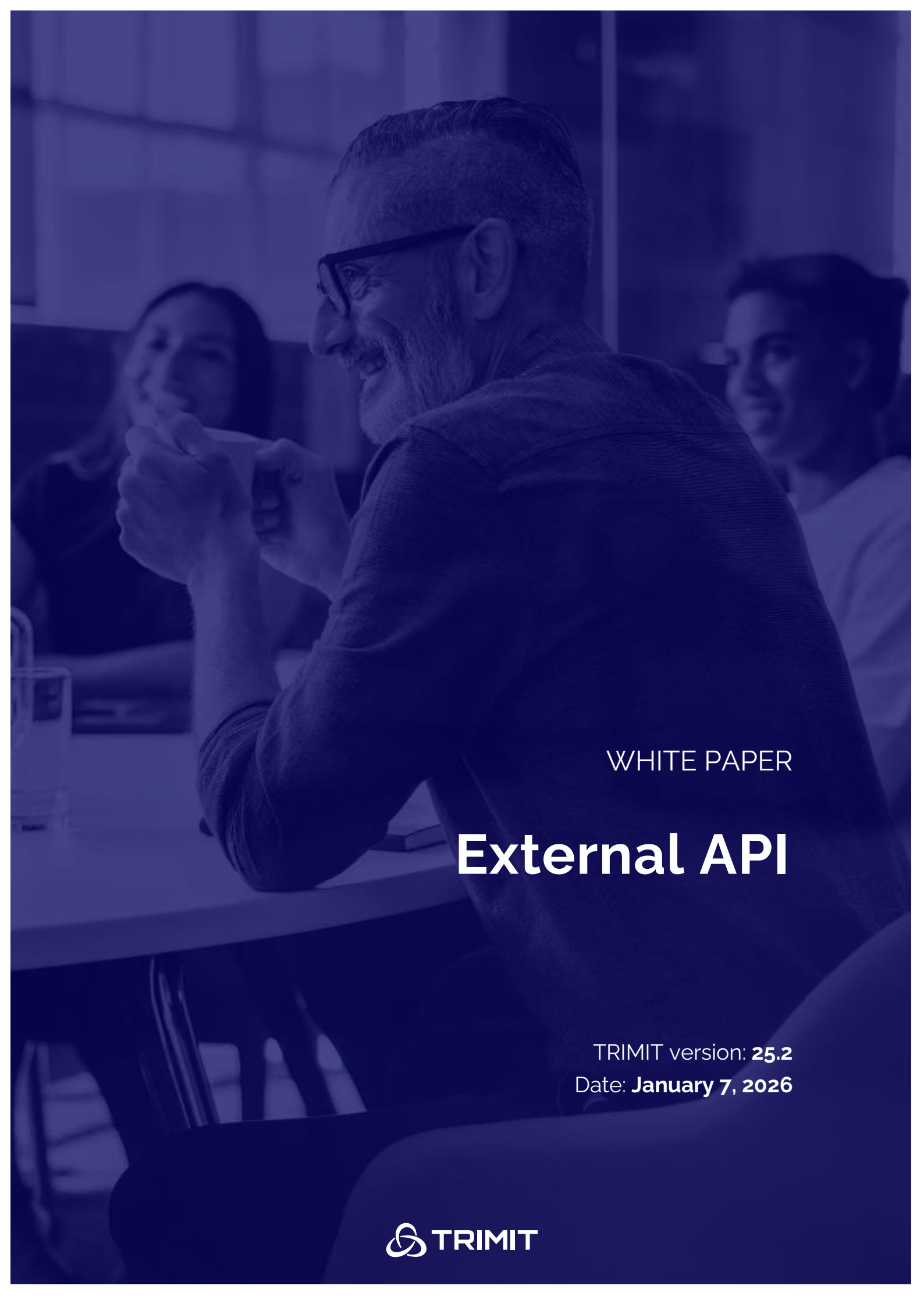




WHITE PAPER

External API

TRIMIT version: **25.2**

Date: **January 7, 2026**

Table of Contents

External API	1
Introduction.....	11
Components.....	12
Integration Setup.....	16
FastTab General.....	17
FastTab Products	18
FastTab Inventory	19
FastTab Sales Prices	21
FastTab Customers.....	21
FastTab Sales Documents.....	23
General Setup	23
Customer Setup	25
Sales Quote Setup.....	26
Sales Order Setup.....	27
Sales Invoice Setup	28
Sales Return Order Setup	29
Sales Credit Memo Setup	29
Reversal Setup	30
Not in the Page, but can be added via Personalize:	31
FastTab Posted Sales Documents.....	33
Actions, Products.....	33
Edit Category Lines.....	33
Category Management.....	33
Category List.....	33
Refresh Products.....	33
Exported Products.....	34
Actions, Documents	34
Create Sales Posting Job Queue Job	34
Sales Documents	35
Posted Sales Documents	39
Actions, Setup.....	41
Request Setup.....	41
Standard API Templates	41
Standard API Template Selection Rules.....	41
Related to the Integration Setup.....	42
Language Setup.....	42

Currency Setup	42
Location Setup	43
Customer Setup	44
Customer Template Setup.....	44
Customer Std. API Template Setup.....	44
Item Setup	45
Additional Data Setup	45
TRIMIT's Postman.json.....	47
Documentation in Postman Collection.....	47
Export of Item References.....	48
Export of Prices.....	49
Export of Extended Text	50
Import of VarDim Order	52
Import Comment Sales Document Line by using Description / Description 2	52
Import Description and Description 2.....	52
Requests	53
Request Setup	53
Requests Running.....	54
Request Log	54
Actions, Requests, Create Request Job Queue Job.....	54
Related, Request	55
Export/Import of Integration Code.....	56
What is exported exactly:.....	57
What is imported exactly:	57
Appendix 1: Codeunit 6036860	58
ExportProductList().....	58
ExportProductList().....	58
ExportProductList().....	59
ExportProduct()	59
ExportProduct()	59
ExportProduct()	60
ExportProduct()	60
ExportProduct()	60
TrackExportedProducts()	61
TrackExportedProduct()	61
TrackExportedProduct()	61
RefreshProductBuffer()	62
GetCategories()	62

ExportInventories()	62
ExportInventories()	62
ExportInventories()	62
ExportInventoryProduct()	63
ExportInventoryProduct()	63
ExportInventoryProduct()	63
ExportItemInventories()	64
ExportItemInventories()	64
ExportPrices()	64
ExportPrices()	64
ExportPrices()	65
ExportPricesProduct()	65
ExportCustomers()	65
ExportCustomers()	66
ExportCustomers()	66
ExportCustomer()	66
ExportCustomer()	67
ImportSalesOrders()	67
ImportSalesOrders()	67
ImportSalesOrders()	67
ImportSalesInvoices()	68
ImportSalesInvoices()	68
ImportSalesInvoices()	68
ImportSalesReturnOrders()	68
ImportSalesReturnOrders()	69
ImportSalesReturnOrders()	69
ImportSalesCreditMemos()	69
ImportSalesCreditMemos()	69
ImportSalesCreditMemos()	70
ClearTempSalesDocuments()	70
ClearTempPostedSalesDocuments()	70
ExportSalesShipments()	71
ExportSalesShipments()	71
ExportSalesShipments()	71
ExportSalesShipments()	71
ExportSalesInvoices()	72
ExportSalesInvoices()	72
ExportSalesInvoices()	72

ExportSalesInvoices()	73
ExportSalesReturnReceipts()	73
ExportSalesReturnReceipts()	73
ExportSalesReturnReceipts()	73
ExportSalesReturnReceipts()	74
ExportSalesCreditMemos()	74
ExportSalesCreditMemos()	74
ExportSalesCreditMemos()	75
ExportSalesCreditMemos()	75
GetDefaultLocationCode()	75
GetLocationCodeWithCheck()	75
GetProductDescriptions()	76
GetProductDescriptions()	76
GetProductDimensions()	76
GetProductDimensions()	77
GetProductStatisticsGroups()	77
GetProductStatisticsGroups()	77
GetProductVarDims()	77
GetProductVarDims()	78
GetProductVarDimOrderChoices()	78
GetProductVarDimOrderChoices()	78
GetProductCompositions()	78
GetProductCompositions()	79
GetProductCarelables()	79
GetProductCarelables()	79
GetProductAdditionalInfo()	79
GetProductAdditionalInfo()	80
GetProductAdditionalTags()	80
GetProductAdditionalTags()	80
GetProductSustainability()	80
GetProductSustainability()	81
GetProductMeasurementCharts()	81
GetProductMeasurementCharts()	81
GetProductCollections()	81
GetProductCollections()	82
GetProductPrices()	82
GetProductPrices()	82
GetProductItems()	82

GetProductItems()	83
GetProductItemAttributes()	83
GetProductItemAttributes()	83
GetProductImages()	84
GetProductImages()	84
GetProductReferences()	84
GetTableData()	84
GetIntegrationProductFromFilter()	84
GetIntegrationProductFromFilter()	85
CopyIntegrationProduct()	85
IsItemValid()	85
CheckProductExportAllowed()	86
CheckInventoryExportAllowed()	86
CheckCustomerExportAllowed()	86
CheckCustomerImportAllowed()	86
CheckSalesQuoteImportAllowed()	86
CheckSalesOrderImportAllowed()	86
CheckSalesInvoiceImportAllowed()	86
CheckSalesCreditMemoImportAllowed()	86
CheckSalesReturnOrderImportAllowed()	87
CheckPostedSalesShipmentExportAllowed()	87
CheckPostedSalesInvoiceExportAllowed()	87
CheckPostedSalesReturnShipmentExportAllowed()	87
CheckPostedSalesCreditMemoExportAllowed()	87
GetCode10FromText20()	87
GetCode10FromCode20()	88
GetCode10FromCode40()	88
GetCode20FromCode30()	88
GetCode20FromCode35()	88
FindIntegrationSetupCode()	89
SetIntegrationSetup()	89
pf_OnAfterIsSaleAllowedCheck()	89
pf_OnAfterGenerateProductList()	90
pf_OnAfterGetProductList()	90
pf_OnAfterGetProductDetails()	90
pf_OnAfterGetExtendedDescription()	90
pf_OnCustomExtendedTextFormat ()	90
pf_OnCustomExtendedTextEncoding ()	91

pf_OnAfterGetDescription()	91
pf_OnAfterGetPrice()	92
pf_OnAfterGetPriceWithFilters()	92
pf_OnAfterGetEAN()	92
pf_OnAfterGetGTIN()	92
pf_OnAfterGetReferences()	93
pf_OnBeforeGetInventories()	93
pf_OnAfterGetInventories()	93
pf_OnAfterGetInventory()	93
pf_OnBeforeGetItemInventories()	93
pf_OnAfterGetItemInventories()	94
pf_OnAfterCalculateInventory()	94
Appendix 2: Codeunit 6036862	95
ProcessDocumentWithSetup()	95
ReverseDocumentWithCheck()	95
ReverseDocument()	95
ReverseDocumentLines()	96
ProcessDocument()	96
ProcessDocument()	96
ProcessDocument()	97
ProcessDocument()	97
ProcessDocument()	97
CreateDocument()	98
SetDocumentNoAndSeries()	98
CreateJournal()	98
CreateDocJournal()	98
ReleaseDocument()	98
CreatePickDocFromOrder()	99
CreateOrderFromQuote()	99
CreateInvoiceFromQuote()	99
CreatePayment()	99
PostDocument()	100
PostDocumentJournal()	100
PostDocWithJournal()	100
SetInvoiceExportedStatus()	100
SetSalesInvoiceExportedStatus()	100
SetCreditMemoExportedStatus()	101
SetSalesCreditMemoExportedStatus()	101

SetShipmentExportedStatus()	101
SetSalesShipmentExportedStatus()	101
SetReturnReceiptExportedStatus()	102
SetSalesReturnReceiptExportedStatus()	102
IsDocumentNoUsed()	102
IsDocumentCreated()	103
HasSellToInfo()	103
HasBillToInfo()	103
HasShipToInfo()	103
CreateOrderExtraInfo()	104
CreateOrderExtraInfo()	104
CopyOrderExtraInfo()	104
GetIntegrationItemWithCheck()	105
GetIntegrationItem()	105
GetShipments()	105
GetShipments()	106
GetInvoices()	106
GetInvoices()	106
GetReturnReceipts()	107
GetReturnReceipts()	107
GetCreditMemos()	107
GetCreditMemos()	107
GetTrackingUrl()	108
GetCustomerNo()	108
GetCustomerNoFromSetup()	108
GetCustomerNoFromTemplateSetup()	109
CreateCustomerFromTemplate()	109
FindTemplateFromCustomer()	109
ApplyTemplateToCustomer()	109
UseConfigurationTemplatesOnCustomer()	109
CreateIntegrationParams()	109
ImportSalesDocuments()	110
GetAdditionalFieldsBlob ()	110
GetAdditionalFieldsBlob ()	110
DeserializeAdditionalFields ()	111
SetAdditionalFieldsOnTarget ()	111
SetIntegrationSetup()	111
Pf_OnAfterImportSalesDocuments()	111

Pf_OnAfterGetShipment()	112
Pf_OnAfterGetIntegrationShipment()	112
Pf_OnAfterGetInvoice()	112
Pf_OnAfterGetIntegrationInvoice()	113
Pf_OnAfterGetReturnReceipt()	113
Pf_OnAfterGetIntegrationReturnReceipt()	114
Pf_OnAfterGetCreditMemo()	114
Pf_OnAfterGetIntegrationCreditMemo ()	114
Pf_OnAfterCreateOrderFromQuote ()	115
Pf_OnAfterCreateInvoiceFromQuote ()	115
Pf_OnAfterCreatePick ()	115
Pf_OnAfterAutoRelease ()	115
Pf_OnAfterManualRelease ()	115
Pf_OnBeforeReverseDocument ()	116
Pf_OnAfterReverseDocument ()	116
Pf_OnBeforeGetCustNo()	117
Pf_OnGetCustomerNo()	117
Pf_OnBeforeGetCustNoFromSetup ()	117
Pf_OnBeforeFindCustNoFromIncomingDoc	117
Pf_OnBeforeSearchCustNo	118
Pf_OnAfterGetCustNoFromSetup()	118
Pf_OnGetCustNoFromSetupFailed()	118
Pf_OnBeforeGetCustNoFromTemplateSetup()	119
Pf_OnMissingCustomerSellToEmail()	119
Pf_OnAfterGetCustNoFromTemplateSetup()	119
Pf_OnGetCustNoFromTemplateSetupFailed()	120
Pf_OnBeforeCreateCustomer()	120
Pf_OnBeforeCreateCustFromTemplate()	120
Pf_OnAfterCreateCustomer()	121
Pf_OnAfterCreateCustFromTemplate()	121
Pf_OnBeforeFindTemplateFromCustomer()	121
Pf_OnAfterFindTemplateFromCustomer()	122
Pf_OnGetTemplateSetupFailed()	122
Pf_OnBeforeApplyTemplateToCustomer()	122
Pf_OnAfterApplyTemplateToCustomer()	123
Pf_OnBeforeCreateIntegrationDoc()	123
Pf_OnAfterCreateDocument()	123
Pf_OnAfterCreateIntegrationDoc()	124

Pf_OnBeforeProcessDocumentWithSetup() 124

Pf_OnBeforeProcessIntegrationDocWithSetup() 124

Pf_OnAfterProcessDocumentWithSetup() 125

Pf_OnAfterProcessIntegrationDocWithSetup() 125

Pf_OnBeforeProcessDocument()..... 125

Pf_OnBeforeProcessIntegrationDoc() 126

Pf_OnAfterProcessDocument() 126

Pf_OnAfterProcessIntegrationDoc()..... 126

Pf_ OnBeforeSetAdditionalField() 127

Pf_OnAdditionalFieldNotFound()..... 127

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)
- ➔ Creating 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.

As of TRIMIT 25.2.15, we also create demo data for the **Integration Setup** and related information in our **TRIMIT Data and Test App**. This can be used for pre-sales or testing TRIMITs External API functionality.

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 25.2 for Business Central W1 BC27.X**.

