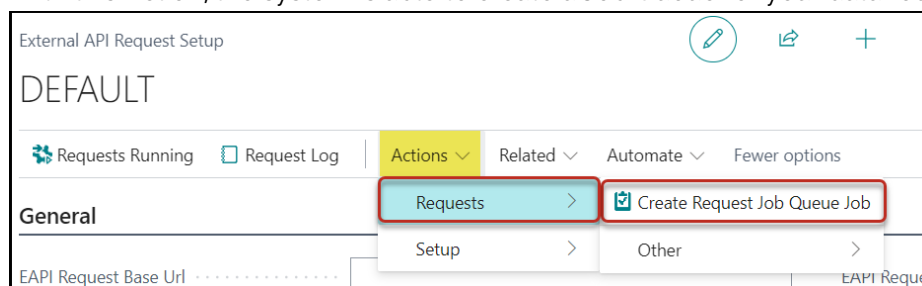
Request Log



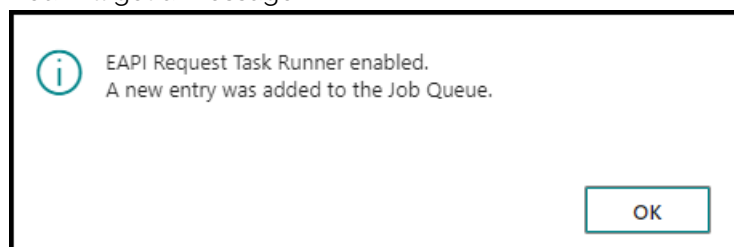
This will show a log of all Requests that have been running.

Actions, Requests, Create Request Job Queue Job

With this Action, the system is able to create a Job Queue for your data requests:



You will get a message:



And in the **Job Queue Entries** you will see a new Entry:

Job Queue Entry Card

Codeunit · 6036874 · trm EAPI Request Task Handling

Home

Job Queue

Actions

Automate

Fewer options

Set Status to Ready

Set On Hold

Restart

Run once (foreground)

Show Error

General

Show less

Object Type to Run

Codeunit

Object ID to Run

6036874

Object Caption to Run

trm EAPI Request Task Handling

Description

Created for automatic running of TRIMIT EAPI Requests

Parameter String

Job Queue Category Code

Priority

Normal

User ID

ADMIN

Maximum No. of Attempts to Run

3

Rerun Delay (sec.)

0

Last Ready State

12/20/2024 10:28 AM

Earliest Start Date/Time

12/30/2024 8:47 AM

Expiration Date/Time

Job Timeout

12 hours

Status

On Hold

Related, Request

Via **Related, Request**, you are able to see all the individual Request for exporting Data that have been developed for this Integration:

External API Request Setup

DEFAULT

Requests Running

Request Log

Actions

Related

Automate

Fewer options

General

Requests

External API Requests

✓ Saved

+ New

Edit List

Delete

Actions

Automate

Fewer options

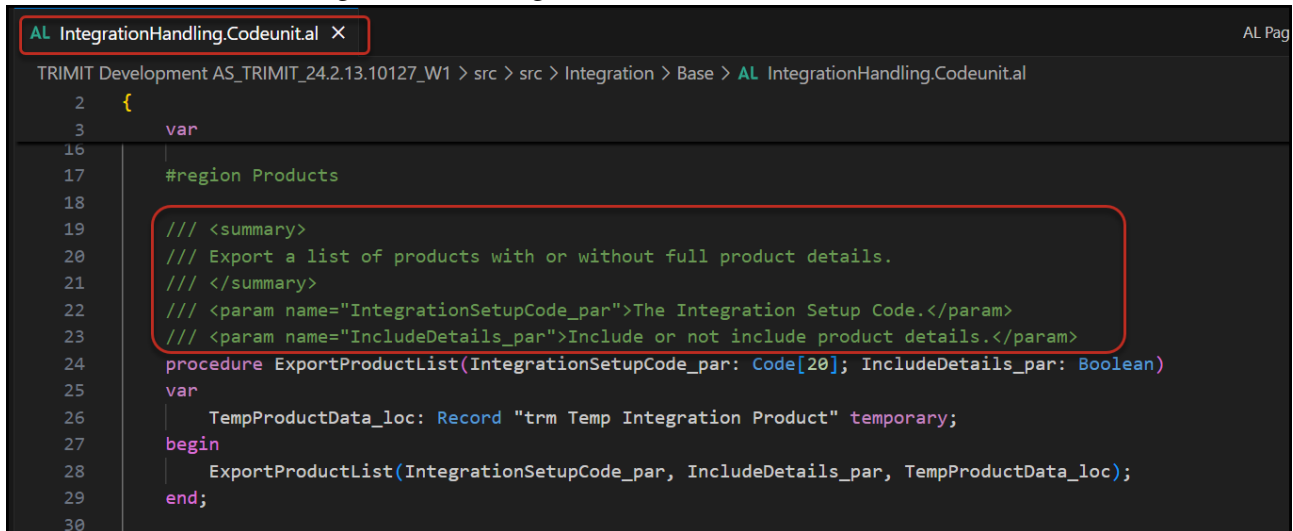
Request Value 1	Name	Method	Path	Query	Authentication Type	Content Type	Expected Response	Task Interval	Task Enabled
GetSale	GetSale	GET	https://.../Data/Sales?filter=SaleDate%20ge%20date%20...	?filter=SaleDate%20ge%20da...	None	application/atom+xml	200		☐
GetSaleLines	GetSaleLines	GET	https://.../Data/SalesLines?from=%7BfromDate%7D&to=...	?from=%7BfromDate%7D&to=...	None	application/atom+xml	200		☐
GetSalePayments	GetSalePayments	GET	https://.../Data/SalesPayments?from=%7BfromDate%7D&to=...	?from=%7BfromDate%7D&to=...	None	application/atom+xml	200		☐
GetProductTransfer	GetProductTransfer	GET	https://.../Data/ProductTransfers?from=%7BfromDate%7D&to=...	?from=%7BfromDate%7D&to=...	None	application/atom+xml	200		☐
GetMovements	GetMovements	GET	https://.../Data/StockmovementsV3?from=%7BfromDate%7D&to=...	?from=%7BfromDate%7D&to=...	None	application/atom+xml	200		☐
GetProducts	GetProducts	GET	https://.../restapi/V2/api/Groups		None	application/json	200		☐
GetStockStatus	GetStockStatus	GET	https://.../restapi/V2/api/Groups		None	application/json	200		☐
GetStoreOrders	GetStoreOrders	GET	https://.../restapi/V2/api/Groups		None	application/json	200		☐
GetSettlements	GetSettlements	GET	https://.../restapi/V2/api/Groups		None	application/json	200		☐