The pictures in this White Paper are based on the demo data for **TRIMIT 25.2**, which is based on Microsoft Dynamics 365 Business Central 2025 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>
	6036892	<i>trm Integration EAPI Request</i>
	6036895	<i>trm EAPI Requests Log</i>
	6036896	<i>trm Integration EAPI Req Setup</i>
	6037017	<i>trm Sales Comment Import List</i>
	6037236	<i>trm Integration Add. Fields</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>

The API Pages

6037813	trm EAPI Salespersons
6037814	trm EAPI Units Of Measure
6037815	trm EAPI Collections
6037816	trm EAPI Delivery Periods
6037817	trm EAPI Contacts
6037818	trm EAPI Contacts Info
6037819	trm EAPI Cust Finance Dtls
6037820	trm EAPI Customers
6037821	trm EAPI Return Order Lines
6037822	trm EAPI Return Orders
6037823	trm EAPI Sales Credit Memos
6037824	trm EAPI Sales Cr.Memo Lines
6037825	trm EAPI Sales Document Lines
6037826	trm EAPI Sales Document
6037827	trm EAPI Sales Invoice Lines
6037828	trm EAPI Sales Invoices
6037829	trm EAPI Sales Order Lines
6037830	trm EAPI Sales Orders
6037831	trm EAPI Sales Quote Lines
6037832	trm EAPI Sales Quotes
6037833	trm EAPI Inventories
6037834	trm EAPI Item Additional Info
6037835	trm EAPI Item Additional Tags
6037836	trm EAPI Item Attribute
6037837	trm EAPI Item Carelabel
6037838	trm EAPI Item Composition
6037839	trm EAPI Item Description
6037840	trm EAPI Item Inventory
6037841	trm EAPI Item Price
6037842	trm EAPI Items
6037843	trm EAPI Item Stat. Group
6037844	trm EAPI Item Sustainability
6037845	trm EAPI Master Add. Info
6037846	trm EAPI Master Add. Tags
6037847	trm EAPI Master Attributes
6037848	trm EAPI Master Carelabel
6037849	trm EAPI Master Collection
6037850	trm EAPI Master Composition
6037851	trm EAPI Master Description
6037852	trm EAPI Master Dimension
6037853	trm EAPI Master Inventories
6037854	trm EAPI Master Item
6037855	trm EAPI Master Measure Chart
6037856	trm EAPI Master Price
6037857	trm EAPI Masters
6037858	trm EAPI Master Stat. Group
6037859	trm EAPI Master Sustainability
6037860	trm EAPI Master VarDim
6037861	trm EAPI Exported Documents
6037862	trm EAPI SalesPostCreditMemos
6037863	trm EAPI SalesPostCrMemoLines

The API Pages	6037864	<i>trm EAPI SalesPostDocLines</i>
	6037865	<i>trm EAPI SalesPostDocuments</i>
	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 Negatives</i>
	6037916	<i>trm EAPI VarDim Translations</i>
	6037917	<i>trm EAPI Item VarDim</i>
	6037919	<i>trm EAPI Additional Fields</i>
	6037920	<i>trm EAPI Sales DocLinesApi</i>

The API Pages	6037921	<i>trm EAPI Sales CrMemoLinesApi</i>
	6037922	<i>trm EAPI Return OrderLinesApi</i>
	6037923	<i>trm EAPI Sales InvoiceLinesApi</i>
	6037924	<i>trm EAPI Sales OrderLinesApi</i>
	6037925	<i>trm EAPI Sales QuoteLinesApi</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>
	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 e-commerce.

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**.

The screenshot displays the TRIMIT Integration Setup interface. At the top, a search bar labeled "Tell me what you want to do" contains the text "Integration". Below the search bar, a list of results is shown, with "Integrations List" highlighted by a red box. The main area shows the "Integrations List" table, which has columns for Code, Description, Integration Type, and User Setup ID. A single row is visible with the code "DEFAULT" and description "Default B2C Integration". A red box highlights the "+ New" button in the table's header. To the right, an "Overview" section provides a summary of various metrics, including Documents, Products, and Setup counts.

Code	Description	Integration Type	User Setup ID
DEFAULT	Default B2C Integration	Default	

Overview

Documents

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

Products

0	0	0
Exported Products	Exported Items	Exported Masters

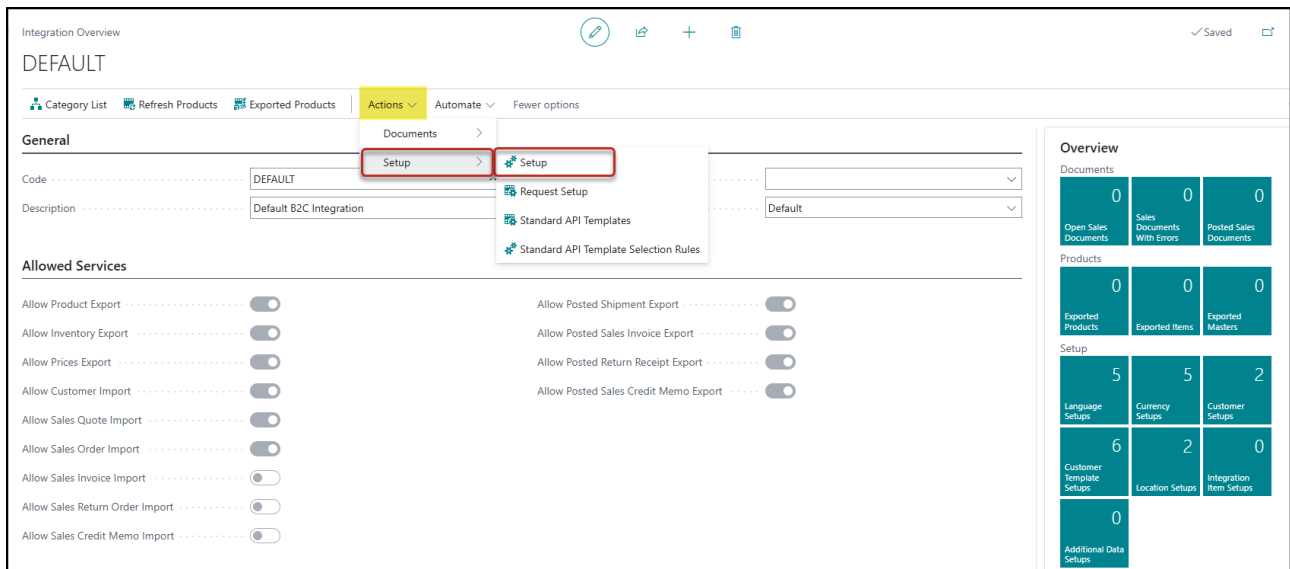
Setup

5	5	2
Language Setups	Currency Setups	Customer Setups
6	2	0
Customer Template Setups	Location Setups	Integration Item Setups
0		
Additional Data 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.

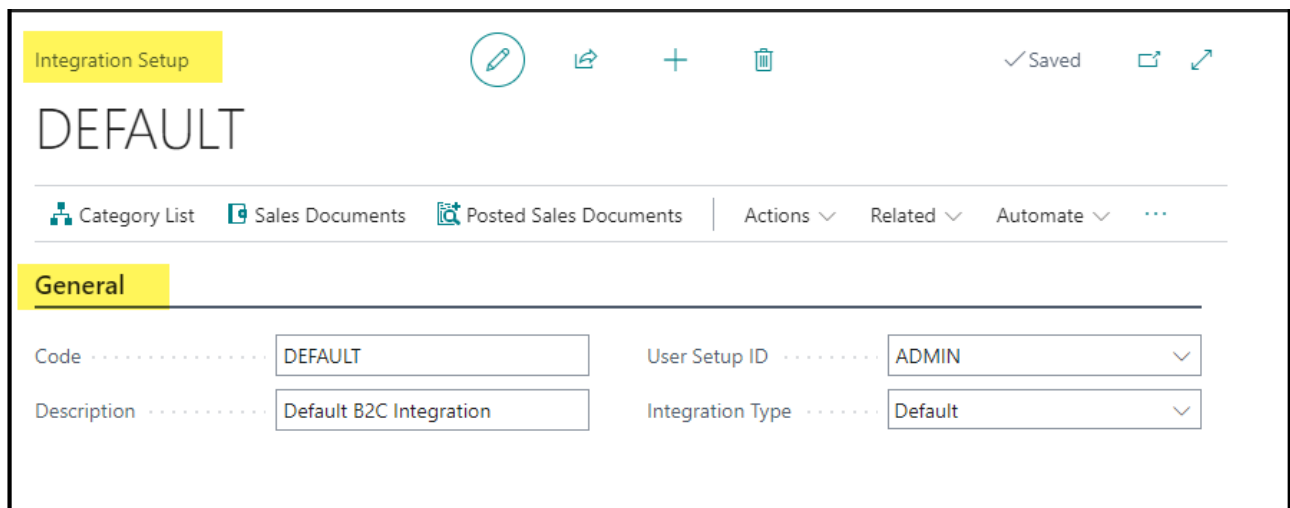
You can create several **Codes** based on the **User Setup ID** for integrations with several external systems.

In the **FactBoxes** you will see the related information to this specific **Code**.



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

FastTab General



Code

The unique Code for this Integration Setup.

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

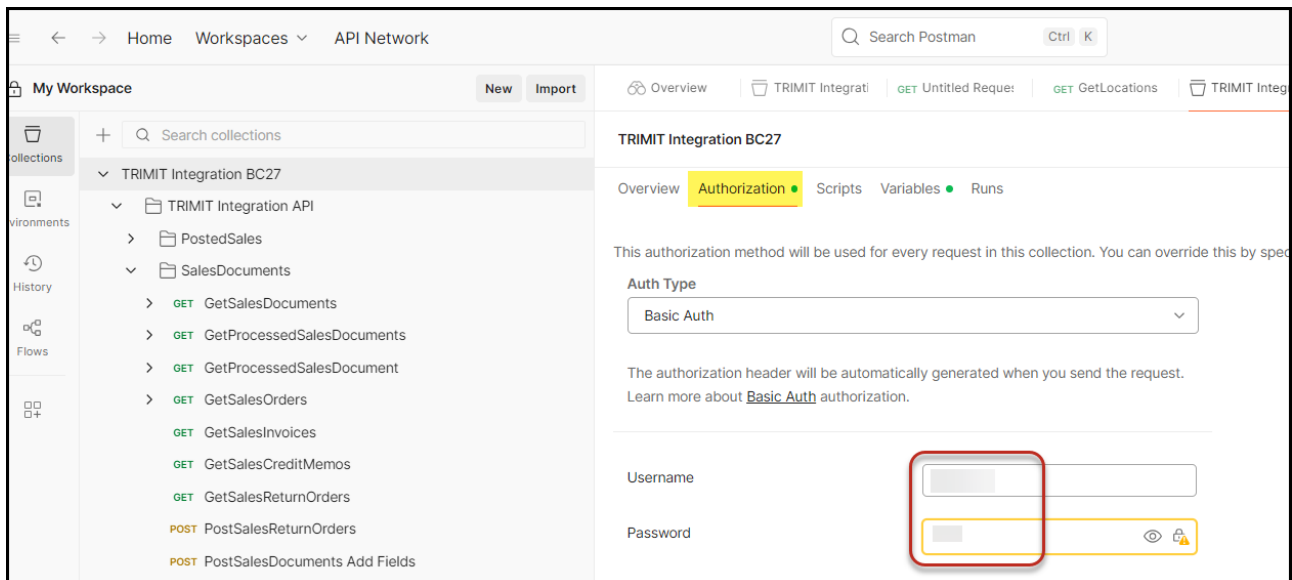
Description

Description for this specific **Code**.

User Setup ID

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

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



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

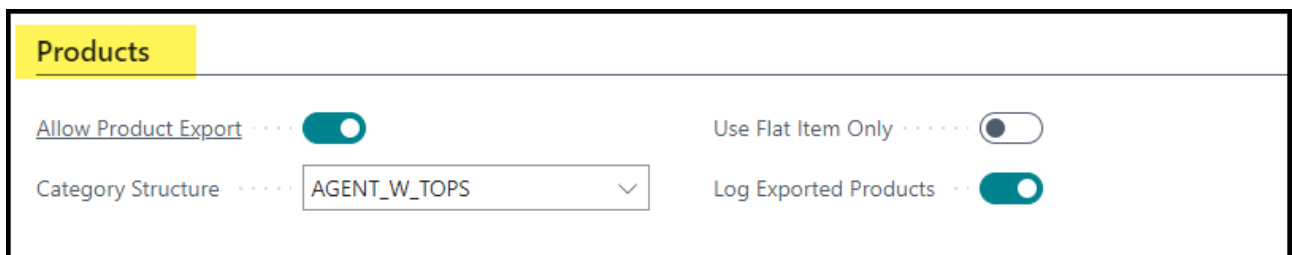
Integration Type

At this point, we only have the option *Default*.

The **Integration Type** can be extended (it is an extendable Enum) by implementing the interface "trm Integration Agent".

Our *Default* implementation can be found in the Codeunit **6036872 trm Integration Agent**.

FastTab Products



Allow Product Export

Is it allowed to export Product Data based on the Masters/Items that are in the **Category Structure**.

To be able to filter the export of Item/Master data, we created this special **Product Export**, which will be based on the **Category Structure** and the **Use Flat Item Only**.

Note

It will always be possible to export all the Items and Masters via the TRIMIT API services, which is an extension of the BC standard APIs with TRIMIT fields.

And of course, you will be able to set filters in the app you use for showing the Data (like i.e., Postman).

Category Structure

You can use the **Category Structure** (Categories in TRIMIT ERP) to filter the Masters/Items for which you want to export the Data if **Allow Product Export** is set to *TRUE*.

For more details, see **White Paper – TRIMIT Category Management**.

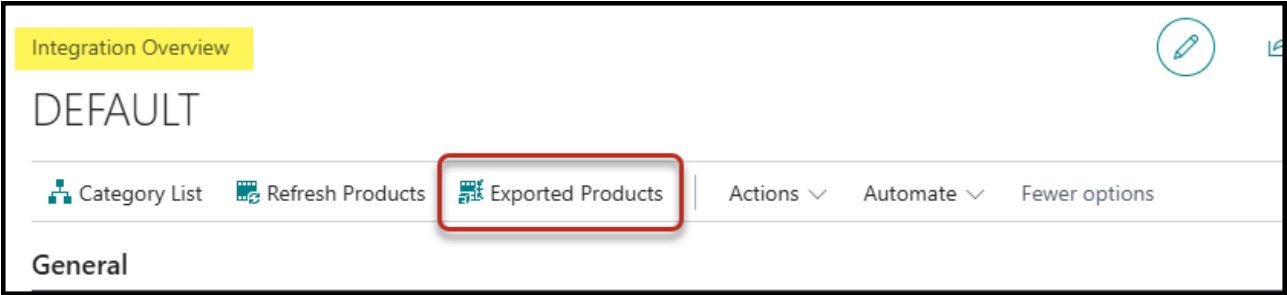
User Flat Item Only

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

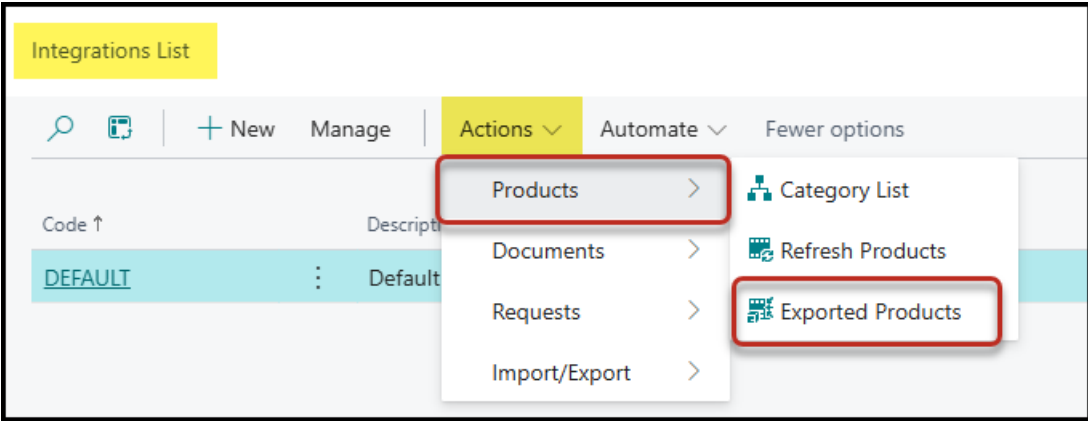
Log Exported Products

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 in the **Integration Overview** via [Exported Products](#).



Or in the **Integrations List** 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 **TRUE**, you are also able to determine which Quantities should be exported. You can choose between the following three options:

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

Inventory Profile Setup Code

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

The **Inventory Profile Setup Code** determines from which Documents the Quantities should be included in the Available Quantities.

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

For more details regarding the Inventory Profile Setup, see **White Paper – TRIMIT Inventory Profile**.

Export Future Deliveries

Available Quantities will be calculated by Current Date, but if you also want to include Sales Document Lines with a Shipment Date > Current Date and/or Purchase Lines with an Expected Receipt Date > Current Date and/or Production Orders with an End Date > Current Date, you need to have a checkmark in **Export Future Deliveries**.

In Postman, it might then look as follows:

TRIMIT Integration BC27 / TRIMIT Integration API / GetInventories

GET

{{dockerURL}} / {{apiURL}} / companies({{companyId}}) / inventories? \$filter=itemNo eq '21103002'

Send

Docs

Params

Authorization

Headers (8)

Body

Scripts

Settings

Query Params

☒	Key	Value	Description
☒	\$filter	itemNo eq '21103002'	
☐	Key	Value	Description

Body

Cookies

Headers (11)

Test Results

200 OK

129 ms

1.46 KB

Save Response

JSON

Preview

Visualize

```
{
  "@odata.context": "https://docbc27w1:7048/BC/api/trimit/integration/v1.1/$metadata#companies(d94ff043-57db-f011-8f93-93df51bd825d)/inventories",
  "value": [
    {
      "@odata.etag": "W/\"JzE50zE3NzMzMzc4MTI0MjQxNTMxNjQxOzAwOyc=\"",
      "itemNo": "21103002",
      "locationCode": "BLUE",
      "date": "2025-12-29",
      "availableQty": 17
    },
    {
      "@odata.etag": "W/\"JzE50zg9MDA5NTY1NzMyODQ0OTk5NTYxOzAwOyc=\"",
      "itemNo": "21103002",
      "locationCode": "BLUE",
      "date": "2026-01-03",
      "availableQty": 47
    },
    {
      "@odata.etag": "W/\"JzIwOzEwODc2MjAzNTc2NjgwMjQxNTMxNzAwMTswMDsn\"",
      "itemNo": "21103002",
      "locationCode": "BLUE",
      "date": "2026-03-11",
      "availableQty": 48
    },
    {
      "@odata.etag": "W/\"JzE50zg5Nzc4MTQ3NDQyMTQ1MzE3MTgxOzAwOyc=\"",
      "itemNo": "21103002",
      "locationCode": "CW",
      "date": "2025-12-29",
      "availableQty": 17
    }
  ]
}
```

FastTab Sales Prices

Sales Prices

Allow Prices Export ☒

Export All Sales Prices ☒

Price Calculation Cust. No.

Allow Prices Export

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

Price Calculation Cust. No.

If **Allow Prices Export** is set to *TRUE*, you also can determine from which Customer the Sales Prices 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 Order Date

Standard Customer STD

Summary Invoice

Filter Summary Invoice None

Summary Invoicing Controlled by Sell-to Customer No.

Generate Summary Order by Batch ... ☐

Blanket Order

Blanket Sales Order Search Until Shipment Date (Ascending)

Blanket Sales Order Search on Locat... ☐

WARNING: no blanket order found Yes

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

Export All Sales Prices

This will export all the records in Table **7002 Sales Prices** of the Masters/Items in the [Category Structure](#), but will not determine Sales Prices based on Search Expressions or Formulas.

Note

The system will export the net Sales Prices, and not the gross Sales Prices and Discounts. So, no Discounts will be exported.

FastTab Customers

Customers

Allow Customer Export ☒

Use Configuration Customer Templates ☒

Allow Customer Import ☒

Allow Customer Export

This field is not shown by default; you need to add it via **Personalize**.

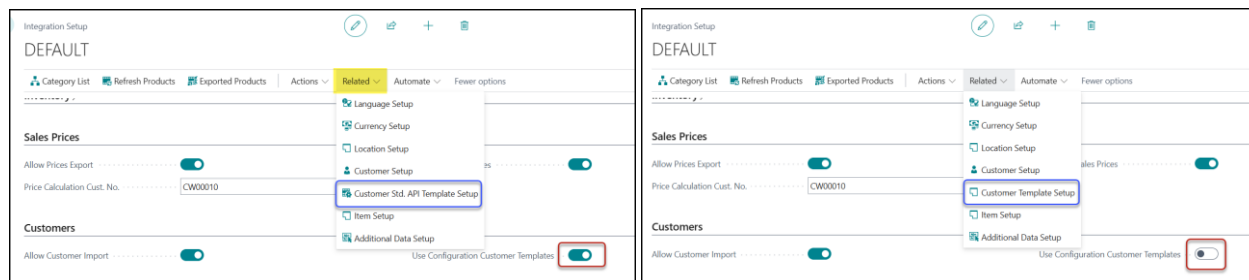
You can determine if it is allowed to export Customer Data for the TRIMIT Service Function. (It is not applicable for the Customer API Page in Postman.)

Allow Customer Import

You can determine if it is allowed to import Customer Data for the TRIMIT Service Function.

Use Configuration Customer Templates

This field determines which Customer Template should be used when the system needs to create a new Customer. Also, the Actions in the **Integration Setup** Page will be determined based on the content of this field.



Without a checkmark, you will make use of the specific TRIMIT Integration Customer Templates, that you can set up via **Related, Customer Template Setup** and can only be related to a **Country/Region Code**. I.e.:

The screenshot shows the 'Customer Template Setup' page with a table of templates. The 'Country/Region Code' column is highlighted. The 'Page Inspection' sidebar on the right shows the page title 'trm Integration Cust. Template (6036877, List)' and the table name 'trm Integration Setup Line (6036841)'.

Template No. ↑	Description	Country/Region Code ↑	Name
CUSTB2C	Trimit B2C Domestic Customer Template	GB	Great Britain
CUSTB2CEU	Trimit B2C Domestic EU Template	DE	Germany
CUSTB2CEU	Trimit B2C Domestic EU Template	DK	Denmark
CUSTB2CEU	Trimit B2C Domestic EU Template	NL	Netherlands
CUSTB2CFOR	Trimit B2C Domestic Foreign Customer Te...	US	USA

With a checkmark in **Use Configuration Customer Templates**, you will make use of the standard BC **Configuration Templates** for the Customers. That also means that you can set it up via **Related, Customer Std. API Template Setup**.

This enables you to determine which Template Code should be used in a more flexible way when you need to create a new Customer. I.e.:

The screenshot shows the 'API Setup' page with a table of API configurations. The 'Conditions' column is highlighted. The 'Page Inspection' sidebar on the right shows the page title 'API Setup (5469, List)' and the table name 'Config. Tmpl. Selection Rules (8620)'.

Order ↑	Table ID	Page ID	Template Code	Description	Conditions
0	18	0	CUSTB2C	Trimit B2C Domestic Customer Templ...	

Clicking on the field **Conditions** will give you the possibility to filter on any Field (even customized Fields) that would be coming in via the External API, to determine the right Template for creating a Customer.

The dialog box shows a filter for 'Customer' with several criteria: No., Credit Limit (LCY), Customer Posting Group, Blocked, Payment Method Code, and Gen. Bus. Posting Group. Each criterion has a dropdown menu. There is also a 'Filter...' button and a 'Filter totals by:' section with another 'Filter...' button.

Because of this Field, the option for [Customer Handling](#) *Use Customer Template Setup* will depend on the content of the **Use Configuration Customer Templates**, where the system will look to find the right Customer Template for creating the new Customer.

Customer Setup

Customer Handling

Use Customer From Incoming Docum...

Customer Sell-to Group

Use Customer Template Setup

Use Customer Setup

Use Customer Template Setup

Apply Customer From Incoming Documents

FastTab Sales Documents

Sales Documents

General Setup

Default Order TypeB2C

Default Sales Channel CodeB2C

No. of Processing Retries5

Document Numbering SourceDefault No. Series

Skip Existing Customer Search

Freeze Unit Price

Freeze Line Discount

Customer Setup

Customer HandlingUse Customer Setup

Use Customer From Incoming Do...

Customer Sell-to GroupB2C

Default Customer Nos.CUSTWEB

Sales Quote Setup

Allow Sales Quote Import

Sales Quote Nos.S-QUO-WC

Auto Create Sales Quote

Auto Release Sales Quote

Sales Order Setup

Allow Sales Order Import

Sales Order Nos.S-ORD-WC

Auto Create Sales Order

Auto Release Sales Order

Auto Create Pick Document From ...

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 TypeReverse on Document Total

Reversal Order ActionOrder is reversed to Return Order

Reversal Return ActionReturn is reversed to Order

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

General Setup

General Setup

Default Order TypeB2C

Default Sales Channel CodeB2C

No. of Processing Retries5

Document Numbering SourceDefault No. Series

Skip Existing Customer Search

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.

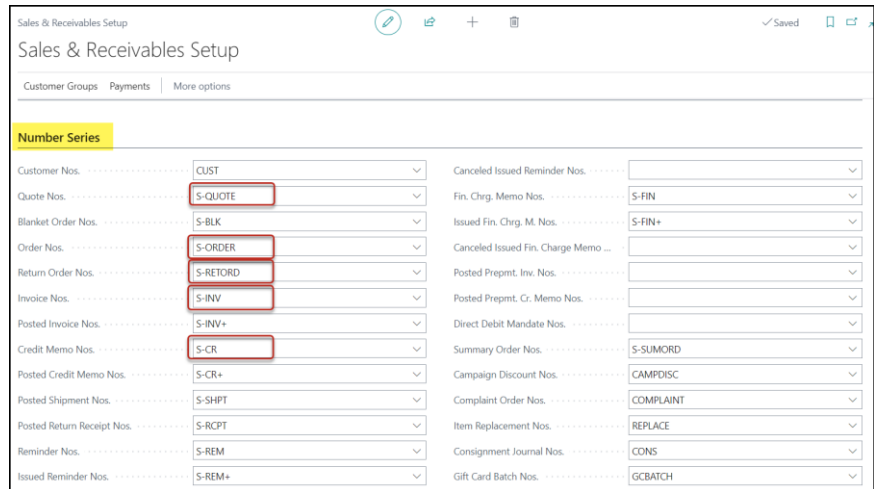
No. of Processing Retries

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. 
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.

Skip Existing Customer Search

If the **Skip Existing Customer Search** is set to *TRUE*, it will bypass the search on **Sell-To E-mail** (and **Customer Sell-to Group**) before a new Customer should be created, either from a Template or from a Setup.

You might know that this search will normally return the existing Customer, but in some cases, you need to bypass this.

Freeze Unit Price

If set to *TRUE*, 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 *TRUE*, 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 *TRUE*, 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 *TRUE*, 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 or by the Customer Std. API Template Setup (depending on the Use Configuration Customer Templates). Based on the 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 *FALSE*.

Sales & Receivables Setup

Customer Groups

Payments

More options

General

Discount Posting

All Discounts

▼

Credit Warnings

Both Warnings

▼

Stockout Warning

Shipment on Invoice

Return Receipt on Credit Memo

Invoice Rounding

Default Item Quantity

Create Item from Item No.

Create Item from Description

Copy Customer Name to Entries

Ext. Doc. No. Mandatory

Appln. between Currencies

All

▼

Logo Position on Documents

No Logo

▼

Default Posting Date

Work Date

▼

Default Quantity to Ship

Remainder

▼

Auto Post Non-inv. via Whole

None

▼

Copy Comments Blanket to Order

Copy Comments Order to Invoice

Copy Comments Order to Shpt.

Copy Cmts Ret.Ord. to Cr. Memo

Copy Cmts Ret.Ord. to Ret. Rcpt

Allow VAT Difference

Calc. Inv. Discount

VAT Bus. Posting Gr. (Price)

DOMESTIC

▼

Exact Cost Reversing Mandatory

Check Preprint. when Posting

Preprint. Auto Update Frequency

Never

▼

Allow Document Deletion Before

▼

Allow Multiple Posting Groups

Check Multiple Posting Groups

Alternative Groups

▼

Ignore Updated Addresses

Skip Manual Reservation

Quote Validity Calculation

▼

Global Dimensions

Brand Code

From Sales Header

▼

Season Code

From Sales Header

▼

Job No.

From Sales Header

▼

Job Sales Order

From Sales Header

▼

Copy Line Descr. to GL Entry

Disable Search by Name

Link Doc. Date to Posting Date

Use Customer From Incoming Documents

Setting this field to *TRUE* 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 newly 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. need 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 🔖 🔍 ↶										
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.

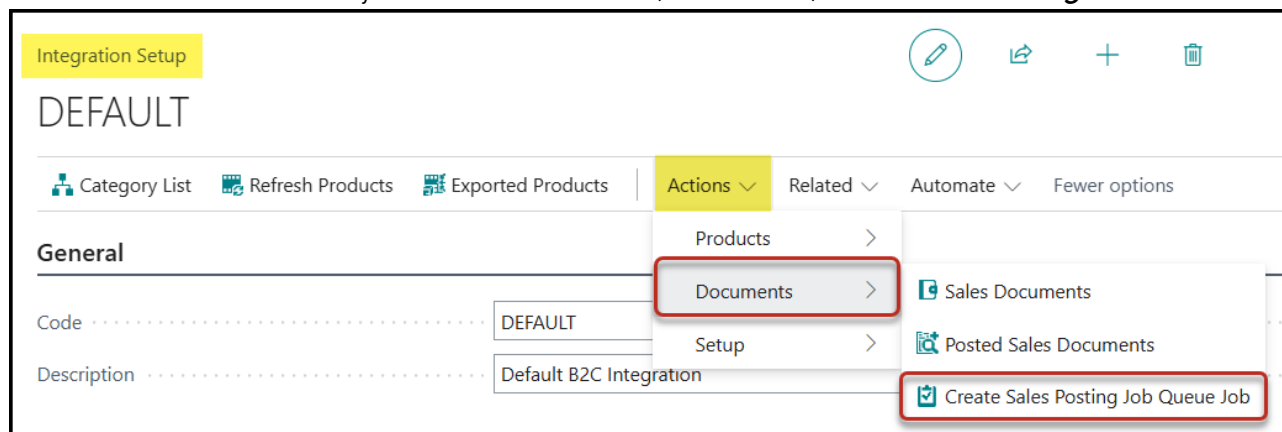
If **Document Numbering Source** is set to *Default No. Series*, you can enter a **No. Series** for the Integration Sales Quotes. Based on this **No. Series** the number for the regular **Sales Quote** will be determined.

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 *TRUE*, and a regular **Sales Quote** is created, you can determine if it also should be *Released* right away.

Sales Order Setup

Sales Order Setup	
Allow Sales Order Import	☒
Sales Order Nos.	S-ORD-WC
Auto Create Sales Order	☒
Auto Release Sales Order	☒
Auto Create Pick Document From Sales Order	☐
Auto Post Shipment and Invoice	☐

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 *TRUE*, 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 *TRUE*, and **Auto Release Sales Order** is set to *TRUE*, 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 *TRUE*, 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 *TRUE*.

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 *TRUE*, 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 *TRUE*, 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 *TRUE*, 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 *TRUE*.