Note

Based on the **Use Sandbox EAPI Request Base Url** the blurred text in the above picture will be the **EAPI Request Base Url** or the **EAPI Sandbox Request Base Url**.

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 X
TRIMIT Development AS_TRIMIT_24.2.13.10127_W1 > src > src > Integration > Base > AL IntegrationHandling.Codeunit.al
2  {
3      var
16
17      #region Products
18
19      /// <summary>
20      /// Export a list of products with or without full product details.
21      /// </summary>
22      /// <param name="IntegrationSetupCode_par">The Integration Setup Code.</param>
23      /// <param name="IncludeDetails_par">Include or not include product details.</param>
24      procedure ExportProductList(IntegrationSetupCode_par: Code[20]; IncludeDetails_par: Boolean)
25      var
26          TempProductData_loc: Record "trm Temp Integration Product" temporary;
27      begin
28          ExportProductList(IntegrationSetupCode_par, IncludeDetails_par, TempProductData_loc);
29      end;
30
```

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 single 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,  
varTempSalesReturnReceiptLine_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"

GetProductVarDimOrderChoices()

Get a varDimOrder ChoiceList for a specific product.

Syntax

```
GetProductVarDimOrderChoices(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"

GetProductVarDimOrderChoices()

Get a varDimOrder ChoiceList for a specific product.

Syntax

```
GetProductVarDimOrderChoices(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

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

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

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

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

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

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

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

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

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

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

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

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_OnAfterGetPriceWithFilters()

Syntax

```
pf_OnAfterGetPriceWithFilters(var TempProductData_par: Record "trm Temp Integration  
Product" temporary; var Price_par: Decimal; var PriceFound_par: Boolean)
```

Parameters

TempProductData_par

Record trm Temp Integration Product

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_OnAfterGetReferences()

Syntax

pf_OnAfterGetGReferences(ItemNo_par: Code[20], var TempProductData_par: Record "trm Temp Integration Product" temporary)

Parameters

ItemNo_par

Type: Code[20]

TempProductData_par

Record trm Temp Integration Product

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

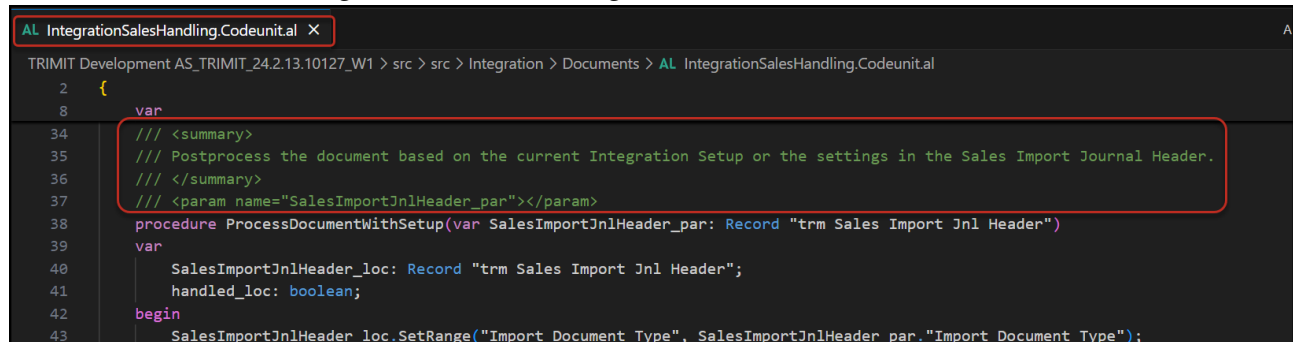
Type: Code[10]

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.Codeunit1 X
TRIMIT Development AS_TRIMIT_24.2.13.10127_W1 > src > src > Integration > Documents > AL IntegrationSalesHandling.Codeunit1
2 {
8   var
34   /// <summary>
35   /// Postprocess the document based on the current Integration Setup or the settings in the Sales Import Journal
36   /// </summary>
37   /// <param name="SalesImportJnlHeader_par"></param>
38   procedure ProcessDocumentWithSetup(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header")
39   var
40     SalesImportJnlHeader_loc: Record "trm Sales Import Jnl Header";
41     handled_loc: boolean;
42   begin
43     SalesImportJnlHeader_loc.SetRange("Import Document Type", SalesImportJnlHeader_par."Import Document Type");
```