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 *TRUE* 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 *TRUE*, 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 **TRUE**, 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.**

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

Posted Documents							
Integration Sales Documents							
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	☐
Use Gen. Journal for Credit Memo Posting	☐
General Journal Template Name	
General Journal Batch Name	

Use Gen. Journal for Sales Invoice Posting

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

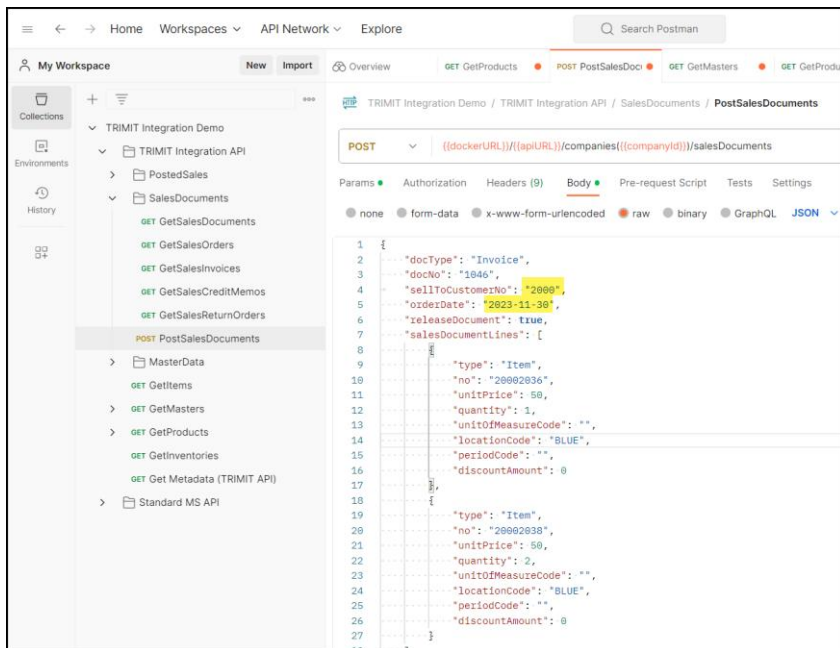
If you do not want to have a regular Sales Invoice created for an Integration Sales Invoice, you can set this field to **TRUE**.

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 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

✓ Saved

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 Type	Bal. Account No.	Bal. Gen. Posting Type
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 **TRUE**.

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** to **TRUE**, you also need to fill in 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 **TRUE**, you also need to fill in the **General Journal Template Name** and the **General Journal Batch Name**.

FastTab Posted Sales Documents

Posted Sales Documents	
Allow Posted Shipment Export	☒
Allow Posted Return Receipt Export	☒
Allow Posted Sales Invoice 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 for a 'DEFAULT' integration. At the top, there are icons for edit, share, add, and delete. Below this is a navigation bar with 'Category List', 'Refresh Products', and 'Exported Products' highlighted with a red box. The 'Actions' menu is open, showing 'Products', 'Documents', and 'Setup' with right-pointing arrows. To the right of the 'Actions' menu, there are buttons for 'Edit Category Lines' and 'Category Management', both highlighted with a red box. Below these, there is a 'Category List' button highlighted with a blue box. At the bottom right, there is a checkbox labeled 'Use short menu' which is checked.

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 TRIMIT e-commerce, this will also influence the products shown in TRIMIT e-commerce.

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**.

Category List

This will show the **Category Tree View** of the content of the [Category Structure](#).

Refresh Products

This will refresh the Category Buffer. Therefore, if 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 *TRUE* 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									
Integration Setup Code	No.	Type	Master No.	Variant 1 Value	Is Assortment	Date Active	SystemId	Created	Modified
DEFAULT	2100	Master	2100		☐	12/29/2025	{6645229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2110	Master	2110		☐	12/29/2025	{6845229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2120	Master	2120		☐	12/29/2025	{6b45229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2130	Master	2130		☐	12/29/2025	{6445229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2600	Master	2600		☐	12/29/2025	{6a45229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2610	Master	2610		☐	12/29/2025	{6545229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2620	Master	2620		☐	12/29/2025	{6945229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2630	Master	2630		☐	12/29/2025	{6c45229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM
DEFAULT	2640	Master	2640		☐	12/29/2025	{6745229b-bee4-f011-979e-f93638d674eb}	12/29/2025 2:59 PM	12/29/2025 2:59 PM

Actions, Documents

Integration Setup

DEFAULT

Category List

Refresh Products

Exported Products

Actions

Related

Automate

Fewer options

General

Code

DEFAULT

Description

Default B2C Integration

Products

Documents

Setup

Sales Documents

Posted Sales Documents

Create Sales Posting Job Queue Job

Create Sales Posting Job Queue Job

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

Note

It is **NOT** possible to create the Job Queue Entry by an "Delegated Administrator". It can only be created by a "Normal Administrator".

This **Job Queue Entry** 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 Entry** is necessary if you have set one or more of the following fields to *TRUE* 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

Home Job Queue More options

Set Status to Ready Set On Hold Restart Run once (foreground) Show Error

General

Object Type to Run Codeunit
Object ID to Run 6036865
Object Caption to Run trm Integration Post SalesDoc
Description Created for automatic processing of TRIMIT Integration Sales Doc...
Parameter String FIND-DOCUMENT
Job Queue Category Code
Priority Normal
User ID

Maximum No. of Attempts to Run 3
Rerun Delay (sec) 0
Last Ready State 12/29/2025 3:22 PM
Earliest Start Date/Time 12/29/2025 3:26 PM
Expiration Date/Time
Job Timeout 12 hours
Status Ready

Recurrence

Recurring Job ☐
Run on Mondays ☐
Run on Tuesdays ☐
Run on Wednesdays ☐
Run on Thursdays ☐
Run on Fridays ☐
Run on Saturdays ☐

Next Run Date Formula
Starting Time
Ending Time
No. of Minutes between Runs 2
Inactivity Timeout Period 5

Sales Documents

Clicking **Actions**, **Documents**, **Sales Documents** in the **Integrations List**, **Integration Overview** or **Integration Setup** will open the Page with the Integration Sales Documents, which will show you only the data that was imported, and not yet posted.

Integrations List

Code ↑ Description

DEFAULT : Default

Products >
Documents >
Requests >
Import/Export >

Sales Documents
Posted Sales Documents

Integration Overview

DEFAULT

Category List Refresh Products Exported Products

Actions > Automate > Fewer options

Documents > Sales Documents
Setup > Posted Sales Documents

Integration Setup

DEFAULT

Category List Refresh Products Exported Products

Actions > Related > Automate > Fewer options

Products >
Documents > Sales Documents
Setup > Posted Sales Documents
Create Sales Posting Job Queue Job

Code DEFAULT
Description Default B2C Integration

Sales Documents									
Integration Sales Documents									
Document Type	No.	Sell-to Customer No.	Sell-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 **Document Type** or **No.** will open the [Integration Sales Document](#) Card.

Sales Documents

Integration Sales Documents

Manage

Related

Posting

Related

Automate

Fewer options

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

Payment

Other

Extra Order Info

Use short menu

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

Ship-to Code

Ship-to Phone

Parcel Shop ID

Ship-to Country/Region Code

GB

Direct Consumer Delivery

Wrapping

Ship-to E-Mail

Order Confirmation E-Mail

Customer

Bill-to Customer No.

Sell-to E-Mail

Sell-to Phone No.

Sell-to Customer 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 e-commerce.

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.

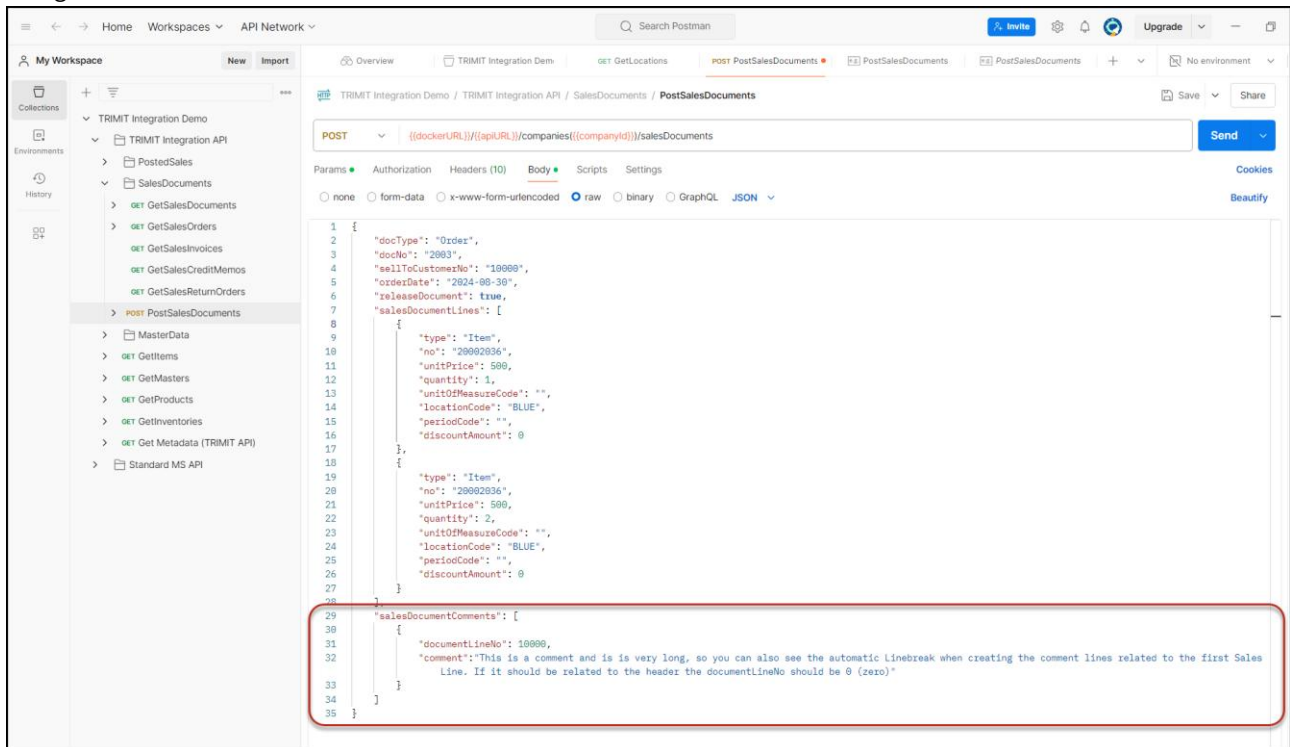
Clicking **Posting, Post All**, will create the regular Sales Documents for all the Lines at once.

Clicking **Posting, Test Order**, will test the data that was imported: Can the **Integration Sales Document** be posted?

Integration Sales Document Card

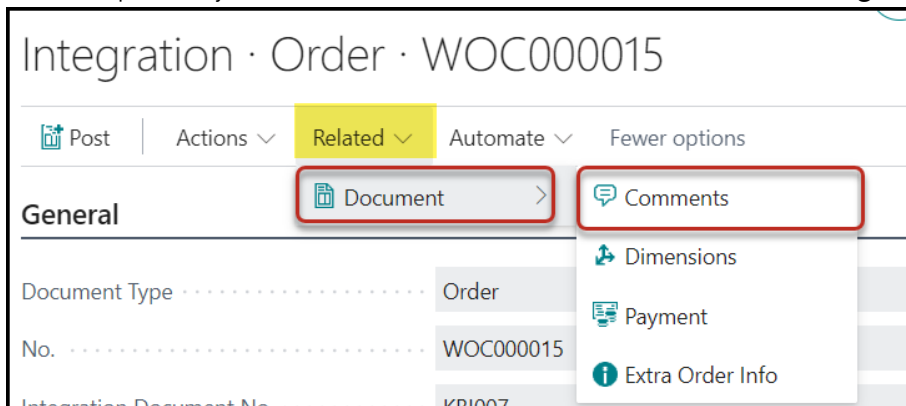
In the **Integration Sales Document** Card, you can see the details of the Integration Sales Document that was imported and not posted yet.

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**.

Lines		Line		Functions	
		Dimensions		Comments	
		VarDim Orders		Set Skip Posting	
Skip Posting	Type	No.	Description	Description 2	
→ ☐	Item	21002004	Lily Blouse	Royal Blue,L	
☐	Item	21002005	Lily Blouse	Royal Blue,XL	

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

X 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

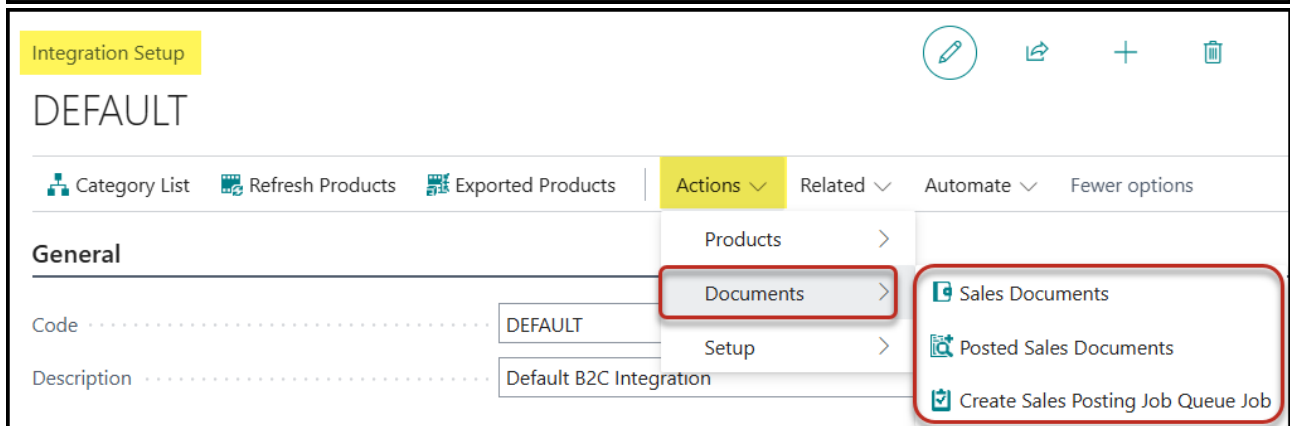
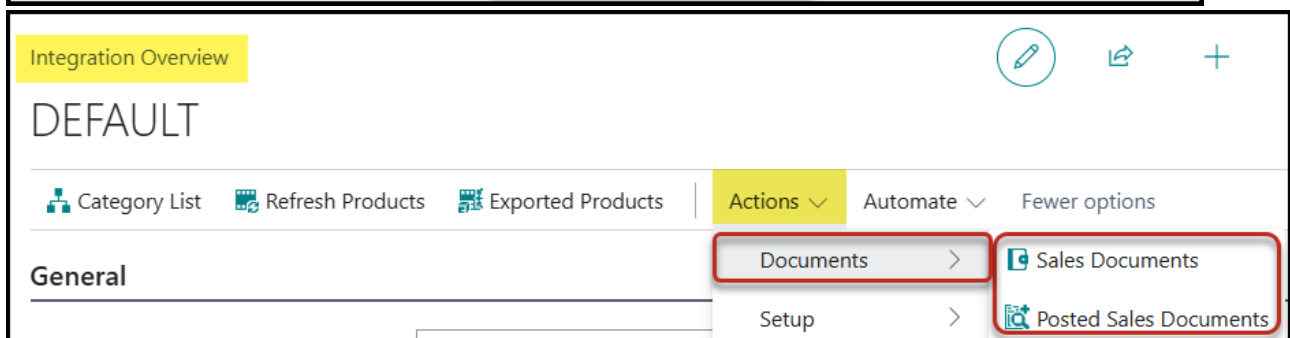
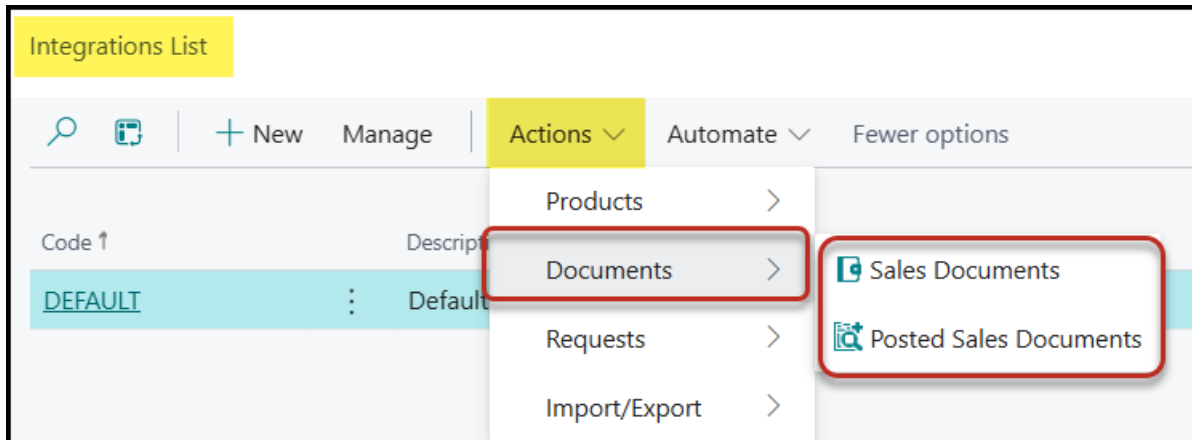
Functions Item Availability by Related Information

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 Entry), you can see it in the Page **Posted Sales Documents** by clicking **Actions, Documents, Posted Sales Documents** in the **Integrations List**, **Integration Overview** or **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.**



Posted Documents							
Integration Sales Documents							
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

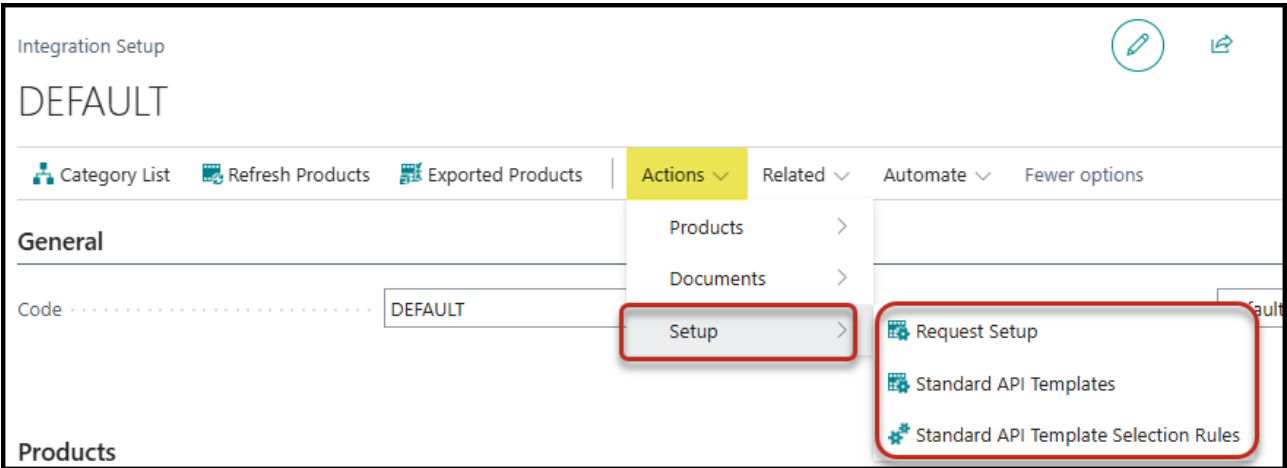
Posted Documents							
Integration Sales Documents							
Document Type	No. ↑	Sell-to Customer No.	Sell-to Customer Name	Location Code	Salesperson Code	Sales Channel	Int. Document No.
Order	WOC000008	CW000010				B2C	2003
Order	WOC000009	CW000010				B2C	KBI001
Order	WOC000010	CW000010				B2C	KBI002
Order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

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 e-commerce.

Posted Sales Document Card

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

Actions, Setup



Request Setup

We will come back to that later in chapter [Request Setup](#).

Standard API Templates

This will show the standard BC Configuration Templates for creating new Customers, where you can create/maintain/delete the Configuration Templates:

The screenshot shows the 'Configuration Templates' table with the following data:

Code ↑	Description
CUSTB2C	Trimit B2C Domestic Customer Template
CUSTB2CEU	Trimit B2C Domestic EU Template
CUSTB2CFOR	Trimit B2C Domestic Foreign Customer Template
CUSTOMER01	Trimit Customer Domestic Template
CUSTOMER02	Trimit Customer EU Template
CUSTOMER03	Trimit Customer Foreign Template

Standard API Template Selection Rules

Per **Conditions** you can set up the default **Template Code** that should be used to create a new Customer for an imported Sales Document.

The screenshot shows the 'API Setup' table with the following data:

Order ↑	Table ID	Page ID	Template Code	Description	Conditions
→	18	0	CUSTB2C	Trimit B2C Domestic Customer Te...	-

Also see [Customer Std. API Template Setup](#).

Related to the Integration Setup

The left screenshot shows the 'Integration Setup' page with the 'Related' dropdown menu open. The 'Customer Template Setup' option is highlighted. The right screenshot shows the same page with the 'Customer Std. API Template Setup' option highlighted.

As mentioned earlier, you can see **Customer Template Setup** or **Customer Std. API Template Setup** based on the setting of the [Use Configuration Customer Templates](#).

Language Setup

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

The screenshot shows the 'Language Setup' page with a table of language codes and names. The table has two columns: 'Language Code' and 'Name'. The 'Language Code' column has a dropdown menu with 'DAN' selected. The 'Name' column lists the corresponding language names.

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:

6037839 trm EAPI Item Description

6037851 trm EAPI Master Description

6037879 trm EAPI Product Description

Currency Setup

For the export of Sales Prices, you can fill in multiple combinations of **Currency Code** and **Customer Price Group**, to export several Sales Prices or Campaign Prices based on the **Cust. Price Group / Campaign**.

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 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					
Currency Code ↑	Description	Type	Cust. Price Group / Campaign ↑		Description
→			Customer Price Group	RETAIL	Retail Prices
DKK	Danish krone	Customer Price Group	RETAIL		Retail Prices
DKK	Danish krone	Campaign	TRM001		Price List Campaign B2C
EUR	Euro	Customer Price Group	RETAIL		Retail Prices
EUR	Euro	Campaign	TRM001		Price List Campaign B2C
GBP	Pound Sterling	Customer Price Group	RETAIL		Retail Prices
GBP	Pound Sterling	Campaign	TRM001		Price List Campaign B2C
USD	US dollar	Customer Price Group	RETAIL		Retail Prices
USD	US dollar	Campaign	TRM001		Price List Campaign B2C

Keep in mind however, if you choose for a **Cust. Price Group / Campaign**, which is a **Campaign Code**, it will only export the Campaign Prices based on Table **7002 Sales Prices**, so it will **NOT** calculate the right Campaign Price based on the different setups that are possible for the TRIMIT Campaigns (like *Mix & Match* or *Get 1 for free*).

This means that only Campaign Prices based on **Campaign Price Calculation Type Price List** in the Price Group Parameters of the Campaign will be exported.

Price Group Parameters

✓ Saved

Campaign · TRM001

Prices

More options

Campaign Setup

Dates

Start Date Campaign Price

End Date Campaign Price

Campaign Price Calculation Type

Limits

Filter Campaign

Filter Price Group

Filter Country/Region Code

Filter Salesperson

Filter Salesperson 2

Portal

Visual Campaign Code

Price List

Location Setup

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

Location Setup				✓ Saved		
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 set up 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* and **Use Configuration Customer Templates** is set to *FALSE*.

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.

Customer Std. API Template Setup

Per **Conditions** you can setup the default **Template Code** that should be used to create a new Customer for an imported Sales Document if **Customer Handling** is set to *Use Customer Template Setup* and **Use Configuration Customer Templates** is set to *TRUE*.

API Setup							✓ Saved	🔗	↗
🔍	📄	+ New	🔧 Edit List	🗑 Delete	Actions ▾	Automate ▾	Fewer options	🔗	🔍
Order ↑		Table ID	Page ID	Template Code	Description	Conditions			
→		18	0	CUSTB2C	Trimit B2C Domestic Customer Te...	—			

Item Setup

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

Item Setup

Not saved

+ New

Edit List

Delete

Automate

Fewer options

Integration Item No. ↑	Country/Region Code ↑	Country/Region Name	Type	G/L Account / Item	Description
1234			Item	20002040	Olivia Trousers
→ TRANSPORT			G/L Account	9996820	Freight Webshop +
			None		
			Item		
			G/L Account		

Additional Data Setup

Export

In **Additional Data Setup**, you are able to add extra Fields from the following Tables that needs to be exported:

- 18 Customer
- 28 Item
- 6037103 Master

The content of these additional Fields will then be shown if you **Expand** the Product Data or Customer Data for the **ConfiguredFields**.

Import

But you are also able to add extra Fields from the following Tables that need to be imported:

- 36 Sales Header
- 37 Sales Line

Th **Additional Data Setup** can also easily be used for Customized Fields in the above-mentioned Tables that should also be exported or imported.

Additional Data Setup

+ New

Edit List

Delete

More options

Table Number ↑	Table Name	Field Number	Field Name
18	Customer	6036961	trm Customer Allocation Prio.
6037103	trm Master	420	Gross Weight
6037103	trm Master	421	Net Weight
→			

Object ID ↑

Object Type ↑

Object Caption

AL Namespace

18

Table

Customer

Microsoft.Sales.Customer

27

Table

Item

Microsoft.Inventory.Item

36

Table

Sales Header

Microsoft.Sales.Document

37

Table

Sales Line

Microsoft.Sales.Document

6037103

Table

Master

Show details

Select from full list

An example for the Export could be:

The screenshot shows the Postman interface with a GET request to the TRIMIT Integration API. The request URL is `{{dockerURL}}/api/trimit/integration/v1.1/$metadata#companies(3189ca11-da89-ee11-b72e-97916988265b)/products?expand=configuredFields&$filter=productNo eq '2110'`. The request body is highlighted with a red box, showing the `$expand` and `$filter` parameters. The response body is highlighted with a blue box, showing the product details for 'Net Weight' and 'Gross Weight'.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

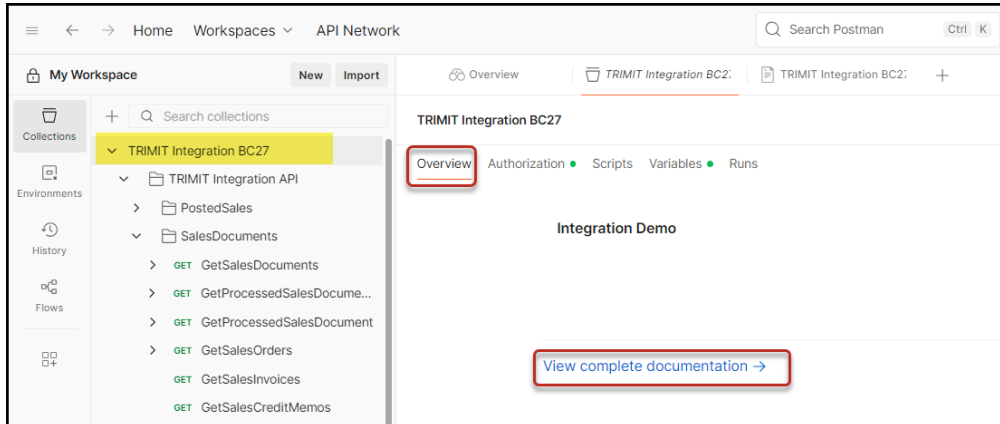
```
{
  "@odata.context": "http://docbc22w1:7040/BC/api/trimit/integration/v1.1/$metadata#companies(3189ca11-da89-ee11-b72e-97916988265b)/products",
  "value": [
    {
      "@odata.etag": "W/\"JzIwOzEyNTA4MzM4MzgzMjc5MTg5MTswMDsn\"",
      "productId": "7d25d4db-3317-ee11-b6b3-c681fd8ee9e9",
      "productType": "Master",
      "productNo": "2110",
      "categoryNode": "{A81AA77C-CB07-4818-9182-71AD0C484988}",
      "lastModifiedDateTime": "2023-06-30T10:51:48.163Z",
      "configuredFields": [
        {
          "@odata.etag": "W/\"JzIwOzEyNTA4MzM4MzgzMjc5MTg5MTswMDsn\"",
          "id": "d7a70fb0-4e93-4919-b8ce-099c5eeb304c",
          "name": "Net Weight",
          "value": "0.5"
        },
        {
          "@odata.etag": "W/\"JzE5OzIzNTQzMzQ4OTk4NDMwMDg5NTcxOzAwOyc\"",
          "id": "07d4cc10-3279-4820-b2bd-8a1c58cd708a",
          "name": "Gross Weight",
          "value": "0.55"
        }
      ]
    }
  ]
}
```

TRIMIT's Postman.json

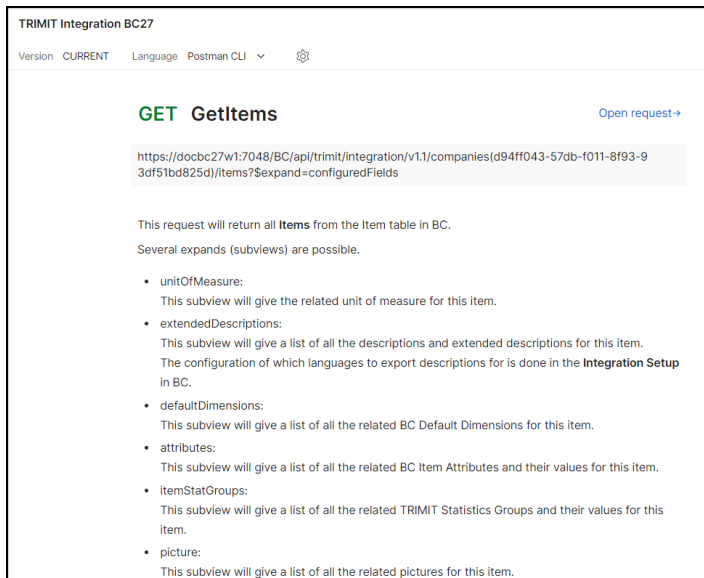
We created a Postman Collection .json file for testing results of importing/exporting data.

Documentation in Postman Collection

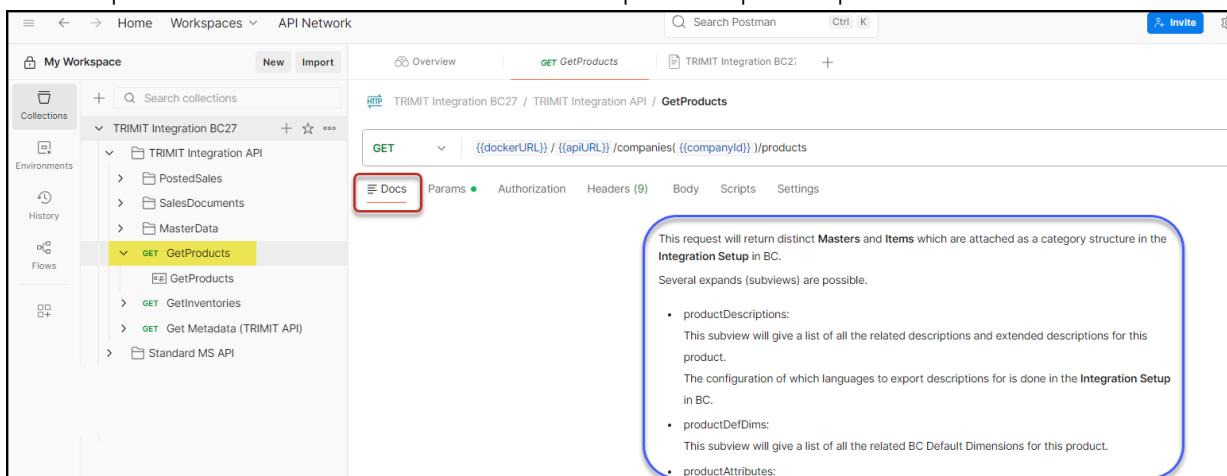
In our **Postman Collection** .json-file we also added documentation. Via **Overview / View complete documentation**, you can find all the documentation we embedded in the **Postman Collection** .json-file.