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

Pf_OnAfterCreateCustFromTemplate()

Syntax

```
Pf_OnAfterCreateCustFromTemplate(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_par: 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_OnBeforeCreateIntegrationDoc()

Syntax

```
Pf_OnAfterCreateCustFromTemplate(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_par: 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_OnAfterCreateDocument()

Syntax

```
Pf_OnAfterCreateDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var SalesHeader_par: Record "Sales Header"; handled_par: Boolean)
```

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

SalesHeader_par

Type: Record "Sales Header"

Handled_loc

Type: Boolean

Pf_OnAfterCreateIntegrationDoc

Syntax

```
Pf_OnAfterCreateIntegrationDoc(var IntegrationSetup_par: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var SalesHeader_par: Record "Sales Header"; Handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

SalesHeader_par

Type: Record "Sales Header"

Handled_loc

Type: Boolean

Pf_OnBeforeProcessDocumentWithSetup()

Syntax

```
Pf_OnBeforeProcessDocumentWithSetup(var IntegrationSetup: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var handled_loc: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Handled_loc

Type: Boolean

Pf_OnBeforeProcessIntegrationDocWithSetup()

Syntax

```
Pf_OnBeforeProcessIntegrationDocWithSetup(var IntegrationSetup_par: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var Handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Handled_loc

Type: Boolean

Pf_OnAfterProcessDocumentWithSetup()

Syntax

```
Pf_OnAfterProcessDocumentWithSetup(var IntegrationSetup: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; handled_loc: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Handled_loc

Type: Boolean

Pf_OnAfterProcessIntegrationDocWithSetup()

Syntax

```
Pf_OnAfterProcessIntegrationDocWithSetup(var IntegrationSetup_par: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; Handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"
SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
Handled_loc
Type: Boolean

Pf_OnBeforeProcessDocument()

Syntax

Pf_OnBeforeProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var ReleaseDoc_par: Boolean; var CreatePickDoc_par: Boolean; var CreatePayment_par: Boolean; var PostDoc_par: Boolean; var handled_loc: Boolean)

Parameters

SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDoc_par
Type: Boolean
CreatePickDoc_par
Type: Boolean
CreatePayment_par
Type: Boolean
PostDoc_par
Type: Boolean
Handled_loc
Type: Boolean

Pf_OnBeforeProcessIntegrationDoc()

Syntax

Pf_OnBeforeProcessIntegrationDoc(var IntegrationSetup_par: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var ReleaseDoc_par: Boolean; var CreatePickDoc_par: Boolean; var CreatePayment_par: Boolean; var PostDoc_par: Boolean; var handled_loc: Boolean)

Parameters

IntegrationSetup_par
Type: Record "trm Integration Setup"
SalesImportJnlHeader_par
Type: Record "trm Sales Import Jnl Header"
ReleaseDoc_par
Type: Boolean
CreatePickDoc_par
Type: Boolean
CreatePayment_par
Type: Boolean
PostDoc_par
Type: Boolean
Handled_loc

Type: Boolean

Pf_OnAfterProcessDocument()

Syntax

```
Pf_OnAfterProcessDocument(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; ReleaseDoc_par: Boolean; CreatePickDoc_par: Boolean; CreatePayment_par: Boolean; PostDoc_par: Boolean; handled_loc: Boolean)
```

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

ReleaseDoc_par

Type: Boolean

CreatePickDoc_par

Type: Boolean

CreatePayment_par

Type: Boolean

PostDoc_par

Type: Boolean

Handled_loc

Type: Boolean

Pf_OnAfterProcessIntegrationDoc()

Syntax

```
Pf_OnAfterProcessIntegrationDoc(var IntegrationSetup_par: Record "trm Integration Setup"; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; ReleaseDoc_par: Boolean; CreatePickDoc_par: Boolean; CreatePayment_par: Boolean; PostDoc_par: Boolean; Handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

ReleaseDoc_par

Type: Boolean

CreatePickDoc_par

Type: Boolean

CreatePayment_par

Type: Boolean

PostDoc_par

Type: Boolean

Handled_loc

Type: Boolean