For instance:

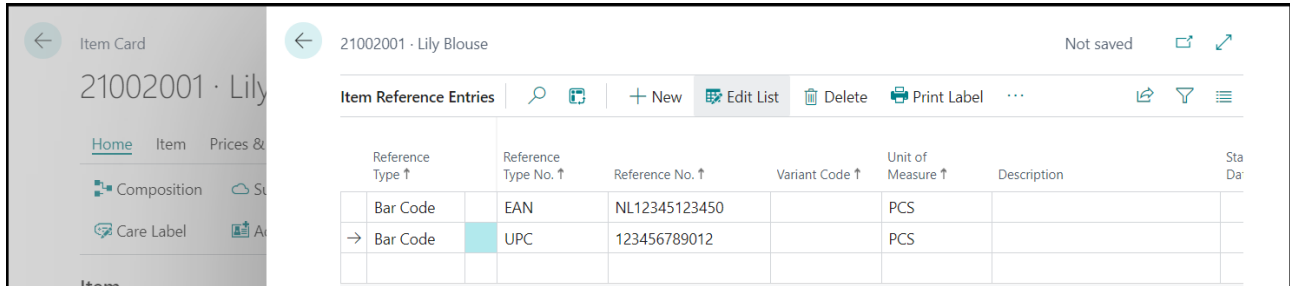


It is also possible to see the Documentation for a specific export/import:



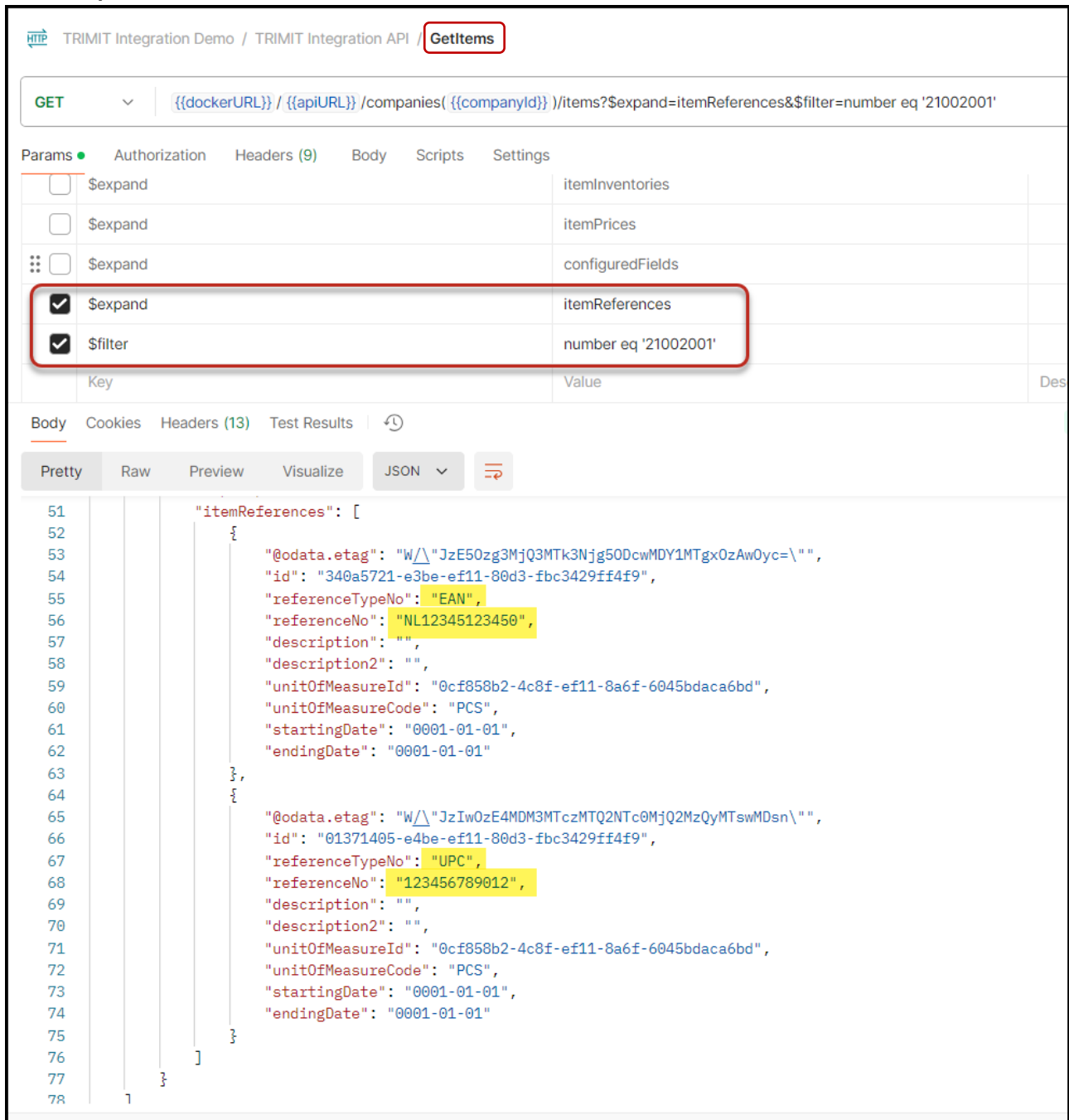
Export of Item References

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:



HTTP 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'

Key Value Des

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 }
```

Export of Prices

It is also possible to filter 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 Calculation Cust. No.** would be 2000, and you would have the following Prices:

Item 2100

Sales Prices

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

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

```
40  vendorItemno :
41
42  "includeInMRP": true,
43  "mmpMethod": "Net Requirement",
44  "mmpPeriod": "",
45  "mmpResponsible": "",
46  "itemPrices": [
47    {
48      "@odata.etag": "W/\"JzE50zMyMDAxMDA1Mjk0NjAyODg0NDUxOzAwOyc=\"",
49      "id": "d2879dab-ed43-4ea3-97cf-dc3cfaeb4909",
50      "customerPriceGroupId": "00000000-0000-0000-0000-000000000000",
51      "customerPriceGroup": "",
52      "currencyId": "66f558b2-4c8f-ef11-8a6f-6045bdaca6bd",
53      "currencyCode": "EUR",
54      "customerId": "00000000-0000-0000-0000-000000000000",
55      "customerNumber": "2000",
56      "unitOfMeasureId": "0cf858b2-4c8f-ef11-8a6f-6045bdaca6bd",
57      "unitOfMeasure": "PCS",
58      "minimumQuantity": 0,
59      "campaignId": "00000000-0000-0000-0000-000000000000",
60      "campaignNo": "",
61      "priceDate": "2024-12-20",
62      "priceIncludesVAT": false,
63      "price": 45.99,
64      "discountPercentage": 0,
65      "externalName": "Calculated price",
66      "calculatedPrice": true
67    }
68  ]
69
70 }
```

Export of Extended Text

It is also possible to export the **Extended Text** of a Master/Item.
Therefore, we added two extra Fields in the Extended Text – which can be added to the Page via **Personalize**:

Extended Text

✓ Saved

2100 Lily Blouse ENU 1

Editor

Automate

Fewer options

General

Language Code

ENU

Ending Date

All Language Codes

Text Format

Html

Description

Integration Format

Text (Default)

Starting Date

Integration Encoding

None

Lines

New Line

Delete Line

Text

→ This is extended Text for the Master 2100 to describe more details of our famous Lily Blouse.

Integration Format

In case the **Extended Text** should also be exported to other systems by using TRIMIT's Integration (External API), you can determine what the format should be for this text:
You can choose between the following two options:

Text (Default)	The Extended Text will be exported in a regular Text format; this is also the default.
Html	The Extended Text will be exported in Html format.

Integration Encoding

In case the **Extended Text** should also be exported to other systems by using TRIMIT's Integration (External API), you can also determine if it should be encrypted.
You can choose between the following two options:

None	The Extended Text will not be encoded.
Base64	The Extended Text will be Base64 encoded.

There is also Batch report **Change Extended Text Integration (Batch) 6036840** available to change these two Fields in the **Extended Text** for many subjects at the same time,
You can find this report via **Search**:

Tell me what you want to do

Change Extended Text

Go to Pages and Tasks

Change Extended Text Integration (Batch)

Administration

Change Extended Text Integration (Batch)

Set New Integration Data Format and Encoding

Integration Data Format: Html

Integration Data Encoding: Base64

Filter: Extended Text Header

× Table Name: Item

× No.: ☐ Standard Text

× Language Code: ☐ G/L Account

☒ Item

☐ Resource

☐ VAT Clause

[+ Filter...](#)

[Schedule...](#) [OK](#) [Cancel](#)

For the **Extended Text** connected to Masters, you need to select **Table Name** *Item*.

By using **\$expand=productDescriptions**, not only the Descriptions of Masters/Items in several [Languages](#) will be exported, but also the Extended Text. I.e.:

TRIMIT Integration BC27 / TRIMIT Integration API **GetProducts** [Save](#) [Share](#)

GET [[{dockerURL}]] / [[{apiURL}]] / companies([[{companyId}]]) / products? \$expand=productDescriptions&\$filter=productNo eq '2100' [Send](#)

Docs Params Authorization Headers (9) Body Scripts Settings [Cookies](#)

Query Params

☐ Key	Value	Description	Bulk Edit
☐ \$orderby	productType,productId		
☒ \$expand	productDescriptions		
☐ \$expand	productImages		

Body Cookies Headers (11) Test Results [200 OK](#) · 91 ms · 2.52 KB [Save Response](#)

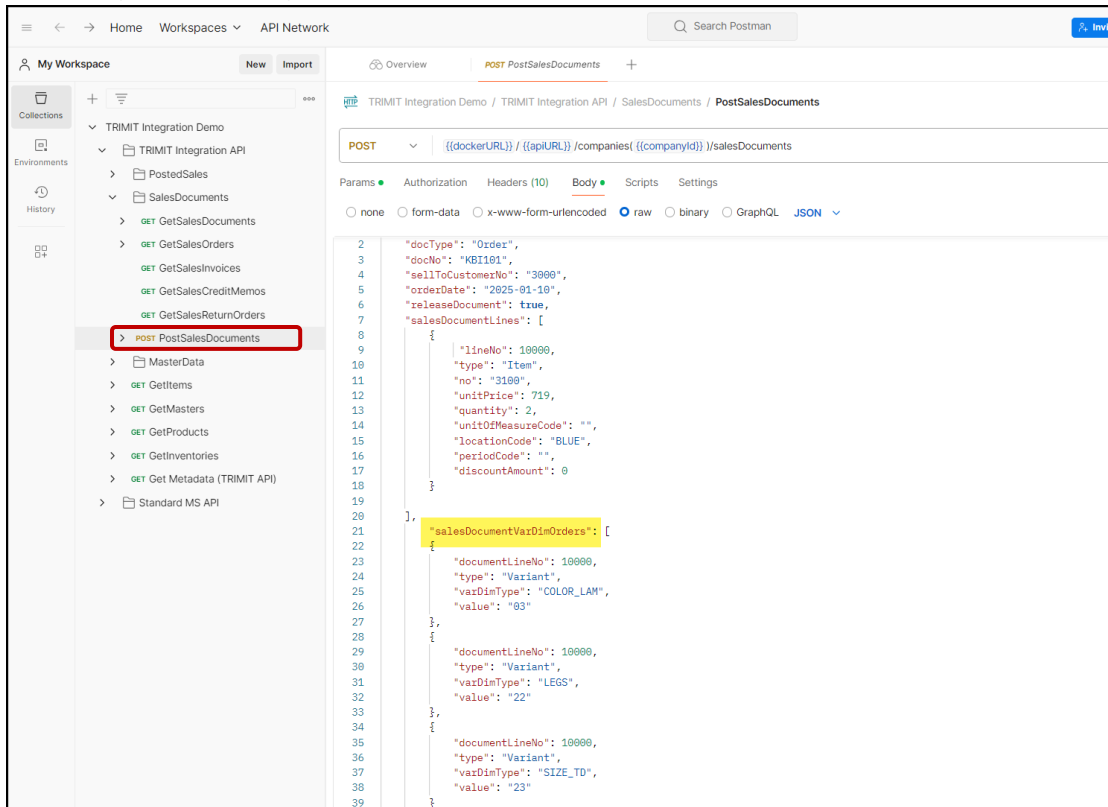
[JSON](#) [Preview](#) [Visualize](#)

```

26      "languageCode": "DEU",
27      "languageId": "79bb61a7-07db-f011-9de7-7ced8d3ebd95",
28      "description": "Lily Blouse",
29      "description2": "",
30      "extDescriptionTitle": "",
31      "extendedDescription": "",
32      "extDescriptionLanguageBlank": true
33    },
34    {
35      "@odata.etag": "W/Δ\"JzE50zExMTM2MDQ0NjM0MjYyNTc0MjIxOzAwOyc=\"",
36      "id": "a380eef-c2f6-4258-b51c-fc3a586fc653",
37      "languageCode": "ENU",
38      "languageId": "80bb61a7-07db-f011-9de7-7ced8d3ebd95",
39      "description": "Lily Blouse",
40      "description2": "",
41      "extDescriptionTitle": "",
42      "extendedDescription": "This is extended Text for the Master 2100 to describe more details of our famousLily Blouse.",
43      "extDescriptionLanguageBlank": false
44    }
  
```

Import of VarDim Order

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



Import Comment Sales Document Line by using Description / Description 2

You are also able to import Sales Document Lines for **Type Comment**.

Therefore, you can use the "description" and the "description2".

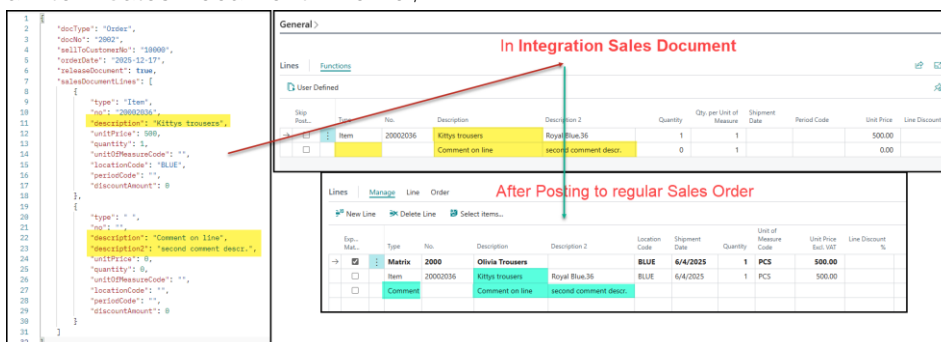
However, in that case, you need to have the **Type** of the Sales Document Line set to " " (empty).

Note

Importing a Sales Document with only Lines for **Type " "** and not Item Lines, will give an error when you try to post the **Integration Sales Document**.

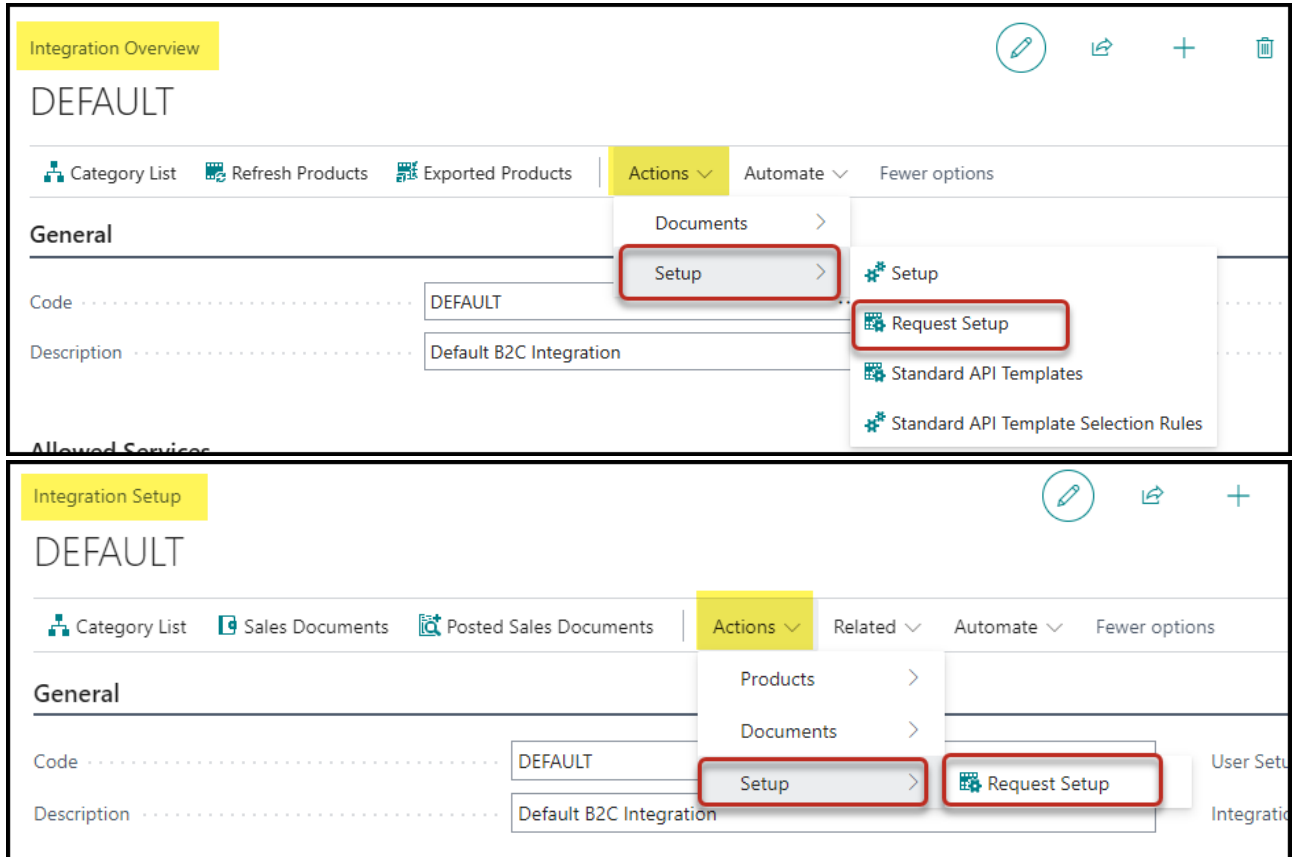
Import Description and Description 2

You can use the "description" and "description2" also for overwriting the Description and Description 2 of an Item Sales Document Line. I.e.,



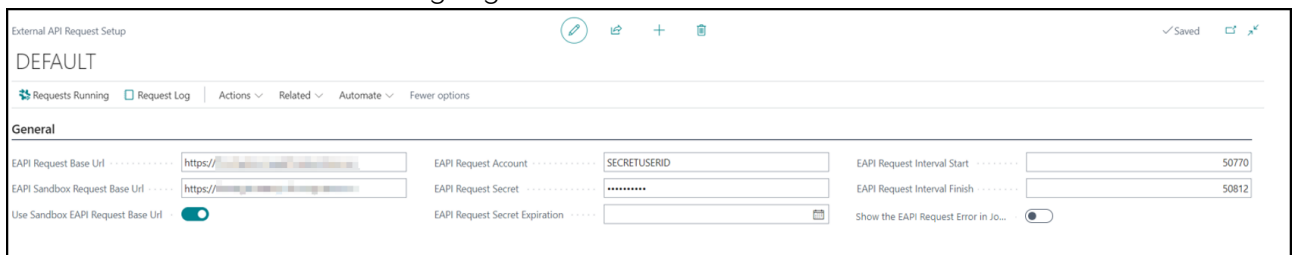
Requests

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 will take 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.

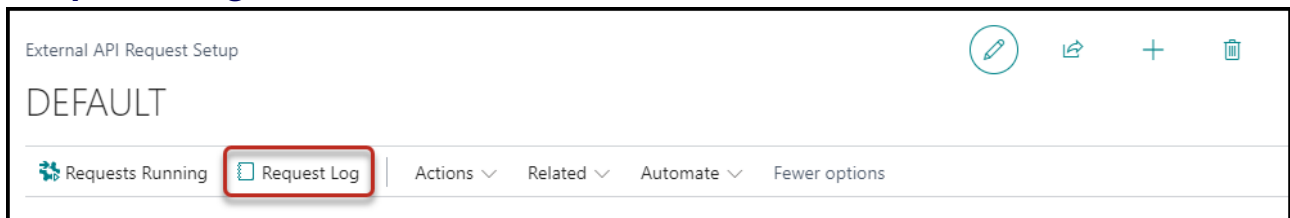
Without a checkmark, it will simply skip the error and continue to the next Request.

Requests Running



This will show a list of all running Requests.

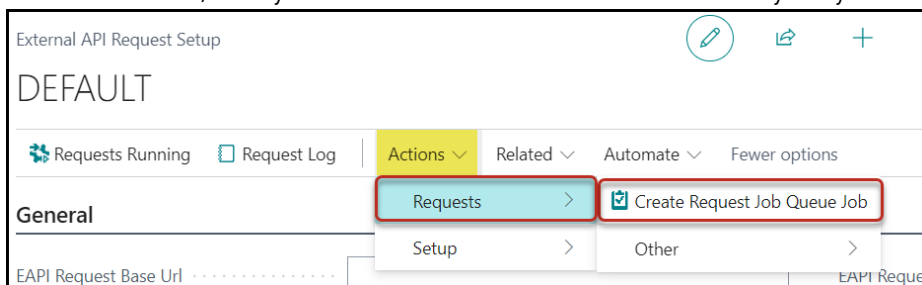
Request Log



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 Entry for your data requests:



You will get a message:



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	Maximum No. of Attempts to Run	3
Object ID to Run	6036874	Rerun Delay (sec.)	0
Object Caption to Run	trm EAPI Request Task Handling	Last Ready State	12/20/2024 10:28 AM
Description	Created for automatic running of TRIMIT EAPI Requests	Earliest Start Date/Time	12/30/2024 8:47 AM
Parameter String		Expiration Date/Time	
Job Queue Category Code		Job Timeout	12 hours
Priority	Normal	Status	On Hold
User ID	ADMIN		

External API Request Setup

DEFAULT

Requests Running

Request Log

Actions

Related

Automate

Fewer options

General

Requests

External API Requests

GetSale

GetSaleLines

GetSalePayments

GetProductTransfer

GetMovements

GetProducts

GetStockStatus

GetStoreOrders

GetSettlements

GET

GET

GET

GET

GET

GET

GET

GET

https://.../oData/Sales?\$filter=SaleDate%20ge%20dateTime...

https://.../oData/SalesLines?from=%7BfromDate%7D&to=...

https://.../oData/SalePayments?from=%7BfromDate%7D&to=...

https://.../oData/ProductTransfers?from=%7BfromDate%7D&to=...

https://.../oData/Stockmovements?from=%7BfromDate%7D&to=...

https://.../restapi/V2/api/Groups

https://.../restapi/V2/api/Groups

https://.../restapi/V2/api/Groups

https://.../restapi/V2/api/Groups

?\$filter=SaleDate%20ge%20dateTime...

?from=%7BfromDate%7D&to=...

?from=%7BfromDate%7D&to=...

?from=%7BfromDate%7D&to=...

?from=%7BfromDate%7D&to=...

None

None

None

None

None

None

None

None

application/atom

application/atom

application/atom

application/atom

application/atom

application/json

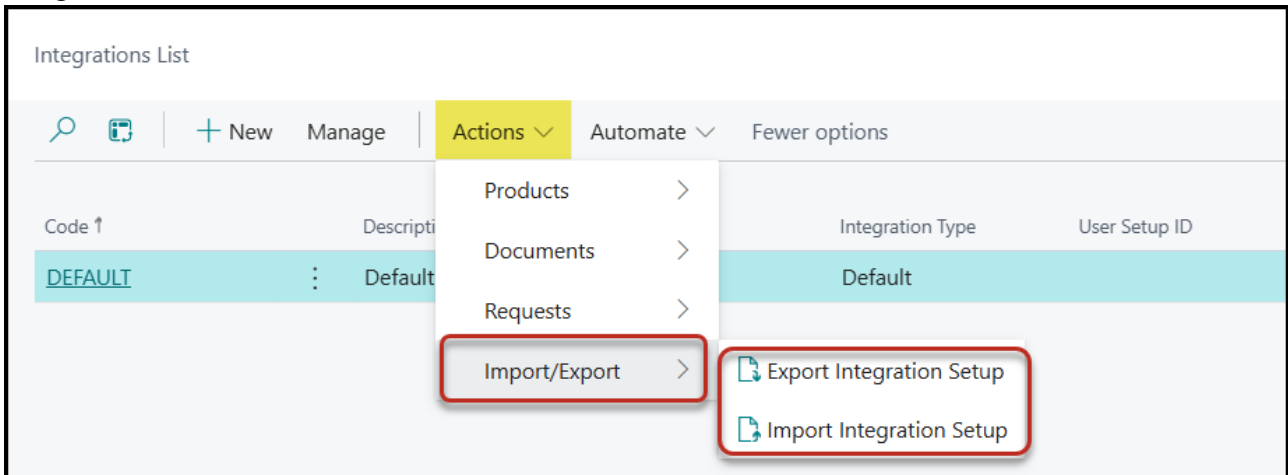
application/json

application/json

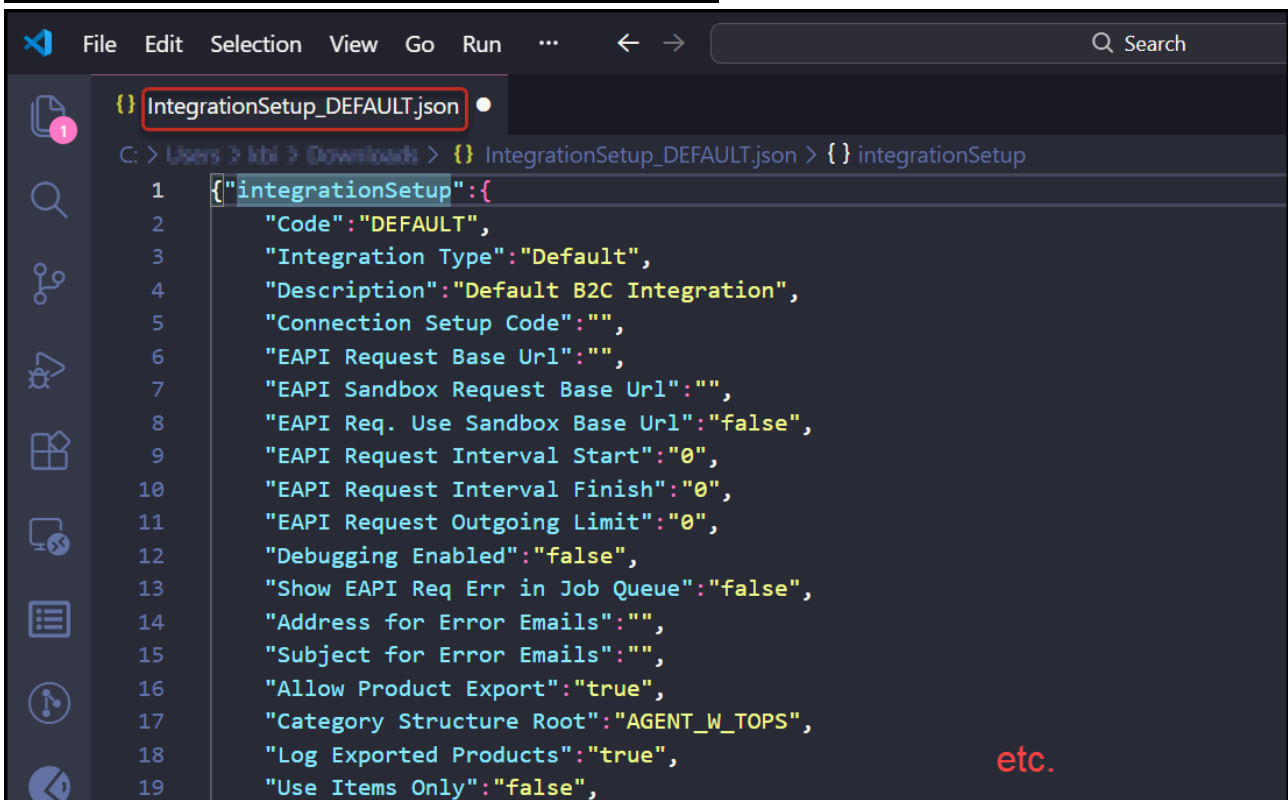
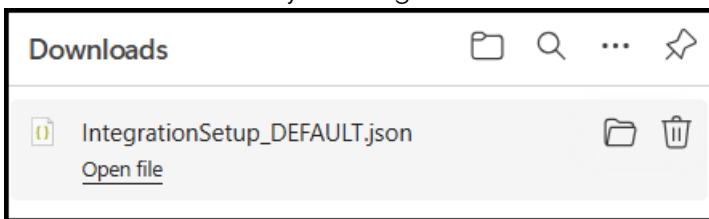
Page 55

Export/Import of Integration Code

You are able to Export and Import the Data from an Integration Setup via **Actions, Import/Export** in the **Integrations List**.



When clicking **Export Integration Setup**, it will create an *IntegrationSetup_XXX.json* file where the XXX stands for the **Code** of your Integration.



What is exported exactly:

- ➔ Content of **Integration Setup**
- ➔ Content of all **Related Information** of the Integration Setup

The screenshot shows the 'Integration Setup' form with the 'DEFAULT' tab selected. The 'Related' dropdown menu is open, showing options: Language Setup, Currency Setup, Location Setup, Customer Setup, Customer Template Setup, Item Setup, and Additional Data Setup. The 'General' section shows 'Code' as 'DEFAULT' and 'Description' as 'Default'. The 'Products' section has a toggle for 'Allow Product Export' which is turned on.

- ➔ Content of the **Requests** and the **External API Request Setup**, but we excluded customer sensitive information like user-/account names and passwords (client id and secret)

The screenshot shows the 'External API Request Setup' form with the 'DEFAULT' tab selected. The 'Requests' dropdown menu is open, showing the 'Requests' option. The 'General' section shows fields for 'Request Base Url', 'Sandbox Request Base Url', 'Use Sandbox Request Base Url', 'Request Account', and 'Request Secret'. The 'Request Secret Expiration' field has a calendar icon. The 'Request Interval Start' and 'Request Interval Finish' fields are empty. The 'Show Request Errors in Job Queue' toggle is turned on.

What is imported exactly:

- ➔ Same content as what was exported – i.e., after you changed some content:

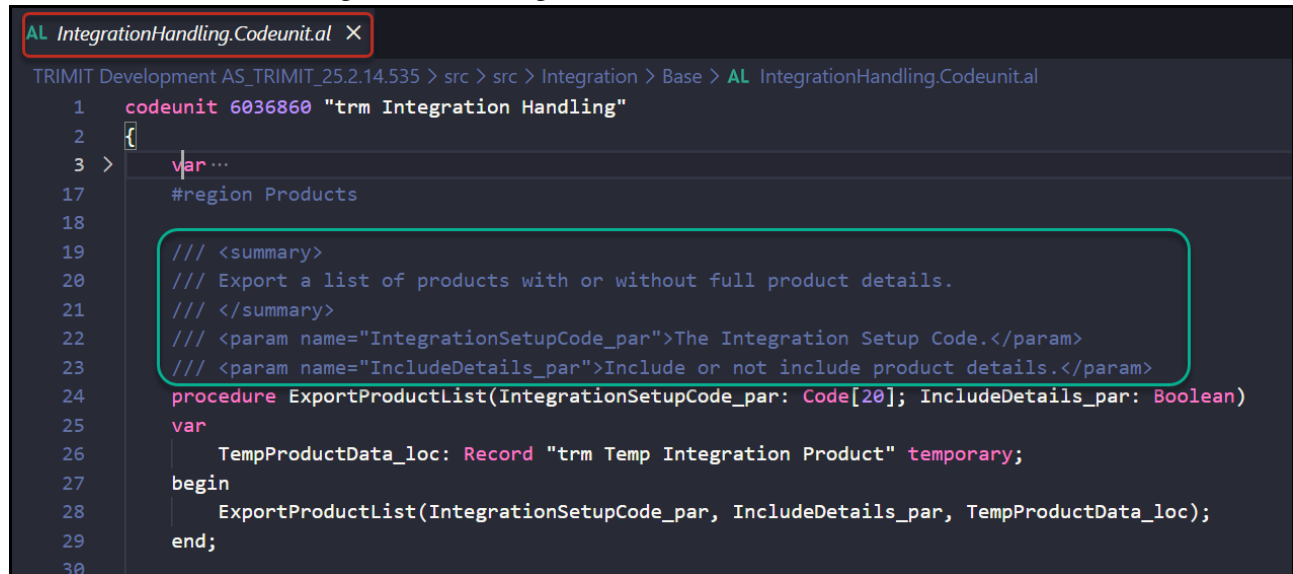
```
{
  "IntegrationSetup": {
    "Code": "TEST",
    "Integration Type": "TEST for Import",
    "Description": "Default B2C Integration",
    "Connection Setup Code": "",
    "EAPI Request Base Url": "",
    "EAPI Sandbox Request Base Url": "",
    "EAPI Req. Use Sandbox Base Url": "false",
    "EAPI Request Interval Start": "0"
  }
}
```

If you do so, make sure you do it also in all the related information, otherwise, you will get an error message while importing. So, you cannot import what was already present. I.e.,

The error message dialog box states: "The record in table Integration Setup Line already exists. Identification fields and values: Integration Setup Code='DEFAULT',Relation 1='Language',Relation 1 Code='DAN',Relation 2='None',Relation 2 Code='". There is a "Share details" link and a feedback section asking "Was this information helpful?" with "Yes" and "No" buttons, and an "OK" button.

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_25.2.14.535 > src > src > Integration > Base > AL IntegrationHandling.Codeunit.al
1  codeunit 6036860 "trm Integration Handling"
2  {
3  >  var ...
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

RefreshProductBuffer()

Refreshes the product buffer used for product exports.

Syntax

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

Parameters

IntegrationSetupCode_par

Type: Code[20]

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]

TemplInventoryData_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]

TemplItemInventoryData_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]

TemplItemInventoryData_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"

GetProductReferences()

Get references for a specific product.

Syntax

```
GetProductReferences(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

IsItemValid()

Check the inactive flag in the matrix for this product.

Syntax

```
IsItemValid(item_par: Record Item): Boolean)
```

Parameters

Item_par

Type: Record "Item"

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()
```

CheckCustomerImportAllowed()

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

Syntax

```
CheckCustomerImportAllowed()
```

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 even though 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 even though 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 even though 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_OnCustomExtendedTextFormat ()

Syntax

```
pf_OnCustomExtendedTextFormat(DataFormat_par: Enum "trm Integration Data Format";  
ItemNo_par: Code[20]; LanguageCode_par: Code[10]; var ExtendedTextHeader_par:  
Record "Extended Text Header"; var ExtendedText_par: Text)
```

Parameters

DataFormat_par

Type: Enum "trm Integration Data Format"

ItemNo_par

Type: Code[20]

LanguageCode_par

Type: Code[10]

ExtendedTextHeader_par

Type: Record "Extended Text Header"

ExtendedText_par

Type: Text

pf_OnCustomExtendedTextEncoding ()

Syntax

```
pf_OnCustomExtendedTextFormat(DataFormat_par: Enum "trm Integration Data Encoding";  
ItemNo_par: Code[20]; LanguageCode_par: Code[10]; var ExtendedTextHeader_par:  
Record "Extended Text Header"; var ExtendedText_par: Text)
```

Parameters

DataFormat_par

Type: Enum "trm Integration Data Encoding"

ItemNo_par

Type: Code[20]

LanguageCode_par

Type: Code[10]

ExtendedTextHeader_par

Type: Record "Extended Text Header"

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_OnAfterGetReferences(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_OnBeforeGetInventories()

Syntax

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

Parameters

TempInventoryData_par

Type: Record "trm Temp Integration Inventory"

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_OnBeforeGetItemInventories()

Syntax

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

Parameters

TempItemInventoryData_par

Type: Record "trm Temp Integration Product"

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

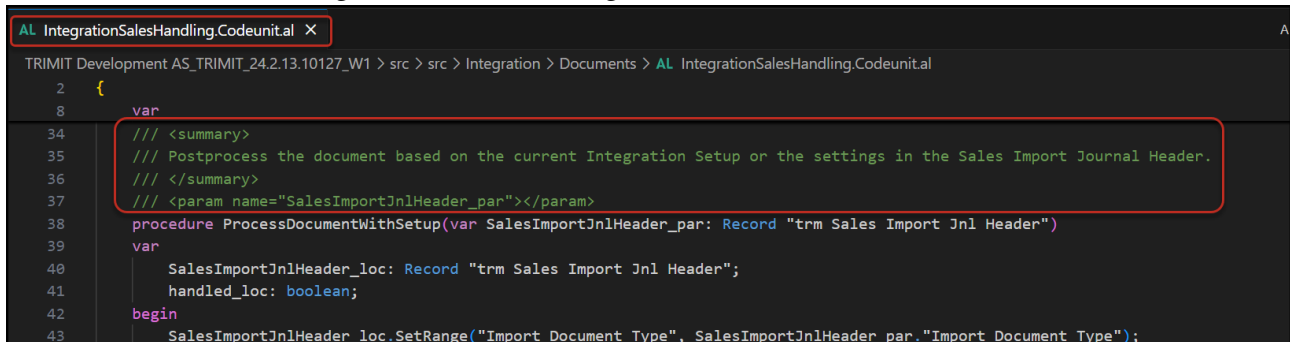
Type: Code[10]

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"

CopyOrderExtraInfo()

Copy Extra Order Information from the referred Sales Import Journal Header to the corresponding Sales Order.

Syntax

```
CopyOrderExtraInfo(var salesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; salesHeaderNotFoundAllowed_par: Boolean): Boolean
```

Parameters

salesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

salesHeaderNotFoundAllowed_par

Type: Boolean

Return

Boolean: True if copy operation was successful

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"

GetTrackingUrl()

Syntax

```
GetTrackingUrl(PackageTrackingNo_par: Code[50]; ShippingAgentCode_par: Code[10]): Text[2048];
```

Parameters

PackageTrackingNo_par

Type: Code[50]

ShippingAgentCode_par

Type: Code[10]

Return

Text[2048]

GetCustomerNo()

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

Syntax

```
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

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

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

GetCustomerNoFromTemplateSetup()

Syntax

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

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

CreateCustomerFromTemplate()

Syntax

```
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"

FindTemplateFromCustomer()

Syntax

```
FindTemplateFromCustomer(var Customer_par: Record Customer): Code[20]
```

Parameters

Customer_par

Type: Record "Customer"

ApplyTemplateToCustomer()

Syntax

```
ApplyTemplateToCustomer(var Customer_par: Record Customer): Code[20]
```

Parameters

Customer_par

Type: Record "Customer"

UseConfigurationTemplatesOnCustomer()

Syntax

```
UseConfigurationTemplatesOnCustomer(): Boolean
```

Return

Boolean: "Use Config. Cust. Templates" from Integration Setup

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"

GetAdditionalFieldsBlob ()

Syntax

```
GetAdditionalFieldsBlob(recVariant: Variant; blobFieldNo_par: Integer; var  
tableId_par: Integer; var systemId_par: Guid; var tempBlob_par: Codeunit "Temp  
Blob")
```

Parameters

recVariant

Type: Variant

blobFieldNo_par

Type: Integer

tableId_par

Type: Integer

systemId_par

Type: Guid

tempBlob_oar

Type: Codeunit "Temp Blob"

GetAdditionalFieldsBlob ()

Syntax

```
GetAdditionalFieldsBlob(var table_par: RecordRef; blobFieldNo_par: Integer; var  
tempBlob_par: Codeunit "Temp Blob")
```

Parameters

Table_par

Type: RecordRef

blobFieldNo_par

Type: Integer

tempBlob_oar

Type: Codeunit "Temp Blob"

DeserializeAdditionalFields ()

Syntax

```
DeserializeAdditionalFields(tableId_par: Integer; systemId_par: Guid; tempBlob_par: Codeunit "Temp Blob"; var additionalFields_par: Record "trm Temp Integration Data" temporary)
```

Parameters

tableID_par

Type: Integer

systemId_par

Type: Guid

tempBlob_oar

Type: Codeunit "Temp Blob"

additionalFields_par

Type: Record "trm Temp Integration Data"

SetAdditionalFieldsOnTarget ()

Syntax

```
SetAdditionalFieldsOnTarget(var additionalFields_par: Record "trm Temp Integration Data" temporary; var recRef_par: RecordRef)
```

Parameters

additionalFields_par

Type: Record "trm Temp Integration Data"

recRef_par

Type: RecordRef

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"

Pf_OnAfterGetShipment()

This publisher is called for every sales shipment that is exported to the external system. Be aware that subscription to this can have performance implications.

Syntax

```
Pf_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_OnAfterGetIntegrationShipment()

This publisher is called for every sales shipment that is exported to the external system. Be aware that subscription to this can have performance implications.

Syntax

```
Pf_OnAfterGetIntegrationShipment(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 SalesTrack" temporary)
```

Parameters

TempSalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

TempSalesImportJnlLine_par

Type: Record "trm Sales Import Jnl Line"

TempSalesTracking_par

Type: Record "trm Temp Integration SalesTrack"

Pf_OnAfterGetInvoice()

This publisher is called for every posted invoice that is exported to the external system. Be aware that subscription to this can have performance implications.

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_OnAfterGetIntegrationInvoice()

This publisher is called for every posted invoice that is exported to the external system. Be aware that subscription to this can have performance implications.

Syntax

```
Pf_OnAfterGetIntegrationInvoice(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 Jnl Line"

TempSalesTracking_par

Type: Record "trm Temp Integration SalesTrack"

Pf_OnAfterGetReturnReceipt()

This publisher is called for every return receipt that is exported to the external system. Be aware that subscription to this can have performance implications.

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_OnAfterGetIntegrationReturnReceipt()

This publisher is called for every return receipt that is exported to the external system. Be aware that subscription to this can have performance implications.

Syntax

```
Pf_OnAfterGetIntegrationReturnReceipt(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 Jnl Line"

TempSalesTracking_par

Type: Record "trm Temp Integration SalesTrack"

Pf_OnAfterGetCreditMemo()

This publisher is called for every credit memo that is exported to the external system. Be aware that subscription to this can have performance implications.

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_OnAfterGetIntegrationCreditMemo ()

This publisher is called for every credit memo that is exported to the external system. Be aware that subscription to this can have performance implications.

Syntax

```
Pf_OnAfterGetIntegrationCreditMemo(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_ 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_part

Type: Decimal

TempSalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header" temprary

TempSalesImportJnlLine_par

Type: Record "trm Sales Import Journal Line" temprary

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_part

Type: Decimal

TempSalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header" temprary

TempSalesImportJnlLine_par

Type: Record "trm Sales Import Journal Line" temprary

IsHandled_loc

Type: Boolean

Pf_OnBeforeGetCustNo()

Syntax

Pf_OnBeforeGetCustNo(var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var CustomerNo_par: Code[20]; var Handled_par: Boolean)

Parameters

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

CustomerNo_par

Type: Code[20]

Handled_par

Type: Boolean

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_OnBeforeGetCustNoFromSetup ()

Syntax

Pf_OnBeforeGetCustNoFromSetup(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var handled_par: Boolean)

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

CustomerNo_par

Type: Code[20]

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Handled_par

Type: Boolean

Pf_OnBeforeFindCustNoFromIncomingDoc

Syntax

Pf_OnBeforeFindCustNoFromIncomingDoc(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var handled_par: Boolean)

Parameters

IntegrationSetup_par

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

Pf_OnBeforeSearchCustNo

Syntax

```
Pf_OnBeforeSearchCustNo(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var handled_par: Boolean)
```

Parameters

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

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_OnGetCustNoFromSetupFailed()

Syntax

```
Pf_OnGetCustNoFromSetupFailed(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var errorHandled_par: Boolean)
```

Parameters

IntegrationSetup_par
Type: Record "trm Integration Setup"

CustomerNo_par

Type: Code[20]

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

errorHandled_par

Type: Boolean

Pf_OnBeforeGetCustNoFromTemplateSetup()

Syntax

```
Pf_OnBeforeGetCustNoFromTemplateSetup(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

CustomerNo_par

Type: Code[20]

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

Handled_par

Type: Boolean

Pf_OnMissingCustomerSellToEmail()

Syntax

```
Pf_OnMissingCustomerSellToEmail(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var errorHandled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

CustomerNo_par

Type: Code[20]

SalesImportJnlHeader_par

Type: Record "trm Sales Import Jnl Header"

errorHandled_par

Type: Boolean

Pf_OnAfterGetCustNoFromTemplateSetup()

Syntax

```
Pf_OnAfterGetCustNoFromTemplateSetup(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; handled_par: Boolean)
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

CustomerNo_par

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

Pf_OnGetCustNoFromTemplateSetupFailed()

Syntax

```
Pf_OnGetCustNoFromTemplateSetupFailed(IntegrationSetup_par: Record "trm Integration Setup"; var CustomerNo_par: Code[20]; var SalesImportJnlHeader_par: Record "trm Sales Import Jnl Header"; var errorHandled_par: Boolean)
```

Parameters

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

Pf_OnBeforeCreateCustFromTemplate()

Syntax

```
Pf_OnBeforeCreateCustFromTemplate(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_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_par
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_par
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_par
Type: Boolean

Pf_OnBeforeFindTemplateFromCustomer()

Syntax

```
Pf_OnBeforeFindTemplateFromCustomer(IntegrationSetup_par: Record "trm Integration Setup"; var Customer_par: Record Customer; var CustomerTemplateNo_par: Code[20]; var Handled_par: Boolean);
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

Customer_par

Type: Record "Customer"

CustomerTemplateNo_par

Type: Code[20]

Handled_par

Type: Boolean

Pf_OnAfterFindTemplateFromCustomer()

Syntax

```
Pf_OnAfterFindTemplateFromCustomer(IntegrationSetup_par: Record "trm Integration Setup"; var Customer_par: Record Customer; var CustomerTemplateNo_par: Code[20]; var Handled_par: Boolean);
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

Customer_par

Type: Record "Customer"

CustomerTemplateNo_par

Type: Code[20]

WasHandled_par

Type: Boolean

Pf_OnGetTemplateSetupFailed()

Syntax

```
Pf_OnGetTemplateSetupFailed(IntegrationSetup_par: Record "trm Integration Setup"; var Customer_par: Record Customer; var CustomerTemplateNo_par: Code[20]; var Handled_par: Boolean);
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

Customer_par

Type: Record "Customer"

CustomerTemplateNo_par

Type: Code[20]

Handled_par

Type: Boolean

Pf_OnBeforeApplyTemplateToCustomer()

Syntax

```
Pf_OnBeforeApplyTemplateToCustomer(IntegrationSetup_par: Record "trm Integration Setup"; var Customer_par: Record Customer; var CustomerTemplateNo_par: Code[20]; var Handled_par: Boolean);
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

Customer_par

Type: Record "Customer"

CustomerTemplateNo_par

Type: Code[20]

Handled_par

Type: Boolean

Pf_OnAfterApplyTemplateToCustomer()

Syntax

```
Pf_OnAfterApplyTemplateToCustomer(IntegrationSetup_par: Record "trm Integration Setup"; var Customer_par: Record Customer; var CustomerTemplateNo_par: Code[20]; var WasHandled_par: Boolean);
```

Parameters

IntegrationSetup_par

Type: Record "trm Integration Setup"

Customer_par

Type: Record "Customer"

CustomerTemplateNo_par

Type: Code[20]

WasHandled_par

Type: Boolean

Pf_ OnBeforeCreateIntegrationDoc()

Syntax

```
Pf_OnBeforeCreateIntegrationDoc(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_par

Type: Boolean

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_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_par

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_par

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_par

Type: Boolean

Pf_OnAfterProcessDocumentWithSetup()

Syntax

```
Pf_OnAfterProcessDocumentWithSetup(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_par

Type: Boolean

Pf_OnAfterProcessIntegrationDocWithSetup()

Syntax

```
Pf_OnAfterProcessIntegrationDocWithSetup(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_par

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

Pf_OnBeforeSetAdditionalField()

Syntax

Pf_OnBeforeSetAdditionalField(var additionalFields_par: Record "trm Temp Integration Data" temporary; var recRef_par: RecordRef; var fieldRef_par: FieldRef; var validateField_par: Boolean; var isHandled_par: Boolean)

Parameters

additionalFields_par

Type: Record "trm Temp Integration Data"

recRef_par

Type: RecordRef

fieldRef_par

Type: FieldRef

validateField_par

Type: Boolean

isHandled_par

Type: Boolean

Pf_OnAdditionalFieldNotFound()

Syntax

Pf_OnAdditionalFieldNotFound(var additionalFields_par: Record "trm Temp Integration Data" temporary; var recRef_par: RecordRef; var isHandled_par: Boolean)
IntegrationSetup_par

Parameters

additionalFields_par

Type: Record "trm Temp Integration Data"

recRef_par

Type: RecordRef

isHandled_loc
Type: Boolean